



**UNIVERSIDAD DEL BÍO-BÍO**  
FACULTAD DE CIENCIAS EMPRESARIALES  
DEPARTAMENTO SISTEMAS DE INFORMACIÓN

# **Sistema de Apoyo al Aprendizaje de Programación Estructurada SAAPE**

**MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL EN INFORMÁTICA**

**AUTORES: CALDERÓN MACKAY, CRISTHIAN S.  
RIVERA SOTO, NATALIA I.**

Profesor Guía: Troncoso Pantoja, Brunny.

**Concepción, Diciembre 2015**

## 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 Civil en Informática. El proyecto titulado “Sistema de Apoyo al Aprendizaje de Programación Estructurada SAAPE”.*

La Universidad del Bío-Bío, específicamente la Facultad de Ciencias Empresariales posee dos carreras del área informática, las que imparten distintas asignaturas relacionadas a la programación estructurada. Estas generan gran dificultad entre los alumnos al dar solución a los ejercicios propuestos para su aprendizaje, los que causan la necesidad de adquirir alguna herramienta capaz de apoyar a los estudiantes en su estudio, facilitando así el entendimiento de las asignaturas.

El presente proyecto se desarrolló para apoyar el correcto aprendizaje de las asignaturas de informática. Por un lado el docente podrá asignar a cada alumno un cuaderno de ejercicios que será individual, con un listado de ejercicios distintos para cada estudiante y corroborando su realización y rendimiento. A su vez el alumno poseerá fechas límites para solucionar los ejercicios asignados. Al momento de finalizar puede verificar sus resultados, registrando su rendimiento y permitiéndole volver a intentarlo hasta que lo haga de una manera totalmente correcta.

Para la realización del Sistema se optó por el lenguaje en programación PHP, que es ideal para la creación de plataformas web y el Framework de PHP “CodeIgniter”, que cuenta con un amplio conjunto de librerías para tareas básicas, siendo excepcionalmente rápido. Usa el enfoque controlador de Modelo, Vista, Controlador; lo que permite llevar un mejor orden y claridad de lo que se está programando. Adecuándose a la Metodología Evolutiva que fue la implementada, ya que se crearon distintas versiones del Sistema para garantizar un mejor control de riesgos asociados al desarrollo, haciendo más eficaz el trabajo.

Al implementar el sistema permitirá mejorar los índices de aprobación de las asignaturas, facilitando la entrega y corrección de los ejercicios que se planteen en la plataforma, además de ayudar a monitorear la realización de estos dentro de los plazos entregados y el rendimiento de cada alumno.

## Abstract

*This project is been presented to satisfy the formalities required by Universidad del Bio-Bio in the Ingeniería Civil en Informática graduation process. The project is called "Learning Support System of Structured Programming SAAPE"*

The University of the Bío Bío, specifically the Faculty of Business has two races of the computer area, they taught different subjects related to structured programming. They generate great difficulty among students to solve exercises proposed for learning, causing the need to acquire a tool able to support students in their study, facilitating the understanding of the subjects.

This project was developed to support the correct learning of computer courses. On the one hand the teacher can assign each student a exercise book that is individual, with a list of different exercises for each student and confirming their development and performance. In turn, the student will have deadlines for solving exercises assigned. Upon completion you can check the results, recording performance and allowing retry to do it in a completely correct manner.

To carry out the system we chose the PHP programming language, which is ideal for creating web platforms and PHP Framework "CodeIgniter", which has an extensive set of libraries for basic tasks, being exceptionally fast. Uses the approach controller Model, View, Controller; allowing you to keep better order and clarity of what is being programmed. They adapting to the evolutionary approach that was implemented, as they were created different versions of the system to ensure better control of risks associated with developing, making the work more efficient.

By implementing the system will improve pass rates of subjects, facilitating the delivery and correction of exercises that arise on the platform, in addition to help monitor the development of these within the deadlines given and performance of each student.

## Índice General

<b>1</b>	<b><u>INTRODUCCIÓN.....</u></b>	<b>10</b>
<b>2</b>	<b><u>DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN .....</u></b>	<b>11</b>
2.1	DESCRIPCIÓN DE LA EMPRESA.....	11
2.2	DESCRIPCIÓN DEL ÁREA DE ESTUDIO.....	14
2.3	DESCRIPCIÓN DE LA PROBLEMÁTICA .....	16
<b>3</b>	<b><u>DEFINICIÓN PROYECTO.....</u></b>	<b>18</b>
3.1	OBJETIVOS DEL PROYECTO.....	18
3.1.1	OBJETIVO GENERAL.....	18
3.1.2	OBJETIVOS ESPECÍFICOS .....	18
3.2	AMBIENTE DE INGENIERÍA DE SOFTWARE .....	18
3.3	DEFINICIONES, SIGLAS Y ABREVIACIONES.....	20
<b>4</b>	<b><u>ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE .....</u></b>	<b>21</b>
4.1	ESTUDIO DE LA SITUACIÓN ACTUAL Y POSIBLES SOLUCIONES .....	21
4.1.1	CLAROLINE .....	22
4.1.2	CHAMILO.....	23
4.1.3	EDU 2.0.....	25
4.1.4	MOODLE.....	28
4.1.5	SOFTWARE A MEDIDA.....	32
4.2	DEFINIR SOLUCIÓN.....	34
4.3	ALCANCES.....	37
4.4	OBJETIVO DEL SOFTWARE .....	38
4.4.1	OBJETIVO GENERAL.....	38
4.4.2	OBJETIVO ESPECÍFICO.....	39
4.5	DESCRIPCIÓN GLOBAL DEL PRODUCTO .....	40
4.5.1	INTERFAZ DE USUARIO .....	40
4.5.2	INTERFAZ DE HARDWARE.....	40
4.5.3	INTERFAZ SOFTWARE .....	40
4.5.4	INTERFACES DE COMUNICACIÓN.....	41
4.6	REQUERIMIENTOS ESPECÍFICOS .....	42
4.6.1	REQUERIMIENTOS FUNCIONALES DEL SISTEMA .....	42
4.6.2	INTERFACES EXTERNAS DE ENTRADA .....	43
4.6.3	INTERFACES EXTERNAS DE SALIDA.....	43
4.6.4	ATRIBUTOS DEL PRODUCTO .....	44
<b>5</b>	<b><u>FACTIBILIDAD.....</u></b>	<b>45</b>
5.1	FACTIBILIDAD TÉCNICA.....	45
5.2	FACTIBILIDAD OPERATIVA. ....	46
5.3	FACTIBILIDAD ECONÓMICA .....	47
5.4	CONCLUSIÓN DE LA FACTIBILIDAD .....	48
<b>6</b>	<b><u>ANÁLISIS.....</u></b>	<b>49</b>

<b>6.1</b>	<b>DIAGRAMA DE CASOS DE USO</b> .....	<b>49</b>
6.1.1	ACTORES .....	49
6.1.2	CASOS DE USO Y DESCRIPCIÓN .....	50
6.1.3	ESPECIFICACIÓN DE LOS CASOS DE USO .....	52
<b>6.2</b>	<b>MODELAMIENTO DE DATOS</b> .....	<b>72</b>
<b>7</b>	<b>DISEÑO</b> .....	<b>73</b>
7.1	DISEÑO DE FÍSICO DE LA BASE DE DATOS .....	73
7.2	DISEÑO DE ARQUITECTURA FUNCIONAL .....	77
7.3	DISEÑO INTERFAZ Y NAVEGACIÓN.....	78
7.4	ESPECIFICACIÓN DE MÓDULOS .....	86
<b>8</b>	<b>PRUEBAS</b> .....	<b>93</b>
8.1	ELEMENTOS DE PRUEBA .....	93
8.2	ESPECIFICACIÓN DE LAS PRUEBAS .....	94
8.3	RESPONSABLES DE LAS PRUEBAS.....	96
8.4	CALENDARIO DE PRUEBAS.....	96
8.5	DETALLE DE LAS PRUEBAS .....	96
8.6	CONCLUSIONES DE PRUEBA .....	96
<b>9</b>	<b>PLAN DE CAPACITACIÓN Y ENTRENAMIENTO</b> .....	<b>97</b>
<b>10</b>	<b>PLAN DE IMPLANTACIÓN Y PUESTA EN MARCHA</b> .....	<b>98</b>
<b>11</b>	<b>RESUMEN ESFUERZO REQUERIDO</b> .....	<b>99</b>
<b>12</b>	<b>CONCLUSIONES</b> .....	<b>100</b>
<b>13</b>	<b>BIBLIOGRAFÍA</b> .....	<b>101</b>
<b>14</b>	<b>ANEXO: PLANIFICACION INICIAL DEL PROYECTO</b> .....	<b>102</b>
14.1.1	ESTIMACIÓN INICIAL DE TAMAÑO .....	106
<b>15</b>	<b>ANEXO: RESULTADOS DE ITERACIONES EN EL DESARROLLO</b> .....	<b>112</b>
15.1	1º ITERACIÓN:.....	112
15.2	2º ITERACIÓN:.....	113
<b>16</b>	<b>ANEXO: MANUAL DE USUARIO</b> .....	<b>114</b>
16.1	ADMINISTRADOR: .....	115
16.2	PROFESOR: .....	121
16.3	ALUMNO:.....	132
<b>17</b>	<b>ANEXO: ESPECIFICACION DE LAS PRUEBAS</b> .....	<b>138</b>
17.1	PRUEBAS DE UNIDAD .....	138
17.1.1	<CREAR DOCENTE> .....	138
17.1.2	<ADMINISTRAR ASIGNATURA> .....	142
17.1.3	<CREAR CURSO>.....	143
17.1.4	<CREAR CUADERNO> .....	144
17.1.5	<CREAR ALUMNO DE FORMA MANUAL> .....	146
17.1.6	<AGREGAR PROBLEMA AL REPOSITORIO>.....	150
<b>18</b>	<b>ANEXO: DICCIONARIO DE DATOS DEL MODELO DE DATOS</b> .....	<b>152</b>

## Índice Tablas

Tabla 1: Tabla Comparativa de Plataformas Web .....	31
Tabla 2: Tabla Comparativa de Moodle y Software a medida .....	33
Tabla 3: Tabla Comparativa de Compiladores.....	36
Tabla 4: Tabla de Requerimientos Funcionales del sistema .....	43
Tabla 5: Tabla de Interfaces Externas de Entrada.....	43
Tabla 6: Tabla de Interfaces Externas de Salida .....	43
Tabla 7: Costo Energía Eléctrica.....	47
Tabla 8: Presupuesto del Proyecto .....	47
Tabla 9: Flujo de Eventos Básicos CU Crear Curso-Sección.....	52
Tabla 10: Flujo de Eventos Alternativo CU Crear Curso-Sección .....	52
Tabla 11: Flujo de Eventos Básicos CU Eliminar Curso-Sección.....	53
Tabla 12: Flujo de Eventos Alternativo CU Eliminar Curso-Sección .....	53
Tabla 13: Flujo de Eventos Básicos CU Ingresar Alumnos .....	54
Tabla 14: Flujo de Eventos Alternativo CU Ingresar Alumnos.....	54
Tabla 15: Flujo de Eventos Básicos CU Ingresar Problema.....	57
Tabla 16: Flujo de Eventos Alternativo CU Ingresar Problema .....	57
Tabla 17: Flujo de Eventos Básicos CU Crear Cuaderno de Ejercicios .....	58
Tabla 18: Flujo de Eventos Alternativo CU Crear Cuaderno de Ejercicios.....	58
Tabla 19: Flujo de Eventos Básicos CU Actualización Cuaderno de Ejercicios.....	59
Tabla 20: Flujo de Eventos Alternativo CU Actualización Cuaderno de Ejercicios .....	59
Tabla 21: Flujo de Eventos Básicos CU Consultar Rendimiento grupal/individual Profesor.....	61
Tabla 22: Flujo de Eventos Alternativo CU Consultar Rendimiento grupal/individual Profesor.....	61
Tabla 23: Flujo de Eventos Básicos CU Consultar Rendimiento individual Alumno .....	61
Tabla 24: Flujo de Eventos Alternativo CU Consultar Rendimiento individual Alumno.....	62
Tabla 25: Flujo de Eventos Básicos CU Emitir Informe Rendimiento grupal/individual Profesor .....	62
Tabla 26: Flujo de Eventos Alternativo CU Emitir Informe Rendimiento grupal/individual Profesor .....	63
Tabla 27: Flujo de Eventos Básicos CU Emitir Informe Rendimiento individual Alumno .....	63
Tabla 28: Flujo de Eventos Alternativo CU Emitir Informe Rendimiento individual Alumno.....	63
Tabla 29: Flujo de Eventos Básicos CU Registrar Profesor .....	64
Tabla 30: Flujo de Eventos Alternativo CU Registrar Profesor .....	64
Tabla 31: Flujo de Eventos Básicos CU Eliminar/Modificar Profesor .....	65
Tabla 32: Flujo de Eventos Alternativo CU Eliminar/Modificar Profesor .....	65
Tabla 33: Flujo de Eventos Básicos CU Ver Cuaderno .....	68
Tabla 34: Flujo de Eventos Alternativo CU Ver Cuaderno.....	68
Tabla 35: Flujo de Eventos Básicos CU Resolver Ejercicio.....	69
Tabla 36: Flujo de Eventos Alternativo CU Resolver Ejercicio .....	69
Tabla 37: Flujo de Eventos Básicos CU Ver Problema Resuelto .....	69
Tabla 38: Flujo de Eventos Alternativo CU Problema Resuelto .....	70
Tabla 39: Flujo de Eventos Básicos CU Autenticar Usuario .....	70
Tabla 40: Flujo de Eventos Alternativo CU Autenticar Usuario.....	70
Tabla 41: Módulo Validar Sección.....	86
Tabla 42: Módulo Verificar Sección.....	86
Tabla 43: Módulo Registrar Curso-Sección .....	86

