



Universidad del Bío-Bío
Facultad de Ciencias Empresariales

**Sistema de apoyo a la enseñanza de la
programación de computadores “basada en
el ejemplo” en entorno colaborativo
(EXEMPLUM)**

Proyecto Título para optar al título de Ingeniero de
Ejecución en Computación e Informática

Viernes, 17 de agosto de 2012
Concepción - Chile

Integrantes:

Jonathan Fierro Sandoval
Iván Nawrath Castillo

Profesor guía:

Dr. Christian Vidal Castro

Resumen

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad de Bío-Bío en el proceso de titulación para a la carrera de Ingeniería de Ejecución en Computación e Informática. El proyecto titulado “Sistema de apoyo a la enseñanza de la programación de computadores “basada en el ejemplo” en entorno colaborativo”.

El proyecto se realiza para la Universidad del Bío-Bío, específicamente para la Facultad de Ciencias Empresariales, donde existe la preocupación de mejorar el aprendizaje de lenguajes de programación en los alumnos de las carreras de Ingeniería de Ejecución en Computación e Informática (IECI) e Ingeniería Civil Informática (ICInf). En este sentido, se intentan utilizar enfoques prácticos y colaborativos para obtener dicho aprendizaje.

Esto se desarrolla, en mayor parte, con PHP y HTML, además de utilizar un patrón de arquitectura de software (Modelo Vista Controlador Orientado a Objetos), permitiendo una mayor organización del código y de mantenimiento sencillo. Además, el sistema permite la obtención de informes del trabajo realizado por los diferentes alumnos, esto para analizar y/o trabajar los diferentes informes mediante la tecnología XML.

Índice General

1	INTRODUCCIÓN	9
2	DEFINICION DE LA EMPRESA O INSTITUCIÓN	10
2.1	DESCRIPCIÓN DE LA EMPRESA	10
2.2	DESCRIPCIÓN DEL ÁREA DE ESTUDIO	13
2.3	DESCRIPCIÓN DE LA PROBLEMÁTICA	13
3	DEFINICIÓN PROYECTO	14
3.1	OBJETIVOS DEL PROYECTO	15
3.2	AMBIENTE DE INGENIERÍA DE SOFTWARE	16
3.3	DEFINICIONES, SIGLAS Y ABREVIACIONES	18
4	ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE	19
4.1	ALCANCES	19
4.2	OBJETIVO DEL SOFTWARE	19
4.3	DESCRIPCIÓN GLOBAL DEL PRODUCTO	20
4.3.1	INTERFAZ DE USUARIO	20
4.3.2	INTERFAZ DE HARDWARE	20
4.3.3	INTERFACES DE COMUNICACIÓN	20
4.4	REQUERIMIENTOS ESPECÍFICOS	21
4.4.1	REQUERIMIENTOS FUNCIONALES DEL SISTEMA	21
4.4.2	INTERFACES EXTERNAS DE ENTRADA	22
4.4.3	INTERFACES EXTERNAS DE SALIDA	22
4.4.4	ATRIBUTOS DEL PRODUCTO	23
5	FACTIBILIDAD	24
5.1	FACTIBILIDAD TÉCNICA	24
5.2	FACTIBILIDAD OPERATIVA	25
5.3	FACTIBILIDAD ECONÓMICA	25
5.4	CONCLUSIÓN DE LA FACTIBILIDAD	25
6	ANÁLISIS	26
6.1	DIAGRAMA DE CASOS DE USO (CU)	26
6.1.1	ACTORES	29
6.1.2	CASOS DE USO Y DESCRIPCIÓN	32
6.1.3	ESPECIFICACIÓN DE LOS CASOS DE USO	38
7	DISEÑO	64
7.1	DISEÑO DE FÍSICO DE LA BASE DE DATOS	64
7.2	DISEÑO DE ARQUITECTURA FUNCIONAL	69
7.3	DISEÑO INTERFAZ Y NAVEGACIÓN	72
7.4	ESPECIFICACIÓN DE MÓDULOS	89
8	PRUEBAS	91
8.1	ELEMENTOS DE PRUEBA	91

8.2	ESPECIFICACIÓN DE LAS PRUEBAS	91
8.3	RESPONSABLES DE LAS PRUEBAS	91
8.4	DETALLE DE LAS PRUEBAS	91
8.5	CONCLUSIONES DE PRUEBA	92
9	<u>PLAN DE CAPACITACIÓN Y ENTRENAMIENTO</u>	<u>93</u>
10	<u>PLAN DE IMPLANTACIÓN Y PUESTA EN MARCHA</u>	<u>95</u>
11	<u>RESUMEN ESFUERZO REQUERIDO</u>	<u>96</u>
12	<u>CONCLUSIONES</u>	<u>97</u>
13	<u>BIBLIOGRAFÍA</u>	<u>98</u>
14	<u>ANEXO: PLANIFICACION INICIAL DEL PROYECTO</u>	<u>99</u>
14.1.1	ESTIMACIÓN INICIAL DE TAMAÑO	101
14.1.2	CONTABILIZACIÓN FINAL DEL TAMAÑO DEL SOFTWARE.....	104
15	<u>ANEXO: ESPECIFICACION DE LAS PRUEBAS</u>	<u>105</u>
15.1	PRUEBAS DE UNIDAD	105
15.1.1	<CREAR USUARIOS>.....	105
15.1.2	<REDACTAR MENSAJES>	105
15.1.3	<CREAR CURSOS>	106
15.1.4	<ELIMINAR CURSOS>	106
15.1.5	<CREAR TEMAS>	107
15.1.6	<CREAR SESIONES>	107
15.1.7	<SUBIR ACTIVIDAD>.....	108
15.1.8	<REVISAR ACTIVIDAD>.....	108
16	<u>ANEXO: DICCIONARIO DE DATOS DEL MODELO DE DATOS</u>	<u>109</u>
17	<u>ANEXO: PATRON DE DESARROLLO Y CODIGOS</u>	<u>114</u>
17.1	¿QUÉ ES MVC?	114
17.1.1	UN PEQUEÑO EJEMPLO:.....	114
17.2	PARA COMENZAR PDO	115
17.2.1	LA CONEXIÓN.....	115
17.2.2	PRIMERA CONSULTA.....	115
17.3	COMENZANDO “MVC”	116

Índice Tablas

TABLA 1 SOFTWARE NECESARIO PARA EL DESARROLLO DEL SISTEMA EXEMPLUM.....	24
TABLA 2 SOFTWARE NECESARIO PARA LA OPERACIÓN DEL SISTEMA EXEMPLUM	24
TABLA 3 HARDWARE NECESARIO PARA EL DESARROLLO DEL SISTEMA EXEMPLUM	24
TABLA 4 DESCRIPCIÓN DE ACTOR: ADMINISTRADOR	29
TABLA 5 DESCRIPCIÓN DE ACTOR: PROFESOR.....	30
TABLA 6 DESCRIPCIÓN DE ACTOR: AYUDANTE.....	31
TABLA 7 DESCRIPCIÓN DE ACTOR: ALUMNO.....	31
TABLA 8 RELACIONES CREADAS COMO ENTIDAD EN EL MR A PARTIR DEL MER	64
TABLA 9 RELACIONES NO CREADAS COMO ENTIDAD EN EL MR A PARTIR DEL MER	65
TABLA 10 ESPECIFICACIÓN DE MÓDULOS: VALIDAR USUARIO	89
TABLA 11 ESPECIFICACIÓN DE MÓDULOS: SUBIR ARCHIVO.....	89
TABLA 12 ESPECIFICACIÓN DE MÓDULOS: DESHABILITAR USUARIO.....	89
TABLA 13 ESPECIFICACIÓN DE MÓDULOS: VALIDAR CAMPOS VACÍOS.....	89
TABLA 14 ESPECIFICACIÓN DE MÓDULOS: COMPROBAR CONTRASEÑA	89
TABLA 15 ESPECIFICACIÓN DE MÓDULOS: COMPROBAR CONEXIÓN.....	89
TABLA 16 ESPECIFICACIÓN DE MÓDULOS: CARGAR ARCHIVO	90
TABLA 17 ESPECIFICACIÓN DE MÓDULOS: COMPROBAR CONTRASEÑA.....	90
TABLA 18 ESPECIFICACIÓN DE MÓDULOS: COMPROBAR CONTRASEÑA.....	90
TABLA 19 ESPECIFICACIÓN DE MÓDULOS: BUSCAR PROFESOR	90
TABLA 20 ESPECIFICACIÓN DE MÓDULOS: BUSCAR AYUDANTE.....	90
TABLA 21 ESPECIFICACIÓN DE MÓDULOS: BUSCAR AYUDANTE.....	90
TABLA 23 DETALLE DE TIEMPO REQUERIDO PARA CAPACITAR AL ADMINISTRADOR	93
TABLA 24 DETALLE DE TIEMPO REQUERIDO PARA CAPACITAR A LOS PROFESORES/AYUDANTES.....	94
TABLA 25 DETALLE DE TIEMPO REQUERIDO PARA CAPACITAR A LOS ALUMNOS.	94
TABLA 26 RESUMEN DE ESFUERZO REQUERIDO	96
TABLA 27 PRUEBA 1.....	105
TABLA 28 PRUEBA 2.....	105
TABLA 29 PRUEBA 3.....	106
TABLA 30 PRUEBA 4.....	106
TABLA 31 PRUEBA 5.....	107
TABLA 32 PRUEBA 6.....	107

TABLA 33 PRUEBA 7.....	108
TABLA 34 PRUEBA 8.....	108

Índice Figuras

FIG. 1, ORGANIGRAMA UNIVERSIDAD DEL BÍO-BÍO, PARTE 1.....	11
FIG. 2, ORGANIGRAMA UNIVERSIDAD DEL BÍO-BÍO, PARTE 2.....	12
FIG. 3, NOTACIÓN DE LENGUAJE DE MODELADO UNIFICADO PARA CASOS DE USO.	16
FIG. 4, NOTACIÓN MODELO ENTIDAD RELACIÓN.	17
FIG. 5, CASOS DE USO GENERALES.....	26
FIG. 6, CASOS DE USO ADMINISTRADOR.....	27
FIG. 7, CASOS DE USO PROFESOR/AYUDANTE	28
FIG. 8, CASOS DE USO ALUMNO	29
FIG. 9, MER MENSAJES.....	60
FIG. 10, MER USUARIOS.....	61
FIG. 11, MER CURSOS.....	62
FIG. 12, MER BITÁCORAS.	63
FIG. 13, MR MENSAJES.....	65
FIG. 14, MR USUARIOS.....	66
FIG. 15, MR CURSOS.....	67
FIG. 16, MR BITÁCORAS.	68
FIG. 17, DISEÑO ARQUITECTURA FUNCIONAL USUARIOS.	69
FIG. 18, DISEÑO ARQUITECTURA FUNCIONAL MENSAJES.	70
FIG. 19, DISEÑO ARQUITECTURA FUNCIONAL CURSOS.	71
FIG. 20, INTERFAZ GENERAL DEL SISTEMA EXEMPLUM	72
FIG. 21, MENÚ ADMINISTRADOR	73
FIG. 22, MENÚ PROFESOR Y/O AYUDANTE	73
FIG. 23, MENÚ ALUMNO.....	73
FIG. 24, JERARQUÍA MENÚ ADMINISTRADOR	74
FIG. 25, JERARQUÍA MENÚ PROFESOR Y/O AYUDANTE	74
FIG. 26, JERARQUÍA MENÚ ALUMNO	75
FIG. 27, PANTALLA BIENVENIDA SISTEMA EXEMPLUM	76
FIG. 28, PANEL DE USUARIO.....	76
FIG. 29, MENSAJES	77
FIG. 30, MENSAJE	77
FIG. 31, REDACTAR O RESPONDER MENSAJES.....	78

FIG. 32, PANTALLA BIENVENIDA ADMINISTRADOR.....	78
FIG. 33, ADMINISTRADOR CURSOS.....	79
FIG. 34, CREAR NUEVO CURSO.....	79
FIG. 35, ADMINISTRADOR USUARIOS.....	80
FIG. 36, CREAR NUEVO USUARIO.....	80
FIG. 37, PANTALLA BIENVENIDA PROFESOR.....	81
FIG. 38, PROFESOR CURSO.....	81
FIG. 39, CREAR NUEVO TEMA.....	82
FIG. 40, CREAR NUEVA SESIÓN.....	82
FIG. 41, PROFESOR REVISAR.....	83
FIG. 42, PROFESOR ALUMNOS ACTIVIDAD REVISAR O CORREGIR.....	83
FIG. 43, PROFESOR CALIFICAR ACTIVIDAD.....	84
FIG. 44, PROFESOR INFORMES.....	84
FIG. 45, LISTADO ALUMNOS PARA VER INFORME.....	85
FIG. 46, INFORME ACTIVIDAD ALUMNO.....	85
FIG. 47, PANTALLA BIENVENIDA ALUMNO.....	86
FIG. 48, ALUMNO CURSO.....	86
FIG. 49, ALUMNO SESIÓN.....	87
FIG. 50, ALUMNO REALIZA ACTIVIDAD.....	88
FIG. 51, ALUMNO ACTIVIDADES.....	88

1 INTRODUCCIÓN

Este proyecto de título tiene como objetivo diseñar e implementar un sistema de apoyo del aprendizaje basado en el software, que apoye a los alumnos en las distintas áreas de la programación, que están contenidas en las diferentes asignaturas de ambas carreras Informáticas del Departamento de Sistemas de Información de la Facultad de Ciencias Empresariales (FACE) de la Universidad del Bío-Bío, sede Concepción.

La mayor parte de las asignaturas de programación utilizan un enfoque tradicional centrado en la cátedra con poca integración de los alumnos. En este caso el software a construir apoya un método que se basa en el aprendizaje colaborativo y basado en el ejemplo, con un énfasis en el “aprender haciendo”.

El Departamento de Sistemas de Información de la FACE esta en una constante actualización de métodos y enfoques de aprendizaje para los nuevos alumnos, adaptándose a nuevos paradigmas del aprendizaje, en busca de una mayor y mejor acreditación para ambas carreras de este departamento.

Para apoyar lo anterior, se ha propuesto desarrollar un sistema de información utilizando las nuevas tecnologías, para así poder brindar un mejor apoyo a las diferentes partes involucradas.

Los capítulos desarrollados durante el informe son: **Definición de la empresa**, donde se describe de forma detallada la empresa; **Definición del proyecto**, donde se muestran los objetivos del proyecto y el ambiente de ingeniería de software sobre el cual se desarrolla; **Especificación de requerimientos de software**, donde especifican los diferentes requerimientos necesarios para llevar a cabo el correcto desarrollo del sistema; **Factibilidad**, donde se detalla el estudio realizado para decidir si es factible realizar el proyecto; **Análisis**, donde se realiza el análisis del proyecto mediante casos de uso y modelamiento de datos; **Diseño**, donde se presenta el diseño físico de la Base de Datos, diseño de arquitectura funcional y diseño de interfaz; **Pruebas**, donde se presentan las pruebas realizadas para comprobar su correcto funcionamiento; **Plan de capacitación y entrenamiento**, donde se detalla la capacitación necesaria para que los usuarios finales hagan uso del sistema; **Plan de implantación y puesta en marcha**, donde se describe la forma en que el sistema es implementado y puesto en marcha; y **Resumen de esfuerzo requerido**, donde indica el tiempo necesario para el desarrollo del software.

2 DEFINICION DE LA EMPRESA O INSTITUCIÓN

2.1 Descripción de la empresa

Antecedentes generales de la Empresa

- Nombre: Facultad de Ciencias Empresariales, Universidad del Bío-Bío
- Dirección: Avda. Collao 1202, Concepción, Chile.
- Rubro: Educacional.
- Servicios Ofrecidos: Educación Superior
- Fono: (56-41) 2731200
- Sitio Web: <http://www.ubb.cl>
- Entorno
- Competencia directa: Universidades públicas y privadas.
- Competencia indirecta: Institutos profesionales y centros de formación técnica.

Misión: La Universidad del Bío-Bío es una institución de educación superior, pública, estatal y autónoma, de carácter regional, que se ha propuesto por misión:

- Formar profesionales de excelencia capaces de dar respuesta a los desafíos de futuro, con un modelo educativo cuyo propósito es la formación integral del estudiante a partir de su realidad y sus potencialidades, promoviendo la movilidad social y la realización personal.
- Fomentar la generación de conocimiento avanzado mediante la realización y la integración de actividades de formación de postgrado e investigación fundamental, aplicada y de desarrollo, vinculadas con el sector productivo, orientadas a áreas estratégicas regionales y nacionales.
- Contribuir al desarrollo armónico y sustentable de la Región del Biobío, a través de la aplicación del conocimiento, formación continua y extensión, contribuyendo a la innovación, productividad y competitividad de organizaciones, ampliando el capital cultural de las personas, actuando de manera interactiva con el entorno y procurando la igualdad de oportunidades.

Visión: Ser reconocida a nivel nacional como una Universidad estatal, pública, regional, autónoma, compleja e innovadora con énfasis en la formación de capital humano, vinculada al desarrollo sustentable de la Región del Biobío y que aporta a la sociedad del conocimiento y al desarrollo armónico del país.

Organigrama:

El organigrama de la Universidad del Bío-Bío, no se muestra en su totalidad, por lo extenso de este, por lo cual se representa en 2 figuras, siendo la segunda figura en donde se destaca la facultad donde se enfoca el proyecto.

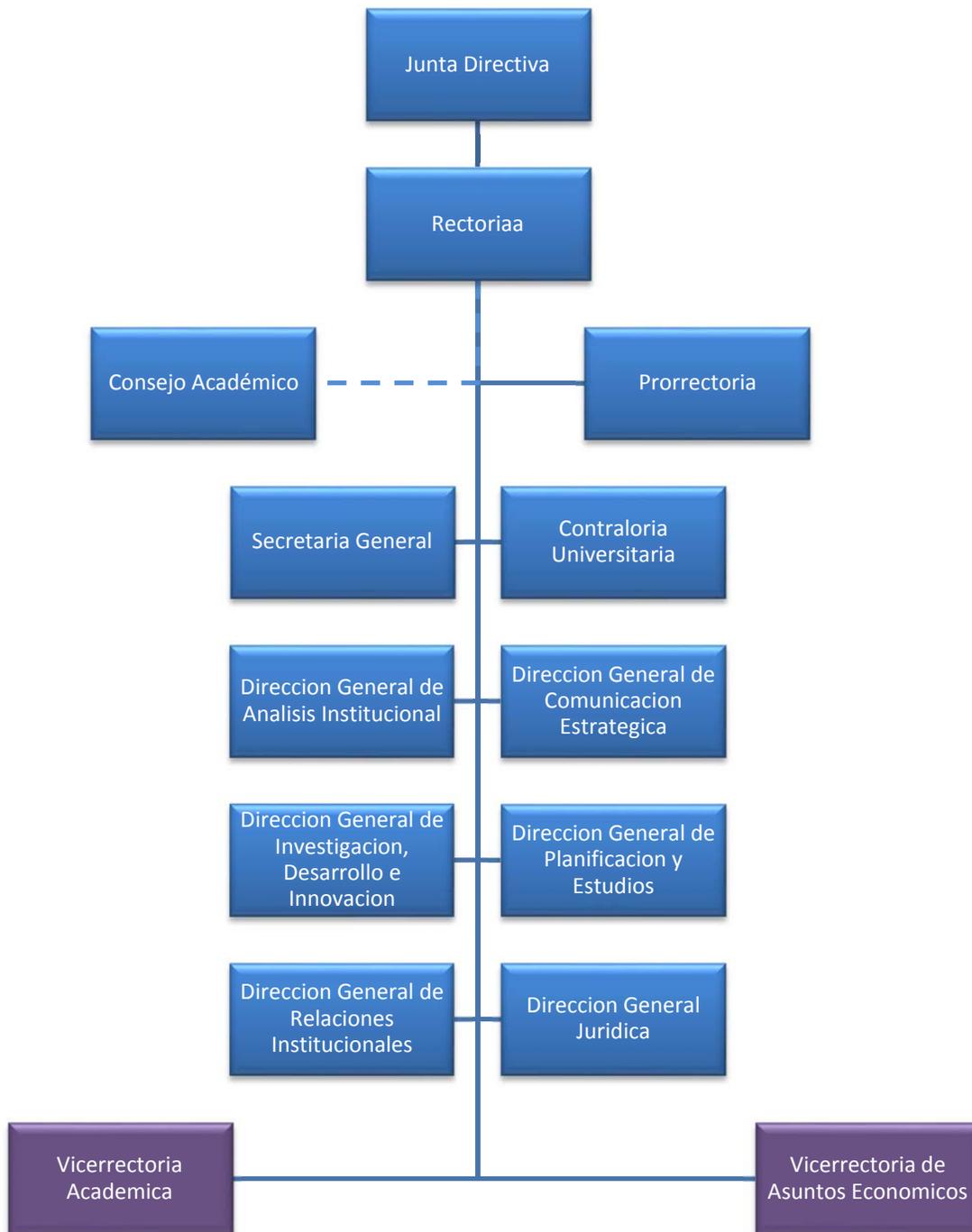


Fig. 1, Organigrama Universidad del Bío-Bío, Parte 1

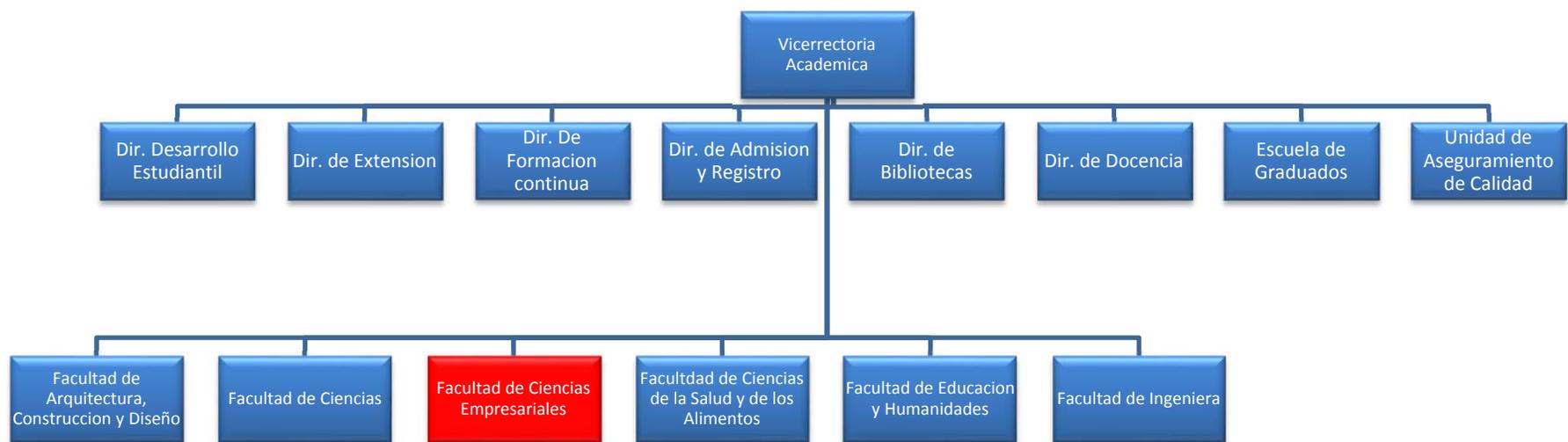


Fig. 2, Organigrama Universidad del Bío-Bío, Parte 2

2.2 Descripción del área de estudio

Este proyecto apoya al área de docencia del Departamento de Sistemas de Información, proporcionando una plataforma que apoya el desarrollo de las actividades docentes de las asignaturas relacionadas con la programación de computadores.

Específicamente, la plataforma construida permite apoyar a asignaturas que buscan que los alumnos aprendan un lenguaje de programación, como es el caso de la asignatura *Introducción a la Programación* de las carreras Ingeniería de Ejecución en Computación e Informática e Ingeniería Civil en Informática así como otras asignaturas relacionadas con la temática que el departamento Sistemas de Información dicta para otras carreras de la Universidad del Bío-Bío.

Los profesores del área de programación son los principales beneficiarios de este proyecto como también los propios alumnos que cursan estas asignaturas.

2.3 Descripción de la problemática

La enseñanza de la programación es un tema de interés en la docencia de la informática, esto está plasmado en diversos trabajos científicos y varios congresos dedicados al tema. Por este motivo, el Departamento de Sistemas de Información de la Facultad de Ciencias Empresariales ha manifestado una importante preocupación por parte de los profesores, para mejorar el proceso de enseñanza y aprendizaje.

Existen pocas herramientas que específicamente apoyen la enseñanza de la programación, por ejemplo DFDSmart, PSint y Laborac, las cuales no permiten desarrollar al máximo el aprendizaje de los alumnos, por lo cual es una preocupación mejorar el nivel de aprendizaje de la programación en ambas carreras informáticas de la FACE.

3 DEFINICIÓN PROYECTO

Actualmente las metodologías de enseñanza de la programación utilizadas principalmente en el Departamento de sistemas de Información y en general en la Universidad del Bío-Bío se basan en la cátedra y en algunos casos en talleres de resolución de problemas principalmente individuales. Este proyecto propone la construcción de una plataforma web denominada “Sistema de apoyo a la enseñanza de la programación de computadores basada en el ejemplo en entorno colaborativo EXEMPLUM” que facilita la tarea del profesor frente al aprendizaje de un lenguaje de programación. Este proyecto implementa un modelo de enseñanza de la programación basado en tres aspectos:

- El aprendizaje basado en el ejemplo: Permitiendo a los estudiantes construir programas utilizando como base programas de ejemplo preparados por el profesor. Estos ejemplos son construidos y/o mejorados por los estudiantes intentando conseguir un objetivo de aprendizaje relacionado con alguna competencia en la construcción de programas de computadores. El aprendizaje se consigue usando el paradigma de “Aprender Haciendo” (Schank, Berman, & Macpherson, 1999).
- Simplificación de la Secuencia de Condiciones: Método proveniente de la Teoría de Elaboración propuesta por Charles Reigeluth (Reigeluth, 1999). Este método propone que el aprendizaje puede lograrse de mejor forma si los estudiantes ejecutan tareas en una secuencia que va aumentando la complejidad de la tarea. En este caso particular, las tareas corresponden a la construcción de programas que aumentan la complejidad de la tarea exigiendo cada vez un mayor nivel de logro de la competencia.
- Colaboración: Los estudiantes no solo analizan y/o mejoran versiones de programas construidas por el profesor sino que además aquellos construidos por estudiantes del mismo grupo más aventajados, resultando en programas construidos implícitamente mediante la colaboración de un grupo de estudiantes. La utilidad de utilizar elementos del aprendizaje colaborativo es que el alumno pueda desarrollar habilidades y capacidades como la iniciativa, la toma de decisiones, el espíritu autocrítico, la creatividad, la flexibilidad, investigación y descubrimiento. También que el profesor logre aumentar la motivación intrínseca por el aprendizaje y que al alumno le enseñe a manejar el “feedback” o retroalimentación como herramienta de refuerzo, apoyo y desarrollo. La principal técnica utilizada en este modelo es la llamada “Learning Together” o “Aprendiendo Juntos” de Johnson y Johnson¹, los elementos esenciales que la componen son los siguientes:
 - La recompensa del grupo: Las recompensas pueden ser de distintas índoles adaptadas a la edad y a la situación, pero substancialmente expresan un reconocimiento público de los resultados conseguidos. En este proyecto específicamente “El mejor alumno es quien recibió el reconocimiento de los compañeros”.
 - La responsabilidad individual: El éxito del grupo depende del nivel de aprendizaje que cada cual está dispuesta alcanzar y realizar. Todos son responsables en este empeño. No de un modo individualista, sino colaborativo: Ayudándose recíprocamente, explicándose aquello no bien comprendido, controlando el propio

¹ JOHNSON, D. W.; JOHNSON, R. Cooperation and competition. Theory and research. Hillsdale, NJ: Addison-Wesley, 1990.

aprendizaje a través de preguntas, asegurándose de que está preparado para afrontar una prueba individual sobre cuanto ha aprendido, etc.

- La misma oportunidad de éxito: La condición cooperativa asegura que todos los miembros tengan la posibilidad de conseguir el éxito mejorando resultados anteriores.

Este proyecto implementa el modelo antes descrito permitiendo:

- A los profesores, conseguir los objetivos de aprendizaje mediante la preparación de ejemplos de programas (muchas veces simples esqueletos de futuros programas) que guían a los estudiantes en la construcción de programas finales totalmente terminados. Permite además crear la secuencia de programas que aumentan cada vez e nivel de complejidad.
- A los estudiantes, conocer, analizar y seguir el ejemplo de programas ya construidos permitiendo un avance más rápido en el aprendizaje de la programación, creando una ambiente que aumenta la motivación de los estudiantes.

Los estudiantes serán grupos heterogéneos (curso) y trabajaran en forma individual sobre una determinada actividad, una vez finalizada está el profesor la calificará y los alumnos podrán comparar sus resultados con las de los compañeros. Lo cual permitirá que los mejores alumnos sean guías en el aprendizaje gracias a sus respectivos trabajos.

Para realizar lo anterior el profesor será el encargado de analizar los temas y tópicos que conforman las unidades de aprendizaje, con el fin de que cada nueva actividad pueda encadenarse con el conocimiento adquirido previamente y ofrecer variedad de tareas de aprendizaje auténtica.

3.1 Objetivos del proyecto

Objetivo general.-

Diseñar y Desarrollar una plataforma de apoyo al aprendizaje de la programación, que utiliza un enfoque basado en ejemplos en entorno colaborativo.

Objetivos específicos.-

- Proponer los componentes del software que permitan realizar la enseñanza de la programación basada en ejemplos, usando como base un sistema desarrollado anteriormente.
- Diseñar sistema web que favorezca el aprendizaje de la programación, utilizando técnicas de usabilidad, tecnologías de la web semántica y estándares e-learning (por ejemplo el concepto de O.A. "Objetos de aprendizaje") para representar los contenidos.
- Construir el sistema basándose en un patrón de arquitectura de software (Modelo Vista Controlador "MVC" orientando a Objetos en PHP), permitiendo la realización de código organizado, adaptable, un mantenimiento sencillo y un fácil entendimiento para futuros desarrollos.
- Conocer las percepciones de los usuarios/estudiantes frente al uso de la plataforma.

3.2 Ambiente de Ingeniería de Software

Métodos de desarrollo:

Para el desarrollo del proyecto se utilizará la Metodología Incremental basada en los modelos evolutivos del software. La metodología involucra las etapas del modelo secuencial aplicadas repetitivamente, con la filosofía interactiva de construcción de versiones y entregando pequeñas funcionalidades del proyecto que van evolucionando durante el desarrollo de éste. Entonces por cada incremento que se presente se considerará una nueva versión, la que estará disponible al usuario para realizar pruebas de usabilidad verificando los requerimientos y permitiendo con la información obtenida retroalimentar el proyecto.

Técnicas y Notaciones:

- **Casos de Uso**

La notación ocupada para la construcción de Casos de Usos es “Lenguaje de modelado unificado (LUM o UML, por sus siglas en inglés, Unified Modeling Language). La Fig. 3, esquematiza la simbología utilizada para describir los Casos de Uso.

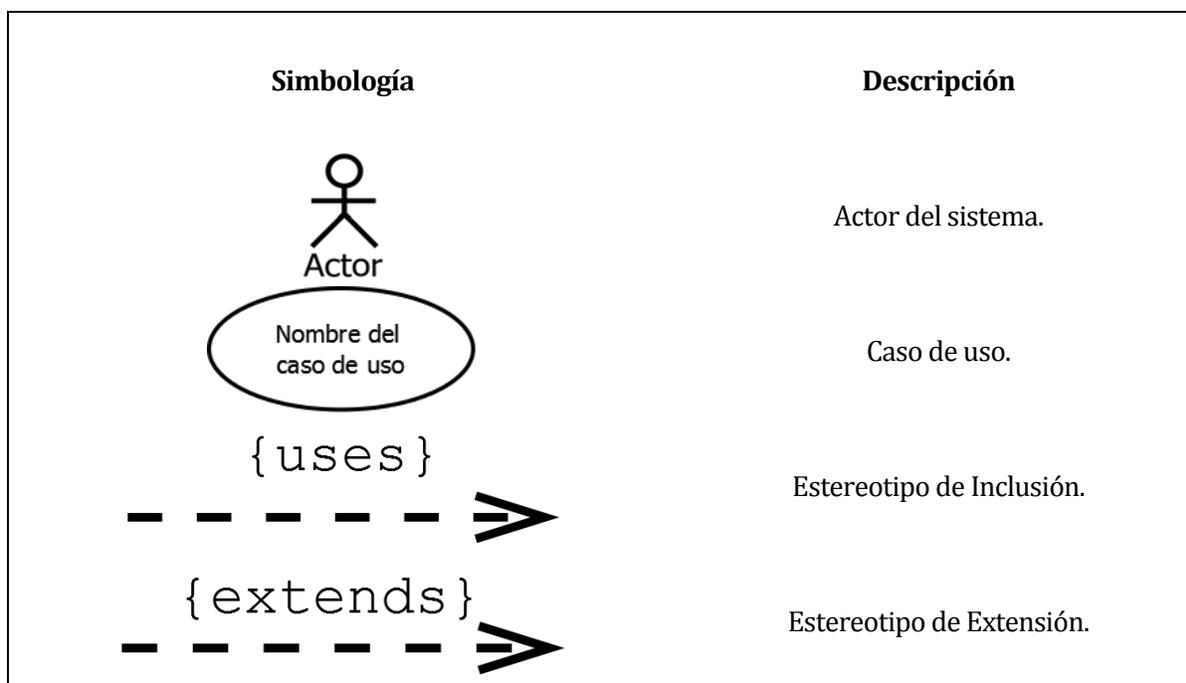


Fig. 3, Notación de lenguaje de modelado unificado para casos de uso.

○ **Modelo Entidad-Relación (MER).**

Para la confección del modelo entidad-relación se utiliza la notación de **Chen**. La Fig. 4, esquematiza la notación ocupada para el MER de **EXEMPLUM**.

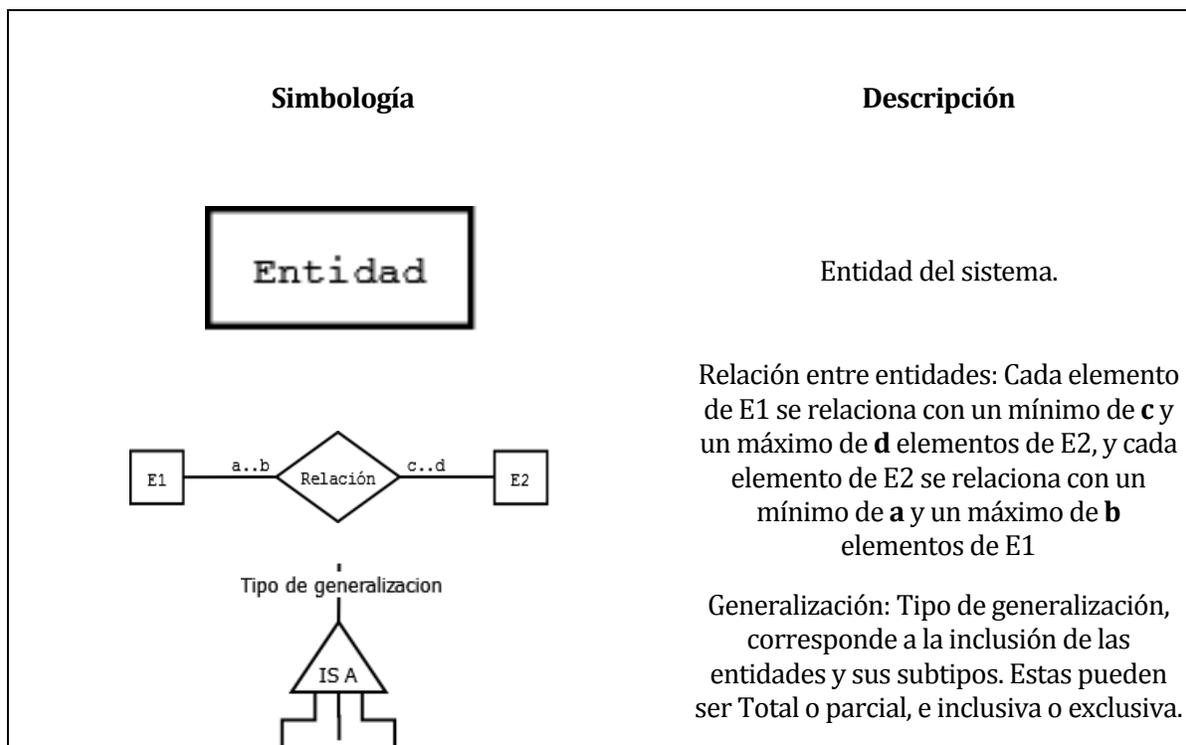


Fig. 4, Notación Modelo Entidad Relación.

Estándares de documentación, producto o proceso:

Los estándares usados para la elaboración del informe proyecto de título son:

- Plantilla de documentación del proyecto de desarrollo de software, 3 de noviembre 2011, Universidad del Bío-Bío, Concepción-Chile.

Herramientas de apoyo al desarrollo del software que son utilizadas:

- **GIT:** sistema de control de versiones distribuido de código abierto, diseñado para manejar proyectos pequeños a grandes con rapidez y eficiencia. Es utilizado para facilitar la integración del código desarrollado por los integrantes del proyecto.
- **Netbeans:** IDE de desarrollo en programación, de código abierto, utilizado para apoyar el desarrollo de PHP y lenguajes webs (JavaScript, CSS, HTML).
- **Pencil:** Utilizado para realizar un pre-diseño del sistema.
- **DIA:** Herramienta para modelamiento de datos.
- Navegadores Web: utilizados para visualizar el aspecto del sistema.

3.3 Definiciones, Siglas y Abreviaciones

Este ítem se incluye la definición de las siglas, abreviaciones, conceptos técnicos o de negocio que son necesarios para el buen entendimiento de este documento:

- **Alumno:** son los estudiantes de la institución educacional.
- **Profesor:** son los docentes de la institución educacional.
- **Ayudante:** son estudiantes de la institución que apoya los contenidos teóricos vistos en clases de manera práctica, por ende, poseen un nivel de conocimiento suficiente para ayudar a los alumnos.
- **Administrador:** el administrador gestiona y/u organiza los recursos y los accesos del sistema, otorgando privilegios a los diferentes usuarios.
- **Curso:** Conjunto de contenidos y actividades, que buscan el logro de objetivos de aprendizaje por parte del alumno.
- **Tema:** Concepto clave a tratar dentro del curso. Es sinónimo de Tópico
- **Sesión:** conjunto de actividades que son realizadas por alumnos y profesor en un tiempo determinado
- **Versión inicial o Ejemplo:** programa inicial o esquema general de un programa que soluciona un problema en estudio
- **Actividad:** enunciado para que el alumno resuelva.
- **Versión:** respuesta a la actividad, en este caso un código ejecutable en un archivo.
- **Calificación:** evaluación realizada sobre una versión construida por un alumno mediante la utilización de símbolos y en una escala definida.
- **CRUD:** acrónimo de crear, leer, actualizar y borrar datos de una tabla de la base de datos.
- **CU:** Casos de uso.
- **Eliminación lógica:** atributo que indica la no vigencia del registro y que permite la no eliminación física del mismo.

4 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

4.1 Alcances

Las principales funcionalidades que el sistema **EXEMPLUM** realiza:

- Permite el ingreso a cuentas con privilegios mediante usuario y contraseña.
- Asigna alumnos en asignaturas determinadas.
- Asigna profesores a diversas asignaturas.
- Asigna ayudantes a diversas asignaturas.
- Permite que el alumno escriba versiones de actividades que realiza.
- Almacena en un archivo el código de una actividad realizada y la guarda en el servidor.
- Almacena los datos relevantes de profesores, ayudantes, alumnos y administradores.
- Almacena las calificaciones que los profesores otorgan a la actividad de un alumno en la plataforma.
- Genera informe de actividad de los alumnos.
- Generar archivos XML, para posterior procesamiento.

4.2 Objetivo del software

Objetivo General:

Facilitar el aprendizaje del alumno gestionando sus programas (Versiones) a través de una plataforma web, la que apoyará la entrega de conocimientos obtenidos a través del profesor, utilizando una metodología de enseñanza basada en ejemplos, permitiendo al alumno desarrollar mayores actitudes al practicar lo aprendido teóricamente.

Objetivos Específicos:

1. Permitirá a los alumnos desarrollar las versiones para las actividades en el sistema.
2. Permitirá a los alumnos tener acceso a las calificaciones que el profesor otorgue a las actividades que realizo.
3. Permitirá a los profesores y/o ayudantes calificar las actividades de los alumnos.
4. Permitirá al profesor obtener informes de las diferentes actividades de los alumnos de un curso respectivamente.
5. Comunicará a todos los usuarios a través de una casilla de mensajes.

4.3 Descripción Global del Producto

4.3.1 Interfaz de usuario

La interfaz será de una forma accesible para que cualquier usuario pueda ingresar y visualizar el contenido del sistema. Algunas características de la Interfaz de usuario son:

- Como el sistema será desarrollado para la Universidad del Bío-Bío se utilizaran colores que representen a la institución, en este caso se partirá utilizando colores como Azul, Blanco y sus combinaciones.
- El logo junto al nombre de la universidad del Bío-Bío se ubicara en la parte superior del sistema (cabecera).
- El enlace al panel de control de cada usuario se encontrara en la parte superior derecha de la pantalla junto a su nombre, para que esta información sea menos invasiva y pueda estar siempre a la mano.
- El menú principal estará a la izquierda de la pantalla para una mayor visibilidad y usabilidad del usuario.

4.3.2 Interfaz De Hardware

El sistema esta considerado para una configuración estándar para un computador de escritorio o computador portátil.

El hardware requerido para el correcto funcionamiento del sistema se presenta a continuación:

- Pantalla con una resolución de 1024 x 768 pixeles o superior.
- Teclado.
- Mouse.
- Procesador de 1.0 GHz o superior.
- 256 Mb de RAM o superior.
- Conexión a internet.

4.3.3 Interfaces de comunicación

La comunicación con sistemas existente es a través de la base de datos que posee la empresa.

- Protocolo Hypertext Transfer Protocol (HTTP).

4.4 Requerimientos Específicos

4.4.1 Requerimientos Funcionales del sistema

A continuación se detalla la lista de los requerimientos funcionales específicos del sistema. Para lo cual hemos clasificado cuatro actores relacionados; Administrador, Alumno, Profesor y/o Ayudante.

Requerimientos funcionales comunes para todos los usuarios:

Id	Nombre	Descripción
001	Ingresar al sistema	Los usuarios se identifican en el sistema mediante usuario y contraseña, si estos son validos, son autorizados para entrar al sistema según sus privilegios.
002	Actualizar datos	Los usuarios pueden modificar su información personal, en su panel de usuario.
003	Redactar mensajes	Los usuarios pueden redactar mensajes a cualquier usuario del sistema.
004	Recibir mensajes	Los usuarios pueden recibir mensajes de cualquier usuario del sistema.
005	Responder mensajes	Los usuarios pueden responder mensajes a cualquier usuario del sistema.
006	Salir del sistema	Los usuarios quedan desautorizados en el sistema.

Requerimientos funcionales del usuario "Administrador":

Id	Nombre	Descripción
001	Crear Curso	Ingresar un curso en el sistema, asignar profesor, ayudante y alumnos.
002	Eliminar Curso	Eliminar lógicamente un curso en el sistema.
003	Modificar Curso	Realizar algún cambio en los datos del curso.
004	Agrega usuario	Ingresar un nuevo usuario al sistema.
005	Eliminar usuario	Eliminar lógicamente a un usuario.
006	Modificar usuario	Realizar algún cambio en los datos del usuario.

Requerimientos funcionales del usuario "Profesor y Ayudante": el usuario ayudante puede estar presente o no, depende del curso, y los permisos son los mismos que los del profesor, siendo este el que lo puede limitar.

Id	Nombre	Descripción
001	Seleccionar Curso	Seleccionar un curso del cual es profesor, para hacer modificaciones.
002	Crear Tema	Crear un nuevo tema dentro de uno de sus cursos.
003	Eliminar Tema	Eliminar lógicamente el tema que considere.
004	Modificar Tema	Realizar modificaciones de datos a los temas de sus cursos.
005	Crear Sesión	Crea una nueva sesión de trabajo, dentro de un tema específico y agrega todas las actividades que el usuario debe realizar.
006	Eliminar Sesión	Eliminar lógicamente la sesión que considere.
007	Modificar Sesión	Realizar modificaciones de datos a las sesiones.
008	Finalizar Sesión	Cerrar el acceso a las actividades de la sesión, ya que en un tiempo determinado los alumnos deben realizarlas con todas.
009	Revisar actividades	Ver que estén todos los alumnos con las actividades realizadas.
010	Calificar actividades	Calificar mediante una nota simbólica las actividades subidas al sistema.
011	Informes de actividad alumnos	Generar un informe con todas las calificaciones de las actividades realizadas por los alumnos.

Requerimientos funcionales del usuario "Alumno":

Id	Nombre	Descripción
001	Ingresar Curso	Seleccionar un curso en el cual está inscrito.
002	Ingresar Tema	Seleccionar un tema dentro del curso al cual pertenece.
003	Ingresar Sesión	Acceder a una sesión de trabajo dentro de un determinado curso.
004	Realizar actividades	Realizar actividades solicitadas dentro de la sesión.
005	Subir actividades	Subir actividad realizada a la plataforma, para obtener una calificación.
006	Ver Actividades Anteriores	Visualizar listado de trabajos anteriores, a modo de ayuda.

4.4.2 Interfaces externas de entrada

La información expuesta a continuación, corresponde a los datos que ingresaran al sistema

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem
IE_01	Ingresar al sistema	Rut, Contraseña
IE_02	Actualizar datos cuenta usuario	Dirección (Calle, Numero, Comuna, Ciudad), Fecha nacimiento, Teléfono (fijo, móvil), Correo electrónico.
IE_03	Redactar mensajes	Destinatario(s), asunto, mensaje.
IE_04	Responder mensajes	Mensaje.
IE_05	Crear curso	Nombre curso, Descripción, Profesor, Ayudante, Alumnos.
IE_06	Modificar curso	Nombre curso, Descripción, Profesor, Ayudante, Alumnos.
IE_07	Agregar usuario	Nombre usuario, Rut, Contraseña, Dirección (Calle, Numero, Comuna, Ciudad), Fecha nacimiento, Teléfono (fijo, móvil), Correo electrónico. Archivo plantilla con datos de muchos usuarios.
IE_08	Modificar usuario	Nombre usuario, Contraseña. Dirección (Calle, Numero, Comuna, Ciudad), Fecha nacimiento, Teléfono (fijo, móvil), Correo electrónico. Archivo plantilla con datos de muchos usuarios.
IE_09	Crear tema	Nombre tema, Descripción.
IE_10	Modificar tema	Nombre tema, Descripción.
IE_11	Crear sesión	Nombre sesión, Descripción, Versión inicial o Ejemplo, Fecha de inicio, Fecha de termino, Hora de inicio, Hora de termino, Actividades a realizar (descripción u objetivo a realizar por cada uno).
IE_12	Modificar sesión	Nombre sesión, Descripción, Versión inicial o Ejemplo, Fecha de inicio, Fecha de termino, Hora de inicio, Hora de termino, Actividades a realizar (descripción u objetivo a realizar por cada uno).
IE_13	Calificar Trabajos	Nota simbólica (puntuación).
IE_14	Realizar trabajo	Código fuente.
IE_15	Subir actividad	Subir archivo del código fuente

4.4.3 Interfaces externas de Salida

La información expuesta a continuación, corresponde a los datos que retornara al sistema.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem	Medio Salida
---------------	------------------	-------------------------------------	--------------

IS_01	Informe actividad alumnos	Rut – nombre – fecha – hora - actividad	Gráfico
IS_02	Archivo XML	Trabajo – alumno – versión	XML

4.4.4 Atributos del producto

Tomando como referencia al modelo ISO/IEC 9126², se indican los atributos de calidad del sistema **EXEMPLUM**. Además se han considerado las heurísticas de usabilidad de Jacob Nielsen³.

- **Funcionalidad:** El sistema mantiene un control de acceso a la funcionalidad a través de autenticación, establecida según los perfiles definidos para los usuarios del sistema (Administrador, Profesor, Ayudante y Alumno).
- **Usabilidad:** Feedback, que siempre se muestre el estado de la aplicación al usuario.
- **Portabilidad:** Este sistema solo necesita de un navegador web con conexión a internet, tener JavaScript habilitado.
- **Fiabilidad:** El sistema no permitirá la pérdida del código escrito que no se ha guardado.
- **Eficiencia:** El sistema debe garantizar un tiempo de carga y de respuesta inferior a 3 segundos, considerando una conexión de red de velocidad estándar de 2 Mbps y 15 usuarios conectados simultáneamente.
- **Flexibilidad:** El sistema debe ser fácil de agregar nuevas características, para esto, se debe presentar una documentación adecuada.

² <http://www.issco.unige.ch/en/research/projects/ewg96/node13.html> - The ISO 9126 Standard

³ <http://www.guiaweb.gob.cl/guia-v2/capitulos/05/aseguramiento.htm> - Aseguramiento de la Calidad

5 FACTIBILIDAD

En este capítulo se especifica el equipamiento y software computacional que son necesarios para el desarrollo e implementación del sistema **EXEMPLUM**. Además se describen los impactos y beneficios que produce la implementación de éstos en los procesos de la empresa.

5.1 Factibilidad técnica.

Software necesario para el desarrollo del sistema EXEMPLUM

<i>Nombre</i>	<i>Versión</i>	<i>Licencia</i>
PHP	5.4.X	PHP License v3.01
MySQL	5.5.X	General Public License (GPLv2)
Netbeans	7.2	Common Development and Distribution License (CDDL) v1.0; General Public License (GPLv2)
GIT	1.7	General Public License (GPLv2)
Pencil Project	1.3	General Public License (GPLv2)
Apache	2.4.X	Apache License v2
MySQL Workbench	5.2	General Public License (GPLv2)

Tabla 1 Software necesario para el desarrollo del sistema EXEMPLUM

Software necesario para la operación del sistema EXEMPLUM.

<i>Nombre</i>	<i>Versión</i>	<i>Licencia</i>
Apache	2.4.X	Apache License v2
PHP	5.4.X	PHP License v3.01
MySQL	5.5.X	General Public License (GPLv2)

Tabla 2 Software necesario para la operación del sistema EXEMPLUM

Hardware necesario para el desarrollo del sistema EXEMPLUM

<i>Hardware</i>	<i>Capacidad</i>
Pantalla	Resolución mínima 1024x768
CPU	Procesador 1,6Ghz.
Memoria RAM	1 Gb.
Accesorios (Mouse, ratón, teclado)	SI

Tabla 3 Hardware necesario para el desarrollo del sistema EXEMPLUM

5.2 Factibilidad operativa.

Con la implementación del sistema **EXEMPLUM** los procesos de la institución educacional serán beneficiados en:

Impactos positivos:

- Mayor contacto profesor-alumno.
- Rapidez en la entrega de información del curso además de rápida actualización.
- Mejorar la velocidad de aprendizaje de los alumnos.
- Mayor recurso para los procesos de acreditación.

5.3 Factibilidad económica.

La realización de este sistema no conlleva gastos para la facultad o la universidad, por motivos de que se utilizan, para el desarrollo, equipos propios, y para la implementación del sistema, equipos existentes en la universidad. Además, todo el software involucrado en el desarrollo e implementación del sistema es libre, por ende no se necesita comprar ninguna licencia. También, es desarrollado bajo el concepto de Proyecto de título, por ende, tendrá un carácter académico.

Beneficios:

Económicamente, es difícil monetizar este sistema, ya que los profesores deben entregar la enseñanza de la programación mediante este método o mediante otros, por ende los beneficios del sistema son más bien intangibles, ya que se puede observar un mejor apoyo por parte del docente para con los alumnos y su aprendizaje de programación, además de analizar el avance y la adquisición de conocimientos por parte de los alumnos.

5.4 Conclusión de la factibilidad

De acuerdo con el análisis realizado previamente se puede concluir que la FACE y el Depto. de Sistemas de Información, cumple con los requerimientos técnicos, operacionales y económicos para el desarrollo e implementación del proyecto, por lo que es factible realizarlo.

6 ANÁLISIS

En este capítulo se presentan los diagramas de caso de uso y su descripción para el sistema **EXEMPLUM**

6.1 Diagrama de casos de uso (CU)

Para aclarar de mejor manera grafica los CU, los distribuimos coincidiendo por cosas en común.