Tabla 44: Módulo Anular Curso-Sección .....	86
Tabla 45: Módulo Verificar Rut (Alumnos) .....	87
Tabla 46: Módulo Validar Rut (Alumnos) .....	87
Tabla 47: Módulo Registrar Alumnos.....	87
Tabla 48: Módulo Actualizar Alumnos.....	87
Tabla 49: Módulo Anular Alumnos .....	88
Tabla 50: Módulo Verificar Rut (Profesor) .....	88
Tabla 51: Módulo Validar Rut (Profesor).....	88
Tabla 52: Módulo Registrar Profesor .....	88
Tabla 53: Módulo Actualizar Profesor .....	89
Tabla 54: Módulo Anular Profesor.....	89
Tabla 55: Módulo Validar Valores de Entrada (Problema) .....	89
Tabla 56: Módulo Validar Valores de Salida (Problema) .....	90
Tabla 57: Módulo Registrar Problemas.....	90
Tabla 58: Módulo Validar Fecha de Caducidad.....	90
Tabla 59: Módulo Registrar Cuaderno .....	91
Tabla 60: Módulo Actualización Cuaderno .....	91
Tabla 61: Módulo Eliminar Cuaderno.....	92
Tabla 62: Especificación de las pruebas.....	96
Tabla 63: Resumen Esfuerzo Requerido.....	99
Tabla 64: Factor de Peso de los Actores sin Ajustar .....	106
Tabla 65: Factor de Peso de los Casos de Uso Sin Ajustar .....	107
Tabla 66: Factores de Complejidad Técnica.....	109
Tabla 67: Factores Ambientales.....	110
Tabla 68: Esfuerzo Horas-Hombre .....	111
Tabla 69: Especificación Prueba Crear Docente .....	142
Tabla 70: Especificación Prueba Administrar Asignatura .....	143
Tabla 71: Especificación Prueba Crear Curso .....	144
Tabla 72: Especificación Prueba Crear Cuaderno.....	146
Tabla 73: Especificación Prueba Crear Alumno de Forma Manual.....	150
Tabla 74: Especificación Prueba Agregar Problema al Repositorio .....	151

## Índice Diagramas

Diagrama 1: Organigrama Universidad del Bío-Bío.....	14
Diagrama 2: Organigrama Facultad de Ciencias Empresariales.....	16
Diagrama 3: Proceso de Negocio del Funcionamiento Actual .....	17
Diagrama 4: Casos de Uso General .....	51
Diagrama 5: Casos de Uso Compilador GCC .....	51
Diagrama 6: Modelo Entidad Relación.....	72
Diagrama 7: Modelo Relación .....	73
Diagrama 8: Árbol de Descomposición funcional .....	77
Diagrama 9: Jerarquía Menú.....	83
Diagrama 10: Esquema de Navegación Administrador .....	84
Diagrama 11: Esquema de Navegación Profesor.....	84
Diagrama 12: Esquema de Navegación Alumno.....	85

## Índice Ilustraciones

Ilustración 1: Diseño de interfaz de usuario .....	78
Ilustración 2: Página de Acceso .....	79
Ilustración 3: Perfil Administrador .....	79
Ilustración 4: Vista Profesores (Administrador) .....	80
Ilustración 5: Perfil Profesor.....	81
Ilustración 6: Vista Cursos (Profesor).....	81
Ilustración 7: Perfil Alumno .....	82
Ilustración 8: Vista Cuaderno (Alumno).....	82
Ilustración 9: Carta Gantt.....	105
Ilustración 10: WampServer .....	112
Ilustración 11: Inicio CodeIgniter .....	112
Ilustración 12: Configuración BD .....	113
Ilustración 13: Menú Administrador Localhost .....	113
Ilustración 14: Pantalla de Acceso .....	114
Ilustración 15: Mensaje LEER .....	114
Ilustración 16: Mensaje Olvidaste tu Contraseña.....	115
Ilustración 17: Menú Administrador Final.....	115
Ilustración 18: Lista de Profesores .....	116
Ilustración 19: Actualizar Docente .....	117
Ilustración 20: Eliminar Docente.....	118
Ilustración 21: Crear Documento.....	118
Ilustración 22: Administrar Tipo de Problema .....	119
Ilustración 23: Administrar Asignaturas.....	120
Ilustración 24: Actualizar Problema .....	120
Ilustración 25: Actualizar Problema 2 .....	121
Ilustración 26: Interfaz Docente .....	121
Ilustración 27: Crear Curso.....	122
Ilustración 28: Ver Curso .....	123
Ilustración 29: Editar Curso .....	124
Ilustración 30: Inscribir Alumnos.....	124
Ilustración 31: Inscribir Alumno Manual.....	125
Ilustración 32: Inscribir Alumno Automático .....	126
Ilustración 33: Lista del Curso Rendimiento.....	127
Ilustración 34: Alumno Rendimiento.....	128
Ilustración 35: Crear Cuaderno de Ejercicios.....	129
Ilustración 36: Agregar Problema al Repositorio.....	130
Ilustración 37: Menú Alumno .....	132
Ilustración 38: Ver Cuaderno.....	133
Ilustración 39: Descripción del Problema .....	134
Ilustración 40: Compilador.....	134
Ilustración 41: Rendimiento del Alumno.....	136
Ilustración 42: Ver problema resuelto.....	137

---

## 1 INTRODUCCIÓN

---

Contemporáneamente la Universidad del Bío-Bío trabaja con diversas plataformas para complementar el aprendizaje de los estudiantes, tanto en apuntes como en guías de trabajo para los alumnos que son impartidos por sus profesores. De esta manera se origina un plan de estudio más moderno y tecnológico. Es así que nace la necesidad de tener una herramienta de trabajo adecuada a los requerimientos de ciertas carreras, como lo son en este caso específico las carreras del área Informática que poseen asignaturas de programación, las que anteriormente trabajaban con la plataforma PVA que sufrió ciertas falencias dejando hoy a sus alumnos y docentes sin herramientas. Es por ello que se pensó en este proyecto generando una nueva plataforma para darle una solución a este problema, brindando así un apoyo fundamental para la comprensión y aprendizaje de los alumnos.

El proyecto se basa en la creación de un sistema web para apoyar a las diversas asignaturas de programación en lenguaje C, de esta manera favorecer el aprendizaje de los estudiantes, haciendo que éstos ejerciten y detectando así sus falencias en ciertos tópicos y a la vez que los puedan mejorar. Así se espera mejorar las tasas de aprobación de estas asignaturas.

Enseguida se presentan las bases teóricas que soportan el proyecto que se va a desarrollar. Para ello se efectúa una investigación de las diversas herramientas disponibles, para conocer y analizar a cada una de ellas, de modo que se escoja la más eficaz y óptima para la construcción del sistema.

El sistema web se describe en los diversos capítulos que se presentarán durante el informe. Estos capítulos se dividen en las siguientes etapas de un software “Especificación de los requerimientos de los usuarios detallando cada aspecto relevante del sistema”, “Tipos de factibilidad dentro del proyecto”, “Análisis del problema y diseño de la solución” y “Pruebas del sistemas y temas posteriores al desarrollo del software”.

---

## 2 DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN

---

### 2.1 Descripción de la empresa

Antecedentes generales de la Empresa

- Nombre: Universidad del Bío-Bío.
- Dirección: Avenida Collao N°1202.
- Rubro: Universidades.
- La universidad del Bío-Bío ofrece la formación integral de los estudiantes, considerando su desarrollo como persona y en la disciplina, aunando para ello en el modelo enseñanza y aprendizaje, el conocimiento, las habilidades y las actitudes que le permitan la integración socio-laboral. Además de que sean competentes y autónomos, que conjuguen en su desempeño laboral las dimensiones de formación humana y disciplinares, que satisfagan las necesidades de la sociedad y de un mercado profesional cambiante en un mundo globalizado.
  
- Entorno:  
La Universidad del Bío-Bío, es una institución de educación superior chilena, de carácter estatal, ubicada en la VIII Región del Biobío, con sedes en Concepción y Chillán. Es una de las veinticinco universidades del Consejo de Rectores de las Universidades Chilenas, y miembro de la Agrupación de Universidades Regionales de Chile.

Se encuentra acreditada por la Comisión Nacional de Acreditación (CNA), durante un periodo de cinco años comprendidos entre agosto de 2009 hasta agosto de 2014 en las áreas de gestión, docencia, investigación y vinculación con el medio.

La UBB figura como la 12ª universidad chilena según la clasificación web-ométrica del CSIC, en julio de 2011,<sup>2</sup> y en el 21º lugar según el ranking de El Mercurio.

En el ranking 2011 del QS World University Ranking<sup>4</sup> se ubicó entre las 200 mejores universidades de Latinoamérica.

▪ Competencia directa:

Actualmente en la Región del Bío-Bío, específicamente en la provincia de Concepción existen diversas universidades que poseen carreras relacionadas con la Informática por lo que existe una amplia competencias, pero las que están directamente en un alto nivel de competencia son las que están dentro consejo de rectores: “Universidad de Concepción”, “Universidad Católica de la Santísima Concepción”, “Universidad Técnica Federico Santa María”. Igualmente se deben incluir las universidades privadas como “Universidad Andrés Bello”, “Universidad Santo Tomás”, “Universidad Tecnológica de Chile INACAP” e institutos como “Instituto Profesional Virginio Gómez”, “Instituto Profesional IPP”, “Instituto Profesional AIEP”.