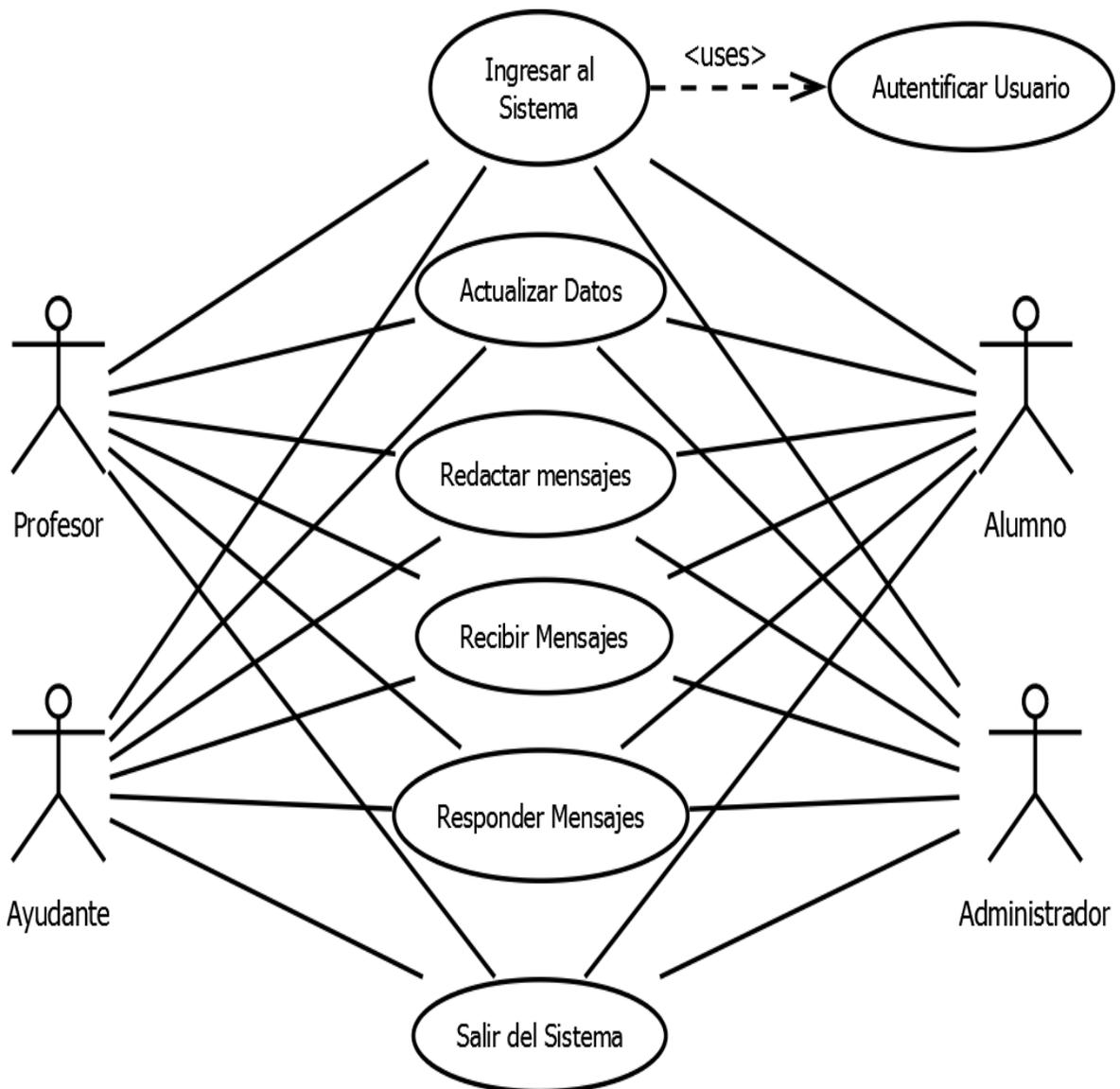


Fig. 5, Casos de Uso GENERALES

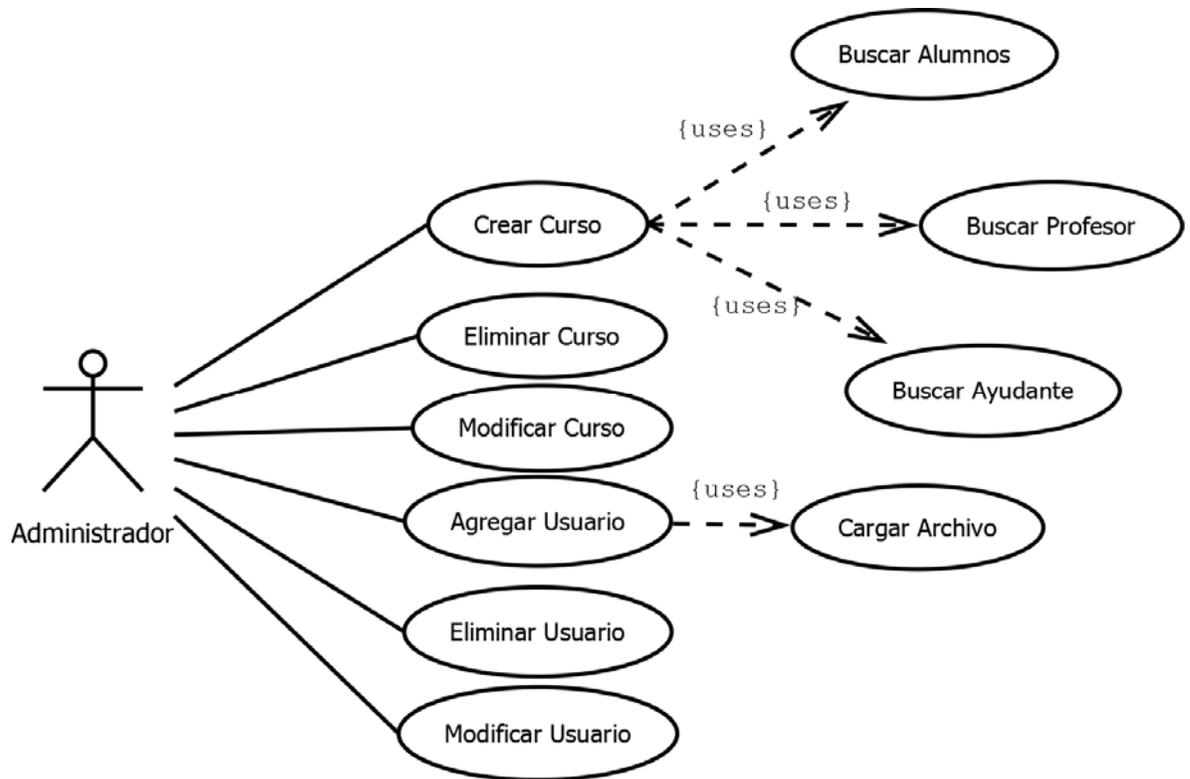


Fig. 6, Casos de Uso Administrador

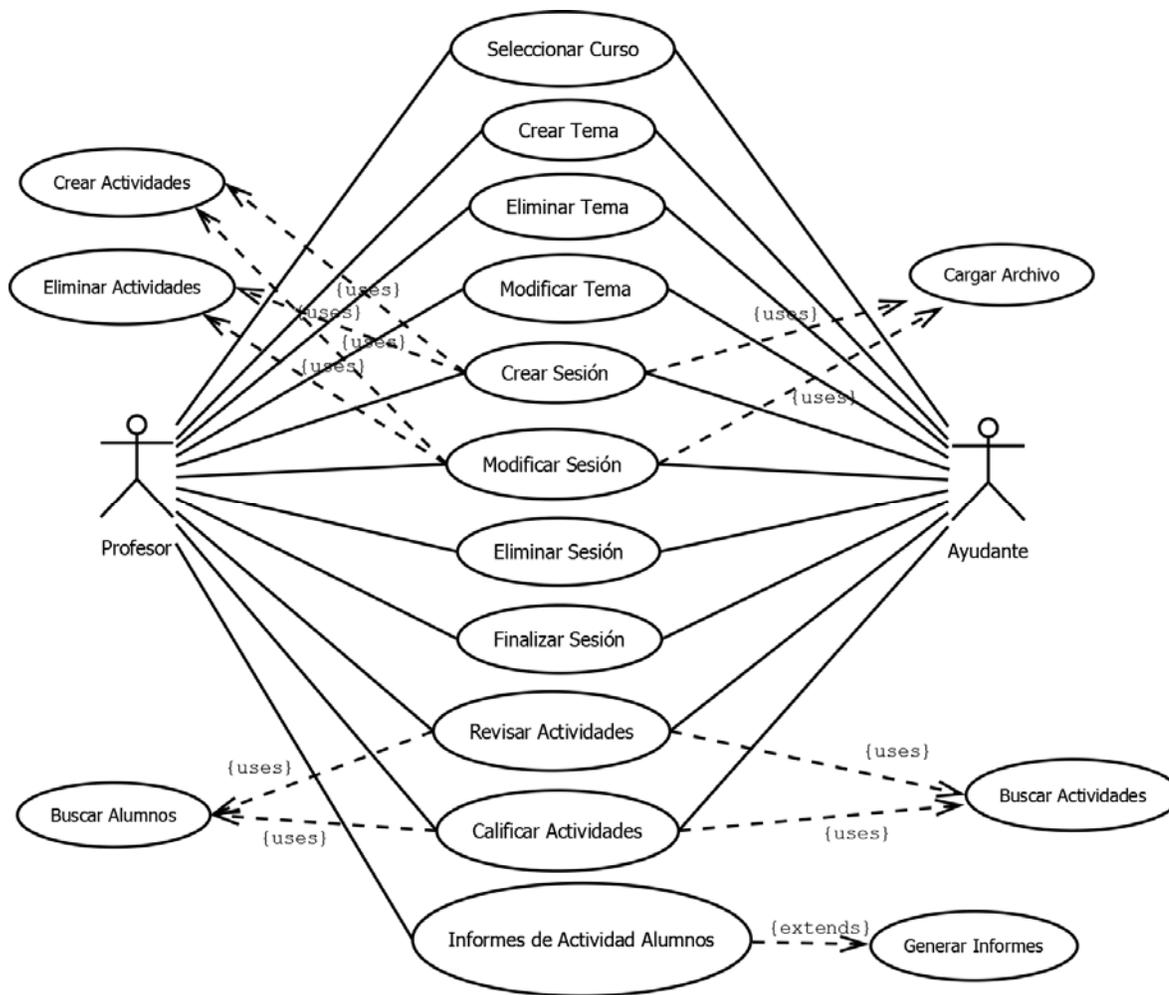


Fig. 7, Casos de Uso Profesor/Ayudante

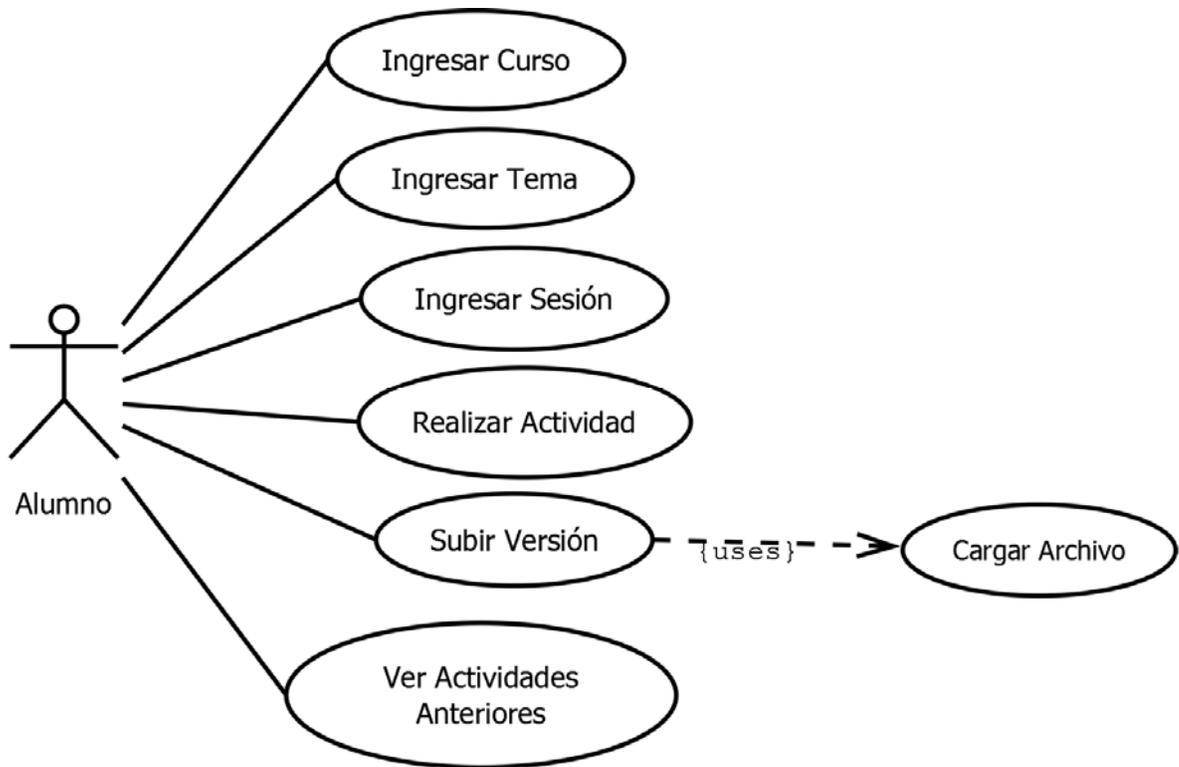


Fig. 8, Casos de Uso Alumno

6.1.1 Actores

Actor	Administrador.
Función dentro de la Institución.	Responsable de registrar usuarios alumnos, profesores, ayudantes y otros administradores, crear los cursos y asignar: (alumnos, profesores y ayudantes).
Nivel de conocimientos técnicos requerido	Manejo básico de navegador web: cargar sitios web, manejo de controles para navegar en la web (atrás, actualizar, detener carga sitio web).
Privilegios en el sistema	<ol style="list-style-type: none"> 1. Crear nuevos usuarios. 2. Modificar usuarios. 3. Eliminar usuarios. 4. Crear cursos. 5. Modificar curso. 6. Eliminar curso. 7. Asignar usuarios al curso. 8. Redactar Mensajes. 9. Recibir mensajes.

Tabla 4 Descripción de Actor: Administrador

Actor	Profesor.
Función dentro de la Institución.	Responsable de organizar el curso, ver el orden de los temas, el tiempo que lleva cada uno y la cantidad de sesiones a trabajar con su respectiva duración.
Nivel de conocimientos técnicos requerido	Manejo básico de navegador web: cargar sitios web, manejo de controles para navegar en la web (atrás, actualizar, detener carga sitio web).
Privilegios en el sistema	<ol style="list-style-type: none"> 1. Seleccionar un curso del cual es profesor. 2. Crear temas dentro de sus cursos. 3. Modificar temas dentro de sus cursos. 4. Eliminar temas dentro de sus cursos. 5. Crear sesiones dentro de un tema. 6. Modificar sesiones dentro de un tema. 7. Eliminar sesiones dentro de un tema. 8. Finalizar las sesiones dentro de un tema. 9. Revisar las actividades de los alumnos. 10. Calificar las actividades de los alumnos. 11. Generar informe de actividad de los alumnos. 12. Redactar mensajes. 13. Recibir mensajes.

Tabla 5 Descripción de Actor: Profesor

Actor	Ayudante.
Función dentro de la Institución.	Estudiantes de la institución Responsable de apoyar en los contenidos teóricos vistos en clases de manera práctica. Ya que poseen un cierto nivel de conocimiento en las materias que apoyan.
Nivel de conocimientos técnicos requerido	Manejo básico de navegador web: cargar sitios web, manejo de controles para navegar en la web (atrás, actualizar, detener carga sitio web).
Privilegios en el sistema	<p>Estos privilegios son los mismos que el profesor a diferencia, que pueden ser limitados por el profesor o el encargado del curso (administrador).</p> <ol style="list-style-type: none"> 1. Seleccionar un curso del cual es profesor. 2. Crear temas dentro de sus cursos. 3. Modificar temas dentro de sus cursos. 4. Eliminar temas dentro de sus cursos. 5. Crear sesiones dentro de un tema. 6. Modificar sesiones dentro de un tema. 7. Eliminar sesiones dentro de un tema. 8. Finalizar las sesiones dentro de un tema. 9. Revisar las actividades de los alumnos. 10. Calificar las actividades de los alumnos. 11. Generar informe de actividad de los alumnos. 12. Redactar mensajes.

13. Recibir mensajes.

Tabla 6 Descripción de Actor: Ayudante

Actor	Alumno.
Función dentro de la Institución.	Son los estudiantes de la institución educacional, siendo los responsables de realizar las sesiones de trabajo y subir las actividades respectivas.
Nivel de conocimientos técnicos requerido	Manejo básico de navegador web: cargar sitios web, manejo de controles para navegar en la web (atrás, actualizar, detener carga sitio web).
Privilegios en el sistema	<ol style="list-style-type: none"> 1. Ingresar a sus cursos. 2. Ingresar a temas de sus cursos. 3. Ingresar a la sesión dentro de los temas de sus cursos. 4. Realizar la actividad. 5. Subir actividad. 6. Ver actividades relacionadas cuando uno realiza una nueva actividad. 7. Redactar mensajes. 8. Recibir mensajes.

Tabla 7 Descripción de Actor: Alumno

6.1.2 Casos de Uso y descripción

Casos de Uso Generales.

6.1.2.1. Caso de uso "Ingresar al Sistema"

Actores	<i>Administrador, Profesor, Ayudante y Alumno</i>
Descripción	El usuario entra los datos de Rut y Contraseña en el sistema, para que sean procesados.

6.1.2.2. Caso de uso "Autenticar Usuario"

	<i>Invocado desde el caso de uso "Ingresar al Sistema"</i>
Descripción	Funcionalidad del sistema que valida los datos ingresados (Rut y Contraseña) y devuelve los permisos a las funcionalidades correspondientes.

6.1.2.3. Caso de uso "Actualizar Datos"

Actores	<i>Administrador, Profesor, Ayudante y Alumno</i>
Descripción	El usuario puede modificar sus datos personales cuando estimen conveniente.

6.1.2.4. Caso de uso "Redactar mensajes"

Actores	<i>Administrador, Profesor, Ayudante y Alumno</i>
Descripción	El usuario envía un mensaje a otro usuario del sistema.

6.1.2.5. Caso de uso "Recibir mensajes"

Actores	<i>Administrador, Profesor, Ayudante y Alumno</i>
Descripción	El usuario recibe un mensaje de otro usuario.

6.1.2.6. Caso de uso "Responder mensajes"

Actores	<i>Administrador, Profesor, Ayudante y Alumno</i>
Descripción	El usuario envía una respuesta a un mensaje anteriormente recibido de otro usuario.

6.1.2.7. Caso de uso "Salir del Sistema"

Actores	<i>Administrador, Profesor, Ayudante y Alumno</i>
Descripción	El usuario acciona la funcionalidad para retirarse del sistema, esto lleva a que se cierre la sesión y sus datos queden registrados.

Casos de Uso Administrador.

6.1.2.8. Caso de uso "Crear Curso"

Actores	<i>Administrador</i>
Descripción	El usuario puede crear un curso asignando la cantidad de usuarios alumnos, profesores y ayudantes respectivamente.

6.1.2.9. Caso de uso "Buscar Alumnos"

	<i>Invocado desde el caso de uso "Crear Curso"</i>
Descripción	Funcionalidad del sistema que busca a los Alumnos, en este caso necesarios para asignarlos a un curso.

6.1.2.10. Caso de uso "buscar Ayudante"

	<i>Invocado desde el caso de uso "Crear Curso"</i>
Descripción	Funcionalidad del sistema que busca a los Ayudantes, en este caso necesarios para asignarlos a un curso.

6.1.2.11. Caso de uso "Buscar Profesor"

	<i>Invocado desde el caso de uso "Crear Curso"</i>
Descripción	Funcionalidad del sistema que busca a los Profesor, en este caso necesarios para asignarlos a un curso.

6.1.2.12. Caso de uso "Eliminar Curso"

Actores	<i>Administrador</i>
Descripción	El usuario Administrador puede eliminar algún curso del sistema por algún motivo en particular.

6.1.2.13. Caso de uso "Modificar Curso"

Actores	<i>Administrador</i>
Descripción	El usuario Administrador puede modificar los cursos, para agregar o quitar un usuario.

6.1.2.14. Caso de uso "Agregar Usuario"

Actores	<i>Administrador</i>
Descripción	El usuario Administrador puede agregar nuevos usuarios de todo tipo.

6.1.2.15. Caso de uso “Cargar Archivo”

	<i>Invocado desde el caso de uso “Agregar Usuario”</i>
Descripción	Funcionalidad del sistema para agregar a los usuarios desde un archivo cargado.

6.1.2.16. Caso de uso “Eliminar Usuario”

Actores	<i>Administrador</i>
Descripción	El usuario Administrador puede eliminar a un usuario de menor rango (Profesor, Ayudante o alumno).

6.1.2.17. Caso de uso “Modificar Usuario”

Actores	<i>Administrador</i>
Descripción	El usuario Administrador puede modificar a un usuario de menor rango (Profesor, Ayudante o alumno).

Casos de Uso Profesor/Ayudante.

6.1.2.18. Caso de uso “Seleccionar Curso”

Actores	<i>Profesor y Ayudante</i>
Descripción	El usuario Profesor o Ayudante debe seleccionar el curso al cual hacerle modificaciones.

6.1.2.19. Caso de uso “Crear Tema”

Actores	<i>Profesor y Ayudante</i>
Descripción	El usuario Profesor o Ayudante puede crear temas en el curso al cual pertenecen.

6.1.2.20. Caso de uso “Eliminar Tema”

Actores	<i>Profesor y Ayudante</i>
Descripción	El usuario Profesor o Ayudante puede eliminar temas en el curso al cual pertenecen.

6.1.2.21. Caso de uso “Modificar Tema”

Actores	<i>Profesor y Ayudante</i>
Descripción	El usuario Profesor o Ayudante puede modificar temas en el curso al cual pertenecen.

6.1.2.22. Caso de uso “Crear Sesión”

Actores	<i>Profesor y Ayudante</i>
Descripción	El usuario Profesor o Ayudante puede crear Sesión de trabajo para los alumnos en el curso al cual pertenecen.

6.1.2.23. Caso de uso “Eliminar Sesión”

Actores	<i>Profesor y Ayudante</i>
Descripción	El usuario Profesor o Ayudante puede eliminar una Sesión de trabajo en el curso al cual pertenecen.

6.1.2.24. Caso de uso “Modificar Sesión”

Actores	<i>Profesor y Ayudante</i>
Descripción	El usuario Profesor o Ayudante puede modificar una Sesión de trabajo en el curso al cual pertenecen.

aso de uso “Cargar Archivo”

	<i>Invocado desde el caso de uso “Crear Sesión” y “Modificar Sesión”</i>
Descripción	El usuario Profesor o Ayudante puede subir un archivo con el código de ejemplo a la sesión de trabajo en el curso al cual pertenecen.

6.1.2.25. Caso de uso “Finalizar Sesión”

Actores	<i>Profesor y Ayudante</i>
Descripción	El usuario Profesor o Ayudante puede finalizar sesión de trabajo de curso al cual pertenecen, con lo cual se bloquean todas las actividades a realizar en la sesión correspondiente.

6.1.2.26. Caso de uso “Revisar Actividades”

Actores	<i>Profesor y Ayudante</i>
Descripción	El usuario Profesor o Ayudante deben revisar las actividades subidas por los alumnos al sistema, por si están correctas.

6.1.2.27. Caso de uso “Calificar Actividades”

Actores	<i>Profesor y Ayudante</i>
Descripción	El usuario Profesor o Ayudante deben calificar las actividades subidas por los alumnos al sistema, con lo cual se podrá hacer un ranking de los alumnos mejor calificados.

6.1.2.28. Caso de uso “Buscar Alumnos”

	<i>Invocado desde el caso de uso “Revisar Actividades” y “Calificar Actividades”</i>
Descripción	Funcionalidad del sistema para encontrar a los usuarios alumnos, para poder revisar las actividades que realizaron y así poder calificarlas.

6.1.2.29. Caso de uso “Buscar Actividades”

	<i>Invocado desde el caso de uso “Revisar Actividades” y “Calificar Actividades”</i>
Descripción	Funcionalidad del sistema para encontrar las actividades que le corresponden a un determinado usuario.

6.1.2.30. Caso de uso “Informes de Actividad Alumnos”

Actores	<i>Profesor y Ayudante</i>
Descripción	El usuario Administrador puede obtener un informe de la cantidad de actividades realizadas por los usuarios alumnos, con los tiempos que se demoraron y en las fechas correspondientes.

6.1.2.31. Caso de uso “Generar Informes”

	<i>Invocado desde el caso de uso “Informe de Ingresos Usuarios”</i>
Descripción	Funcionalidad del sistema para generar el informe, en este caso obtiene los datos de los usuarios sobre actividad realizada y los ingresos a distintos lugares del sistema.

Casos de Uso Alumno.

6.1.2.32. Caso de uso “Ingresar Curso”

Actores	<i>Alumno</i>
Descripción	El usuario Alumno puede ingresar a un curso al cual pertenece.

6.1.2.33. Caso de uso “Ingresar Tema”

Actores	<i>Alumno</i>
Descripción	El usuario Alumno puede ingresar a un tema de un curso al cual pertenece.

6.1.2.34. Caso de uso “Ingresar Sesión”

Actores	<i>Alumno</i>
Descripción	El usuario Alumno puede ingresar a una Sesión de trabajo dentro de un tema de un curso al cual pertenece.

6.1.2.35. Caso de uso “Realizar Actividad”

Actores	<i>Alumno</i>
Descripción	El usuario Alumno puede realizar actividades dentro de las sesiones de trabajo

6.1.2.36. Caso de uso “Subir Actividad”

Actores	<i>Alumno</i>
Descripción	El usuario Alumno después de realizar una actividad de una actividad dentro de una sesión de trabajo, debe subir esta para que pueda ser calificada.

6.1.2.37. Caso de uso “Cargar Archivo”

	<i>Invocado desde el caso de uso “Subir Actividad”</i>
Descripción	Funcionalidad del sistema para subir un archivo con el código de una actividad realizada en la sesión de trabajo en el curso al cual pertenece.

6.1.2.38. Caso de uso “Ver Actividades Anteriores”

Actores	<i>Alumno</i>
Descripción	El usuario Alumno puede tener acceso a las actividades anteriores que el mismo realizo, esto a modo de apoyo para el desarrollo de las nuevas actividades.

6.1.3 Especificación de los Casos de Uso

Casos de Uso Generales.

6.1.3.1 Caso de Uso: <Ingresar al sistema>

Descripción	<i>Su objetivo es validar a los usuarios que ingresan al sistema.</i>		
Pre Condición	<i>El usuario debe estar registrado en el sistema.</i>		
Post Condición	<i>El usuario se encuentra autenticado en el sistema "EXEMPLUM"</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresar al sistema mediante su dirección web.	2.- Despliega el formulario de autenticación.		
3.- El usuario completa el formulario con su Rut y Contraseña del sistema.	4.a.- El sistema valida los datos ingresado por el usuario.		4.b.- No encuentra los datos ingresado por el usuario en el sistema y devuelve error.
	5.a.- El sistema despliega la página de inicio del sistema, al usuario ya autorizado.	5.b.- El sistema despliega un mensaje de error al usuario.	

6.1.3.2 Caso de Uso:<Autenticar Usuario>

Descripción	<i>Su objetivo es comprobar que la Contraseña y Rut ingresados por el usuarios sean correctos.</i>		
Pre Condición	<i>El usuario debe ingresar Contraseña y Rut al formulario de ingreso</i>		
Post Condición	<i>El usuario es activado como valido y se obtienen sus permisos dentro del sistema.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
	1.- El sistema recibe el RUT y la contraseña desde el formulario de ingreso.		
	2.- El sistema revisa si coinciden los datos con el de la base de datos.		
	3.a.- El sistema encuentra la coincidencia y retorna los permisos que el usuario posee.		3.b.- No se encuentran los datos y se devuelve un mensaje de error.

6.1.3.3 Caso de Uso:<Actualizar Datos>

Descripción	<i>Su objetivo es que el usuario pueda modificar su información personal.</i>		
Pre Condición	<i>El usuario debe estar autorizado dentro del sistema.</i>		
Post Condición	<i>El usuario tiene nueva información personal</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresa al panel de usuario a través a través del menú y selecciona la opción "Actualizar datos"	2.- Despliega el formulario con los datos ya ingresados posiblemente a modificar y los campos vacíos a ingresar, si se desea actualizar.		
3.- El usuario modifica los datos que desee y acciona actualizar.	4.a.- El sistema hace las validaciones correspondientes a los distintos campos y los ingresa al sistema.		4.b.- El sistema encuentra errores en los datos o no se puede conectar a la base de datos.
5.a- Se despliega un mensaje de que los datos fueron actualizados correctamente.		5.b- Se despliega un mensaje de error en los datos o conexión a la base datos.	

6.1.3.4 Caso de Uso:<Redactar Mensajes>

Descripción	<i>Su objetivo es que un usuario pueda interactuar con otros usuarios a través de un mensaje.</i>		
Pre Condición	<i>El usuario debe estar autorizado dentro del sistema.</i>		
Post Condición	<i>Uno o varios usuarios reciben el mensaje del usuario emisor.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresa a "mensajes" a través del menú y selecciona "redactar mensaje".	2.- Despliega el formulario correspondiente a redactar mensaje.		
3.- El usuario selecciona a los destinatarios, escribe el mensaje y presiona enviar.	4.a.- El mensaje es enviado correctamente y se despliega un mensaje correcto.		4.b.-El mensaje no puede ser enviado y se despliega un mensaje de error.

6.1.3.5 Caso de Uso:<Recibir Mensajes>

Descripción	<i>Su objetivo es que un usuario ver los mensajes que otros usuarios le enviaron.</i>		
Pre Condición	<i>El usuario es destinatario del mensaje que otro usuario redacto.</i>		
Post Condición	<i>El usuario leyó el mensaje y tiene la posibilidad de responder.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresa a “mensajes” a través del menú y selecciona “mensajes entrantes”.	2.a.- Se verifican que hayan mensajes como destinatario y se despliegan.		2.b.- No habrían mensajes como destinatario.
3.a.- El usuario puede leer el mensaje que desee.		3.b.- Se despliega un mensaje que no hay entradas.	

6.1.3.6 Caso de Uso:<Responder Mensajes>

Descripción	<i>Su objetivo es que un usuario pueda responder a los mensajes que otros usuarios le enviaron.</i>		
Pre Condición	<i>El usuario es destinatario del mensaje que otro usuario redacto.</i>		
Post Condición	<i>El usuario respondió el mensaje.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresa a “mensajes” a través del menú y selecciona “mensajes entrantes”.	2.a.- Se verifican que hayan mensajes como destinatario y se despliegan.		2.b.- No habrían mensajes como destinatario.
3.a.- El usuario puede leer el mensaje que desee.		3.b.- Se despliega un mensaje que no hay entradas.	
4.- El usuario selecciona responder	5.- Se despliega el formulario para responder.		
6.- El usuario escribe el mensaje y presiona enviar.	7.a.- El mensaje es enviado correctamente.		7.b.- El mensaje no se puede enviar.
8.a.- se despliega un mensaje de envío correcto en pantalla.		8.b.- se despliega un mensaje de error en pantalla	

6.1.3.7 Caso de Uso:<Salir del Sistema>

Descripción	<i>Su objetivo es que un usuario pueda salir del sistema.</i>		
Pre Condición	<i>El usuario debe estar autorizado dentro del sistema.</i>		
Post Condición	<i>El usuario pierde la autorización.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Selecciona la opción salir del menú.	2.- Elimina el estado autorizado al usuario y lo manda a la pantalla de ingreso.		

Casos de Uso Administrador.

6.1.3.8 Caso de Uso:<Crear Curso>

Descripción	<i>Su objetivo es crear un curso con su estructura base y asignar los alumnos, profesores y ayudantes.</i>		
Pre Condición	<i>El usuario debe estar autorizado como administrador dentro del sistema.</i>		
Post Condición	<i>Hay un nuevo curso dentro del sistema.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresa a "cursos" a través del menú y selecciona "Crear curso".	2.- Se despliega el formulario correspondiente a crear curso.		
3.- El usuario ingresa los datos, asigna usuarios y presiona crear.	4.a- El sistema valida que estén llenos los campos correspondientes, la conexión a la base de datos y crea la estructura del curso.		4.b.- El sistema encuentra errores de conexión o hay campos vacíos.
5.a.- El usuario visualiza un mensaje de creado correctamente el curso.		5.b.- El usuario visualiza un mensaje de error en la creación, problemas de conexión o datos vacíos.	

6.1.3.9 Caso de Uso:<Buscar Alumnos>

Descripción	<i>Su objetivo es encontrar a los usuarios Alumnos que se necesitan asignar a un curso.</i>		
Pre Condición	<i>El usuario debe estar autorizado como administrador dentro del sistema.</i>		
Post Condición	<i>Se encuentra uno o más Alumnos y se les asigna un curso.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
	1.- Busca a un usuario alumno para asignarlo a un curso.		
	2.a- El sistema encuentra al alumno y retorna el RUT, para que puedan asignarlo a un curso.		2.b.-El sistema no encuentra al alumno y retorna un mensaje de error.

6.1.3.10 Caso de Uso:<Buscar Ayudante>

Descripción	<i>Su objetivo es encontrar a los usuarios Ayudantes que se necesitan asignar a un curso.</i>		
Pre Condición	<i>El usuario debe estar autorizado como administrador dentro del sistema.</i>		
Post Condición	<i>Se encuentra uno o más Ayudantes y se les asigna un curso.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
	1.- Busca a un usuario ayudante para asignarlo a un curso.		
	2.a- El sistema encuentra al ayudante y retorna el RUT, para que puedan asignarlo a un curso.		2.b.-El sistema no encuentra al ayudante y retorna un mensaje de error.

6.1.3.11 Caso de Uso:<Buscar Profesor>

Descripción	<i>Su objetivo es encontrar a los usuarios profesores que se necesitan asignar a un curso.</i>		
Pre Condición	<i>El usuario debe estar autorizado como administrador dentro del sistema.</i>		
Post Condición	<i>Se encuentra uno o más profesores y se les asigna un curso.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
	1.- Busca a un usuario profesor para asignarlo a un curso.		
	2.a- El sistema encuentra al profesor y retorna el RUT, para que puedan asignarlo a un curso.		2.b.-El sistema no encuentra al profesor y retorna un mensaje de error.

6.1.3.12 Caso de Uso:<Eliminar Curso>

Descripción	<i>Su objetivo es eliminar lógicamente un curso.</i>		
Pre Condición	<i>El usuario debe estar autorizado como administrador dentro del sistema.</i>		
Post Condición	<i>Un curso cambia a estado oculto, esto quiere decir que no es visible por profesores, ayudantes y alumnos.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresa a “cursos” a través del menú, del listado de cursos selecciona uno y aplica la acción “eliminar”.	2.- Se despliega el un mensaje sobre si esta seguro de realizar la acción.		
3.- El usuario selecciona que si esta seguro	4.a- El sistema elimina la estructura del curso lógicamente.		4.b.- El sistema encuentra errores de conexión.
5.a- El usuario visualiza un mensaje de eliminado correctamente el curso.		5.b.- El usuario visualiza un mensaje de error en la conexión.	

6.1.3.13 Caso de Uso:<Modificar Curso>

Descripción	<i>Su objetivo es actualizar la información referente al curso y agregar o quitar usuarios.</i>		
Pre Condición	<i>El usuario debe estar autorizado como administrador dentro del sistema.</i>		
Post Condición	<i>Un curso adquiere nueva información y/o usuarios.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresa a “cursos” a través del menú, del listado de cursos selecciona uno y aplica la acción “modificar”.	2.- Se despliega el formulario con los datos que posee el curso, así el usuario puede modificarlos y quitar o agregar usuarios.		
3.- El usuario actualiza los datos que desea y selecciona guardar cambios.	4.a- El sistema valida que estén llenos los campos correspondientes, la conexión a la base de datos e ingresa los nuevos datos a la estructura del curso.		4.b.- El sistema encuentra errores de conexión o hay campos vacíos.
5.a.- El usuario visualiza un mensaje de modificado correctamente el curso.		5.b.- El usuario visualiza un mensaje de error en la modificación, problemas de conexión o datos vacíos.	

6.1.3.14 Caso de Uso:<Agregar Usuario>

Descripción	<i>Su objetivo es ingresar al sistema un nuevo usuario.</i>		
Pre Condición	<i>El usuario debe estar autorizado como administrador dentro del sistema.</i>		
Post Condición	<i>Hay un nuevo usuario dentro del sistema.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresa a "usuarios" a través del menú y selecciona "agregar nuevo usuario".	2.- Se despliega el formulario correspondiente a agregar nuevo usuario.		
3.- El usuario ingresa los datos del nuevo usuario y presiona agregar.	4.a- El sistema valida que estén llenos los campos correspondientes, la conexión a la base de datos y agrega el nuevo usuario.		4.b.- El sistema encuentra errores de conexión o hay campos vacíos.
5.a.- El usuario visualiza un mensaje de agregado correctamente.		5.b.- El usuario visualiza un mensaje de error, problemas de conexión o datos vacíos.	

6.1.3.15 Caso de Uso:<Cargar Archivo>

Descripción	<i>Su objetivo es ingresar un archivo plantilla que contenga los usuarios ordenados, para poder agregarlos al sistema.</i>		
Pre Condición	<i>El usuario debe estar autorizado como administrador dentro del sistema.</i>		
Post Condición	<i>Habrán nuevos usuarios dentro del sistema.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
	1.a.- carga el archivo plantilla con los datos de nuevos usuarios en el servidor.		1.b.- no puede cargar el archivo en el servidor
	2.a.- El sistema procesa los datos del archivo y los ingresa al sistema.		2.b.- El sistema no puede procesar los datos.
	3.a.- El sistema retorna un mensaje de cargados los usuarios correctamente desde el archivo.		3.b.- El sistema retorna un mensaje de error.

6.1.3.16 Caso de Uso:<Eliminar Usuario>

Descripción	<i>Su objetivo es eliminar lógicamente a un usuario, impidiendo el ingreso al sistema.</i>		
Pre Condición	<i>El usuario debe estar autorizado como administrador dentro del sistema.</i>		
Post Condición	<i>Un usuario pierde el privilegio de ingresar al sistema.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresa a "usuarios" a través del menú, del listado de usuarios selecciona uno y aplica la acción "eliminar".	2.- Se despliega el un mensaje sobre si esta seguro de realizar la acción.		
3.- El usuario selecciona que si esta seguro	4.a- El sistema elimina los datos del usuario lógicamente.		4.b.- El sistema encuentra errores de conexión.
5.a.- El usuario visualiza un mensaje de eliminado correctamente.		5.b.- El usuario visualiza un mensaje de error en la conexión.	

6.1.3.17 Caso de Uso:<Modificar Usuario>

Descripción	<i>Su objetivo es actualizar los datos de un determinado usuario.</i>		
Pre Condición	<i>El usuario debe estar autorizado como administrador dentro del sistema.</i>		
Post Condición	<i>Un usuario tiene nueva información o nuevos privilegios dentro del sistema.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresa a "usuarios" a través del menú, del listado de usuarios selecciona uno y aplica la acción "modificar".	2.- Se despliega el formulario con los datos que posee el usuario a modificar.		
3.- El usuario actualiza los datos que desea y selecciona guardar cambios.	4.a- El sistema valida que estén llenos los campos correspondientes, la conexión a la base de datos e ingresa los nuevos datos.		4.b.- El sistema encuentra errores de conexión o hay campos vacíos.
5.a.- El usuario visualiza un mensaje de modificado correctamente.		5.b.- El usuario visualiza un mensaje de error en la modificación, problemas de conexión o datos vacíos.	

Casos de Uso Profesor/Ayudante.

6.1.3.18 Caso de Uso:<Seleccionar Curso>

Descripción	<i>Su objetivo es que el usuario profesor y/o ayudante pueda elegir el curso al cual desea hacerle modificaciones.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>El usuario profesor y/o ayudante esta dentro de un curso a los que pertenece.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Selecciona el curso al cual pertenece.	2.- El sistema despliega todas las opciones correspondientes a ese curso.		
3.- El usuario visualiza el menú del curso.			

6.1.3.19 Caso de Uso:<Crear Tema>

Descripción	<i>Su objetivo es crear un nuevo tema dentro de un curso.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>Un curso posee un nuevo tema.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- dentro del curso seleccionado se ingresa a "temas" a través del menú y se selecciona "agregar nuevo tema".	2.- Se despliega el formulario correspondiente a agregar nuevo tema.		
3.- El usuario ingresa los datos del nuevo tema y presiona agregar.	4.a- El sistema valida que estén llenos los campos correspondientes, la conexión a la base de datos y agrega el nuevo tema.		4.b.- El sistema encuentra errores de conexión o hay campos vacíos.
5.a.- El usuario visualiza un mensaje de agregado el tema correctamente.		5.b.- El usuario visualiza un mensaje de error, problemas de conexión o datos vacíos.	

6.1.3.20 Caso de Uso:<Eliminar Tema>

Descripción	<i>Su objetivo es eliminar lógicamente un tema dentro de un curso.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>El tema queda deshabilitado para ser visualizado por los alumnos.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- dentro del curso seleccionado se ingresa a “temas” a través del menú, del listado de temas selecciona uno y aplica la acción “eliminar”.	2.- Se despliega el un mensaje sobre si esta seguro de realizar la acción.		
3.- El usuario selecciona que si esta seguro	4.a- El sistema elimina los datos del tema lógicamente.		4.b.- El sistema encuentra errores de conexión.
5.a.- El usuario visualiza un mensaje de eliminado el tema correctamente.		5.b.- El usuario visualiza un mensaje de error en la conexión.	

6.1.3.21 Caso de Uso:<Modificar Tema>

Descripción	<i>Su objetivo es actualizar la información de un tema dentro de un curso.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>El tema tiene nueva información.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Ingresa a “usuarios” a través del menú, del listado de usuarios selecciona uno y aplica la acción “modificar”.	2.- Se despliega el formulario con los datos que posee el tema a modificar.		
3.- El usuario actualiza los datos que desea y selecciona guardar cambios.	4.a- El sistema valida que estén llenos los campos correspondientes, revisa la conexión a la base de datos e ingresa los nuevos datos.		4.b.- El sistema encuentra errores de conexión o hay campos vacíos.
5.a.- El usuario visualiza un mensaje de modificado correctamente.		5.b.- El usuario visualiza un mensaje de error en la modificación, problemas de conexión o datos vacíos.	

6.1.3.22 Caso de Uso:<Crear Sesión>

Descripción	<i>Su objetivo es generar una sesión de trabajo dentro de un tema de un curso.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>Hay una nueva sesión dentro de un tema de un curso.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- dentro del curso seleccionado se ingresa a “temas” a través del menú y se selecciona “agregar nueva sesión”.	2.- Se despliega el formulario correspondiente a agregar nueva sesión.		
3.- El usuario ingresa los datos de la nueva sesión y presiona agregar.	4.a- El sistema valida que estén llenos los campos correspondientes, revisa la conexión a la base de datos y agrega la nueva sesión.		4.b.- El sistema encuentra errores de conexión o hay campos vacíos.
5.a.- El usuario visualiza un mensaje de agregada la sesión correctamente.		5.b.- El usuario visualiza un mensaje de error, problemas de conexión o datos vacíos.	

6.1.3.23 Caso de Uso:<Eliminar Sesión>

Descripción	<i>Su objetivo es eliminar lógicamente una sesión.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>La sesión queda deshabilitada para ser visualizada por los alumnos.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- dentro del curso seleccionado se ingresa a “temas” a través del menú, del listado de temas selecciona uno, se despliegan las sesiones se selecciona una y se aplica la acción “eliminar”.	2.- Se despliega el un mensaje sobre si esta seguro de realizar la acción.		
3.- El usuario selecciona que si esta seguro	4.a- El sistema elimina los datos de la sesión lógicamente.		4.b.- El sistema encuentra errores de conexión.

6.1.3.24 Caso de Uso:<Modificar Sesión>

Descripción	<i>Su objetivo es actualizar datos de la sesión.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>La sesión adquirió nuevos información.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- dentro del curso seleccionado se ingresa a “temas” a través del menú, del listado de temas selecciona uno, se despliegan las sesiones se selecciona una y se aplica la acción “modificar”.	2.- Se despliega el formulario con los datos que posee la sesión a modificar.		
3.- El usuario actualiza los datos que desea y selecciona guardar cambios.	4.a- El sistema valida que estén llenos los campos correspondientes, revisa la conexión a la base de datos e ingresa los nuevos datos.		4.b.- El sistema encuentra errores de conexión o hay campos vacíos.
5.a.- El usuario visualiza un mensaje de modificado correctamente.		5.b.- El usuario visualiza un mensaje de error en la modificación, problemas de conexión o datos vacíos.	

6.1.3.25 Caso de Uso:<Cargar Archivo>

Descripción	<i>Su objetivo es subir un archivo con el ejemplo de versión a realizar dentro de una sesión.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>Hay un ejemplo de versión para los alumnos.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
	1.a.- carga el archivo con el código de ejemplo para los alumnos en el servidor.		1.b.- no puede cargar el archivo en el servidor
	2.a.- El sistema retorna un mensaje de cargados los usuarios correctamente desde el archivo.		2.b.- El sistema retorna un mensaje de error.

6.1.3.26 Caso de Uso:<Finalizar Sesión>

Descripción	<i>Su objetivo es cerrar el acceso al alumno para realizar las versiones que hay de tarea dentro de una sesión.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>La sesión aparece como finalizada al alumno.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- dentro del curso seleccionado se ingresa a “temas” a través del menú, del listado de temas selecciona uno, se despliegan las sesiones se selecciona una y se aplica la acción “finalizar”.	2.a.- Se cierra el acceso a todos los usuarios para realizar las actividades de los ejercicios propuestos dentro de la sesión.		2.b.- No se puede realizar la acción por problemas de conexión.
3.a.- El usuario visualiza un mensaje de finalizada correctamente.		3.b.- El usuario visualiza un mensaje de problemas en la conexión.	

6.1.3.27 Caso de Uso:<Revisar Actividades>

Descripción	<i>Su objetivo es comprobar que la actividad sea revisada por el profesor y/o ayudante.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>La actividad ha sido revisada por el profesor.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- dentro del curso seleccionado se ingresa a “Actividades” a través del menú, del listado de actividades selecciona las últimas realizadas y presiona “revisar” sobre una.	2.a.- Se despliega la actividad el alumno al cual pertenece y las opciones para revisar.		2.b.- No se puede obtener la actividad.
3.a.- El usuario visualiza la actividad, la cual puede compilar y dar por revisada.		3.b.- El usuario visualiza un mensaje de error.	

6.1.3.28 Caso de Uso:<Calificar Actividades>

Descripción	<i>Su objetivo es que el profesor y/o ayudante pueda revisar la actividad y darle una puntuación</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>La actividad ha sido calificada.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- dentro del curso seleccionado se ingresa a "Actividades" a través del menú, del listado de actividades selecciona las ultimas realizadas y presiona "calificar" sobre una.	2.a.- Se despliega la actividad el alumno al cual pertenece y las opciones para calificar.		2.b.- No se puede obtener la actividad.
3.a.- El usuario califica y presiona guardar.	4.a.- El sistema guarda la calificación en el sistema.	3.b.- El usuario visualiza un mensaje de error.	4.b.- El sistema no puede guardar datos.
5.a.- El usuario visualiza un mensaje de guardado correctamente.		5.b.- El usuario visualiza un mensaje de error.	

6.1.3.29 Caso de Uso:<Buscar Alumnos>

Descripción	<i>Su objetivo es que el profesor y/o ayudantes puedan visualizar al alumno que pertenece cada actividad realizada.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>El usuario visualiza al alumno correspondiente a una versión.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
	1.- Recibe el RUT del usuario que se busca.		
	2.a.- El sistema encuentra el usuario y retorna los datos correspondientes.		2.b.- El sistema no encuentra el usuario o no hay conexión y retorna un mensaje de error.

6.1.3.30 Caso de Uso:<Buscar Actividades>

Descripción	<i>Su objetivo es que el profesor y/o ayudantes puedan visualizar la actividad que pertenece a un alumno.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>El usuario visualiza la actividad correspondiente a un alumno.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
	1.- Recibe el identificador de la actividad.		
	2.a.- El sistema encuentra la actividad y devuelve los datos correspondientes.		2.b.- El sistema no encuentra la actividad o no hay conexión y retorna un mensaje de error.

6.1.3.31 Caso de Uso:<Informes de Actividad Alumnos>

Descripción	<i>Su objetivo es que el profesor y/o ayudante obtener un informe de las actividades que realizaron los alumnos.</i>		
Pre Condición	<i>El usuario debe estar autorizado como profesor y/o ayudante dentro del sistema.</i>		
Post Condición	<i>Hay generado un informe de actividades de los alumnos al cual se le puede hacer un análisis.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Dentro del curso seleccionado, ingresa a “informes” a través del menú y selecciona “informe actividad de alumnos”.	2.a.- Se genera el informe con los datos correspondientes a cada alumno.		2.b.- No se pueden obtener los datos para generar el informe.
3.a.- El usuario visualiza el informe por pantalla.		3.b.- El usuario visualiza un mensaje de error.	

6.1.3.32 Caso de Uso:<Generar Informes>

Descripción	<i>Su objetivo es generar un informe con los datos correspondientes a lo que se desee.</i>		
Pre Condición	<i>El usuario debe estar autorizado como administrador dentro del sistema.</i>		
Post Condición	<i>Hay un nuevo informe al cual se le puede realizar un análisis.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
	1.- Recibe la petición de los datos que necesita par el informe.		
	2.a.- El sistema busca todos los datos necesarios.		2.b.- hay problemas con la conexión.
	3.a.- Se genera el informe para mostrarlo al usuario.		3.b.- Se genera un mensaje de error.

Casos de Uso Alumno.

6.1.3.33 Caso de Uso:<Ingresar Curso>

Descripción	<i>Su objetivo es que el usuario alumno ingrese al curso que pertenece.</i>		
Pre Condición	<i>El usuario debe estar autorizado como alumno dentro del sistema.</i>		
Post Condición	<i>El usuario alumno esta dentro de un curso al cual pertenece.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Selecciona un curso al cual pertenece desde el menú.	2.- El sistema le despliega todas las opciones correspondientes a su curso.		
3.- El usuario visualiza el menú del curso y las actividades por realizar.			

6.1.3.34 Caso de Uso:<Ingresar Tema>

Descripción	<i>Su objetivo es que el usuario alumno ingrese a un tema dentro de un curso.</i>		
Pre Condición	<i>El usuario debe estar autorizado como alumno dentro del sistema.</i>		
Post Condición	<i>El usuario alumno esta dentro de un tema de un curso al cual pertenece.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Selecciona el “tema” que debe trabajar, dentro del curso en cual está.	2.- El sistema le despliega todas las opciones correspondientes a su tema.		
3.- El usuario visualiza las sesiones por realizar dentro del tema.			

6.1.3.35 Caso de Uso:<Ingresar Sesión>

Descripción	<i>Su objetivo es que el usuario alumno ingrese a una sesión dentro de un tema.</i>		
Pre Condición	<i>El usuario debe estar autorizado como alumno dentro del sistema.</i>		
Post Condición	<i>El usuario alumno esta dentro de una sesión.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Selecciona la “sesión de trabajo” que debe realizar dentro del tema en cual está.	2.- El sistema le despliega todas las opciones correspondientes a la sesión.		
3.- El usuario visualiza la sesión de trabajo con todas sus actividades.			

6.1.3.36 Caso de Uso:<Realizar Actividad>

Descripción	<i>Su objetivo es que el usuario alumno escribe el código de una actividad que debe realizar.</i>		
Pre Condición	<i>El usuario debe estar autorizado como alumno dentro del sistema.</i>		
Post Condición	<i>El usuario alumno tendrá realizada una Actividad.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Selecciona la "actividad" que debe realizar dentro del curso en cual está.	2.- El sistema le despliega todas las opciones correspondientes para realizar la actividad.		
3.- El usuario visualiza las opciones para realizar la actividad y escribe todo lo que corresponde.			

6.1.3.37 Caso de Uso:<Subir Actividad>

Descripción	<i>Su objetivo es que el usuario alumno cargue el código de una actividad que realizo en el servidor.</i>		
Pre Condición	<i>El usuario debe estar autorizado como alumno dentro del sistema.</i>		
Post Condición	<i>El usuario tiene su código en servidor, con esto el profesor puede revisarlo.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Al tener realizada la actividad presiona subir al servidor.	2.a.- El sistema sube el código correctamente.		2.b.- El sistema no tiene conexión o demora en el tiempo de ejecución.
3.a.- El usuario visualiza un mensaje de todo correcto.		3.b.- El usuario visualiza un mensaje de error.	

6.1.3.38 Caso de Uso:<Cargar Archivo>

Descripción	<i>Su objetivo es que el usuario alumno pueda subir el archivo con el código de la actividad que realizo.</i>		
Pre Condición	<i>El usuario debe estar autorizado como alumno dentro del sistema.</i>		
Post Condición	<i>En el servidor hay un archivo con el código de la actividad que realizo el usuario alumno.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
	1.a.- carga el archivo con el código de la actividad correspondiente en el servidor.		1.b.- no puede cargar el archivo en el servidor
	2.a.- El sistema procesa los datos del archivo y los ingresa al sistema.		2.b.- El sistema no puede procesar los datos.
	3.a.- El sistema retorna un mensaje de cargados los usuarios correctamente desde el archivo.		3.b.- El sistema retorna un mensaje de error.

6.1.3.39 Caso de Uso:<Ver Actividades Anteriores>

Descripción	<i>Su objetivo es que el usuario pueda ver las actividades que realizo para recordar lo que ha hecho, esto le permitirá que su trabajo sea más rápido.</i>		
Pre Condición	<i>El usuario debe estar autorizado como alumno dentro del sistema.</i>		
Post Condición	<i>El usuario alumno reviso sus actividades anteriores.</i>		
Curso Normal		Curso Alternativo	
Usuario	Sistema	Usuario	Sistema
1.- Esta realizando una actividad.	2.- el sistema despliega en un costado las versiones realizadas anteriormente.		
3.- El usuario visualiza la actividad anterior.			

6.1.3.40 Caso Modelamiento de datos

Para la utilización del sistema **EXEMPLUM**, se implementa una Base de Datos dedicada a este, para un funcionamiento autónomo.

A continuación se da una breve descripción de las entidades y relaciones que posee el Modelo Entidad-Relación de la Base de Datos:

- Usuarios: esta entidad contiene los datos relevantes de los usuarios que utilizan el sistema. Además, por generalización, corresponde a una súper clase del tipo Total inclusiva, ya que un usuario debe estar presente en al menos 1 subclase (Alumno, Ayudante, Profesor y Administrador) y también, un alumno podría ser un ayudante, por lo tanto puede tomar mas de una subclase a la vez.
- Cursos: esta entidad almacena la información de los cursos que se dictan o se han dictado y que necesitan utilización del sistema **EXEMPLUM**.
- Etiquetas: esta identidad almacena la información correspondiente a palabras claves que son relacionadas con los distintos temas, lo cual facilitara la búsqueda de ayudas relacionadas al tema que esta revisando.
- Temas: esta entidad almacena la información referente a las unidades que se dictan dentro del curso.
- Sesiones: esta entidad almacena la información de las sesiones de trabajo que tiene cada curso.
- Actividades: esta entidad almacena la información correspondiente a cada trabajo que deben realizar durante las sesiones.
- Versiones: esta entidad almacena la información correspondiente a cada archivo que suben los alumnos a la sesión, además de almacenar si el archivo es compilado o no, con sus respectivos mensajes de compilación (si es que existen).
- Ayudante: entidad que almacena el identificador de los ayudantes.
- Profesor: entidad que almacena el identificador de los profesores que existen dentro del sistema.
- Administrador: entidad que almacena el identificador de los administradores que existen dentro del sistema.
- Alumno: entidad que almacena el identificador de los alumnos que utilizan el sistema.
- Mensajes: entidad que almacena los mensajes que envían los usuarios.
- Envíos: entidad que almacena la información correspondiente a que usuarios va el mensaje, quien es la persona que envía el mensaje y cual es el contenido del mensaje.

- Incluye: relación que permite obtener la información de que envíos que se realizan con un determinado mensaje.
- Efectúa: representa la acción del usuario al enviar un mensaje a una cierta cantidad de personas.
- Recibe: representa la acción del usuario al recibir un mensaje enviado por otro usuario.
- Pertenece: representa los cursos a los que están registrados los alumnos.
- Tiene: representa la acción de los temas que pertenecen a ciertos cursos.
- Posee: representa las distintas etiquetas que pueden estar unidas con ciertos temas.
- Contiene: representa la acción de las sesiones que pertenecen a ciertos temas.
- Trabaja: representa la acción de las actividades que se realizan o trabajan durante la duración de cada sesión.
- Crea: representa la acción de agregar versiones a cada actividad.
- Realiza: representa la acción de la creación de una(s) versión(es) por parte de un usuario.
- Imparte: representa los cursos a los cuales esta asignado cierto profesor.
- Apoya: representa los cursos a los cuales esta asignado cierto ayudante y sus respectivos permisos para realizar modificaciones dentro del curso al cual este apoyando.

Para mostrar el Modelo Entidad-Relación del sistema **EXEMPLUM** se divide en 4 partes, desde la Fig. 9 a la Fig. 12.

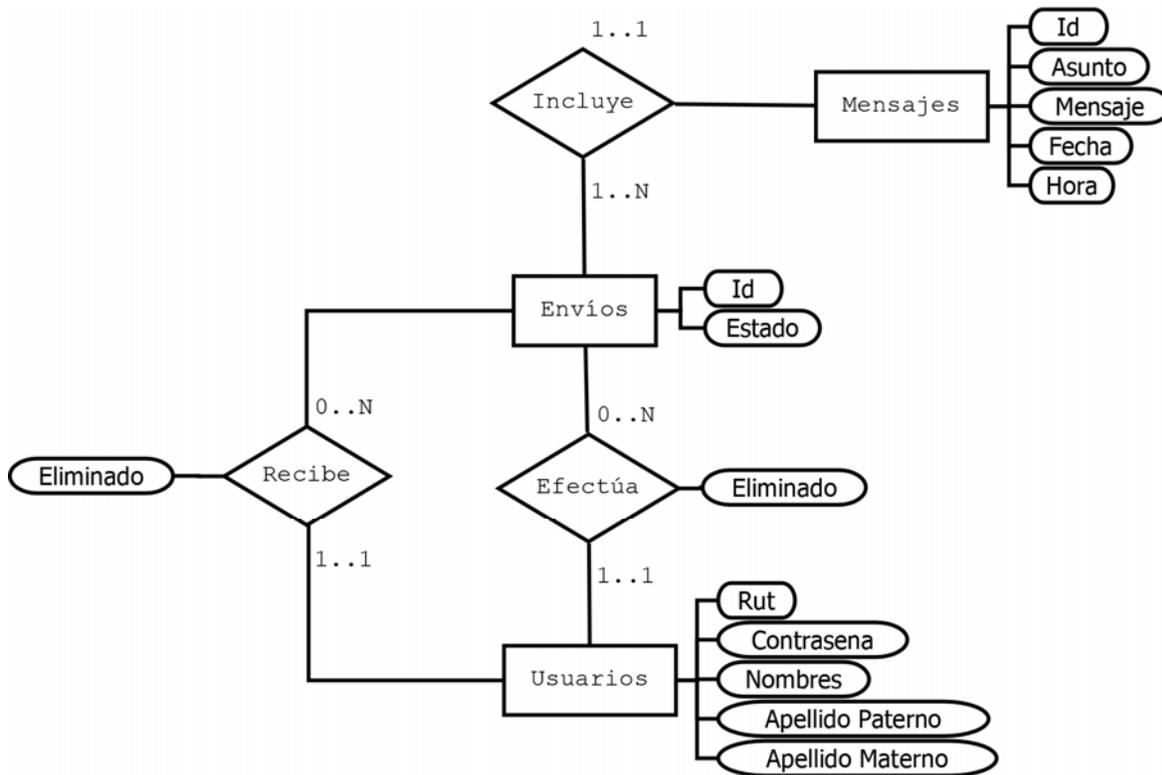


Fig. 9, MER Mensajes.