▪ 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:

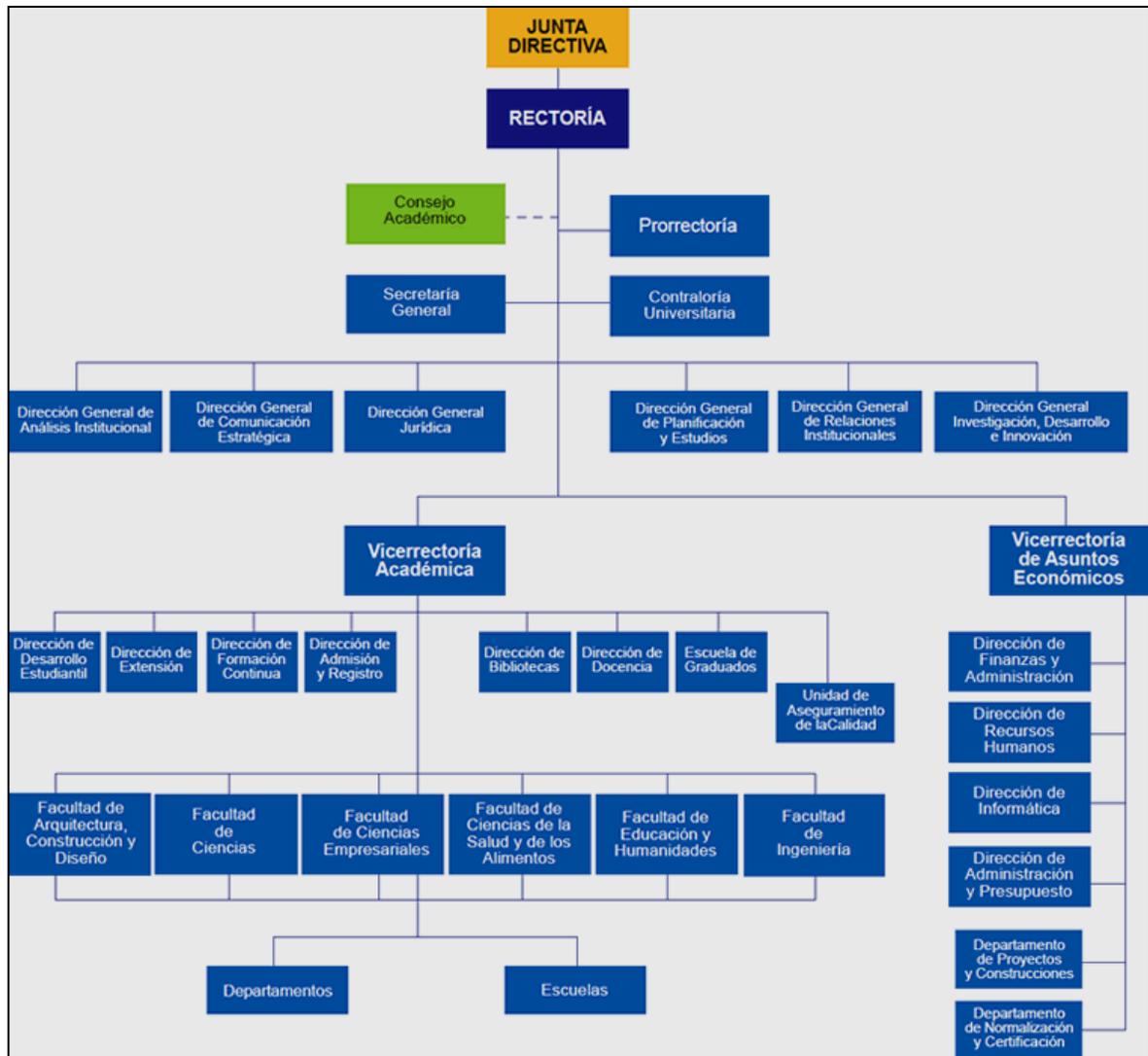
- Formar profesionales de excelencia capaces de dar respuesta a los desafíos de futuro, con un modelo educativo cuyo propósito, es la formación integral del estudiante a partir de su realidad y sus potencialidades, promoviendo la movilidad social y la realización personal.
- Fomentar la generación de 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.

- Desarrollar una gestión académica y administrativa moderna, eficiente, eficaz y oportuna, centrada en el estudiante, con estándares de calidad certificada que le permiten destacarse a nivel nacional y avanzar en la internacionalización.

- 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.

- Estructura organizativa:



## Diagrama 1: Organigrama Universidad del Bío-Bío

### 2.2 Descripción del área de estudio

El sistema web está destinado a la Facultad de Ciencias Empresariales de la Universidad del Bío-Bío, específicamente al departamento de Sistemas de Información donde se encuentran involucradas las carreras de Ingeniería Civil en Informática e Ingeniería de Ejecución en Computación e Informática, con sus correspondientes directores de escuela quienes delegan las asignaturas a dictar en cada semestre y los docentes que las dictarán. De manera que como el sistema está orientado para el aprendizaje y ejercitación en la programación estructurada de computadores de las diversas asignaturas de informáticas existentes, está vinculado con los diferentes profesores encargados de estas asignaturas.

Además como las estructuras de las carreras Ingeniería Civil en Informática e Ingeniería de Ejecución en Computación e Informática son pertenecientes a escuelas, están relacionadas con el Decano de la Facultad de Ciencias Empresariales.

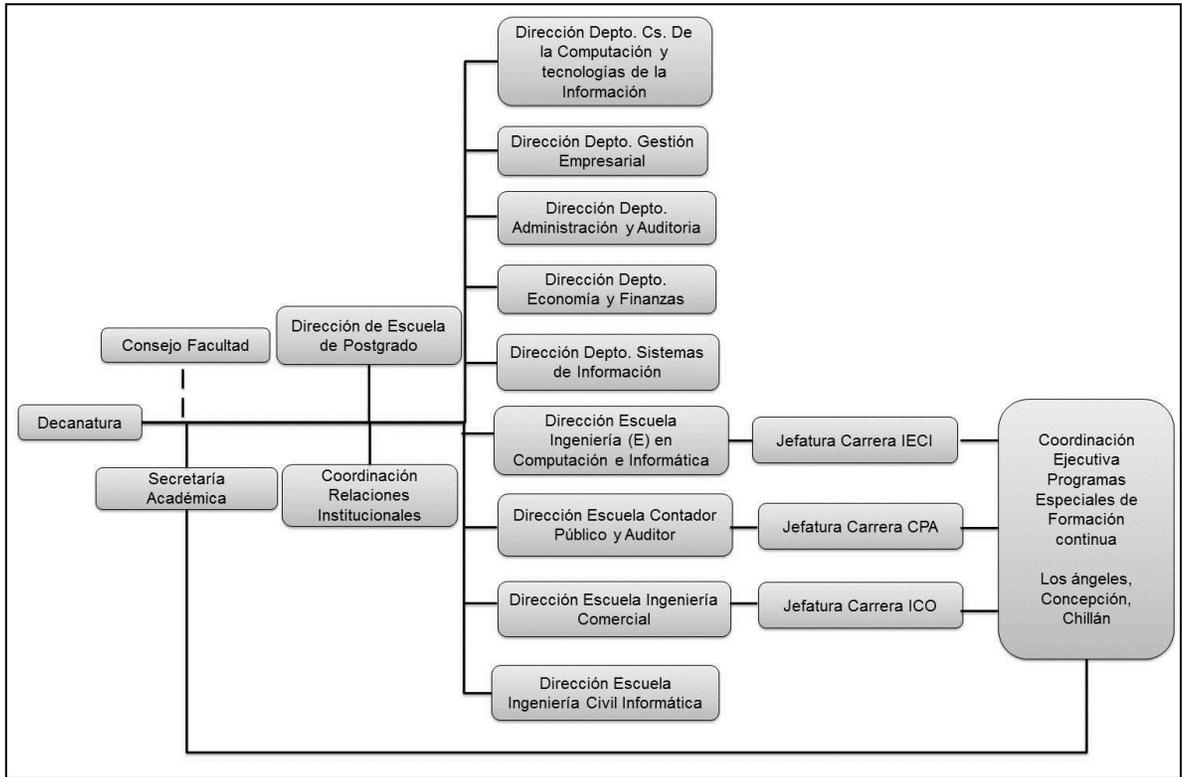
- Objetivo del área:

El desarrollo académico de la Ciencias de la Computación e Informática y la participación activa en la formación, capacitación y asistencia técnica de profesionales en informática, particularmente en las áreas de la informática aplicada a la gestión.

Una creciente actividad de investigación relevante, buscando la formación de equipos en áreas donde se pueda destacar.

Un permanente perfeccionamiento de sus cuadros académicos y capacitación del personal administrativo, para el mejoramiento de la calidad.

Establecimiento de relaciones con el entorno, que permita una mayor pertinencia y contribución a su quehacer.



## Diagrama 2: Organigrama Facultad de Ciencias Empresariales

- El Departamento de Sistemas de Información se fortalece para elevar su quehacer académico en Ciencias de la Computación e Informática y Gestión, para responder ante nuevos escenarios que genera la aplicación de tecnologías de información, en las organizaciones públicas, privadas y a la sociedad en general. Además se provee de la dotación y el perfeccionamiento de los cuadros académicos y su preocupación docente es el uso estratégico de la información y la incorporación de las tecnologías en las organizaciones, estructura la docencia y la investigación.
- El decano es el responsable de las distintas direcciones de departamentos y escuelas, dirige y coordina las escuelas y docentes.
- Los jefes de carrera, son los responsables de organizar las actividades de docencia y alumnado, así como el desarrollo de los proyectos de extensión.

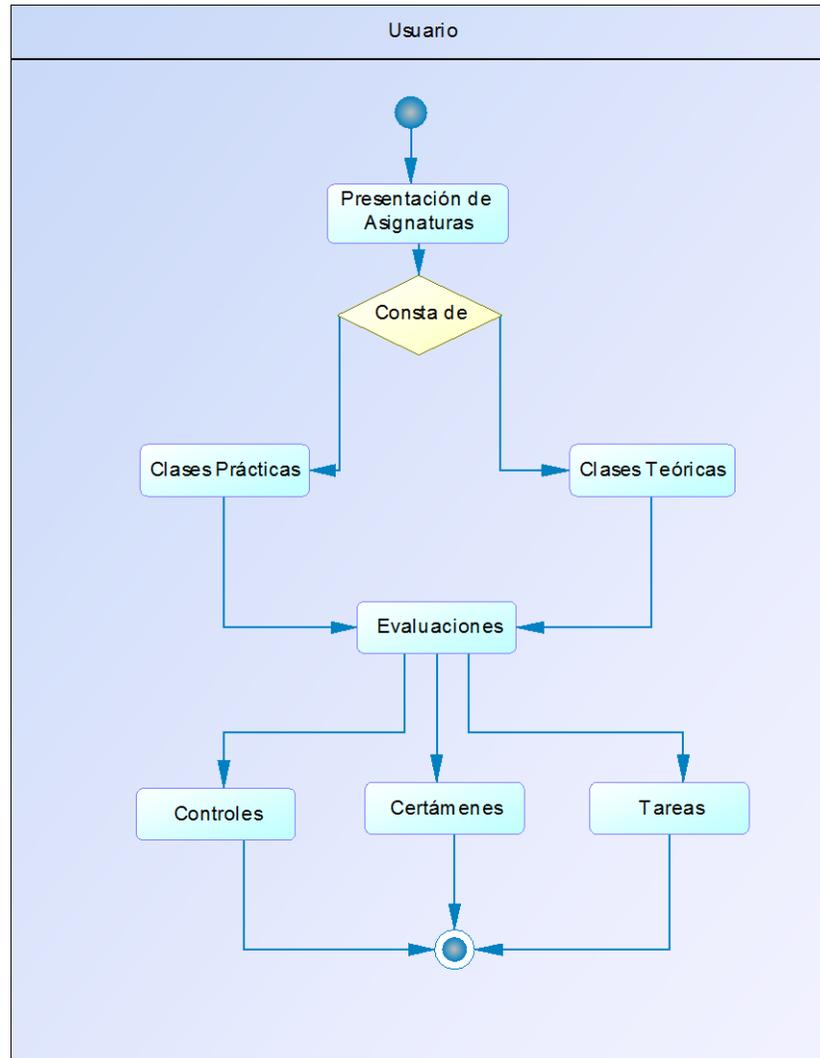
### 2.3 Descripción de la problemática

Actualmente los ramos de programación por lo general utilizan la misma metodología de aprendizaje. Éstas consisten en clases teóricas con un docente, en las que van revisando las diferentes unidades del plan de estudios con ejemplos y pequeños ejercicios que se intentan resolver en las horas de clases. Por otro lado existen las ayudantías que se realizan en los laboratorios de la facultad, donde se trata de realizar la mayor cantidad de ejercicios prácticos posibles.

A la hora de analizar esta metodología se pueden encontrar muchas falencias, esto debido en gran medida al poco tiempo con el que se cuenta para asimilar y practicar algo tan complejo como es la programación, sobre todo en lo que se refiere a las prácticas en laboratorio, ya que para un ayudante es muy complicado revisar y aclarar todas las dudas para tantos alumnos en 90 minutos a la semana.

Por lo general para corroborar el aprendizaje realizado durante el semestre académico se realiza un trabajo de programación, el que muchas veces le resulta muy complicado

al alumnado debido a la falta de práctica a lo largo del curso, lo que conlleva a copias o solicitar ayuda externa, así se está muy lejos de lograr un aprendizaje y asimilación de los contenidos, provocando vacíos y dificultades en las siguientes asignaturas de programación.



**Diagrama 3: Proceso de Negocio del Funcionamiento Actual**

---

## 3 DEFINICIÓN PROYECTO

---

### 3.1 Objetivos del proyecto

#### 3.1.1 Objetivo General

Diseñar y construir un Sistema de Apoyo para el Aprendizaje y la Ejercitación en la Programación Estructurada de Computadores para alumnos de asignaturas de Informática de la Facultad de Ciencias Empresariales en la Universidad del Bío-Bío.