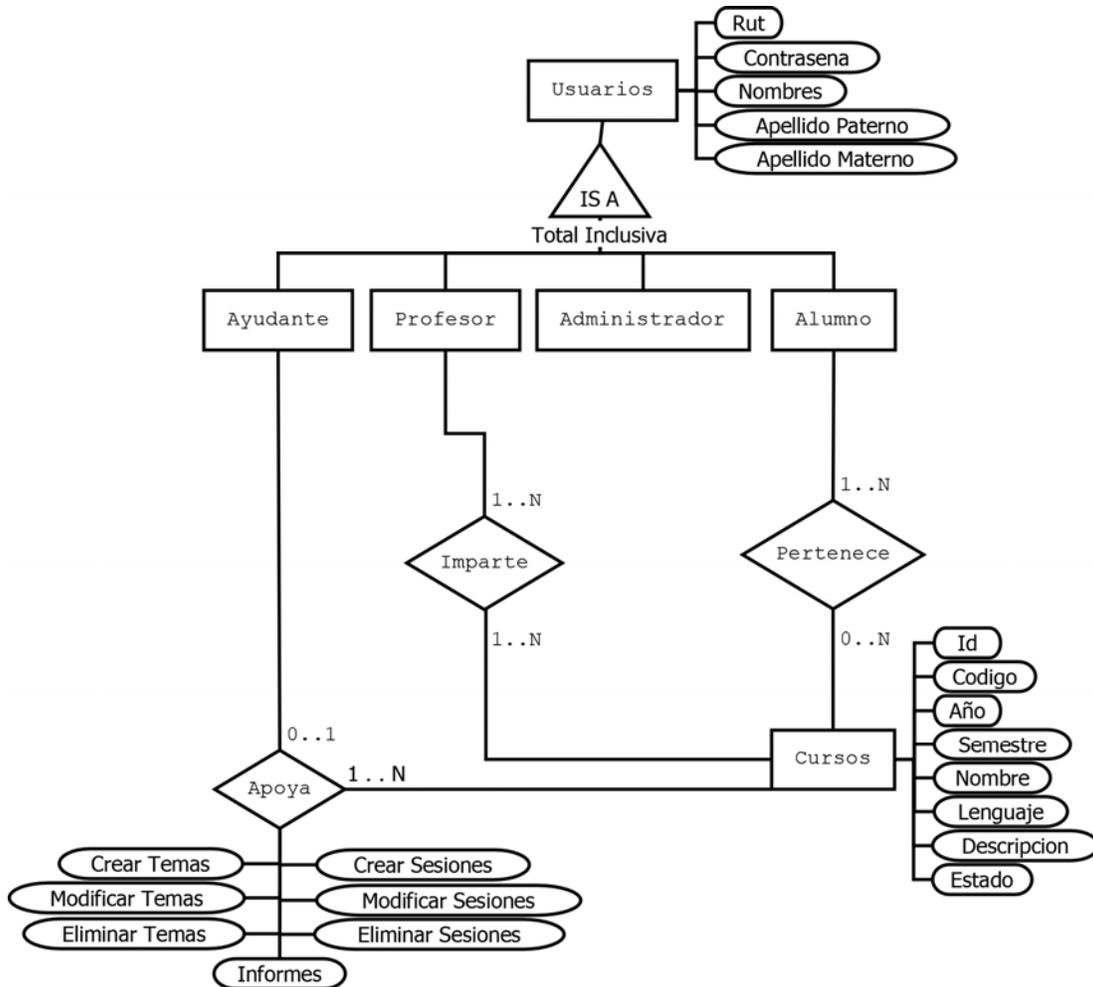


Fig. 10, MER Usuarios

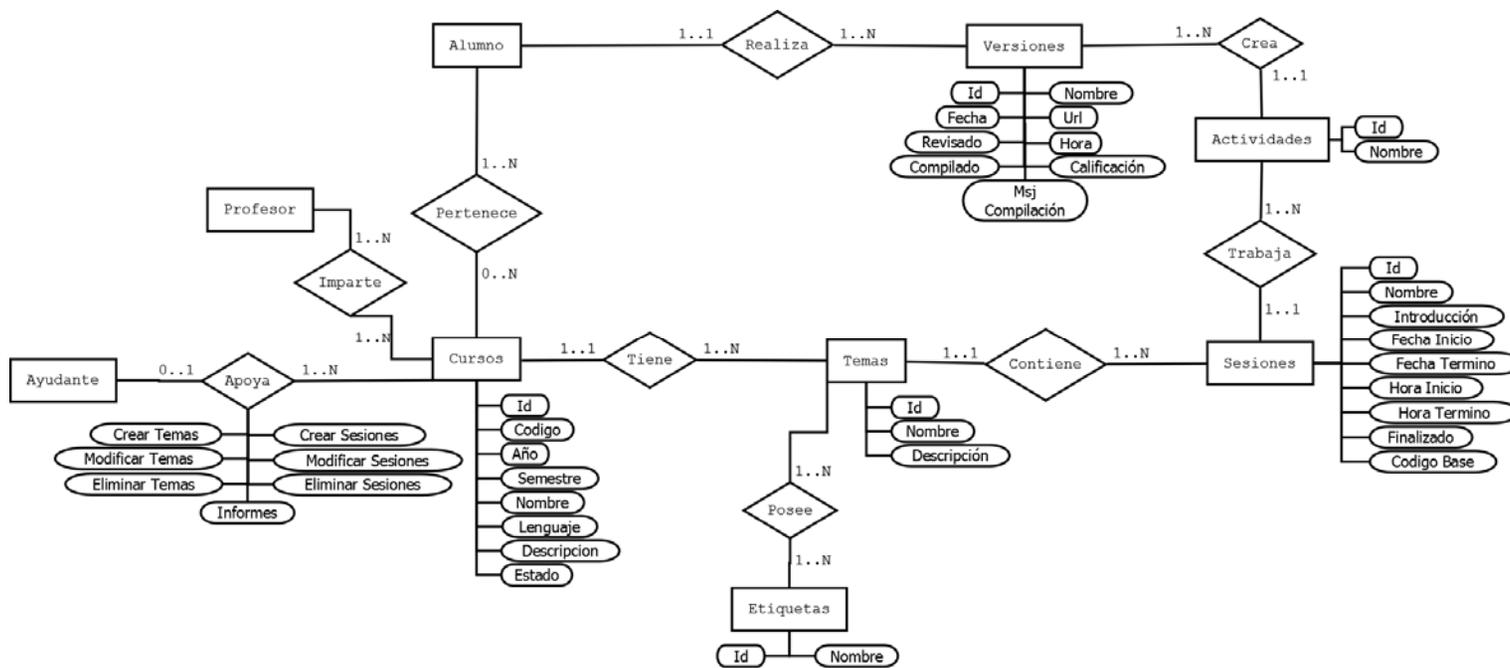


Fig. 11, MER Cursos

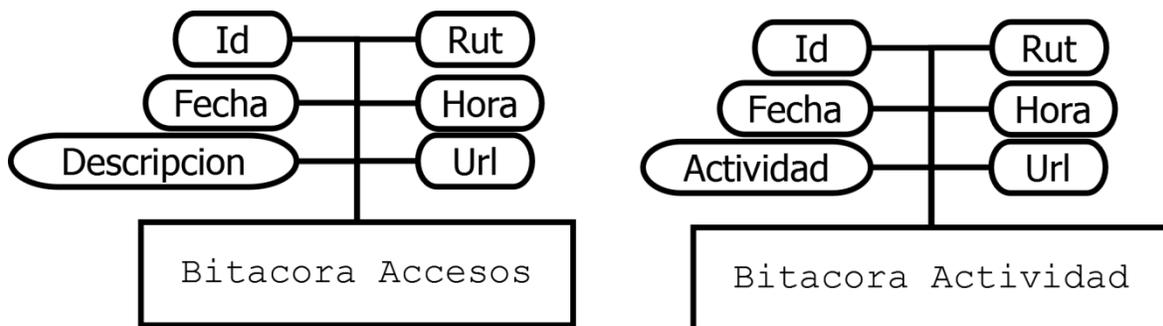


Fig. 12, MER Bitácoras.

7 DISEÑO

7.1 Diseño de Físico de la Base de datos

Dadas las características del traspaso, del Modelo Entidad-Relación a Modelo Relacional, queda como se indica en las **Fig. 13** a **Fig. 16**.

De las entidades descritas en el MER se han creado las siguientes tablas en el MR:

- Usuarios
- Mensajes
- Envíos
- Alumno
- Ayudante
- Profesor
- Administrador
- Cursos
- Temas
- Etiquetas
- Sesiones
- Actividades
- Versiones

De las relaciones descritas en el MER se han creado las siguientes entidades en el MR que se describen en la **Tabla 8**

Nombre Relación	Entidades Relacionadas	Cardinalidad
Pertenece	Alumno – Cursos	0 a N
Posee	Temas – Etiquetas	N a N
Apoya	Ayudante – Cursos	0 a N
Imparte	Profesor – Cursos	N a N

Tabla 8 Relaciones creadas como entidad en el MR a partir del MER

De las relaciones descritas en el MER no se han creado las entidades en el MR que se describen en la **Tabla 9**

Nombre Relación	Entidades Relacionadas	Cardinalidad
Incluye	Mensajes – Envíos	1 a N
Efectúa	Usuarios – Envíos	1 a 0.N
Recibe	Usuarios – Envíos	1 a 0.N
Tiene	Cursos – Temas	1 a N
Contiene	Temas – Sesiones	1 a N
Trabaja	Sesiones – Actividades	1 a N
Crea	Actividades – Versiones	1 a N
Realiza	Usuarios – Versiones	1 a 0.N

Tabla 9 Relaciones no creadas como entidad en el MR a partir del MER

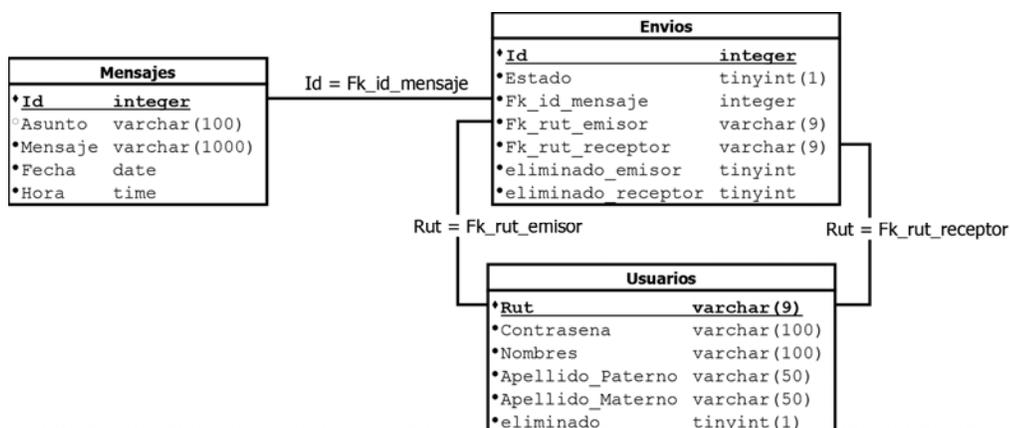


Fig. 13, MR Mensajes.

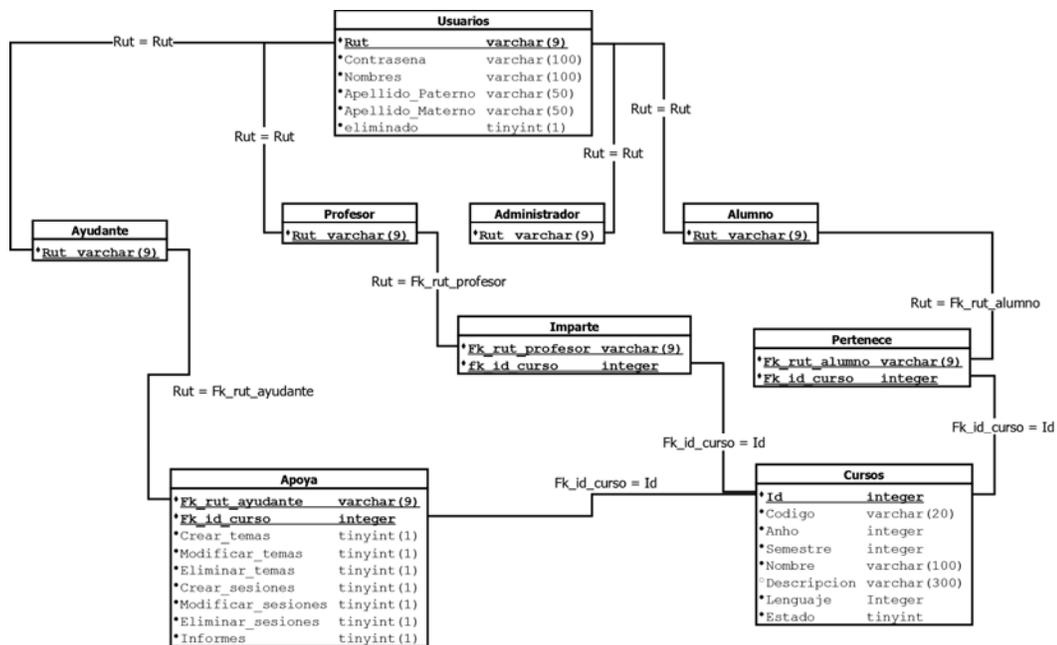


Fig. 14, MR Usuarios

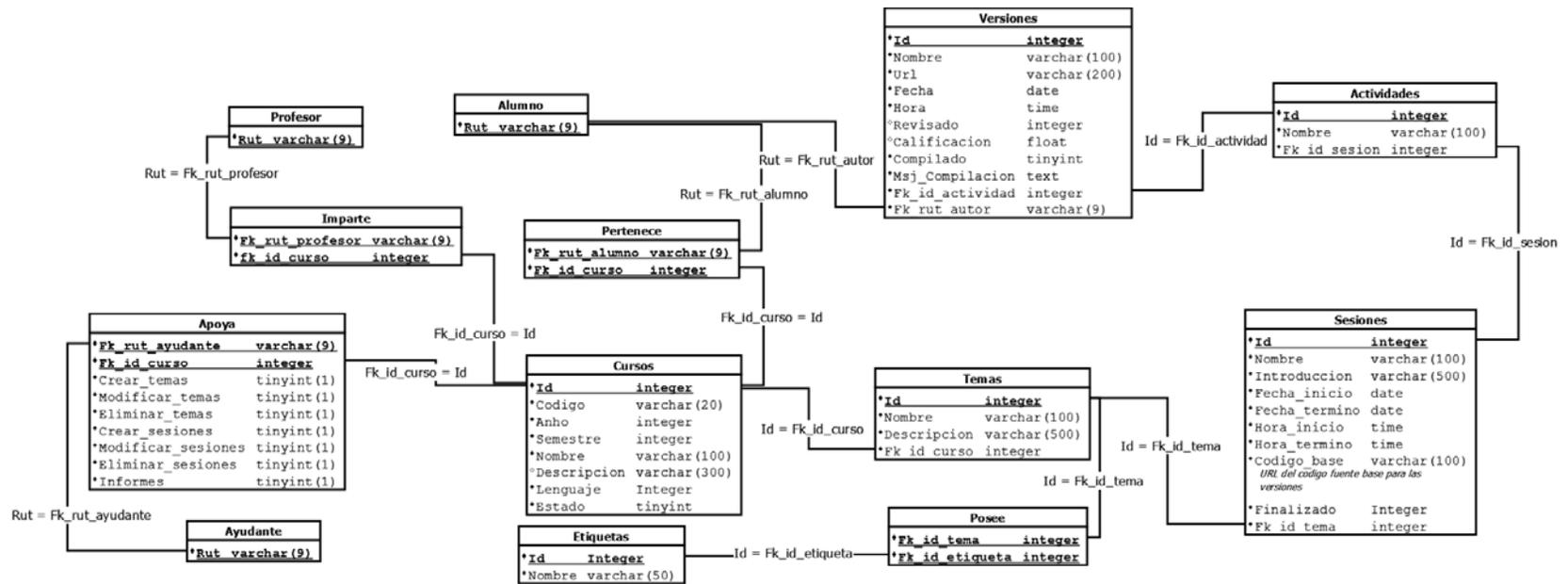


Fig. 15, MR Cursos

Bitacora_Accesos	
*Id	integer
*Rut	varchar(9)
*Descripcion	varchar(100) <i>Corresponde a los lugares de la plataforma donde accede el usuario</i>
*Fecha	date
*Hora	time
*url	varchar(100)

Bitacora_Actividad	
*Id	integer
*Rut	varchar(9)
*Actividad	varchar(200) <i>Referida a la Sesiones, versiones y todo lo relacionado a ellas</i>
*Fecha	date
*Hora	time
*Url	varchar(100)

Fig. 16, MR Bitácoras.

7.2 Diseño de arquitectura funcional

En este ítem se presenta la funcionalidad que presenta el sistema mediante arboles de descomposición funcional, en las **Fig. 17**, **Fig. 18** y **Fig. 19**.

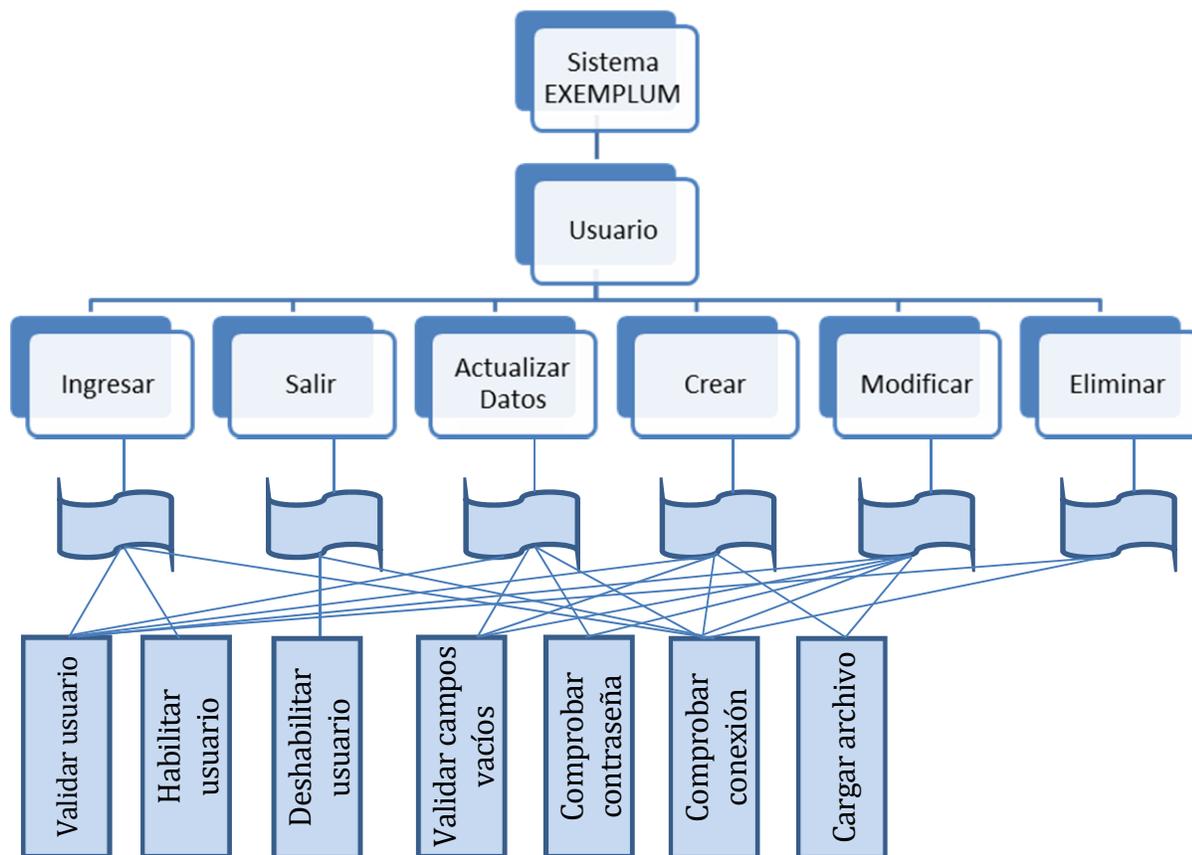


Fig. 17, Diseño Arquitectura Funcional Usuarios.

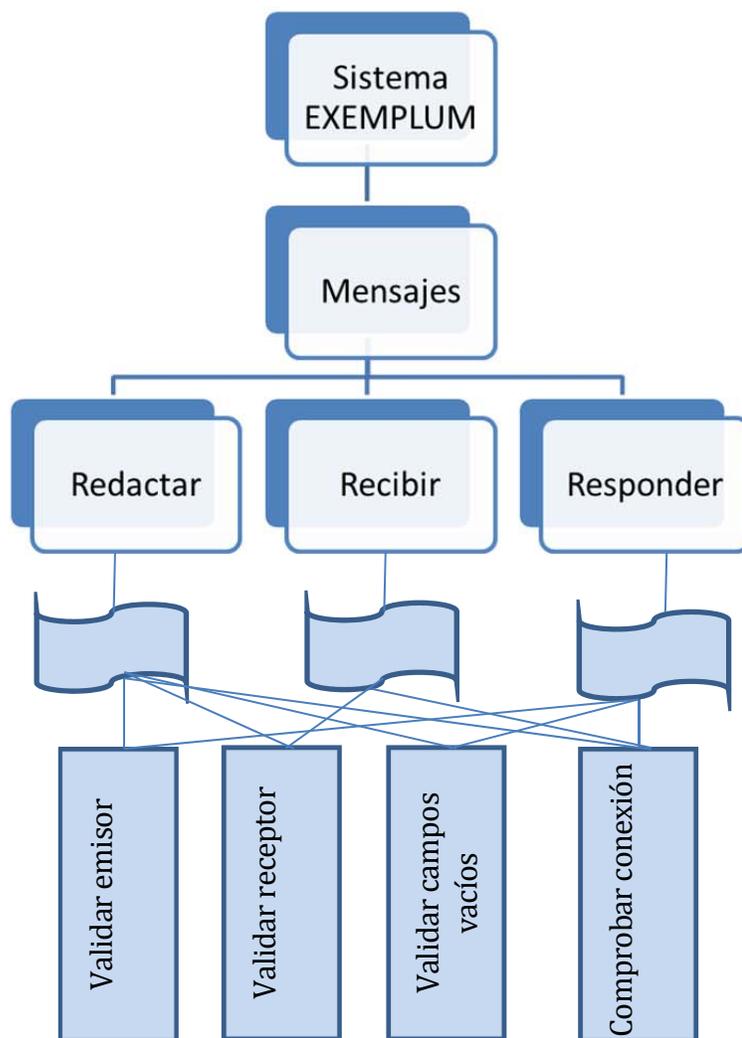


Fig. 18, Diseño Arquitectura Funcional Mensajes.

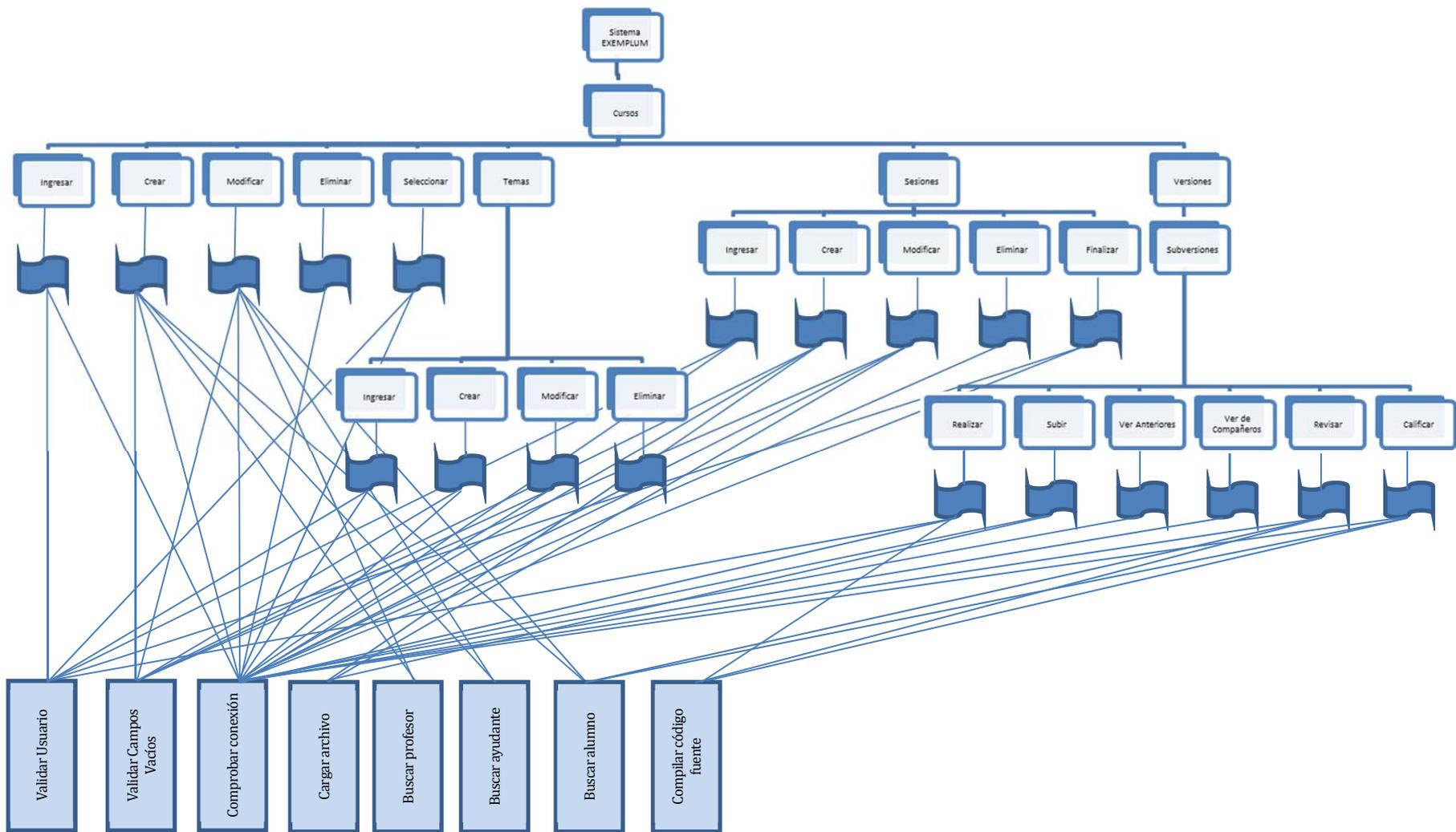


Fig. 19, Diseño Arquitectura Funcional Cursos.

7.3 Diseño interfaz y navegación

En este capítulo se describe el diseño general del sistema **EXEMPLUM** que es respetado en todas las pantallas. Además se consideran aspectos de la interfaz como botones, logos, tamaño de letras, mensajes y navegación entre pantallas.

Interfaz General del Sistema EXEMPLUM

El diseño de la interfaz posee los bloques básicos de una página web, está diseñada para que sólo cambie la sección “contenido” manteniendo la cabecera y menú lateral presentes, así el usuario podrá acceder a otra opción siempre que lo desee.

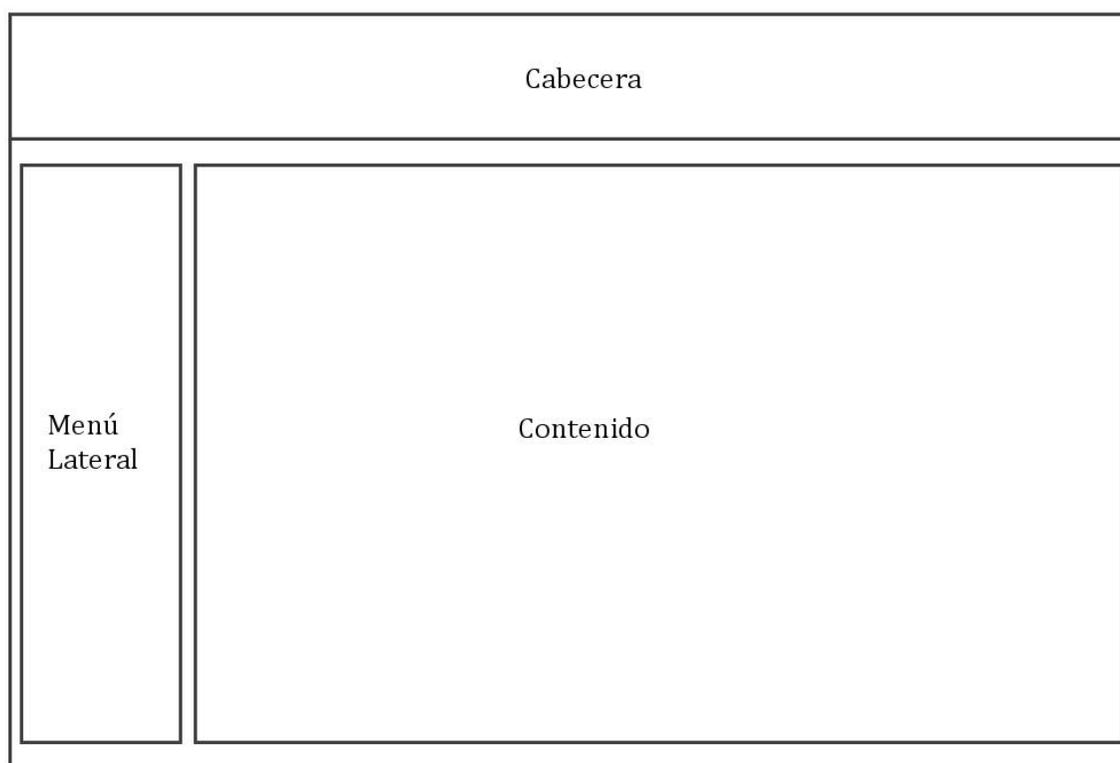


Fig. 20, Interfaz General del Sistema EXEMPLUM

Cabecera

En esta sección se encuentran los siguientes elementos:

- Logo de la Institución: Ubicado en la esquina superior izquierda de la página.
- Nombre Institución: Se sitúa bajo el logo.
- Datos de usuario autenticado: Presenta el nombre del usuario autenticado, enlace para salir del sistema y un enlace para ver perfil (acceder a modificar información personal).
- Nombre del módulo: Presenta el nombre del módulo a cual acceso el usuario, con el fin de informar donde se encuentra.

Menú Lateral

En esta sección se encuentra el menú de acciones que puede tomar el usuario. Las opciones del menú cambian según el tipo de usuario que ingresa al sistema, como se muestra en las siguientes figuras.

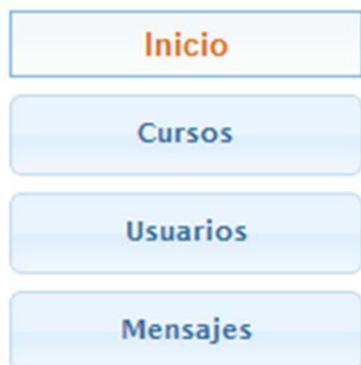


Fig. 21, Menú Administrador

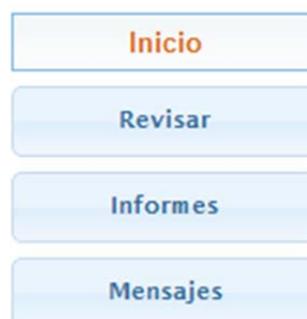


Fig. 22, Menú Profesor y/o Ayudante

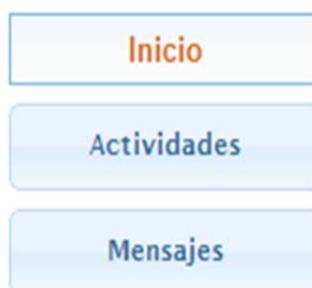


Fig. 23, Menú Alumno

La jerarquía de menú que presenta el sistema para el usuario administrador es la siguiente:

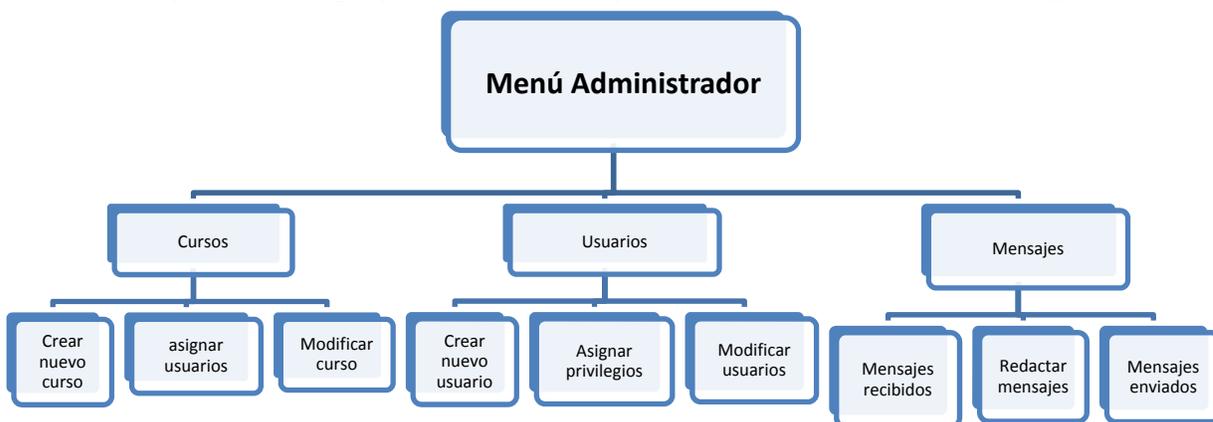


Fig. 24, jerarquía menú Administrador

La jerarquía de menú que presenta el sistema para el usuario Profesor y/o Ayudante es la siguiente:

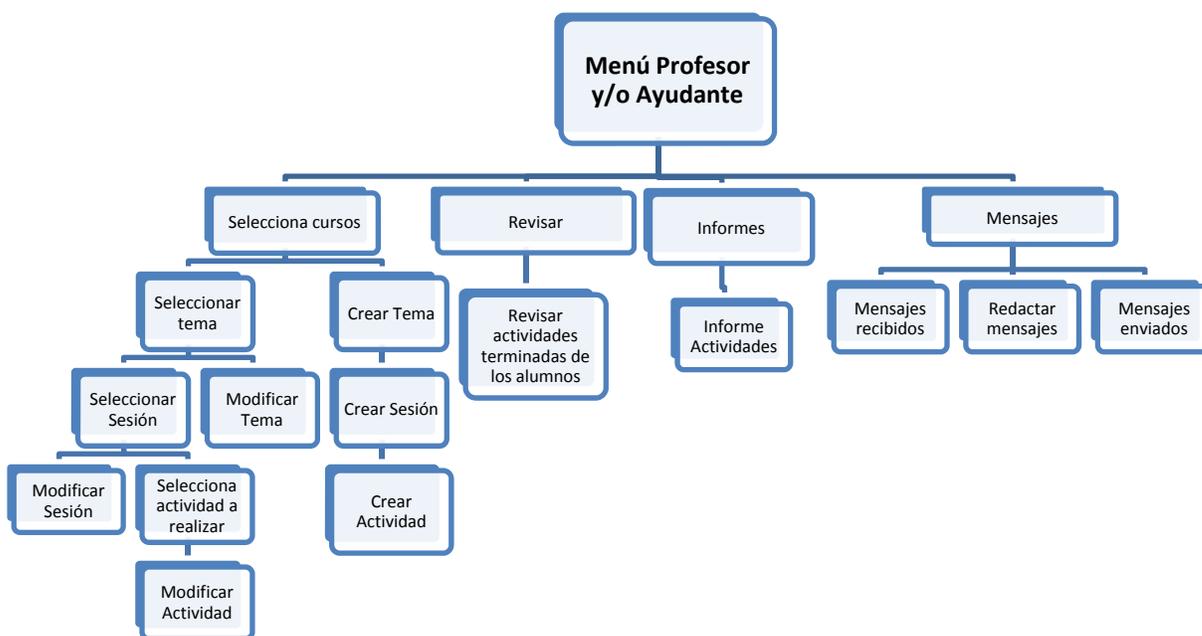


Fig. 25, jerarquía menú Profesor y/o Ayudante

La jerarquía de menú que presenta el sistema para el usuario Profesor y/o Ayudante es la siguiente:

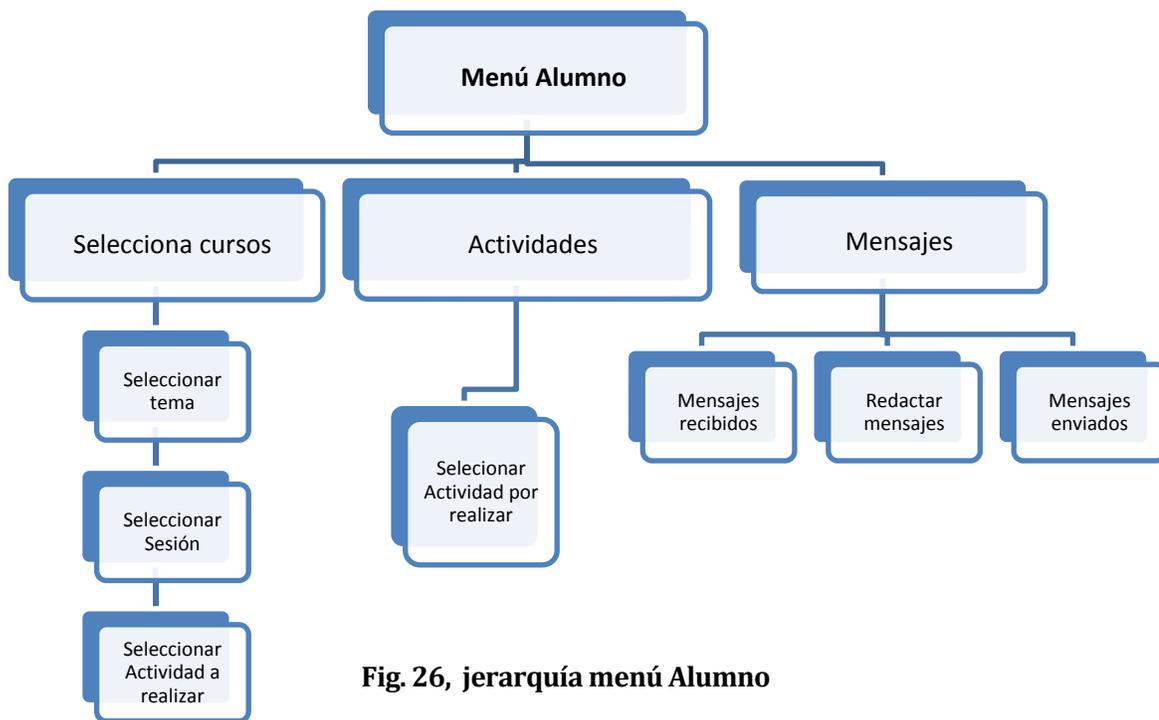


Fig. 26, jerarquía menú Alumno

Contenido

En esta sección se despliegan los formularios para ingreso de datos al sistema y la información referente a un curso, los temas, las sesiones, las actividades dependiendo la opción que el usuario ha seleccionado en el menú lateral y el tipo de usuario que ingreso al sistema.

A continuación se presentan las distribuciones de los elementos del sistema en maquetas de representación web.

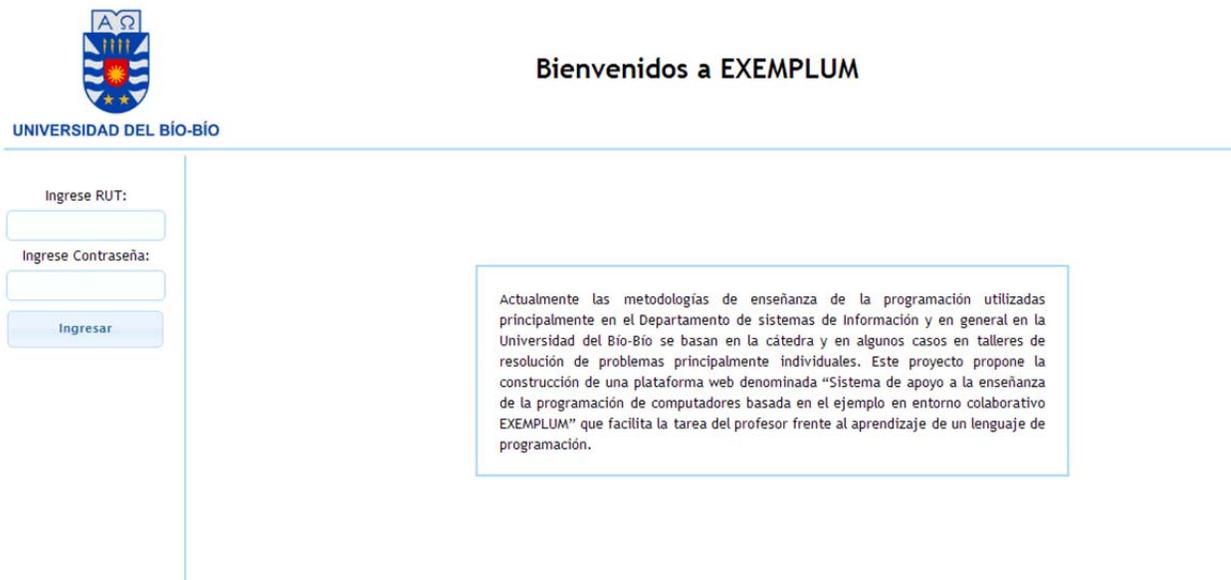


Fig. 27, Pantalla Bienvenida Sistema EXEMPLUM

Pantalla de bienvenida (Fig. 27): contiene los elementos básicos de la cabecera (logo, nombre institución, nombre sistema), en la parte del menú tiene el formulario de ingreso al sistema, y en la parte de contenidos, tiene una pequeña descripción de lo que es **EXEMPLUM**.



Fig. 28, Panel de Usuario

Panel de Usuario (Fig. 28): se despliega la información referente al usuario.



Fig. 29, Mensajes

Mensajes (Fig. 29): esta opción en el menú es la única que para todos los tipos de usuario aparecerá siempre, su finalidad es desplegar los últimos mensajes de entradas, las opciones para redactar un nuevo mensaje y ver aquellos enviados.



Fig. 30, Mensaje

Mensaje (Fig. 30): despliega el usuario que envió el mensaje, el texto de éste y el asunto que lo convoca.

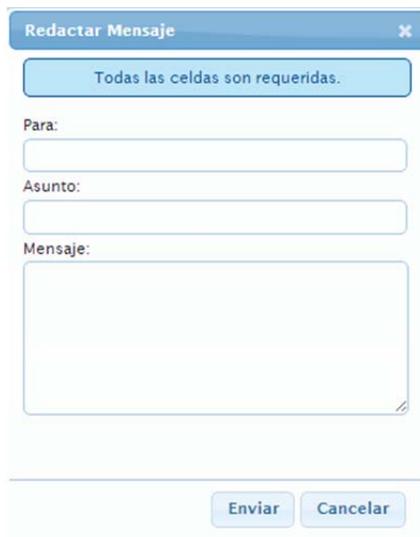


Fig. 31, Redactar o Responder Mensajes

Redactar o Responder Mensajes (Fig. 31): se despliega al usuario el formulario correspondiente, con las celdas requeridas para el mensaje (destinatario, asunto y mensaje) y las opciones correspondientes a enviar el mensaje o cancelar respectivamente.

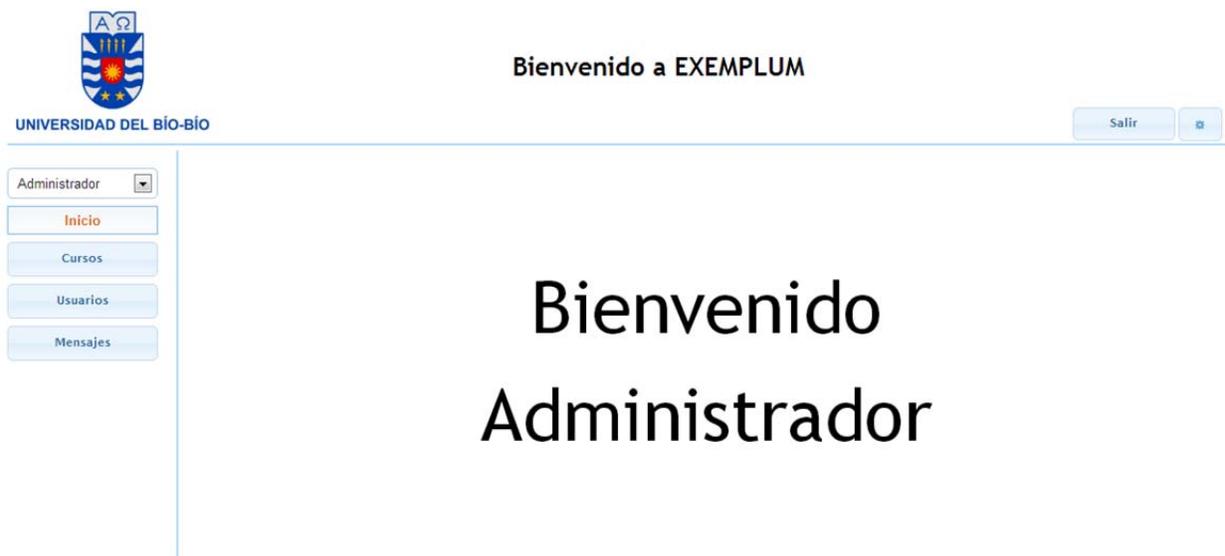


Fig. 32, Pantalla Bienvenida Administrador

Pantalla Bienvenida Administrador (Fig. 32): muestra un mensaje de bienvenida al administrador y le despliega sus opciones de trabajo en el menú lateral izquierdo.

Cursos

Salir

Listado de cursos: Nuevo Curso Cursos Eliminados

Código	Nombre	Descripción	Lenguaje programación	Año	Semestre		
IP	Introducción a la Programación	Este es un ramo entretenido :P	C	2012	2	Editar	Eliminar
01	Programación C 1	Primer curso programación C	C	2013	1	Editar	Eliminar
02	Programación JAVA 1	Primer curso programación JAVA. -Conceptos básicos	Java	2013	1	Editar	Eliminar
03	Programación C 2	Segundo curso programación C. - Conceptos avanzados.	C	2013	1	Editar	Eliminar
04	Programación JAVA 2	Segundo curso programación JAVA-Conceptos avanzados.	Java	2013	1	Editar	Eliminar

Fig. 33, Administrador Cursos

Administrador Cursos (Fig. 33): se listan todos los cursos dentro del sistema, con sus respectivas opciones de modificar y eliminar. También el botón de acceso a crear nuevo curso.

Crear Nuevo Curso

Fig. 34, Crear Nuevo Curso

Crear Nuevos Cursos (Fig. 34): se despliega el formulario para ingreso de nombre y seleccionador de lenguaje de programación, profesores, ayudante y alumnos.

Usuarios

UNIVERSIDAD DEL BÍO-BÍO

Salir

Administrador

Inicio

Cursos

Usuarios

Mensajes

Listado de Usuarios: Nuevo Usuario Cargar Archivo Usuarios Eliminados

Rut	Nombre	Apellido Paterno	Apellido Materno	Tipo		
111111111	prueba	prueba	prueba	Normal	Editar	Eliminar
173947755	Jonathan	Fierro	Sandoval	Administrador	Editar	Eliminar
175705228	Ivan Edgardo	Nawrath	Castillo	Administrador	Editar	Eliminar
222222222	prueba	prueba	prueba	Normal	Editar	Eliminar

Fig. 35, Administrador Usuarios

Administrador Usuarios (Fig. 35): se listan los usuarios con sus datos y las opciones de modificar y eliminar respectivamente. También la opción de crear uno nuevo, cargarlos desde un archivo y ver los usuarios eliminados.

Crear nuevo usuario

todas las celdas con requeridas.

Rut

Nombres

Apellido Paterno

Apellido Materno

Password

Tipo

Normal

Guardar Cancel

Fig. 36, Crear Nuevo Usuario

Crear Nuevo Usuario (Fig. 36): se despliega el formulario para ingresar los datos del nuevo usuario.

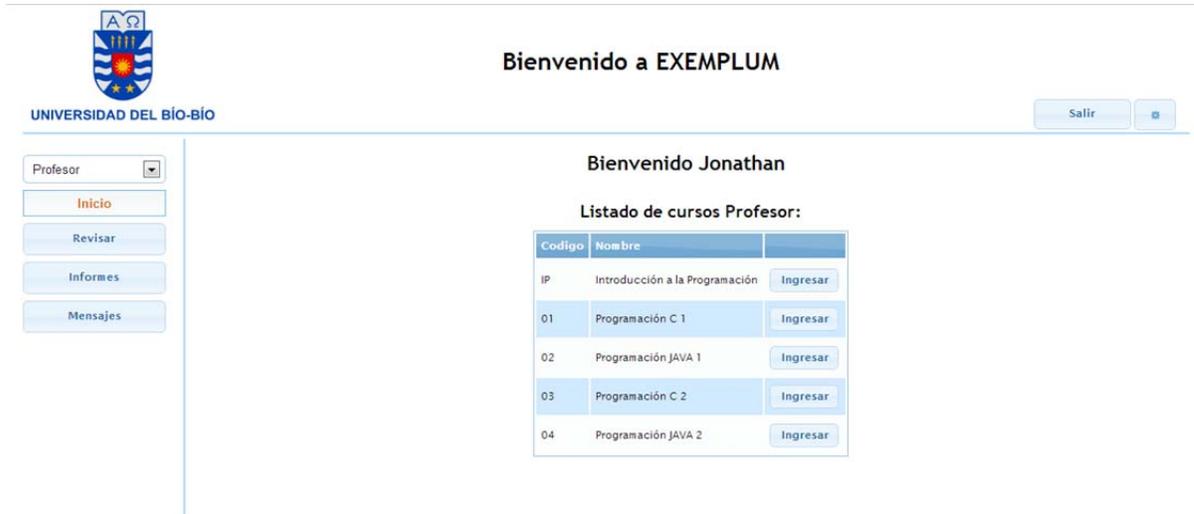


Fig. 37, Pantalla Bienvenida Profesor

Pantalla Bienvenida Profesor (Fig. 37): se despliega un mensaje de bienvenida y se le listan los cursos al cual pertenece como profesor, para que pueda seleccionar y realizar las modificaciones que estime conveniente.



Fig. 38, Profesor Curso

Profesor Curso (Fig. 38): se listan los temas con las opciones de editar o eliminar, también pueden ser expandidos y ver las sesiones que contiene el tema, las cuales también tienen las opciones de editar o eliminar.

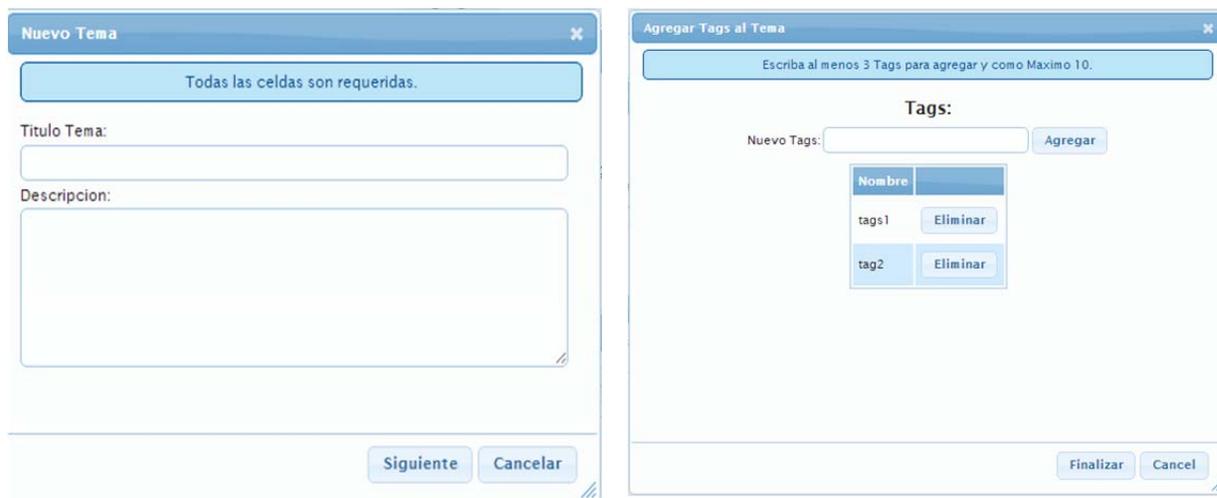


Fig. 39, Crear Nuevo Tema

Crear Nuevo Tema (Fig. 39): se despliega el formulario de ingreso de datos del temam, junto al listado de etiquetas que se le van asignando para una futura búsqueda.

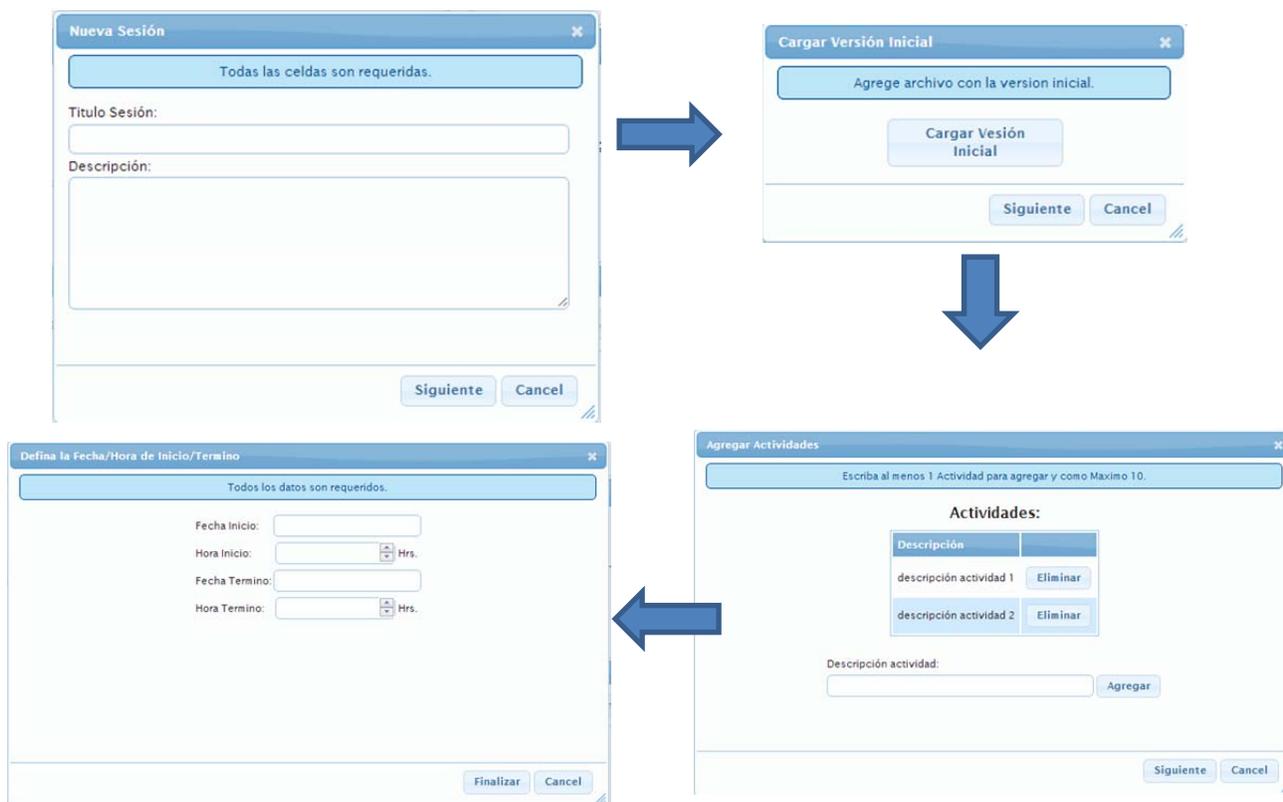


Fig. 40, Crear Nueva Sesión

Crear Nueva Sesión (Fig. 40): se despliega el formulario con los datos correspondientes a la sesión, junto al listado de actividades que el alumno deberá realizar dentro de esta.



Fig. 41, Profesor Revisar

Profesor Revisar (Fig. 41): 2 listados de actividades por revisar, el de la izquierda son las No calificadas y el de la derecha están todas las revisadas en caso de corrección de alguna calificación.



Fig. 42, Profesor alumnos actividad revisar o corregir

Profesor alumnos actividad revisar o corregir (Fig. 42): se despliegan al profesor los alumnos de la actividad a revisar.



Fig. 43, Profesor Calificar Actividad

Profesor Calificar Actividad (Fig. 43): se despliega al profesor la actividad del alumno, y las opciones para calificar.

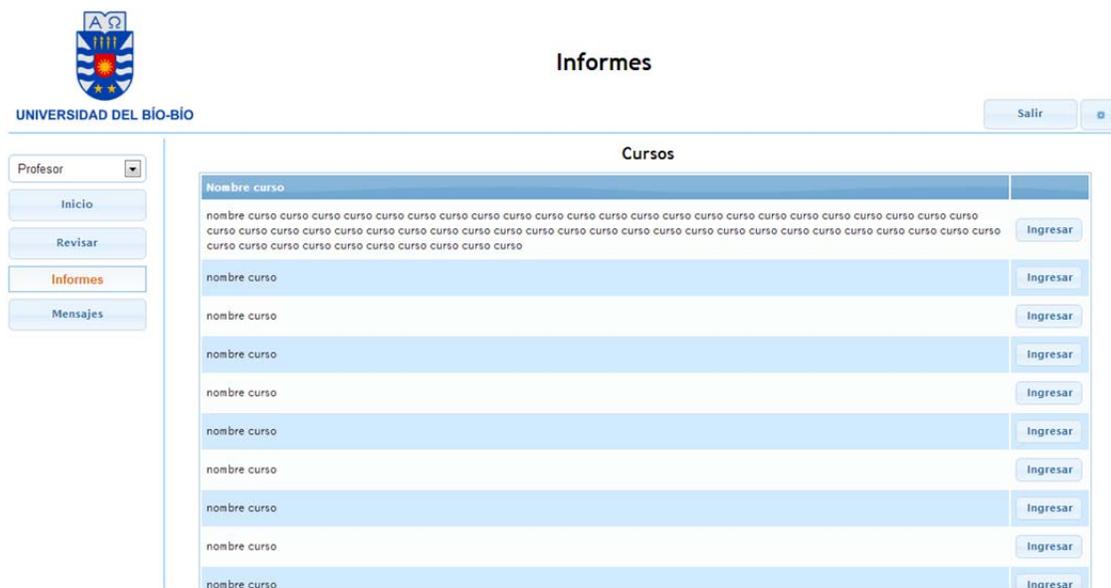


Fig. 44, Profesor Informes

Profesor Informes (Fig. 44): se listan los cursos del profesor con la opción de ver informe, el cual es de las actividades de los alumnos en comparación con el curso.