#### 3.1.2 Objetivos Específicos

- Desarrollar un breve estudio del arte del desarrollo de plataformas educativas similares, sus aplicaciones y características técnicas a fin de seleccionar las herramientas de desarrollo y capturar ideas de utilidad para el proyecto.
- Diseñar y construir un sistema web para los alumnos de las asignaturas de informática, en cuanto al apoyo y ejercitación de la programación estructurada.
- Probar el Software, siguiendo estándares establecidos en la ingeniería del software.

### 3.2 Ambiente de Ingeniería de Software

- Metodología de Desarrollo:

La metodología a utilizar en este proyecto será la “evolutiva”, ya que con esto se privilegia el desarrollo del Software, y al ser iterativa permite ir creando versiones del sistema a corto plazo, con lo que se garantiza un mejor control de riesgos asociados al desarrollo.

Otro factor a favor de esta metodología es que se asume que los requerimientos pueden cambiar en cualquier momento, lo que favorece a la inexperiencia de los desarrolladores, así junto con el cliente se puede ir construyendo un mejor sistema.

- Técnicas y notaciones:
  - Modelo Entidad Relación: para la modelación de los datos
  - Modelo Relacional: para la gestión de la base de datos
  - Diagramas de casos de uso: para ver la funcionalidad del sistema
  - Diagrama de procesos: para diagramar el proceso de negocio
  - Esquema de navegación: para visualizar la organización de menús y submenús.
  - Reuniones constantes con profesora guía: Brunny Troncoso Pantoja.
  
- Estándares de Documentación, Producto o Proceso:

Una buena documentación otorga al software un plus para facilitar su utilización, ya que lo hace comprensible para personas que no han participado en su desarrollo y principalmente para los usuarios de éste. Por otra parte y quizás más importante aspecto de una buena documentación es que facilita el desarrollo de nuevas versiones y futuras mantenciones del software.

La documentación implementada será:

- Plantilla de Documentación Técnica del Proyecto
- Plantilla de Pruebas.
- Registro de Reuniones.

En nuestro paquete de documentos irán incluidos los siguientes elementos:

- Diagrama de Casos de Uso.
- Diagrama de Procesos.
- MER /MR.

- Herramientas de apoyo al desarrollo de software que serán utilizadas:
  - Power Designer: Programa utilizado para crear los diagramas de casos de uso, diagrama de procesos, Modelo Entidad Relación, diagrama de jerarquía de menú, esquema de navegación, Modelo Relacional.
  - Ms Project: Software que permite crear una Carta Gantt.
  - Wamp: Servidor que permite probar el sistema de forma local .

### 3.3 Definiciones, Siglas y Abreviaciones

- URL: Uniform Resource Locator, Localizador de Recursos Uniforme. Es una secuencia de caracteres, que se usa para nombrar recursos en internet para su localización o identificación (12).
- PHP: Es un lenguaje de programación, originalmente diseñado para el desarrollo web de contenido dinámico.
- SGBD: Sistema de Gestión de Base de datos Distribuida.
- LMS: Learning Management System, o Sistema de Gestión de Aprendizaje.
- SAAPE: Sistema de Apoyo al Aprendizaje de Programación Estructurada.
- ICINF: Ingeniería Civil en Informática.
- IECI: Ingeniería Ejecución en Computación e Informática.

---

## 4 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

---

### 4.1 Estudio de la Situación Actual y Posibles Soluciones

Para dar solución a la problemática planteada anteriormente se ha optado por comparar dos de las alternativas más viables. Por un lado las plataformas LMS (Learning Management System), o Sistema de Gestión de Aprendizaje, y por otro la construcción de un “Software a medida”.

Una plataforma virtual de aprendizaje es un software instalado generalmente en un servidor web, que se emplea para crear, aprobar, administrar, almacenar, distribuir y gestionar un aprendizaje virtual.

Se investigaron alrededor de 20 de estos softwares, entre los que se destacan los siguientes:

- ATutor
- Chamilo
- Claroline
- Dckeos
- Moodle
- Sakai
- BlackBoard
- Edu 2.0
- Nixty
- Udemmy

Entre estos se escogieron las 4 mejores opciones para este caso, las que se detallan a continuación:

### 4.1.1 Claroline

Es una plataforma de aprendizaje y trabajo virtual de software libre y código abierto (open source) que permite a los formadores construir cursos online y gestionar las actividades de aprendizaje y colaboración en la web. Está escrito en el lenguaje de programación PHP, utiliza MySQL como SGBD.

Está disponible para plataformas (Linux) y navegadores libres (Mozilla, Netscape), y plataformas (Unix, Mac OS X y Windows) y navegadores propietarios (Internet Explorer). Está traducido a 35 idiomas y tiene una gran comunidad de desarrolladores y usuarios en todo el mundo. Se inició en el año 2001 por la UCL (Universidad católica de Louvain, Bélgica). En mayo de 2007 se creó el Consorcio Claroline formado por varias universidades de Bélgica, España, Canadá y Chile cuyos objetivos son organizar el desarrollo y la promoción de la plataforma en un nivel técnico y pedagógico.

Desde el sitio de Claroline, refieren que su funcionamiento no requiere conocimientos técnicos especiales, es fácil de instalar y de usar.

#### Características

- Publicación de recursos en cualquier formato de archivo.
- Foros de discusión públicos y privados.
- Administración de listas de enlaces.
- Creación de grupos de estudiantes.
- Confección de ejercicios.
- Agenda con anuncios, tareas y plazos.
- Publicación de anuncios vía email o portada del curso.
- Gestión de los envíos de los estudiantes.
- Administración de chats.
- Supervisión de acceso y progreso de estudiantes.
- Agrupación de los contenidos en temas o módulos.
- Gestión de estadísticas de cursos.

#### Ventajas

- No tiene límite de usuarios.
- Las tareas de administración son muy sencillas.
- La interfaz es funcional, intuitiva y con elementos básicos que facilitan la navegación.
- Cuida la estética de los cursos.

Desventajas:

- Cuenta con pocos módulos y plugins para descargar.
- Su personalización es un tanto dificultosa.
- La herramienta de chat es algo lenta.
- Los servicios que puede configurar el administrador son muy limitados, con respecto a otras plataformas. Por ejemplo, no se tiene acceso a realizar una copia de seguridad del curso, ni encuestas, entre otros.
- Algo dificultoso a la hora de abrir archivos.

#### **4.1.2 Chamilo**

Es una herramienta más bien moderna, fue lanzada en 2010, tiene ya una comunidad bastante grande y muy buena acogida por parte de las instituciones educativas y las empresas.

Es una plataforma de aprendizaje virtual, de código abierto y software libre (bajo la licencia GNU/GPLv3) que le permite a los docentes construir cursos en línea como soporte a la modalidad presencial o netamente virtuales.

Se puede instalar en diferentes plataformas operativas como Linux, Windows, OS-X, desarrollado con lenguaje PHP y motor de base de datos MySQL, también de software libre.

Chamilo es un LMS que organiza los diferentes procesos de enseñanza-aprendizaje mediante diseño instruccional y colaborativo y está implementado de tal forma que permite al profesor escoger entre una serie de metodologías pedagógicas, siendo una de ellas el constructivismo social.

Tiene traducciones abiertas para 55 idiomas, y alrededor de 25 de ellas están en más del 80% completada.

El proyecto está protegido por una asociación sin fines de lucro, lo que significa que es literalmente imposible para cualquier empresa tomar la propiedad del proyecto y cerrarlo. Pero también significa que cada uno es libre de desarrollar sus propias extensiones o mejoras, y compartirlo (o no) con la comunidad. La asociación Chamilo proporciona un sistema de recompensa para alentar a las empresas a contribuir con código al proyecto.

### Características

Entre las funcionalidades de que dispone se pueden destacar:

- Interacción (foros, chats, compartir archivos, anuncios, grupos, tareas, wiki, usuarios, encuestas, notas personales, redes sociales, glosarios).
- Contenido (lecciones, gestionar un curso, evaluaciones, asistencia, enlaces, glosario, administración de documentos, avances temáticos, ejercicios (en forma de preguntas y exámenes con control de tiempo).
- Administración (gestión de blogs, configuración y mantenimiento de cursos, informes, documentos).

### Ventajas

- Usabilidad: muy fácil de usar tanto por el docente como por los estudiantes.
- Sus vistas son muy limpias, lo que hace que el estudiante no se distraiga tan fácilmente.
- Es de licencia GNU/GLP (software libre) lo cual da libertad para: usar, modificar, mejorar, distribuir.
- Trabaja bajo los principios pedagógicos constructivistas.
- Facilidad para crear contenidos.
- Soporta multi-idomas.
- Seguimiento de actividades y usuarios mediante informes gráficos.
- Manejo de videoconferencia.
- Manejo de actividades tanto sincrónicas como asincrónicas.
- Contiene de forma integral herramientas de autor.

- Genera certificaciones.
- Interfaces personalizables.
- Es estructurado y de fácil comprensión.
- Ayuda a mejorar las destrezas comunicativas a nivel individual y grupal.
- Permite al estudiante trabajar a su propio ritmo.
- Permite crear y subir audio.

#### Desventajas

- Lleva tiempo instalarlo e implementarlo
- Debido a lo reciente de esta plataforma no existen muchas extensiones ni una comunidad tan grande.

### 4.1.3 Edu 2.0

Edu2.0 nos ofrece las mismas alternativas que otras plataformas con la ventaja de no utilizar un servidor ya que el acceso es en línea.

Está ideado para ser usado por los profesores que trabajan habitualmente de forma presencial y que desean incluir elementos digitales online.

Es un sistema LMS que las instituciones dedicadas a la educación pueden utilizar para divulgar y gestionar cursos en la web, siendo gratuito para aquellas con menos de 2000 alumnos registrados.

Edu2.0 es una sencilla, potente, y moderna plataforma elearning, fácil de usar en el sistema de gestión de aprendizaje (LMS) que simplifica la educación en línea. Ofrece un plan gratuito para escuelas y un plan Premium para usuarios expertos.

Puede ser utilizada por instituciones educativas públicas o privadas, capacitación de profesores y en la educación en casa ya que proporciona un sistema completo de gestión del aprendizaje para los padres y sus hijos. También se dirige a las empresas, organizaciones profesionales, compañías dedicadas a la formación, profesores privados y organizaciones sin ánimo de lucro.

A menudo se utiliza para la formación interna de las empresas, formación del cliente y la certificación en línea.

## Características

- Gratuito (en algunos casos) y fácil de utilizar: incluye transmisión de noticias parecidas a las de Facebook y otras redes sociales. Para los usuarios potentes se cuenta con planes premium de bajo costo o costo bajo demanda.
- Portal: Cada organización recibe un portal propio para personalizar.
- Moderno e intuitivo: La interfaz moderna y abierta es agradable y fácil de usar.
- Fuentes y red de contactos: Para estar al día con las noticias (tareas, anuncios y eventos).
- Las fuentes animan a los miembros de la clase a comunicarse abiertamente para que tanto los alumnos como los profesores puedan participar.
- Integral: Un conjunto integral de funciones, al cual se le añaden cada semana nuevas funciones a petición de los usuarios.
- Móvil: Se puede acceder desde cualquier dispositivo móvil.
- Mundial: Disponible en más de 10 idiomas con traducción automática integrada entre los miembros.
- Gran capacidad de evaluación: las herramientas de evaluación incluyen pruebas, bancos de preguntas y otros siete tipos de tareas.
- Servicio Baremos: El soporte transparente de baremos le permite puntuar las tareas de manera rápida y consistente.
- Libro de calificaciones: Fácil de utilizar con aprobaciones personalizadas, periodos académicos, etc.
- Planes de estudio: Permite crear planes de estudios, relacionar las competencias con las lecciones y tareas, y seguir el progreso de sus estudiantes.
- Multimedia: Permite incrustar medios audiovisuales dentro de las clases, incluso imágenes, audio, videos, presentaciones, etc.
- Colaborativo: Tanto los grupos, como los blogs, wikis, foros y chats están integrados.
- Comercio electrónico: El soporte al comercio electrónico le permite cobrar a los alumnos que se matriculen en una clase.
- Hay soporte de pago para las principales tarjetas de crédito y Paypal.
- SCORM: Soporte SCORM integrado facilita la reutilización de los módulos estándar de aprendizaje.

- Aprender y colaborar por internet.
- Personalización: Permite personalizar el banner, el URL, la página de inicio, la combinación de colores y el logotipo de la organización.
- Correo electrónico y SMS: sistema de mensajería segura se integra con el correo electrónico y los SMS.
- Los padres: Los padres pueden contar con una cuenta propia para poder acceder con facilidad a las notas, tareas e informes de sus hijos.
- Bibliotecas: Los profesores pueden guardar los recursos dentro de la biblioteca personal, del centro o del distrito para poderlos compartir con facilidad.
- Carpetas: Cada alumno recibe una carpeta para guardar y mostrar sus mejores trabajos.
- Seguridad: Para poder mantener la privacidad de sus materiales cuenta con características especiales, muy seguras, con conexión SSL y en conformidad con Ley de Confidencialidad y Derechos de Educación de la Familia (FERPA).
- Monitoreo: Se pueden monitorizar todas las comunicaciones dentro de la organización en tiempo real, y se reciben informes diarios sobre cualquier comunicación que pueda ser ofensiva.
- La navegación en la plataforma se realiza por pestañas.

## Demo y Pago

Se ha creado una escuela de demostración en desde la perspectiva del profesor sobre el uso de edu2.0.

Una vez que la prueba gratuita de 30 días ha terminado, es necesario inscribirse en un plan de pago recurrente para continuar utilizando el sitio. Cada plan de pago permite hasta un cierto número de estudiantes activos por mes.

## Ventajas

- Tiene un diseño atractivo.
- Fácil de usar por cualquier usuario.
- Puede utilizarse en cualquier idioma.
- Facilidad para el diseño de cursos.
- Organización secuencial.

- No requiere instalación.
- Integración total de los recursos.
- Se aceptan sugerencia de los usuarios para futuras actualizaciones.

#### Desventajas

- Los blogs y las wikis son un punto débil.
- No siempre funcionan las invitaciones por email ya que llegan sin el link para registrarse en el sistema.
- No permite iniciar sesión en más de un centro a la vez.

#### 4.1.4 Moodle

Es un sistema de gestión de cursos de código abierto (*Open Source Course Management System, CMS*), bajo la Licencia Pública General de GNU 2.2 . Básicamente esto significa que Moodle tiene derechos de autor, pero que tiene libertades adicionales. Usted está autorizado a copiar, usar y modificar Moodle siempre que se comprometa a proporcionar la fuente a otros, no modificar o eliminar la licencia original y los derechos de autor, y aplicar esta misma licencia a cualquier trabajo derivado.

Es compatible con otros formatos (SCORM, IMS, entre otros). Puede ser instalado en cualquier ordenador que pueda ejecutar PHP, y puede soportar una base de datos tipo SQL (por ejemplo MySQL). Se ejecuta sin modificaciones en Unix, GNU/Linux, OpenSolaris, FreeBSD, Windows, Mac OS X, NetWare y otros sistemas que soportan PHP, incluyendo la mayoría de proveedores de alojamiento web.

Es una aplicación web gratuita que los educadores pueden utilizar para crear sitios de aprendizaje efectivo en línea o como complemento del aprendizaje presencial.

Moodle permite una amplia gama de modos de enseñanza. Puede ser utilizado para generar contenido de manera básica o avanzada (por ejemplo páginas web) o evaluación.

Es utilizado por una gran variedad de instituciones educativas y no educativas y por educadores independientes.

### Características:

- Se basa en una aproximación constructiva del aprendizaje enfatizando que tanto los estudiantes como los profesores pueden contribuir a la experiencia educativa de varias maneras, ya sea comentando entradas de bases de datos o trabajando colaborativamente en una wiki.
- Forma parte de una gran comunidad en constante crecimiento, haciendo el sistema muy dinámico.
- Existen alrededor de 20 tipos diferentes de actividades disponibles en Moodle: foros, glosarios, wikis, tareas, quizzes, encuestas, bases de datos (entre otras) y cada una puede ser adaptada a las necesidades propias de cada curso.
- Permite combinar las actividades en secuencias y grupos, ayuda al docente a guiar a los participantes.

### Ventajas

- El profesor tiene absoluto control sobre los contenidos del curso.
- Normalmente, se establecen plazos de entrega de actividades y el profesor monitorea el desarrollo.
- Permite colocar como recurso enunciados de exámenes, y la posibilidad de subir su resultado como archivos adjuntos, con horario de plazo de entrega.
- Completa información del trabajo realizado por los alumnos.
- Reutilización de los cursos.
- Posibilidad de compartir cursos y/o recursos.
- Posibilidad de crear cursos conjuntamente con otros compañeros profesores del mismo o diferente centro.
- Permite colocar recursos variados para formar una unidad de contenidos: etiquetas, archivos en formato variable (texto, audio, vídeo, hoja de cálculo).
- Facilidad de comunicación con sus alumnos y con el resto de profesores del curso.
- Las encuestas que se pueden realizar son de gran utilidad para evaluar el conocimiento inicial de los alumnos en una materia específica o para calificar el desempeño del tutor o profesor del curso.

- La evaluación es continua y permanente: todo se comenta por todos y se evalúa. El profesor da feedback continuo y los estudiantes demandan esta actividad.
- Dispone de varios temas o plantillas que permiten al administrador del sitio personalizar colores y tipos de letra a su gusto o necesidad. Estas plantillas son fáciles de modificar y ampliar.
- Es posible cambiar el modo de edición de profesor a vista del alumno. De esta forma, permite asegurarse que los alumnos vean en la plataforma sólo que deben ver y ocultar el resto.
- Se encuentra traducido a más de 70 idiomas.
- Los estudiantes se familiarizan rápidamente con el entorno de la plataforma.
- Permite que cada estudiante tenga su propio ritmo de trabajo.
- Feedback inmediato en muchas actividades, incluida la evaluación.
- En los exámenes tipo “múltiple choice”, puede verse el resultado inmediatamente después de que el alumno lo terminó.
- Los alumnos pueden participar en la creación de glosarios, y en todas las lecciones se generan automáticamente enlaces a las palabras incluidas en estos.

#### Desventajas

- Prescinde de algunas herramientas pedagógicas, como por ejemplo crucigramas y juegos de roles (role playing).
- Su interfaz necesita mejorarse.
- Hay desventajas asociadas a la seguridad, dependiendo en dónde se esté alojando la instalación de Moodle, cuáles sean las políticas de seguridad y la infraestructura tecnológica con la cual se cuente durante la instalación Zapata, (2010)
- No integra automáticamente el uso de videoconferencias.
- La estructura de navegación, tanto para la creación de contenidos como para la administración del sitio, es poco amigable y utiliza muchos recursos de la red, provocando lentitud en el acceso.
- Por estar basado en tecnología PHP, la configuración de un servidor con muchos usuarios debe ser cuidadosa para obtener un mayor desempeño.

- No tiene la posibilidad de realizar la gestión económica-financiera de alumnos en línea, sobre todo cuando un mismo alumno está inscrito en varios cursos.

Para poder encontrar la mejor plataforma de acuerdo a las necesidades de este proyecto se han establecido criterios y comparado entre las mejores opciones.

Criterio	Moodle	Edu 2.0	Chamilo	Claroline
Gratuidad	SI	SI (Hasta 2000 usuarios)	SI	SI
Agenda	SI	SI	SI	SI
Fácil de usar	SI	SI	SI	SI
Interfaz personalizable	SI	SI	SI	SI
Disponibilidad de idiomas	SI	SI	SI	SI
Pruebas, banco de preguntas, tareas	SI	SI	SI	NO
Plazo de entrega de actividades	SI	NO	SI	NO
Respaldo del Sistema	SI	NO	SI	NO
Facilidad de personalización	SI	NO	NO	NO
Facilidad de incorporar plugins	SI	NO	SI	NO
Vistas Alumno/Profesor	SI	NO	SI	NO
Interacción de la comunidad	ALTA	BAJA	MEDIA	BAJA

**Tabla 1: Tabla Comparativa de Plataformas Web**

Como resolución a toda la investigación expuesta anteriormente se concluye que la plataforma “Moodle” es la mejor opción entre las plataformas e-learning, esto debido que a pesar de ser muy similar a otras, esta cuenta con varios puntos fuertes, como son la seguridad, la cantidad de años que lleva en desarrollo, la gran cantidad de extensiones ya existentes en la web y además el apoyo de una comunidad con millones de usuarios y/o desarrolladores activos cada día.

#### **4.1.5 Software a medida**

Otras de las alternativas viables como solución a la problemática planteada es la creación de un software a medida.

Un software a medida consiste en la creación y fabricación de sistemas informáticos que busca complacer todas las necesidades y adaptarse lo mejor posible a lo que una empresa o cliente necesita, siendo más exclusivos y resolviendo problemas concretos.

Sus principales características son:

- Tiene su tiempo de desarrollo.
- Se adapta a las necesidades específicas de la empresa.
- Es probable que pueda contener errores y se deba mejorar, así como que si el negocio sufre cambios el software también.
- En general, es más costoso que el software estándar.
- Solución personalizada.
- Posee rapidez, facilidad, flexibilidad e integración.
- Se pueden desarrollar e implementar por partes o módulos, asimismo desarrollar adaptaciones o módulos de comunicación hacia otras herramientas.

Para complementar se analizará un caso real donde existía una problemática similar y se dió por solución la construcción de un software a medida.

En el curso de programación C al que se hace referencia en el artículo “A Web Support System for Submission and Handling of Programming Assignments” se utiliza un software realizado a medida para un total de 115 estudiantes pertenecientes a la Universidad de Aalborg en el otoño de 2010.

El programa presenta múltiples perfiles, uno de éstos es el “estudiante”, el cual posee una página web sencilla, con un módulo de identificación para poder tener acceso y una serie de ejercicios que deben realizar periódicamente.

Desde el punto de vista del profesor, la interfaz principal del sistema es una gran tabla donde los estudiantes se enumeran verticalmente, y los ejercicios se enumeran horizontalmente, pudiendo asignar a cada alumno un número indefinido de estos.

Al momento de culminar el curso los alumnos participantes presentaron 501 programas individuales en total. De estas presentaciones 484 tenían una auto-estimación de beneficio para el estudiante y 474 contenían un auto-estimación razonable del consumo de tiempo.

La mayoría de los estudiantes (20 de 25), que no pasaron el examen final, hicieron menos de cuatro programas durante el curso. También se observa que más del 90% de los estudiantes, que resuelven cuatro o más ejercicios de tarea durante el curso, aprobó el examen final.”

Al analizar el artículo, se reafirma la gran ayuda que puede prestar un sistema de estas características para motivar a alumnos que están comenzando con la programación en lenguaje C y lo importante que es la utilización de esta herramienta.

A continuación se presenta una tabla en la que se compara Moodle con el desarrollo a medida:

	Moodle	Software a medida
Tiempo de desarrollo	Medio	Alto
Adaptabilidad a las necesidades	Bajo	Alto
Escalabilidad	Baja	Alta
Estabilidad	Alta	Media

**Tabla 2: Tabla Comparativa de Moodle y Software a medida**

## 4.2 Definir Solución

Luego de hacer la comparación entre la mejor plataforma e-learning de acuerdo a esta investigación y el desarrollo de un software a medida. Se escoge como solución final el desarrollo del software a medida, ya que a pesar de tener un mayor riesgo o tal vez emplear más tiempo de desarrollo los beneficios que puede entregar son muy superiores, como por ejemplo una muy alta escalabilidad, lo que abre paso a posibles mejoras o agregación de módulos no necesariamente realizadas por el equipo de desarrollo actual.

Para la realización del Software a medida se ha optado por la utilización del Framework de php “CodeIgniter”, debido a que este cuenta con un amplio conjunto de librerías para tareas básicas además de ser excepcionalmente rápido (considerado uno de los más veloces en la actualidad). Este Framework usa el enfoque controlador de MVC (Modelo, Vista, Controlador), lo que permite llevar un mejor orden y claridad de lo que se está programando.

Por otro lado, se efectúa una investigación en cuanto a herramientas de compiladores online, compiladores que se caracterizan por recibir código fuente escrito en diferentes lenguajes de programación (junto con las Entradas y Salidas esperadas) y entregar los resultados de su compilación en tiempo real.