Fig. 45, Listado alumnos para ver informe

Listado alumnos para ver informe (Fig. 45): se despliega el listado de alumnos del curso seleccionado en la pantalla anterior para visualizar su gráfico.

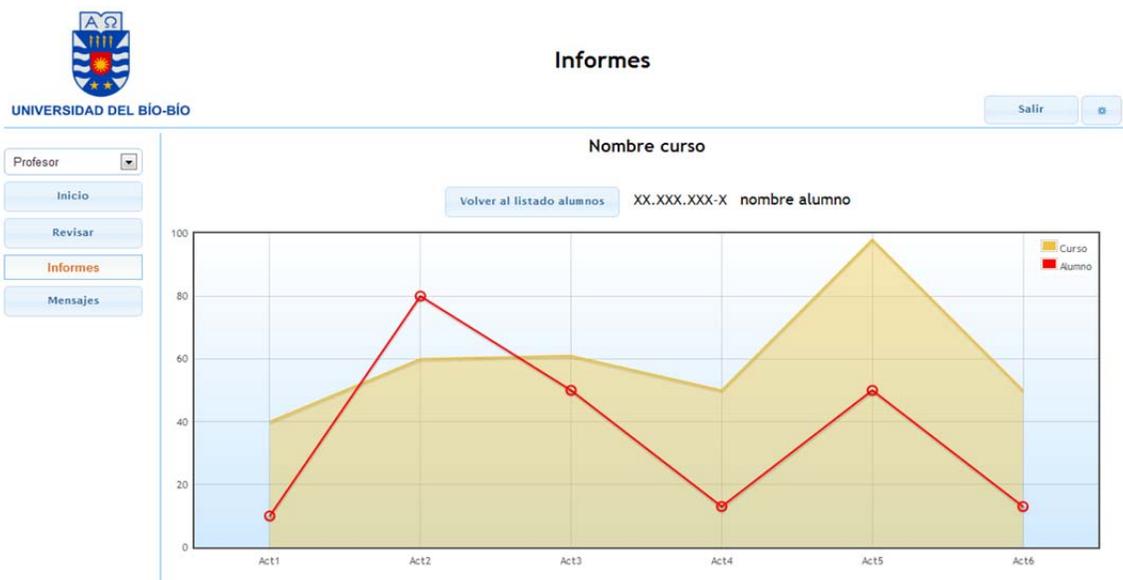


Fig. 46, Informe Actividad alumno

Informe Actividad Alumno (Fig. 46): se despliega un gráfico en comparación con la media del curso al cual pertenece el alumno.

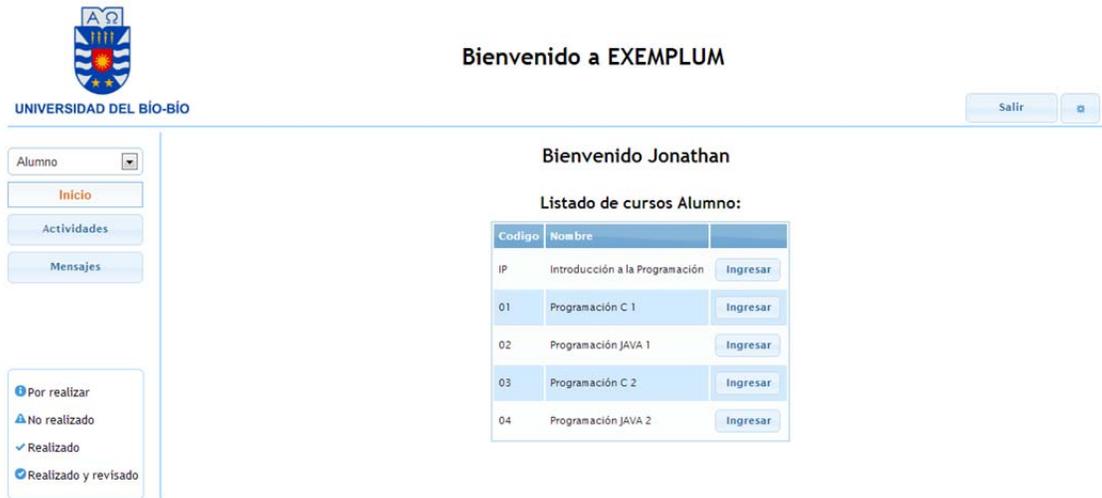


Fig. 47, Pantalla Bienvenida Alumno

Pantalla Bienvenida Alumno (Fig. 47): se despliega un mensaje de bienvenida al alumno y se le listan los cursos a los que pertenece para que ingrese.



Fig. 48, Alumno Curso

Alumno Curso (Fig. 48): dentro del curso el alumno puede ver todos los temas que lo componen e ingresar a las sesiones a desarrollar correspondientes actividades.

Sesión

UNIVERSIDAD DEL BÍO-BÍO

Fecha inicio: 31-01-2013 Hora inicio: 00:00:00 **sesión 1** Fecha término: 31-01-2013 Hora término: 23:59:00

Alumno [dropdown]

Inicio

Actividades

Mensajes

descripción sesión 1

Código base: Ver Código

Listado de Actividades:

Descripción		
Realice hola mundo	?	Realizar
Que el hola mundo se repita 2 veces	?	Realizar

Por realizar
 No realizado
 Realizado
 Realizado y revisado

Fig. 49, Alumno Sesión

Alumno Sesión (Fig. 49): dentro de la sesión seleccionada, el alumno puede ver la descripción de esta, la fecha de inicio y término, un ejemplo inicial (versión inicial), las actividades a desarrollar, la calificación que poseen cuando la actividad está terminada y revisada.



Fig. 50, Alumno Realiza Actividad

Alumno Realiza Actividad (Fig. 51): se despliega al alumno el problema a resolver, las actividades relacionadas que realizo antes, que están relacionadas al tema y las opciones de escribir el código en la misma pagina o cargar desde un archivo.



Fig. 51, Alumno Actividades

Alumno Actividades (Fig. 53): se despliega al alumno un listado con las actividades y la opción de realizar.

7.4 Especificación de módulos

A continuación se detalla cada uno de los procesos del último nivel de descomposición del diseño arquitectónico funcional que deberán corresponder a los módulos de programas que serán construidos en la codificación.

Nombre módulo: Validar usuario			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
Rut	Numérico	Usuario	Objeto
Contraseña	Texto		

Tabla 10 Especificación de módulos: Validar usuario

Nombre módulo: Habilitar usuario			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
Rut	Numérico	Usuario	Objeto

Tabla 11 Especificación de módulos: subir archivo

Nombre módulo: Deshabilitar usuario			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
Rut	Numérico		

Tabla 12 Especificación de módulos: Deshabilitar usuario

Nombre módulo: Validar campos vacíos			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
Formulario	Objeto	Campos vacíos	Objeto

Tabla 13 Especificación de módulos: Validar campos vacíos

Nombre módulo: Comprobar contraseña			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
Contraseña	Texto	Valido	Booleano

Tabla 14 Especificación de módulos: Comprobar contraseña

Nombre módulo: Comprobar conexión			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
		Valido	Booleano

Tabla 15 Especificación de módulos: Comprobar conexión

Nombre módulo: Cargar archivo			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
Archivo	Plantilla	Cargado correcto	Booleano
Archivo	Código		

Tabla 16 Especificación de módulos: Cargar archivo

Nombre módulo: Validar emisor			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
Rut	Numérico	Valido	Booleano

Tabla 17 Especificación de módulos: Comprobar contraseña

Nombre módulo: Validar receptor			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
Rut	Numérico	Valido	Booleano

Tabla 18 Especificación de módulos: Comprobar contraseña

Nombre módulo: Buscar Profesor			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
Rut	Numérico	Profesor	Objeto

Tabla 19 Especificación de módulos: Buscar Profesor

Nombre módulo: Buscar Ayudante			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
Rut	Numérico	Ayudante	Objeto

Tabla 20 Especificación de módulos: Buscar Ayudante

Nombre módulo: Buscar Alumno			
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo de dato	Nombre	Tipo de datos
Rut	Numérico	Alumno	Objeto

Tabla 21 Especificación de módulos: Buscar Ayudante

8 PRUEBAS

En este capítulo se describen las pruebas realizadas al sistema **EXEMPLUM**.

8.1 Elementos de prueba

A continuación se detallan los módulos a los que se realizan las pruebas de funcionalidad.

- **Crear Usuarios:** consiste en agregar usuarios al sistema mediante un formulario o a través de un archivo.
- **Redactar Mensajes:** consiste en escribir y enviar mensajes a distintos usuarios del sistema.
- **Crear Curso:** consiste en la creación de un curso para su realización en el sistema.
- **Eliminar Cursos:** consiste en la eliminación de un curso del sistema.
- **Crear Temas:** consiste en la creación de temas dentro de un curso creado con anterioridad en el sistema.
- **Crear Sesiones:** consiste en la creación de sesiones dentro de un tema creado con anterioridad en el sistema.
- **Subir Actividad:** consiste en la subida de un código al sistema para su próxima revisión y/o calificación.
- **Revisar Actividad:** consiste en la revisión de un código anteriormente subido por un alumno.

8.2 Especificación de las pruebas

A continuación se indican las características que serán probadas en el sistema.

- Las pruebas realizadas son de Unidad.
- El principal objetivo de los casos de pruebas es determinar que el sistema interactúe de correcta forma con los distintos usuarios, además de verificar que se cumplan los requerimientos funcionales.
- Se utiliza el enfoque de Caja Negra, el que consiste en ingresar datos y obtener una salida, la cual puede ser o no la esperada, determinando así el éxito o fracaso de prueba.
- Las técnicas de pruebas que se utilizara será la de “valores y límites y partición equivalente”.
- El criterio de cumplimiento que tienen estas pruebas es cuando se asegura el correcto funcionamiento según los requerimientos.

8.3 Responsables de las pruebas

Los responsables de realizar las pruebas de los sistemas son los desarrolladores Jonathan Fierro e Iván Nawrath.

8.4 Detalle de las pruebas

El detalle de las pruebas se adjunta en el anexo 16.

8.5 Conclusiones de Prueba

Las realizadas a los distintos módulos de fueron con un enfoque de caja negra, donde lo que importa es que el sistema entregue la información definida en los requerimientos del proyecto, según los datos que se ingresan y las respuestas que estos provocaban.

Los datos usados en las pruebas son obtenidos de casos reales de la universidad, por lo que al arrojar un resultado satisfactorio, se puede confiar en que el sistema funciona de manera correcta.

Los resultados de las pruebas con enfoque de caja negra de los módulos arrojaron la salida esperada, fallando al ingresar datos erróneos y funcionando cuando eran correctos. Los módulos presentaron un comportamiento estable durante el ingreso masivo de datos.

Por lo tanto se puede concluir que los sistemas están validando los datos erróneos y presenta el comportamiento esperado. De esta forma se puede da paso a la implementación y puesta en marcha del proyecto.

9 PLAN DE CAPACITACIÓN Y ENTRENAMIENTO

En este capítulo se indica la capacitación a realizar a los usuarios del sistema **EXEMPLUM**, con la cual se les enseñara utilizar las funcionalidades de manera sencilla.

Los usuarios finales que utilizarán el sistema y necesitan ser capacitados son:

- Administrador
- Profesores
- Ayudantes
- Alumnos

El tipo de capacitación realizado **será en el caso del administrador y profesores en su puesto de trabajo, a los alumnos y ayudantes será durante el desarrollo de la primera clase.**

Para la realización de la capacitación se deben diferenciar las funcionalidades que realizan los distintos tipos de usuarios, se organizaron como se muestra a continuación:

Administrador:

Funcionalidad	Tiempo capacitación
Ingresar al sistema	10 minutos
Actualizar datos	15 minutos
Redactar mensajes	15 minutos
Recibir mensajes	10 minutos
Responder mensajes	10 minutos
Salir del sistema	5 minutos
Crear Curso	20 minutos
Eliminar Curso	5 minutos
Modificar Curso	20 minutos
Agrega usuario	20 minutos
Eliminar usuario	5 minutos
Modificar usuario	20 minutos

Tabla 23 Detalle de tiempo requerido para capacitar al Administrador

Profesores o Ayudantes:

Funcionalidad	Tiempo capacitación
Ingresar al sistema	10 minutos
Actualizar datos	15 minutos
Redactar mensajes	15 minutos
Recibir mensajes	10 minutos
Responder mensajes	10 minutos
Salir del sistema	5 minutos
Seleccionar Curso	5 minutos
Crear Tema	20 minutos
Eliminar Tema	5 minutos
Modificar Tema	20 minutos
Crear Sesión	20 minutos
Eliminar Sesión	5 minutos
Modificar Sesión	20 minutos
Finalizar Sesión	10 minutos
Revisar actividades	20 minutos
Calificar actividades	30 minutos
Informes de actividad alumnos	30 minutos

Tabla 24 Detalle de tiempo requerido para capacitar a los Profesores/Ayudantes.

Alumnos:

Funcionalidad	Tiempo capacitación
Ingresar al sistema	10 minutos
Actualizar datos	15 minutos
Redactar mensajes	15 minutos
Recibir mensajes	10 minutos
Responder mensajes	10 minutos
Salir del sistema	5 minutos
Ingresar Curso	5 minutos
Ingresar Tema	5 minutos
Ingresar Sesión	5 minutos
Realizar actividades	30 minutos
Subir actividades	10 minutos
Ver actividades Anteriores	20 minutos

Tabla 25 Detalle de tiempo requerido para capacitar a los Alumnos.

Los responsables de realizar la capacitación son Jonathan Fierro Sandoval e Iván Nawrath Castillo desarrolladores del sistema, apoyados por el profesor guía Christian Vidal Castro.

Los recursos necesarios para realizar la capacitación son:

- 1 Proyector
- 1 Computador por usuario

10 PLAN DE IMPLANTACIÓN Y PUESTA EN MARCHA

El proyecto se implementó en un servidor institucional, para que cada modulo de creación fuera probado independiente de la etapa de desarrollo, ya que este se realizo mediante la metodología incremental y de esta forma es posible probar los módulos cuando estén terminados y no es necesario esperar al término total del proyecto.

Según la planificación de la Carta Gantt El sistema se debería poner en marcha durante el transcurso de un día y su instalación fue contemplada para ser realizada en 8 hrs.

La mantención será responsabilidad de los desarrolladores.

11 RESUMEN ESFUERZO REQUERIDO

En la **Tabla 26** Resumen de Esfuerzo Requerido se indica el tiempo real usado para desarrollar el proyecto.

Actividades/fases	N° Horas
Reunión sobre planeación proyecto y obtener requerimientos	2
Análisis requerimientos	22
Diseñar Base de datos	31
Determinar Arquitectura de desarrollo	32
Diseño interfaz	38
Integración de los conceptos de enseñanza basada en ejemplos	86
Programación	207
Despliegue	9
Pruebas de usuario	20
Análisis de resultados	40
Documentación	72
TOTAL	559

Tabla 26 Resumen de esfuerzo requerido

12 CONCLUSIONES

Una vez terminado el proyecto se puede concluir respecto a los objetivos planteados inicialmente, los cuales son “Diseñar y Desarrollar una plataforma de apoyo al aprendizaje de la programación, que utiliza un enfoque basado en ejemplos en entorno colaborativo, proponer los componentes del software que permitan realizar la enseñanza de la programación basada en ejemplos, usando como base un sistema desarrollado anteriormente, construir el sistema basándose en un patrón de arquitectura de software (Modelo Vista Controlador “MVC” orientando a Objetos en PHP), permitiendo la realización de código organizado, adaptable, un mantenimiento sencillo y un fácil entendimiento para futuros desarrollos”

Estos fueron cumplidos a cabalidad, sistema se desarrolló basándose en el patrón de arquitectura de software Modelo Vista Controlador “MVC” orientado a objetos en PHP, utilizando librerías PHP como PDO (PHP DATA OBJECT), la cual permite un mejor manejo de las consultas SQL y Protección de la base de datos. Por el lado del cliente se utilizó Framework JavaScript llamado JQuery y JQuery UI con su tema Redmond, para un mayor manejo y mejor visualización de la interfaz. También para la distribución de contenidos se utilizó “960 Grid System” Framework para maquetación CSS. Todo esto permitió un desarrollo más limpio y más organizado del sistema **EXEMPLUM** lo cual permitirá añadir nuevas funcionalidades cuando se estime conveniente.

Los conocimientos adquiridos durante la carrera de *Ingeniería de Ejecución en Computación e Informática* fueron la base necesaria para desarrollar el proyecto de forma exitosa y comprender de manera correcta los nuevos conceptos que aparecían al leer sobre las herramientas, Framework y librerías utilizadas durante el desarrollo.

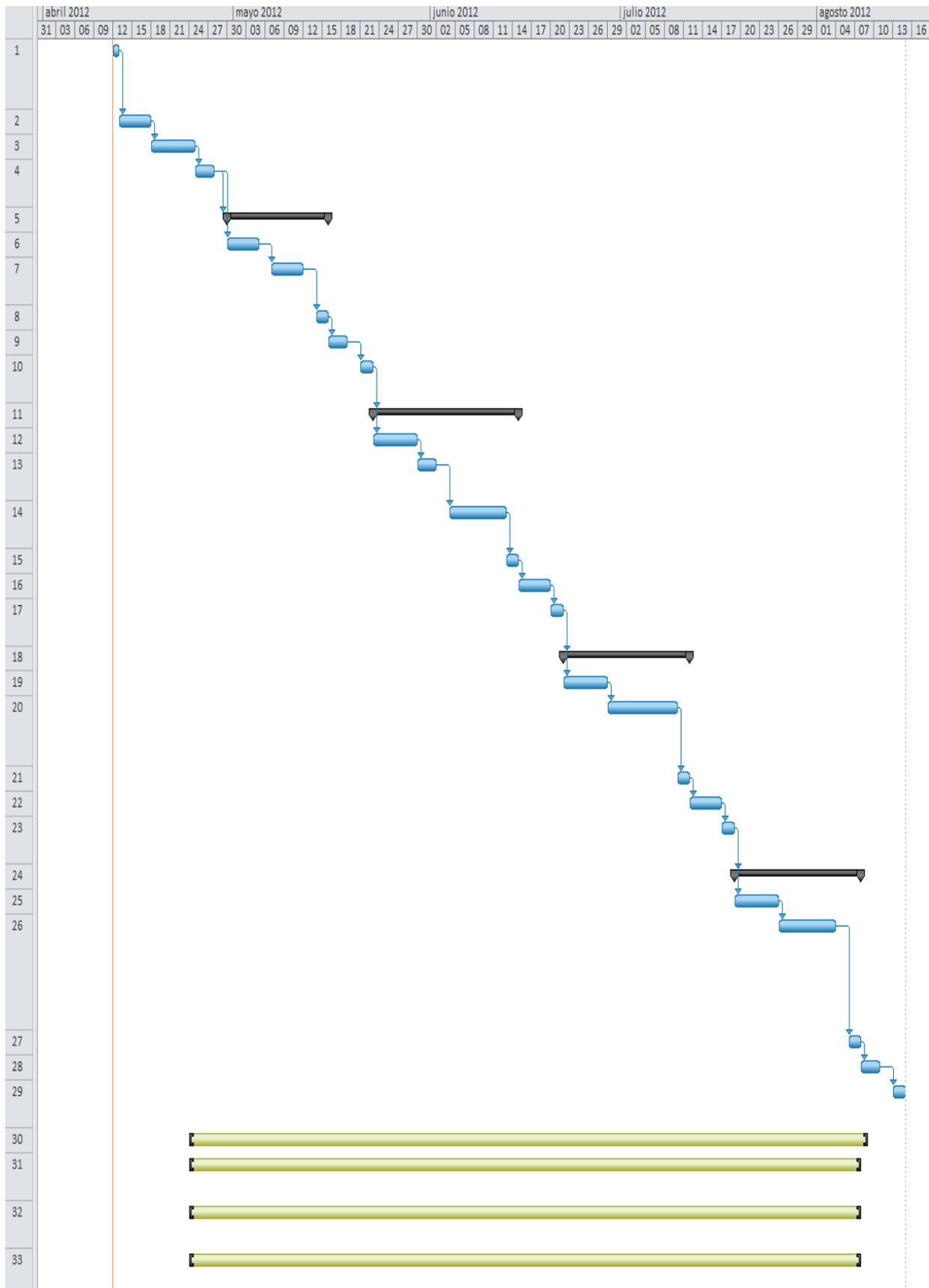
13 BIBLIOGRAFÍA

- Reigeluth, C. (1999). *Instructional-Design Theories and Models, Volume II: A New Paradigm of Instructional Theory*. Mahwah, NJ: Lawrence Erlbaum Assoc.
- Schank, R., Berman, T., & Macpherson, K. (1999). *Learning by Doing*. In C. Reigeluth (Ed.), *Instructional-Design Theories and Models. A New paradigm of Instructional Theory. Volume II.* : Lawrence Erlbaum Assoc.
- Johnson, D. W., Johnson, R., & Holubec, E. *Circles of learning* (4th ed.). Edina, MN: Interaction Book Company, 1993.
- Pressman Roger S. *Ingeniería de Software: Un Enfoque Práctico*, McGraw-Hill Interamericana, 6^a Edición, 2005.
- Minera, Francisco José *Php Avanzado: Domine El Lenguaje Más Confiable Y Estable*, Fox Andina, 1^a Edición, 2011.

14 ANEXO: PLANIFICACION INICIAL DEL PROYECTO

A continuación se da a conocer la calendarización programada.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Reunión sobre planeación	1 día	jue 12-04-12	jue 12-04-12	
2	Revisar requerimientos	3 días	vie 13-04-12	mar 17-04-12	1
3	Diseñar Base de Datos	5 días	mié 18-04-12	mar 24-04-12	2
4	Determinar Arquitectura	3 días	mié 25-04-12	vie 27-04-12	3
5	1° incremento	12 días	lun 30-04-12	mar 15-05-12	4
6	Diseño interfaz	5 días	lun 30-04-12	vie 04-05-12	4
7	Adaptación laborac a i	5 días	lun 07-05-12	vie 11-05-12	6
8	Despliegue	2 días	lun 14-05-12	mar 15-05-12	7
9	Pruebas de usuario	3 días	mié 16-05-12	vie 18-05-12	8
10	Analisis resultado prueba	2 días	lun 21-05-12	mar 22-05-12	9
11	2° incremento	17 días	mié 23-05-12	jue 14-06-12	10
12	arreglos post pruebas	5 días	mié 23-05-12	mar 29-05-12	10
13	Estudio Objetos de Ap	3 días	mié 30-05-12	vie 01-06-12	12
14	integración del conce	7 días	lun 04-06-12	mar 12-06-12	13
15	Despliegue	2 días	mié 13-06-12	jue 14-06-12	14
16	Pruebas de usuario	3 días	vie 15-06-12	mar 19-06-12	15
17	Analisis resultado prueba	2 días	mié 20-06-12	jue 21-06-12	16
18	3° incremento	14 días	vie 22-06-12	mié 11-07-12	17
19	arreglos post pruebas	5 días	vie 22-06-12	jue 28-06-12	17
20	Desarrollo de evaluad	7 días	vie 29-06-12	lun 09-07-12	19
21	Despliegue	2 días	mar 10-07-12	mié 11-07-12	20
22	Pruebas de usuario	3 días	jue 12-07-12	lun 16-07-12	21
23	Analisis resultado prueba	2 días	mar 17-07-12	mié 18-07-12	22
24	4° incremento	14 días	jue 19-07-12	mar 07-08-12	23
25	arreglos post pruebas	5 días	jue 19-07-12	mié 25-07-12	23
26	Desarrollo de generad	7 días	jue 26-07-12	vie 03-08-12	25
27	Despliegue	2 días	lun 06-08-12	mar 07-08-12	26
28	Pruebas de usuario	3 días	mié 08-08-12	vie 10-08-12	27
29	Analisis resultado prueba	2 días	lun 13-08-12	mar 14-08-12	28
30	Documentación	77 días	mar 24-04-12	mié 08-08-12	
31	Analisis de requerimient	76 días	mar 24-04-12	mar 07-08-12	
32	Estudios de Ingenieria w	76 días	mar 24-04-12	mar 07-08-12	
33	Estudios sobre metodos	76 días	mar 24-04-12	mar 07-08-12	



14.1.1 Estimación inicial de tamaño

A continuación, se realiza la estimación de tamaño del proyecto, el cual se ejercitara el método de PUNTO DE CASO DE USO.

- 1. Clasificación de actores:** mediante el análisis de cada actor que se relaciona en el sistema podremos obtener el Factor de peso de los actores sin ajustar (UAW).

Tipo de actor	Descripción	Factor de peso	Número de actores	Total
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación	1	0	0
Intermedio	Otro sistema interactuando a través de un protocolo (ej. TCP/IP) o una persona interactuando a través de una interfaz en modo texto	2	0	0
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica (GUI).	3	4	12
UAW				12

- 2. Clasificación de casos de usos:** Analizaremos a cada caso de uso que se encuentre en el sistema para de obtener el Factor de peso de los CU sin ajustar (UUCW) que nos ayudara para el cálculo del Factor de los CU sin ajustar (UUCP).

Tipo de caso de uso	Descripción	Factor de peso	Número de CU	Total
Simple	1 - 3 transacciones	5	29	145
Intermedio	4 - 7 transacciones	10	7	70
Complejo	Más de 7 transacciones	15	0	0
UUCW				215

- 3. Factores técnicos:** Mediante la siguiente tabla se evalúa la complejidad de los modelos del sistema y así poder obtener el factor técnico (TFactor).

Factor	Descripción	Peso	Influencia	Resultado
T1	Sistema distribuido	2	2	4
T2	Objetivos de performance o tiempo de respuesta	1	6	6
T3	Eficiencia del usuario final	1	4	4
T4	Procesamiento interno complejo.	1	2	2
T5	El código debe ser reutilizable.	1	6	6
T6	Facilidad de instalación	0.5	2	1
T7	Facilidad de uso	0.5	6	3
T8	Portabilidad	2	1	2
T9	Facilidad de cambio	1	6	6
T10	Concurrencia	1	1	1
T11	Incluye objetivos especiales de seguridad	1	1	1
T12	Provee acceso directo a terceras partes	1	1	1
T13	Se requiere facilidades especiales de entrenamiento a usuario	1	1	1
TFactor				38

4. **Factores de entorno:** se evalúa la complejidad de los factores ambientales

Factor	Descripción	peso	Influencia	Resultado
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	4.5
E2	Experiencia en la aplicación	0.5	6	3
E3	Experiencia en orientación a objetos	1	2	3
E4	Capacidad del analista líder.	0.5	2	1
E5	Motivación	1	7	7
E6	Estabilidad de los requerimientos	2	4	8
E7	Personal part-time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	4	-4
EFactor				21,5

5. Cálculos Finales de Punto de Caso de Uso

- Calcular UUCP(Punto de Caso de Uso sin ajustar)

$$UUCP = UAW + UUCW$$

$$UUCP = 12 + 215$$

$$UUCP = 227$$

- Calcular TCF (Factor de Complejidad Técnica)

$$TCF = 0,6 + (0,1 * \text{Factor Total Técnico})$$

$$TCF = 0,6 + (0,01 * 38)$$

$$TCF = 0,98$$

- Calcular ECF (Factor de Complejidad Ambiental)

$$ECF = 1,4 + (-0,03 * \text{Factor Ambiental Total})$$

$$ECF = 1,4 + (-0,03 * 21,5)$$

$$ECF = 0,755$$

- Calculando los UCP

$$UCP = UUCP * TCF * ECF$$

$$UCP = 227 * 0,98 * 0,755$$

$$UCP = 167,9573$$

- Esfuerzo De Trabajo

Este cálculo se realiza para tener una aproximación del esfuerzo de grupo de trabajo, pensando en el desarrollo según las funcionalidades de los casos de uso.

Se deben contabilizar la cantidad de factores ambientales entre E1 Y E6 que tienen una puntuación menor a 3 y también contabilizar de E7 Y E8 los que tienen puntuación mayor a 3

$$E1 \text{ a } E6 \Rightarrow 1$$

$$E7 \text{ a } E8 \Rightarrow 1$$

- Se evalúa el resultado obtenido dependiendo de la siguiente tabla:

<i>Horas -Persona</i>	<i>Descripción de LOE</i>
20	Si el valor es ≤ 2
28	Si el valor es 3 0 4
Reconsiderar el Proyecto	Si el valor es ≥ 5

Entonces el **CF = 20**

- El esfuerzo Horas – Persona viene dado por

$$E = UCP * CF$$

$$E = 167,9573 * 20$$

$$E = 3359,146$$

- Por lo tanto el esfuerzo estimado en Horas – Persona es de

33459,146 horas/persona

- Así al contabilizar unas 8 horas de trabajo diarias, y teniendo en cuenta que son 2 desarrolladores, se estima que el esfuerzo en horas para cada persona es de:

1679 Horas/persona aproximado por desarrollador.

14.1.2 Contabilización final del tamaño del Software

El proyecto concluye con 7113 líneas de código implementadas, considerando todas las líneas, incluidos comentarios y líneas en blanco, siendo:

- 1708 correspondientes a HTML.
- 238 correspondientes a CSS.
- 3121 correspondiente a JavaScript (jQuery).
- 2046 correspondientes a PHP.