En el caso del sistema en construcción, aunque el trabajo está orientado a la programación en lenguaje C, el sistema debe considerar utilizar un compilador multilenguaje para sus futuras versiones, que pueden incluir trabajos en otros lenguajes.

A continuación algunas herramientas de esta índole investigadas:

- **Compileonline:**  
Plataforma web que compila códigos fuentes, tiene incorporado 72 lenguajes de programación, tecnologías web, Javascript Frameworks, Text Formatting, Online Text Editors, Useful Utilities.

- **Ideone:**  
Plataforma web que compila y depura códigos fuentes en línea, posee más de 40 lenguajes de programación para compilar y ejecutar.
  
- **Onlinecompiler:**  
Plataforma web que compila los códigos fuentes, devuelve el archivo correspondiente del lenguaje o también un archivo en .exe (ejecutable de windows), soporta C/C++, FORTRAN, JAVA, PASCAL, BASIC.
  
- **Compilr:**  
Plataforma web que compila códigos fuentes de lenguaje PHP, C, C++, Ruby, Java, C# y VB. Requiere registrarse.
  
- **Codetwist:**  
Plataforma web que compila/interpreta en línea, donde se puede compilar código en C, C ++ o Java y ejecutarlo en línea y ver los resultados.
  
- **GCC (GNU Compiler Collection):**  
Conjunto de compiladores de software libre, considerados estándar para los sistemas operativos derivados de UNIX, es capaz de compilar actualmente para para C, C++, Objective C y Fortran.

Para escoger uno de los compiladores online presentados anteriormente se presenta una tabla comparativa:

Criterio	Compileonline	Ideone	Onlinecompiler	Compilr	Codetwist	GCC
Incorpora lenguaje c	Si	Si	Si	Si	Si	Si
Idioma	Ingles	Ingles/español	Ingles	Ingles	Ingles	Ingles
Ejemplos de códigos	Si	Si	Si	No	No	No
Entrada Estándar (Stdin)	Si	Si	No	Si	Si	Si
Identificación de errores	Si	Si	Si	Si	Si	Si
Descripción de errores	Especificación de línea y columna, descripción con símbolos	Especificación de línea y columna.	Especificación de línea y columna			
Plantilla fijada	Si	Si	Si	Si	No	No
Gratuidad	Si	Si	Si	No	Si	Si
Solicita registro de usuario	No	No	No	Si	No	No
Tipo de Compilador	Online	Online	Online	Online	Online	Local

**Tabla 3: Tabla Comparativa de Compiladores**

Al analizar estos compiladores, los que cumplen con la mayor cantidad de parámetros de selección son Compileonline e Ideone, sin embargo existe uno con una característica en particular, el cual es GCC, el que trabaja de manera local, es decir se ejecuta desde el mismo servidor en el que se encuentre la aplicación.

Se ha optado por GCC, que funciona perfectamente con todo el código C probado, es compatible con diferentes plataformas web, además de poseer un gran punto a su favor: trabaja desde un entorno local, por lo que no se depende de un servidor externo para el correcto funcionamiento del software.

### **4.3 Alcances**

Se implementa un sistema web, el cual permita a los docentes plantear ejercicios de programación, para ello se desarrolla un repositorio on-line con distintos problemas base, a través del cual se permite la creación de cuadernos de ejercicios para cada uno de los alumnos de una asignatura, que contiene problemas dictados por el profesor.

Por otra parte permitir la gestión curso-sección (importación y/o exportación de información). Así como dar soporte a la corrección automática, informando a los estudiantes de su resultado de evaluación y en el caso de malos resultados dar la oportunidad de volver ingresar para obtener una respuesta correcta, que va en el cuaderno resuelto.

Además permitir la generación de reportes del desempeño del alumno en cuanto a los ejercicios resueltos, de manera individual o colectiva. Incluso el docente podrá establecer fechas de caducidad para los ejercicios asignados.

El sistema no realiza la corrección de ejercicios sin que antes el alumno haya ingresado su respuesta, además no permite el uso de otro léxico no especificado debe seguir patrón establecido al momento de compilar códigos como respetar un espacio por cada palabra y utilizar las mismas variables declaradas en el enunciado, además que no permitirá que el alumno realice modificaciones a las pautas establecidas por el profesor.

## 4.4 Objetivo del software

### 4.4.1 Objetivo General

- Desarrollar un repositorio on-line de distintos **problemas base**, que facilite a los profesores de las asignaturas de Informática ingresar y compartir nuevos problemas, catalogados según los temas tratados y el nivel de dificultad de cada uno de ellos.
- Dar soporte a la **corrección automática** de los problemas. Para ello, cada problema debe llevar asociado las pautas que hagan posible su corrección automática (test de pruebas con entradas y salidas esperadas para cada una de ellas).
- Ofrecer, a cada alumno de un curso-sección, un **cuaderno de problemas** para su desarrollo; el cuaderno estará organizado por temas en los que se irán incorporando diferentes problemas según los tópicos que contemple una asignatura. Los ejercicios, para cada tema, se distribuirán aleatoriamente tomados directamente desde el repositorio.

#### 4.4.2 Objetivo Específico

- Recibir, desde cada estudiante, las soluciones propuestas para los problemas asignados, informándoles de la evaluación que éstas obtienen, brindándole la oportunidad de volver a ingresar una respuesta hasta obtener una que sea correcta para todos los casos de prueba. Estas respuestas deben ir formando parte del **cuaderno resuelto** de cada estudiante.
- Permitir a los profesores de Informática gestionar un curso-sección dentro del sistema, importando y/o exportando la información de los alumnos del curso.
- Permitir al docente establecer fechas de caducidad para los ejercicios asignados, de esta manera puede asegurarse que el alumno trabaje a lo largo del semestre, evitando que los resuelva al final del mismo.
- Generar reportes del desempeño académico de los estudiantes, tanto de manera individual como colectiva (como curso).

## **4.5 Descripción Global del Producto**

### **4.5.1 Interfaz de usuario**

La interfaz de usuario contendrá:

- Título.
- Banner.
- Menú de navegación.
- Espacio libre.
- Sidebar.
- Pie de página.

### **4.5.2 Interfaz De Hardware**

Dado que el sistema opera bajo un entorno web, es necesario una capacidad hardware suficiente para poder procesar un explorador web tradicional, para ello las recomendaciones mínimas de hardware que el computador debe poseer son las siguientes:

-Para entornos Windows y Linux:

- Procesador Pentium 4 o superior, compatible con SSE2
- 512 MB de RAM
- 200 MB de espacio en disco

En el caso de profesores en la universidad no es necesario adquirir nuevos dispositivos para implementar el sistema, ya que existen plataformas establecidas en el servidor “..” que trabajan con Moodle.

En cuanto a los alumnos debe considerar los aspectos anteriores, más mouse, teclado, monitor.

### **4.5.3 Interfaz Software**

Para el funcionamiento del sistema, se requiere la instalación de uno de los siguientes Navegadores web:

- Google Chrome:
  - Abreviación: no tiene.
  - Versión: 37.0.2062.120 m

- Mozilla Firefox:
  - Abreviación: no tiene.
  - Versión: 32.0.1
  
- Internet Explorer:
  - Abreviación: no tiene.
  - Versión: 11.0.3
  
- Opera:
  - Abreviación: no tiene.
  - Versión: 24.0
  
- Windows:
  - Abreviación: Win 7 – Win 8
  - Versión: 7 - 8
  
- Linux Ubuntu:
  - Abreviación: no tiene.
  - Versión: 14.04
  
- GCC:
  - Abreviación: GCC
  - Versión: 4.7.2-5

#### 4.5.4 Interfaces de comunicación

El sistema se basa en protocolos de comunicación de internet como:

- TCP/IP: Es un protocolo de comunicación de datos que establece una conexión y realiza intercambio de datos entre ordenadores de distinta red. Tiene gran fiabilidad, proporciona más seguridad y eficiencia en la utilización de recursos. Además está diseñado para enrutar y tener acceso a internet y servidores web.
- HTTP: Es un protocolo de transferencia de hipertexto, que intercambia información en la **World Wide Web**. De esta manera se transfieren las páginas web a un ordenador, realizando así la transferencia de datos entre el servidor y el cliente.

- FTP: Es un protocolo de transferencia de archivos entre sistemas conectados a un red TCP. Su funcionalidad es descargar o subir un archivo a un servidor.
- SSH: Es un protocolo de comunicación en red que permite la administración e intercambio de estos. Sirve para acceder a maquinas remotas a través de una red. Permite la copia de datos de manera segura.
- SOAP: Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.
- HTML: Es la base para la creación de páginas web. Es usado para describir la estructura y el contenido en forma de texto.

## 4.6 Requerimientos Específicos

### 4.6.1 Requerimientos Funcionales del sistema

Id	Nombre	Descripción
1	Autenticar usuario	Pueden acceder al Sistema solo usuarios registrados en la plataforma.
2	Gestionar curso - sección	El Profesor podrá crear un curso - sección en el sistema y eliminar estos que ya han sido creados.
3	Gestionar alumnos	El Profesor podrá ingresar, modificar y eliminar alumnos a sus cursos creados.
4	Gestionar problemas	El Profesor ingresara un problema junto a sus respectivos datos y casos de pruebas. El Administrador podrá modificar problema.
5	Gestionar cuaderno de ejercicio	El Profesor podrá crear un cuaderno de ejercicios con diversos problemas y tópicos para cada alumno ingresado, aparte se irá actualizando cada vez que se ingresen nuevos tópicos al cuaderno y también podrá ser eliminado.
6	Resolver ejercicio	Alumno escoge un problema y plantea una solución propia.
7	Evaluar solución	El Sistema analizara los parámetros ingresados por el alumno consultando con la base de datos y

		retornara la compatibilidad existente con los parámetros registrados en la base de datos de posible solución. Los ejercicios deben ser resueltos en un tiempo determinado, si es que así lo requirió el Profesor en la opción de caducidad.
8	Emitir informes	El Sistema generará distintos tipos de informes.
9	Gestionar profesor	El Administrador podrá registrar, modificar y eliminar un usuario profesor.

**Tabla 4: Tabla de Requerimientos Funcionales del sistema**

#### 4.6.2 Interfaces externas de entrada

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem
DU	Datos de Usuario	Perfil, Username, Password
DP	Datos de Problema	ID, Tipo, Nombre, Descripción, Valor de entrada1, Valor de entrada2, Valor de entrada3, Valor de salida1, Valor de salida2, Valor de salida3
DC	Datos de Cuaderno	Username, ID, Tipo, Progreso, Intentos, Caducidad, Estado

**Tabla 5: Tabla de Interfaces Externas de Entrada**

#### 4.6.3 Interfaces externas de Salida

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem	Medio Salida
IDC	Informe de Rendimiento Curso	Rut, Nombre, Apellido Progreso	Archivo PDF Pantalla
IDA	Informe de Rendimiento Alumno	Nombre, Tipo, Descripción, Intentos, Estado	Archivo PDF Pantalla

**Tabla 6: Tabla de Interfaces Externas de Salida**

#### 4.6.4 Atributos del producto

Tomando como referencia el modelo ISO/IEC 9126, se indican los atributos de calidad del producto software que se desarrolla. Considere que no es necesario que todos los atributos sean aplicables al producto, solo indique los que se consideran críticos. Defina en términos claros como se concretan cada uno de los atributos de calidad de forma que sean fáciles de comprobar.

Por ejemplo:

- **USABILIDAD - OPERABILIDAD:** El Sistema debe otorgar comprensión y operatividad para que los usuarios logren reconocer el uso y control del software. Además de una interactiva y amigable interfaz, que facilita aún más su uso mediante link, campos de textos y opciones de menú. En cuanto a los errores muestra mensajes de forma clara mediante una ventana externa sobre la pantalla.
- **EFICIENCIA - TIEMPO DE EJECUCIÓN/RESPUESTA:** El Sistema debe presentar una alta estabilidad entre el desempeño del software y los recursos necesarios. Teniendo un buen comportamiento en el tiempo de respuesta y desempeño, que dependerá de la conexión de internet disponible, teniendo una demora como máximo de 30 segundos.
- **FUNCIONALIDAD - SEGURIDAD:** El Sistema debe mantener un control de acceso a la funcionalidad a través de login –password establecido según los perfiles definidos para los usuarios del sistema. Es por ello que cada usuario registrado posee distintos privilegios y podrá acceder a cada tipo de información de acuerdo a sus privilegios, además de que si no posee cuenta no podrá acceder.
- **PORTABILIDAD:** El Sistema debe poseer la capacidad de instalación para un ambiente específico y la adaptabilidad en cuanto a funcionar en diferentes ambientes sin aplicar otras acciones. Es por ello que es compatible tanto nivel servidor como cliente en cualquier Sistema Operativo.
- **RENDIMIENTO:** El Sistema deberá comprobar si su rendimiento es óptimo, de acuerdo a su funcionamiento como en el caso de ingresar códigos fuentes para el análisis de las posibles soluciones, si logra verificar un buen o mal resultado estará realizando un excelente desempeño como ayuda a las asignaturas de informática.