15 ANEXO: ESPECIFICACION DE LAS PRUEBAS

15.1 Pruebas de Unidad

A continuación se especifican las pruebas de unidad realizadas a los sistemas.

15.1.1 <Crear Usuarios>

Para realizar las pruebas es necesario tener en cuenta:

- Las pruebas se realizan con el navegador web Chrome de Google, teniendo habilitado JavaScript, sistema operativo Windows XP y una conexión de red de 2 Mbps a Internet.

Descripción	Mostrar formulario en pantalla para crear un nuevo usuario.
Precondiciones	⇒ El usuario debe tener permisos de Administrador.
Datos de entrada	⇒ Rut: 11111111-1 ⇒ Nombre: Jorge ⇒ Apellido Paterno: Bustos ⇒ Apellido Materno: González ⇒ Contraseña: Generada automáticamente por el sistema ⇒ Tipo: Alumno
Salida esperada	Mensaje de éxito al crear el usuario Jorge Bustos González.
Salida obtenida	Mensaje de éxito al crear el usuario.
Resultado	Éxito: El sistema arrojo el mensaje esperado.
Observaciones	

Tabla 27 Prueba 1

15.1.2 <Redactar mensajes>

Para realizar las pruebas es necesario tener en cuenta:

- Las pruebas se realizan con el navegador web Chrome de Google, teniendo habilitado JavaScript, sistema operativo Windows XP y una conexión de red de 2 Mbps a Internet.

Descripción	Mostrar el formulario en pantalla para enviar el mensaje.
Precondiciones	⇒ Deben existir al menos 2 usuarios en el sistema.
Datos de entrada	⇒ Para: Jorge Bustos González ⇒ Asunto: Asunto de prueba ⇒ Mensaje: "Este es un mensaje de prueba para el Usuario Jorge Bustos González"
Salida esperada	Mensaje de éxito el enviar el mensaje al usuario Jorge Bustos González.
Salida obtenida	Mensaje de éxito el enviar el mensaje.
Resultado	Éxito: El sistema arrojo el mensaje esperado.
Observaciones	

Tabla 28 Prueba 2

15.1.3 <Crear Cursos>

Para realizar las pruebas es necesario tener en cuenta:

- Las pruebas se realizan con el navegador web Chrome de Google, teniendo habilitado JavaScript, sistema operativo Windows XP y una conexión de red de 2 Mbps a Internet.

Descripción	Muestra el formulario para crear el curso y asignar los accesos necesarios.
Precondiciones	⇒ El usuario debe tener permisos de Administrador. ⇒ Debe existir al menos 1 usuario Profesor. ⇒ Debe existir al menos 1 Usuario Alumno.
Datos de entrada	⇒ Nombre: Introducción a la Programación ⇒ Código: 620109 ⇒ Profesor: Selección a usuario 17570521-K ⇒ Ayudante: no seleccionado ningún usuario. ⇒ Alumno: Selección a Usuario 11111111-1 ⇒ Año: 2012 ⇒ Semestre: 2 ⇒ Lenguaje: Java
Salida esperada	Mensaje de éxito al crear el curso “Introducción a la programación”.
Salida obtenida	Mensaje de éxito al crear el curso.
Resultado	Éxito: el sistema arrojó el mensaje esperado.
Observaciones	

Tabla 29 Prueba 3

15.1.4 <Eliminar Cursos>

Para realizar las pruebas es necesario tener en cuenta:

- Las pruebas se realizan con el navegador web Chrome de Google, teniendo habilitado JavaScript, sistema operativo Windows XP y una conexión de red de 2 Mbps a Internet.

Descripción	Muestra el formulario con el listado de los cursos para su posible eliminación.
Precondiciones	⇒ El usuario debe tener permisos de Administrador. ⇒ Debe existir al menos 1 Curso.
Datos de entrada	⇒ Curso a eliminar: - Introducción a la programación. - Código: 620109
Salida esperada	Mensaje de éxito al eliminar el curso “Introducción a la programación”
Salida obtenida	Mensaje de éxito al eliminar el curso.
Resultado	Éxito: el sistema arrojó el mensaje esperado.
Observaciones	

Tabla 30 Prueba 4

15.1.5 <Crear Temas>

Para realizar las pruebas es necesario tener en cuenta:

- Las pruebas se realizan con el navegador web Chrome de Google, teniendo habilitado JavaScript, sistema operativo Windows XP y una conexión de red de 2 Mbps a Internet.

Descripción	Muestra el formulario para la creación de un tema dentro de un curso.
Precondiciones	<ul style="list-style-type: none"> ⇒ El curso debe estar creado con anterioridad. ⇒ El usuario debe tener permisos de Profesor o Ayudante, y pertenecer al curso.
Datos de entrada	<ul style="list-style-type: none"> ⇒ Título: Introducción a punteros ⇒ Descripción: “En este tema se trabaja lo fundamental en punteros, como reconocerlos y trabajar de manera sencilla con ellos”. ⇒ Etiquetas: “Punteros”, “C”.
Salida esperada	Mensaje de éxito al crear el tema de “Introducción a punteros”
Salida obtenida	Mensaje de éxito al crear el tema.
Resultado	Éxito: el sistema arrojo el mensaje esperado.
Observaciones	

Tabla 31 Prueba 5

15.1.6 <Crear Sesiones>

Para realizar las pruebas es necesario tener en cuenta:

- Las pruebas se realizan con el navegador web Chrome de Google, teniendo habilitado JavaScript, sistema operativo Windows XP y una conexión de red de 2 Mbps a Internet.

Descripción	Muestra el formulario para la creación de una sesión dentro de un tema.
Precondiciones	<ul style="list-style-type: none"> ⇒ El tema debe estar creado con anterioridad. ⇒ El usuario debe tener permisos de Profesor o Ayudante, y pertenecer al curso.
Datos de entrada	<ul style="list-style-type: none"> ⇒ Título: “Punteros Parte 1” ⇒ Descripción: “En esta versión se trabaja con la declaración de punteros, operaciones aritméticas, y llamadas a funciones pasando parámetros por referencia.” ⇒ Actividad 1: “Realizar un programa que demuestre la declaración de punteros e impresión por pantalla de punteros” ⇒ Actividad 2: “Realizar un programa con menú para las 4 operaciones aritméticas utilizando punteros” ⇒ Actividad 3: “Realizar un programa que permite la declaración y el llamado de funciones utilizando parámetros por referencias”
Salida esperada	Mensaje de éxito al crear la sesión “Punteros parte 1”
Salida obtenida	Mensaje de éxito al crear la sesión.
Resultado	Éxito: el sistema arrojo el mensaje esperado.
Observaciones	

Tabla 32 Prueba 6

15.1.7 <Subir Actividad>

Para realizar las pruebas es necesario tener en cuenta:

- Las pruebas se realizan con el navegador web Chrome de Google, teniendo habilitado JavaScript, sistema operativo Windows XP y una conexión de red de 2 Mbps a Internet.

Descripción	Muestra el formulario para la subida de código del alumno a la correspondiente actividad.
Precondiciones	⇒ La sesión debe estar creada con anterioridad. ⇒ El alumno debe pertenecer al curso donde esta la sesión.
Datos de entrada	⇒ Código: “ #include <stdio.h> #include <stdlib.h> int main() { printf("Hola mundo\n"); return 0; } ”
Salida esperada	Mensaje de éxito al subir el código correctamente al sistema.
Salida obtenida	Mensaje de éxito al subir el código.
Resultado	Éxito: el sistema arrojo el mensaje esperado.
Observaciones	

Tabla 33 Prueba 7

15.1.8 <Revisar Actividad>

Para realizar las pruebas es necesario tener en cuenta:

- Las pruebas se realizan con el navegador web Chrome de Google, teniendo habilitado JavaScript, sistema operativo Windows XP y una conexión de red de 2 Mbps a Internet.

Descripción	Muestra el formulario donde aparece el código para revisión.
Precondiciones	⇒ El Usuario debe tener permisos de Profesor o Ayudante.
Datos de entrada	⇒ Calificación: Barra de desplazamiento que otorgara una nota simbólica al alumno.
Salida esperada	Mensaje de éxito al calificar la Actividad.
Salida obtenida	Mensaje de éxito al calificar.
Resultado	Éxito: el sistema arrojo el mensaje esperado.
Observaciones	

Tabla 34 Prueba 8

16 ANEXO: DICCIONARIO DE DATOS DEL MODELO DE DATOS

A continuación se presentan las especificaciones de las tablas creadas para el proyecto.

Se omiten las tildes, y letras Ñ, en los Atributos y Tipo de Datos, para asegurar la compatibilidad de la Base de Datos.

Estructura de tablas y relaciones del sistema **EXEMPLUM**

- Usuarios

Atributo	Descripción	Tipo de Dato	Restricción
Rut	Identificador Único de un Usuario	Varchar(9)	Clave Primaria
Contrasena	Clave del Usuario	Varchar(100)	
Nombres	Primer y segundo nombre del Usuario	Varchar(100)	
Apellido_Paterno	Apellido Paterno del Usuario	Varchar(50)	
Apellido_Materno	Apellido Materno del Usuario	Varchar(50)	

- Mensajes

Atributo	Descripción	Tipo de Dato	Restricción
Id	Código Autoincremental	Integer	Clave Primaria
Asunto	Un pequeño resumen descriptivo del mensaje	Varchar(100)	
Mensaje	Contenido del mensaje	Varchar(1000)	
Fecha	Fecha del mensaje	Date	
Hora	Hora del mensaje	Time	

- Envíos

Atributo	Descripción	Tipo de Dato	Restricción
Id	Código Autoincremental	Integer	Clave Primaria
Estado	Verifica si el mensaje ha sido leído o no.	Tinyint(1)	
Fk_id_mensaje	Mensaje que ha sido enviado.	Integer	Clave Foránea (Mensajes)
Fk_rut_emisor	Usuario que envía el mensaje.	Varchar(9)	Clave Foránea (Usuarios)
Fk_rut_receptor	Usuario que recibe el mensaje	Varchar(9)	Clave Foránea (Usuarios)
Eliminado_emisor	Indica si el mensaje fue eliminado por el emisor	Tinyint	
Eliminado_receptor	Indica si el mensaje fue eliminado por el receptor	Tinyint	

- Administrador

Atributo	Descripción	Tipo de Dato	Restricción
Rut_adm	Identificador único de un administrador	Varchar(9)	Clave Primaria, Clave Foránea (Usuarios)

- Profesor

Atributo	Descripción	Tipo de Dato	Restricción
Rut_profesor	Identificador único de un profesor	Varchar(9)	Clave Primaria, Clave Foránea (Usuarios)

- Ayudante

Atributo	Descripción	Tipo de Dato	Restricción
Rut_ayudante	Identificador único de un ayudante	Varchar(9)	Clave Primaria, Clave Foránea (Usuarios)

- Alumno

Atributo	Descripción	Tipo de Dato	Restricción
Rut_alumno	Identificador único de un alumno	Varchar(9)	Clave Primaria, Clave Foránea (Usuarios)

- Cursos

Atributo	Descripción	Tipo de Dato	Restricción
Id	Código Autoincremental	Integer	Clave Primaria
Codigo	Código del curso	Varchar(20)	
Nombre	Nombre asignado al curso	Varchar(100)	
Descripcion	Breve descripción de la que tratara	Varchar(300)	
Anho	Año en que se dicta el curso	Integer	
Semestre	Semestre en la que se dicta el curso	Integer	
Lenguaje	Lenguaje de programación que se usara en el curso	Integer	
Estado	Identifica si esta eliminado o no	Tinyint	

- Apoya

Atributo	Descripción	Tipo de Dato	Restricción
Fk_rut_ayudante	Identificador único de un ayudante	Varchar(9)	Clave Primaria, Clave Foránea (Ayudante)
Fk_id_curso	Identificador del curso	Integer	Clave Primaria, Clave Foránea (Cursos)
Crear_temas	Esta o no autorizado para crear temas	Tinyint(1)	
Modificar_temas	Esta o no autorizado para modificar temas	Tinyint(1)	
Eliminar_temas	Esta o no autorizado para eliminar temas	Tinyint(1)	
Crear_sesiones	Esta o no autorizado para crear sesiones	Tinyint(1)	
Modificar_sesiones	Esta o no autorizado para modificar sesiones	Tinyint(1)	
Eliminar_sesiones	Esta o no autorizado para eliminar sesiones	Tinyint(1)	
Informes	Esta o no autorizado para revisar los informes.	Tinyint(1)	

- Pertenece

Atributo	Descripción	Tipo de Dato	Restricción
Fk_rut_alumno	Identificador único de un alumno	Varchar(9)	Clave Primaria, Clave Foránea (Alumno)
Fk_id_curso	Identificador del curso	Integer	Clave Primaria, Clave Foránea (Cursos)

- Imparte

Atributo	Descripción	Tipo de Dato	Restricción
Fk_rut_profesor	Identificador único de un profesor	Varchar(9)	Clave Primaria, Clave Foránea (Profesor)
Fk_id_curso	Identificador del curso	Integer	Clave Primaria, Clave Foránea (Cursos)

- Temas

Atributo	Descripción	Tipo de Dato	Restricción
Id	Código Autoincremental	Integer	Clave Primaria
Nombre	Nombre del tema	Varchar(100)	
Descripcion	Breve descripción de lo que tratara el tema a revisar.	Varchar(500)	
Fk_id_curso	Identifica a que curso corresponde el tema.	Integer	Clave Foránea (Cursos)

- Etiquetas

Atributo	Descripción	Tipo de Dato	Restricción
Id	Código Autoincremental	Integer	Clave Primaria
Nombre	Nombre de la etiqueta	Varchar(50)	

- Posee

Atributo	Descripción	Tipo de Dato	Restricción
Fk_id_tema	Identificador del tema	Integer	Clave Primaria, Clave Foránea (Temas)
Fk_id_etiqueta	Identificador de la etiqueta	Integer	Clave Primaria, Clave Foránea (Etiqueta)

- Sesiones

Atributo	Descripción	Tipo de Dato	Restricción
Id	Código Autoincremental	Integer	Clave Primaria
Nombre	Nombre de cada sesión	Varchar(100)	
Introduccion	Breve introducción al trabajo que revisaran en la sesión	Varchar(500)	
Fecha_inicio	Fecha en que empezara la sesión	Date	
Fecha_termino	Fecha de termino de la sesión	Date	
Hora_inicio	Hora de inicio de la sesión	Time	
Hora_termino	Hora de termino de la sesión	Time	
Codigo_base	Url del código fuente base para las versiones a trabajar en la sesión	Varchar(100)	
Fk_id_tema	Identifica a que tema corresponde cada tema.	Integer	Clave Foránea (Sesiones)

- Actividades

Atributo	Descripción	Tipo de Dato	Restricción
Id	Código Autoincremental	Integer	Clave Primaria
Nombre	Nombre de la actividad	Varchar(100)	
Fk_id_sesion	Identifica a que sesión pertenece cada actividad	Integer	Clave Foránea (Sesiones)

- Versiones

Atributo	Descripción	Tipo de Dato	Restricción
Id	Código Autoincremental	Integer	Clave Primaria
Nombre		Varchar(200)	
Url	Dirección donde esta guardada la versión	Varchar(200)	
Fecha	Fecha de subida	Date	
Hora	Hora de subida	Time	
Revisado	Permite saber si el archivo ha sido o no revisado por el profesor o ayudante.	Integer	
Calificacion	Corresponde a la nota obtenida por la versión. Esta nota es solo simbólica y no corresponde a notas del semestre	Float	
Fk_id_actividad	Identifica a que actividad corresponde la versión	Integer	Clave Foránea (Versiones)
Fk_rut_autor	Identifica al autor de la versión	Integer	Clave Foránea (Usuarios)

- Bitacora_Accesos

Atributo	Descripción	Tipo de Dato	Restricción
Id	Código Autoincremental	Integer	Clave Primaria
Rut	Corresponde al RUT del usuario que realizo el acceso	Integer	
Descripcion	Se refiere a que parte del sistema esta entrando, curso, tema, etc.,	Varchar(100)	
Fecha	Fecha del acceso	Date	
Hora	Hora del acceso	Time	
Url	Dirección que visito el usuario	Varchar(100)	

- Bitacora_Actividad

Atributo	Descripción	Tipo de Dato	Restricción
Id	Código Autoincremental	Integer	Clave Primaria
Rut	Corresponde al RUT del usuario que realizo alguna actividad	Integer	
Actividad		Varchar(200)	
Fecha	Fecha del acceso	Date	
Hora	Hora del acceso	Time	
Url	Dirección que visito el usuario	Varchar(100)	

17 ANEXO: PATRON DE DESARROLLO Y CODIGOS

Este anexo presenta de manera introductoria el patrón de diseño MVC y cómo llegamos a utilizarlo. Podrás encontrar los códigos PHP realizados en el proyecto desde sus comienzos.

17.1 ¿Qué es MVC?

MVC viene de Model, View, Controller, o bien: Modelo, Vista y Controlador. Es un patrón de diseño cuya idea básica es separar nuestros sistemas en 3 capas, El Modelo, La Vista y el Controlador.

El **Modelo** se encarga de todo lo que tiene que ver con la persistencia de datos. Guarda y recupera la información del medio persistente que utilizamos, ya sea una base de datos, ficheros de texto, XML, etc.

La **Vista** presenta la información obtenida con el modelo de manera que el usuario la pueda visualizar.

El **Controlador**, dependiendo de la acción solicitada por el usuario, es el que pide al modelo la información necesaria e invoca a la vista que corresponda para que la información sea presentada.

17.1.1 Un pequeño ejemplo:

1. “Usuario” entra a nuestro sitio mediante la URL www.example.com/items/listar.
2. Se carga el Controlador Items para ejecutar la acción de Listar.
3. El controlador solicita al modelo que le entregue un arreglo con todos los items que hay almacenados en la base de datos.
4. Una vez que posee dicha información le indica a la vista el correspondiente listado de ítems.
5. La vista, por su parte, toma el listado de items y los muestra uno a uno.
6. Finalmente “Usuario” recibe el listado de items; lo observa un instante y decide que quiere agregar un nuevo ítem, por lo que hace click en un enlace que lo lleva a la URL www.example.com/items/agregar.
7. Se repite el proceso desde el paso 1 pero con la nueva URL

17.2 Para comenzar PDO

PDO o PHP Data Objects es una interface de acceso a datos que nos permite, mediante varios drivers, conectarnos a diferentes bases de datos y olvidarse de esto, eso o incluso aquello. Esta librería escrita en C viene activada por defecto desde PHP 5.1 por lo cual se podrá utilizar en la mayoría de los servidores que actualmente soportan PHP5.

17.2.1 La conexión

Para todos los ejemplos se utilizara MySQL, pero también se podría utilizar cualquier otra de las bases de datos soportadas adaptando un poco el código que sigue:

```
1 $db = new PDO('driver:host=servidor;dbname=bd', user, pass);
```

El ejemplo práctico:

```
1 $db = new PDO('mysql:host=localhost;dbname=pruebas', 'root', '');
```

Ahora en `$db` tenemos una instancia de `PDO_MySQL`

17.2.2 Primera consulta

Para la primera consulta haremos uso de `prepare`, `execute` y `fetch`.

```
1 require 'conexion.php';
2
3 //Nos conectamos
4 $db = new PDO('mysql:host=' . $servidor . ';dbname=' . $bd, $usuario,
5 $contrasenia);
6
7 //Preparamos la consulta para dejarla lista para su ejecución
8 $consulta = $db->prepare('SELECT * FROM items');
9
10 //Ejecutamos la consulta
11 $consulta->execute();
12
13 //Recorremos el set de resultados mostrando la información
14 while($fila = $consulta->fetch())
15 {
16     echo $fila[0] . ' ' . $fila[1] . '<br />';
17 }
18
19 //Cerramos la conexión a la vez que destruimos nuestra instancia de PDO
20 $db = null;
```

Como se puede notar, no es muy complicado y es muy similar a lo que nos acostumbramos a hacer con las clásicas funciones `mysql_`.

Con las funciones MySQL también debíamos validar estrictamente los parámetros de entrada para evitar inyecciones SQL. En este caso, PDO lo hará por nosotros siempre y cuando utilicemos alguna de las varias formas que nos provee para realizar consultas parametrizadas.

Este es un ejemplo de una consulta preparada con PDO:

```

1 //Preparamos la consulta marcando donde irán los parametros con ?
2 $consulta = $db->prepare('SELECT * FROM items WHERE id_item = ? OR id_item =
  ?');
3
4 //Ejecutamos la consulta incluyendo los parámetros en el mismo orden en el que
  deben incluirse
5 $consulta->execute(array(2, 4));

```

El ejemplo anterior generará una consulta de la siguiente manera:

```

1 SELECT * FROM items WHERE id_item = '2' OR id_item = '4'

```

Otra manera de hacer lo mismo:

```

1 $id = 6;
2
3 //Esta vez utilizamos un nombre-clave para cada parámetro
4 $consulta = $db->prepare('SELECT * FROM items WHERE id_item = :id');
5
6 //Con dicho nombre-clave, agregamos el valor del parámetro
7 $consulta->bindParam(':id', $id);
8
9 //Y ejecutamos la consulta
10 $consulta->execute();

```

```

1 SELECT * FROM items WHERE id_item = '6'

```

Ahora bien, intenten inyectar SQL concatenando alguna sentencia en la variable \$id y verán los resultados. Para que puedan confiar más en la protección de PDO.

17.3 Comenzando “MVC”

Para ir de a poco se realizó un código similar a los utilizados anteriormente para PDO y se fue separando en capas paso a paso. El código inicial es el siguiente:

```

1 < ?php
2 require 'conexion.php';
3 $db = new PDO('mysql:host=' . $servidor . ';dbname=' . $db, $usuario,
  $contrasenia);
4 $consulta = $db->prepare('SELECT * FROM items WHERE id_item = ? OR id_item =
  ?');
5 $consulta->execute(array(2, 4));
6 $items = $consulta->fetchAll();
7 $db = null;
8 ?>
9 < !DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
10
11 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
12 <head>
13     <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
14     <title>PDO</title>
15 </head>
16 <body>
17 <table>

```

```

18     <tr>
19         <th>ID</th>
20         <th>Item</th>
21     </tr>
22     < ?php
23     foreach($items as $item)
24     {
25     ?>
26     <tr>
27         <td>< ?php echo $item['id_item']?></td>
28         <td>< ?php echo $item['item']?></td>
29     </tr>
30     < ?php
31     }
32     ?>
33 </table>
34 <a href="index.php">Menú</a>
35 </body>
36 </html>

```

Es un simple listado común presentado en una tabla HTML. Separaremos dicho ejemplo, por el momento, en 3 ficheros. Uno corresponderá al **modelo**, otro a la **vista** y el tercero será el **controlador**.

La primera inquietud fue ¿cuál es el modelo en este ejemplo? como se mencionó más arriba, el modelo es el que se ocupa, básicamente, de todo lo que tiene que ver con el acceso a la información. Sin dudar, en este ejemplo PDO es quien cumple el papel de Modelo.

modelo.php

```

1 < ?php
2 $db = new PDO('mysql:host=' . $servidor . ';dbname=' . $bd, $usuario,
3 $contrasena);
4 $consulta = $db->prepare('SELECT * FROM items');
5 $consulta->execute();
6 $items = $consulta->fetchAll();
7 ?>

```

Segunda inquietud ¿cuál es la vista?. La vista es quien representa la información para que el usuario la pueda entender, en este caso, el HTML, la tabla y todo lo usado para mostrar la información forma parte de la vista.

vista.php

```

1 < !DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
4 <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
6     <title>PDO - Jourmoly</title>
7 </head>
8 <body>
9 <table>
10     <tr>
11         <th>ID</th>
12         <th>Item </th>
13     </tr>

```

```

14     < ?php
15     foreach($items as $item)
16     {
17     ?>
18     <tr>
19         <td>< ?php echo $item['id_item']?></td>
20         <td>< ?php echo $item['item']?></td>
21     </tr>
22     < ?php
23     }
24     ?>
25 </table>
26 </body>
27 </html>

```

Finalmente ¿el controlador?. El controlador es el que permite que todo funcione.

controlador.php

```

1 < ?php
2 //Se incluye el modelo
3 require 'modelo.php';
4
5 //En $items tenemos un arreglo con todos los items gracias al modelo
6
7 //Ahora la vista recibe dicho arreglo para mostrarlo por pantalla
8 require 'vista.php';
9 ?>

```

Lo anterior está bien para un primer acercamiento, pero las cosas no son tan sencillas como en este caso, una sola sección o elemento (items), una sola acción (listar), etc. Lo más normal es que necesitemos de varios controladores y que cada controlador tenga varias acciones. A su vez, cada controlador puede utilizar uno o más modelos como así también plantillas. Para lograr todo esto, es necesario automatizar un poco el código anterior para que admita, en principio, varios controladores y acciones.

Como primera medida vamos a crear una estructura de ficheros para que todo quede más ordenado:

- controladores/
 - *itemsControlador.php*
- modelos/
 - *itemsModelo.php*
- vistas/
 - *listar.php*
- *index.php*

Donde agregamos un nuevo archivo *index.php* que lo único que hará es incluir algunas variables de configuración y nuestro controlador. *Listar.php* equivale a *vista.php*, *itemsModelo.php* equivale a *modelo.php*, *itemsControlador.php* a *controlador.php* de nuestro código anterior con algunos cambios:

itemsModelo.php

```

1 < ?php
2 global $servidor, $bd, $usuario, $contrasenia;
3 $db = new PDO('mysql:host=' . $servidor . ';dbname=' . $bd, $usuario,
4 $contrasenia);
5 function buscarTodosLosItems($db)
6 {
7     $consulta = $db->prepare('SELECT * FROM items');
8     $consulta->execute();
9     return $consulta->fetchAll();
10 }
11 ?>

```

itemsControlador.php

```

1 < ?php
2 function listar()
3 {
4     //Incluye el modelo que corresponde
5     require 'modelos/itemsModelo.php';
6
7     //Le pide al modelo todos los items
8     $items = buscarTodosLosItems($db);
9
10    //Pasa a la vista toda la información que se desea representar
11    require 'vistas/listar.php';
12 }
13 ?>

```

Como se puede observar los únicos cambios han sido armar los scripts con funciones, de modo que cada archivo pueda tener más de una de ellas y puedan ser llamadas en cualquier momento e independientemente.

Ahora, index.php será quien se encargue de averiguar cuál es el controlador y acción que busca el usuario, incluirá los archivos que sean necesarios y ejecutara la acción solicitada. Todos los accesos a nuestro sistema serán por medio de index.php y las URL serán similares a las siguientes:

```

www.example.com/index.php?controlador=items&accion=listar
www.example.com/index.php?controlador=items&accion=agregar
www.example.com/index.php?controlador=items&accion=eliminar
www.example.com/index.php?controlador=usuarios&accion=listar

```

Ahora solo queda hacer el script que interprete nuestra URL y llame al controlador y la acción que corresponda.

index.php

```

1 < ?php
2 //Primero algunas variables de configuracion
3 require 'conexion.php';
4
5 //La carpeta donde buscaremos los controladores
6 $carpetaControladores = "controladores/";

```

```
7
8 //Si no se indica un controlador, este es el controlador que se usará
9 $controladorPredefinido = "items";
10
11 //Si no se indica una accion, esta accion es la que se usará
12 $accionPredefinida = "listar";
13
14 if(! empty($_GET['controlador']))
15     $controlador = $_GET['controlador'];
16 else
17     $controlador = $controladorPredefinido;
18
19 if(! empty($_GET['accion']))
20     $accion = $_GET['accion'];
21 else
22     $accion = $accionPredefinida;
23
24 //Ya tenemos el controlador y la accion
25
26 //Formamos el nombre del fichero que contiene nuestro controlador
27 $controlador = $carpetaControladores . $controlador . 'Controlador.php';
28
29 //Incluimos el controlador o detenemos todo si no existe
30 if(is_file($controlador))
31     require_once $controlador;
32 else
33     die('El controlador no existe - 404 not found');
34
35 //Llamamos la accion o detenemos todo si no existe
36 if(is_callable($accion))
37     $accion();
38 else
39     die('La accion no existe - 404 not found');
40 ?>
```

Logramos implementar un script separado en 3 capas y dimos el primer paso con MVC usando programación estructurada/funcional. Se le pueden hacer muchas mejoras, como utilizar programación orientada a objetos y algunas funcionalidades extras como aplicar URL amigables.