---

## 5 FACTIBILIDAD

---

### 5.1 Factibilidad técnica.

El Departamento de Sistemas de información consta con un adecuado equipamiento y dispositivos que posibilitan un correcto uso del Sistema de Apoyo al Aprendizaje de Programación Estructurada, por lo tanto no es necesaria la adquisición de algún software con su versión y tipo de licencia.

En este caso el servidor habilitado por el Laboratorio Transferencia Tecnológica perteneciente al departamento de Sistemas de Información es más que suficiente, ya se compone con los siguientes recursos:

- Servidor: HP Proliant ML 150G6 – Torre.
- Procesador: 1 Procesador Intel Xeon QUAD core E5504 (2.0 GHZ/1X4 Mb cache L3).
- Memoria: 8 GB RAM
- Almacenamiento: Dos Discos de 450 Gb SAS 15000 rpm
- Sistema Operativo: Linux Debian 7.0 – 64 Bits
- Soporte: Soporta Sistemas Raid.
- Postgresql 9.1
- Php 5.4
- PhPpgAdmin 5.0.4

Por ende con estas características se puede inferir que el funcionamiento del software desarrollado es eficiente y óptimo, sin tener que conseguir algún nuevo dispositivo o equipamiento.

También al poseer las habilidades, técnicas y conocimientos con respecto a estos recursos, se logra un pleno desarrollo del software sin recurrir a capacitaciones, más bien con mínimas fuentes de información con respecto al lenguaje Php y el framework CodeIgniter utilizados en esta elaboración.

Por lo tanto se concluye que la factibilidad técnica con los recursos descritos anteriormente es aceptable para un software con estas particularidades.

## 5.2 Factibilidad operativa.

El desarrollo de este software tiene los siguientes impactos positivos:

- Facilitar a las asignaturas de programación en lenguaje C un sistema de trabajo online, donde podrán publicar guías de trabajo para los alumnos, creando un cuaderno para cada asignatura con sus ejercicios correspondiente.
- Permitir el control de los avances que los alumnos realicen en sus cuadernos de trabajo, para que así el profesor logre reforzar los contenidos con mayor dificultad del curso correspondiente.
- Facultar la posibilidad al profesor de establecer un fecha de caducidad para que los alumnos realicen sus ejercicios del cuaderno.
- La repartición del tiempo en cuanto a las funciones que debe ejercer el profesor serán optimizadas, ya que en este caso las correcciones de los ejercicios se harán de manera online.
- Los alumnos podrán obtener una respuesta de corrección inmediata en cuanto a sus soluciones propuestas para cada ejercicio, teniendo además la alternativa de resolverlo más veces hasta obtener un resultado correcto, registrando a la vez la cantidad de intentos realizados.
- El sistema proporciona la asignación de distintos ejercicios por alumno, por lo que le ayuda al docente a disminuir la probabilidad de copia entre ellos.

Por lo tanto se concluye que la factibilidad operativa con las características positivas descritas anteriormente es aceptable para un software con estas particularidades.

### 5.3 Factibilidad económica.

Los beneficios esperados al desarrollar el proyecto son que sea una herramienta útil para los docentes, ya que se automatiza el proceso de ejercitación en lenguaje C para las distintas asignaturas que trabajen con él, además de la generación de informes para los docentes en cuanto al rendimiento de los alumnos en los ejercicios. Además el departamento consta con las herramientas necesarias para el desarrollo y uso del software como el servidor, por lo que no se necesita invertir en la adquisición de herramientas de trabajo para la implementación, ya que se utiliza un software libre. Por lo que se excluyen del gasto de desarrollo del sistema.

A continuación se presenta una tabla con el detalle de gasto eléctrico del total de horas invertidas del proyecto.

Equipo	Cantidad	Consumo de energía (W/h)	Tiempo de consumo(hrs)	Total de Consumo
Notebook	2	50	1690	169000
Router	2	6	1690	20280
Luz Eléctrica	2	33	1690	111540
<b>Total W</b>				300820
<b>Total KW</b>				300.82
<b>Valor (KW/h)</b>				118
<b>Total en \$</b>				35.496

**Tabla 7: Costo Energía Eléctrica**

Una idea para cuantificar este sistema, es estimar un costo del software en base al total de horas invertidas en el desarrollo correspondientes a 1690 horas y un valor de hora hombre de \$6.250, lo que arroja un total de \$10.562.500 en mano de obra.

Recurso	Total
Recurso Humano	\$10.562.500
Recurso Eléctrico	\$35.496
Servicio Internet	\$353.280
<b>Total</b>	<b>\$10.951.276</b>

**Tabla 8: Presupuesto del Proyecto**

## 5.4 Conclusión de la factibilidad

Por medio del análisis realizado se manifiestan las necesidades de las diversas asignaturas de programación de crear un sistema que ayude con el aprendizaje de la programación estructurada, en cuanto al desarrollo de ejercicios prácticos y el monitoreo del avance de estos mismos.

Por lo que se espera que este sistema cuente con la creación de un cuaderno de ejercicios para los alumnos, corrección automática de los ejercicios, monitoreo del avance por ejercicios, disminución del tiempo de corrección y reprobación de asignaturas e identificar los tópicos con mayor facilidad.

Con cada factibilidad vista se puede concluir que la facultad cuenta con el equipo de software y hardware necesario para la implementación del sistema, sin la necesidad de invertir en estos puntos.

En el ámbito operativo se mejora la manipulación de información y automatiza los procesos de las distintas asignaturas para tener una forma de trabajo más ameno, en cuanto a la enseñanza y ejercitación de la programación.

Y en el ámbito económico se concluye que la facultad no necesita costear el sistema, ya que cuenta con los recursos necesarios para su implementación y funcionamiento.

Por lo tanto se concluye que el proyecto es factible de realizar al no tener costos de implementación y desarrollo, además de ser beneficioso para quienes lo utilicen, colaborando con su trabajo y aprendizaje.

---

## 6 ANÁLISIS

---

### 6.1 Diagrama de casos de uso

#### 6.1.1 Actores

- **Administrador:**  
Es el encargado de gestionar al profesor en el sistema. Es decir, podrá registrar, modificar y eliminar al usuario profesor en el sistema siempre que se requiera.  
En cuanto a nivel de conocimiento técnico debe poseer nivel usuario y su nivel de privilegio es bajo por el rol que cumple.
- **Profesor:**  
Es el encargado de gestionar curso-sección, alumnos, problemas y cuaderno de ejercicio. Es decir, podrá ingresar, modificar y eliminar cada uno de estos casos en el sistema de acuerdo a las necesidades de cada asignatura de programación en lenguaje C.  
En cuanto a nivel de conocimiento técnico debe poseer nivel usuario y su nivel de privilegio es alto por el rol que cumple.
- **Alumno:**  
Son los estudiantes de las carreras ICINF o IECI que inscriben las asignaturas de programación en lenguaje C. Ellos deben resolver los ejercicios planteados en el cuaderno y proponer una solución propia para cada problema.  
En cuanto a nivel de conocimiento técnico debe poseer nivel usuario y su nivel de privilegio es bajo por el rol que cumple.
- **Sistema Web SAAPE:**  
Es donde trabajarán las distintas asignaturas de programación en lenguaje C, registrando el material de ejercicios de cada ramo. Además de evaluar la solución propuesta por cada estudiante analizando los parámetros ingresados por el alumno consultando con la base de datos y retornará la

compatibilidad existente con los parámetros registrados en la base de datos de posible solución.

También emitirá informes del rendimiento académico y autentificará al usuario.

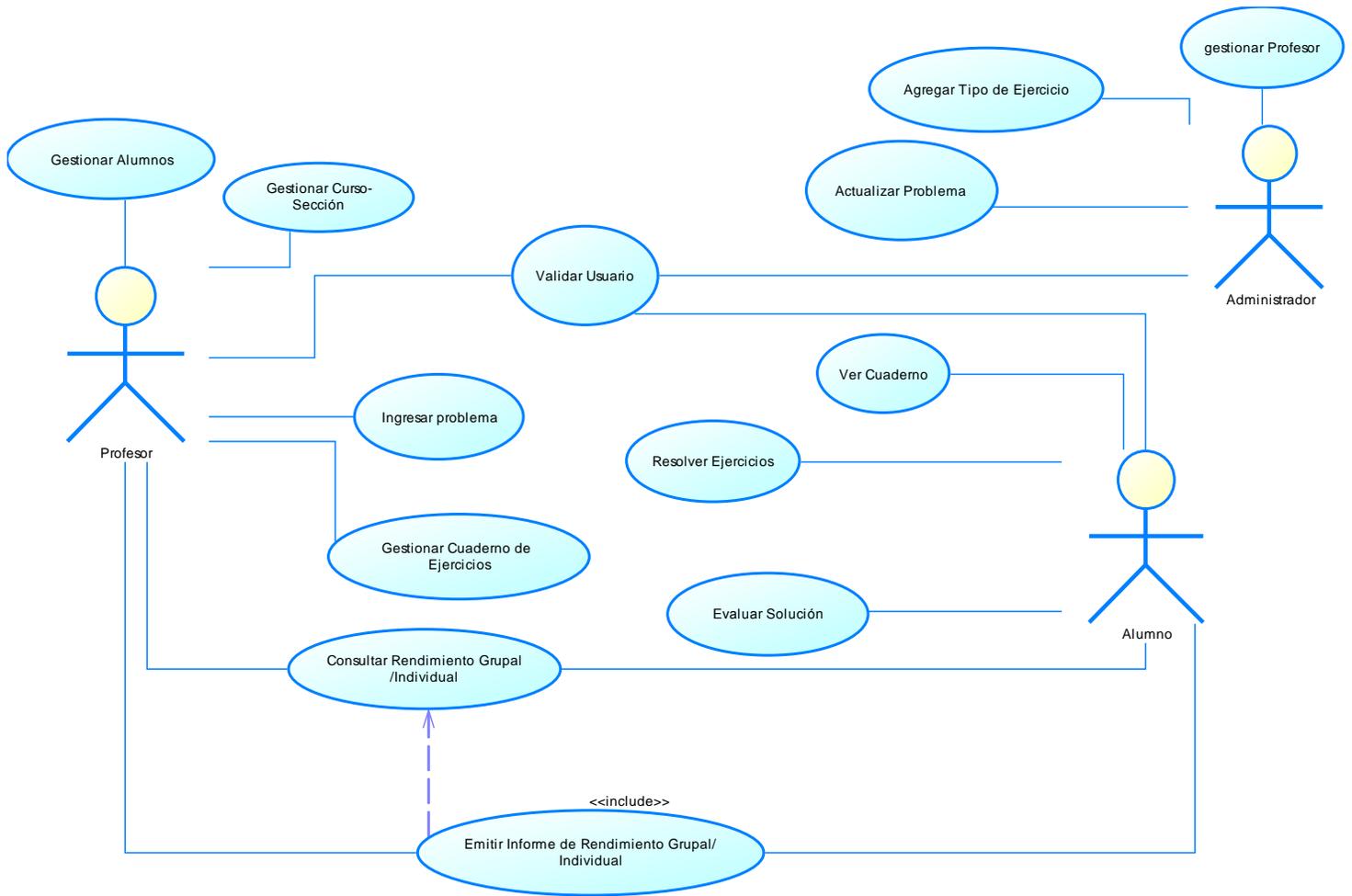
- GCC:  
Es el encargado de compilar y ejecutar la solución propuesta por los alumnos de algún problema específico, entregando la respuesta obtenida al sistema.

### **6.1.2 Casos de Uso y descripción**

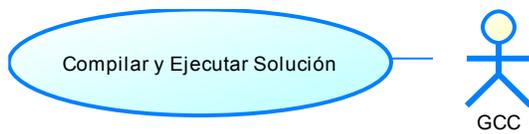
En el diagrama de Casos de uso se refleja la interacción entre el sistema y los actores Profesor, Administrador, Alumno, Sistema Web SAAPE e GCC.

Los actores Profesor, Administrador y Alumno se relacionan con el sistema a través de computador conectado a internet, en donde están registrados con un usuario correspondiente y autenticándolos cada vez que realizan el ingreso al sistema. En donde una vez ingresados podrán visualizar los distintos cursos de las diversas asignaturas de programación en lenguaje C con sus cuadernos con ejercicios para cada alumno.

El actor GCC es un compilador local que admite una multitud de lenguajes, que en este caso compila y ejecuta las soluciones propuestas por los alumnos a cada problema planteado en sus cuadernos de trabajo.



**Diagrama 4: Casos de Uso General**



**Diagrama 5: Casos de Uso Compilador GCC**

### 6.1.3 Especificación de los Casos de Uso

#### 6.1.3.1 Caso de Uso: <Gestionar Curso-Sección>

Este caso de uso implica dos acciones:

##### 6.1.3.1.1 “Crear Curso-Sección”:

- Descripción: El Profesor podrá crear un curso - sección en el sistema, de alguna asignatura específica que se imparta.
  
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Profesor.
  
- Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona la opción “Administrar Curso”	2) Despliega el menú con las opciones para curso.
3) Selecciona la opción “Crear Curso”	4) Despliega formulario para Completar con datos del curso.
5) El actor completa el formulario y selecciona la opción guardar los datos ingresados.	6) El sistema muestra mensaje de éxito.

**Tabla 9: Flujo de Eventos Básicos CU Crear Curso-Sección**

- Flujo de Eventos Alternativo:

Al actor	El sistema
5(a) El actor ingresa datos erróneos dentro de los campos del formulario	7) El sistema muestra un mensaje de error.
5(b) El actor no completa todos los datos que son obligatorios para hacer el correcto registro del curso.	8) El sistema muestra un mensaje de datos incompletos.

**Tabla 10: Flujo de Eventos Alternativo CU Crear Curso-Sección**

- Post-Condiciones: La base de datos se actualiza con los datos registrados.

**6.1.3.1.2 “Eliminar Curso-Sección”**

- Descripción: El Profesor podrá eliminar un curso - sección en el sistema, de alguna asignatura específica que se imparta.
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Profesor.

- Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona la opción “Administrar Curso”	2) Despliega el menú con las opciones para curso.
3) Selecciona la opción “Eliminar Curso”	4) Elimina Curso-Sección.

**Tabla 11: Flujo de Eventos Básicos CU Eliminar Curso-Sección**

- Flujo de Eventos Alternativo:

Al actor	El sistema
5(a)	7)

**Tabla 12: Flujo de Eventos Alternativo CU Eliminar Curso-Sección**

- Post-Condiciones: La base de datos anula los datos registrados del curso, alumnos, cuadernos.

**6.1.3.2 Caso de Uso: <Gestionar Alumnos>**

Este caso de uso implica tres acciones:

**6.1.3.2.1 “Ingresar Alumnos”**

- Descripción: El Profesor podrá ingresar alumnos a un curso determinado registrado en el sistema.

- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Profesor.
  
- Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona la opción “Administrar Curso”	2) Despliega el menú con las opciones para curso.
3) Selecciona la opción “Ver Curso”	4) Despliega listado de cursos creados por el profesor.
5) Selecciona el curso para inscribir alumnos	6) Visualiza los datos del curso seleccionado, con opción “Inscribir Alumnos”.
7) Selecciona opción “Inscribir Curso”	8) Muestra dos opciones “Inscripción manual” e “Inscripción automática”

**Tabla 13: Flujo de Eventos Básicos CU Ingresar Alumnos**

- Flujo de Eventos Alternativo:

Al actor	El sistema
3(a) Selecciona la opción “Ver Curso”	9) No visualiza el curso en el que desea inscribir alumnos.

**Tabla 14: Flujo de Eventos Alternativo CU Ingresar Alumnos**

- Post-Condiciones: La base de datos registra los alumnos dentro del curso.

#### 6.1.3.2.2 “Modificar Alumnos”

- Descripción: El Profesor podrá modificar alumnos a un curso determinado registrado en el sistema.
  
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Profesor.

- Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona la opción "Administrar Curso"	2) Despliega el menú con las opciones para curso.
3) Selecciona la opción "Ver Curso"	4) Despliega listado de cursos creados por el profesor.
5) Selecciona el curso para modificar alumnos	6) Visualiza el listado con alumnos correspondientes al curso escogido, donde aparece la opción de "actualizar".
7) Selecciona opción "Actualizar"	8) Muestra los campos editables del alumno escogido.
9) Selecciona opción "Guardar"	10) Vuelve a mostrar lista de curso.

**Tabla 15: Flujo de Eventos Básicos CU Actualizar Alumnos**

- Flujo de Eventos Alternativo:

Al actor	El sistema
*No aplica	

**Tabla 16: Flujo de Eventos Alternativo CU Actualizar Alumnos**

- Post-Condiciones: La base de datos actualiza los datos del alumno.

### 6.1.3.2.3 "Eliminar Alumnos"

- Descripción: El Profesor podrá eliminar alumno en un curso determinado registrado en el sistema.
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Profesor.

- Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona la opción "Administrar Curso"	2) Despliega el menú con las opciones para curso.
3) Selecciona la opción "Ver Curso"	4) Despliega listado de cursos creados por el profesor.
5) Selecciona el curso para modificar alumnos	6) Visualiza el listado con alumnos correspondientes al curso escogido, donde aparece la opción de "borrar".
7) Selecciona opción "borrar"	8) Muestra un mensaje de confirmación para eliminar alumno.

**Tabla 17: Flujo de Eventos Básicos CU Eliminar Alumnos**

- Flujo de Eventos Alternativo:

Al actor	El sistema
*No aplica	

**Tabla 18: Flujo de Eventos Alternativo CU Eliminar Alumnos**

- Post-Condiciones: La base de datos elimina alumno de su registro.

### 6.1.3.3 Caso de Uso: <Ingresar Problemas>

- Descripción: El Profesor ingresara un problema junto a sus respectivos datos y casos de pruebas.
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Profesor.

- Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona la opción “Crear Problema”	2) Despliega formulario con los datos que debe completar.
3) El actor completa el formulario y selecciona la opción guardar los datos ingresados.	4) El sistema muestra mensaje de éxito.

**Tabla 19: Flujo de Eventos Básicos CU Ingresar Problema**

- Flujo de Eventos Alternativo:

Al actor	El sistema
3(a) El actor ingresa datos erróneos dentro de los campos del formulario	7) El sistema muestra un mensaje de error.
3(b) El actor no completa todos los datos que son obligatorios para hacer el correcto registro de los problemas.	8) El sistema muestra un mensaje de datos incompletos.

**Tabla 20: Flujo de Eventos Alternativo CU Ingresar Problema**

- Post-Condiciones: La base de datos registra el problema en el repositorio.

#### 6.1.3.4 Caso de Uso: <Gestionar Cuaderno de Ejercicios>

Este caso de uso implica tres acciones:

##### 6.1.3.4.1 “Crear Cuaderno de ejercicios”

- Descripción: El Profesor podrá crear un cuaderno de ejercicios con diversos problemas y tópicos para cada alumno registrado en un curso determinado.
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.

- El usuario debe iniciar sesión como Profesor.

▪ Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona la opción “Administrar Curso”	2) Despliega el menú con las opciones para curso.
3) Selecciona la opción “Crear Cuaderno de Ejercicios”	4) Despliega formulario con los datos que debe completar.
3) El actor completa el formulario y selecciona la opción guardar los datos ingresados.	4) El sistema muestra mensaje de éxito.

**Tabla 21: Flujo de Eventos Básicos CU Crear Cuaderno de Ejercicios**

▪ Flujo de Eventos Alternativo:

Al actor	El sistema
3(a) El actor ingresa datos erróneos dentro de los campos del formulario	7) El sistema muestra un mensaje de error.
3(b) El actor no completa todos los datos que son obligatorios para hacer el correcto registro del cuaderno.	8) El sistema muestra un mensaje de datos incompletos.

**Tabla 22: Flujo de Eventos Alternativo CU Crear Cuaderno de Ejercicios**

▪ Post-Condiciones: La base de datos registra el cuaderno para los alumnos.

**6.1.3.4.2 “Actualización Cuaderno de Ejercicios”**

▪ Descripción: El Profesor actualizara el cuaderno de ejercicios a medida que incorpore nuevos tópicos.

▪ Pre-Condiciones:

- El usuario debe estar registrado en el Sistema Web SAAPE.
- El usuario debe iniciar sesión como Profesor.

▪ Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona la opción “Administrar Curso”	2) Despliega el menú con las opciones para curso.
3) Selecciona la opción “Crear Cuaderno de Ejercicios”	4) Despliega formulario con los datos que debe completar.
3) El actor completa el formulario y selecciona la opción guardar los datos ingresados.	4) El sistema muestra mensaje de éxito.

**Tabla 23: Flujo de Eventos Básicos CU Actualización Cuaderno de Ejercicios**

▪ Flujo de Eventos Alternativo:

Al actor	El sistema
3(a) El actor ingresa datos erróneos dentro de los campos del formulario	7) El sistema muestra un mensaje de error.
3(b) El actor no completa todos los datos que son obligatorios para hacer el correcto registro del cuaderno.	8) El sistema muestra un mensaje de datos incompletos.

**Tabla 24: Flujo de Eventos Alternativo CU Actualización Cuaderno de Ejercicios**

- Post-Condiciones: La base de datos actualiza el cuaderno para los alumnos.

**6.1.3.4.3 “Eliminar Cuaderno de Ejercicios”**

Descripción: El Profesor elimina el cuaderno.

▪ Pre-Condiciones:

- El usuario debe estar registrado en el Sistema Web SAAPE.
- El usuario debe iniciar sesión como Profesor.

- Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona la opción “ver Curso”	2) Despliega el menú con la lista de cursos existentes.
3) Selecciona la opción “editar” del curso correspondiente.	4) Muestra la opción borrar curso, ya que en esta opción se elimina cuaderno.
3) Selecciona opción “Borrar”	4) El sistema muestra mensaje de éxito.

**Tabla 25: Flujo de Eventos Básicos CU Eliminar Cuaderno de Ejercicios**

- Flujo de Eventos Alternativo:

Al actor	El sistema
*No aplica	

**Tabla 26: Flujo de Eventos Alternativo CU Eliminar Cuaderno de Ejercicios**

- Post-Condiciones: La base de datos elimina cuadernos.

#### 6.1.3.5 Caso de Uso: <Consultar Rendimiento grupal/individual>

- Descripción: El Profesor o Alumno puede consultar por el rendimiento, de acuerdo a su perfil podrá visualizar el progreso grupal del curso completo o individual por alumno.
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Profesor o Alumno.

- Flujo de Eventos Básicos Profesor:

Al actor	El sistema
1) Selecciona la opción “Administrar Curso”	2) Despliega el menú con las opciones para curso.

3) Selecciona la opción "Ver Curso"	4) Despliega listado de cursos creados por el profesor.
5) Selecciona la opción "Rendimiento"	6) Visualiza una tabla con todos los alumnos pertenecientes al curso y su progreso.
7) Selecciona un alumno específico	8) Visualiza una tabla con el progreso de este alumno.

**Tabla 27: Flujo de Eventos Básicos CU Consultar Rendimiento grupal/individual Profesor**

- Flujo de Eventos Alternativo Profesor:

Al actor	El sistema
3(a) Selecciona la opción "Ver Curso"	9) No registra el curso deseado.
5(a) Selecciona la opción "Rendimiento"	10) No se visualiza ningún alumno, ya que no están inscritos.

**Tabla 28: Flujo de Eventos Alternativo CU Consultar Rendimiento grupal/individual Profesor**

- Flujo de Eventos Básicos Alumno:

Al actor	El sistema
1) Selecciona opción "Rendimiento"	2) Visualiza la información de progreso que el alumno ha desarrollado en su cuaderno.

**Tabla 29: Flujo de Eventos Básicos CU Consultar Rendimiento individual Alumno**

- Flujo de Eventos Alternativo Alumno:

Al actor	El sistema
1(a) Selecciona opción "Rendimiento"	3) No registra cuaderno de ejercicios.

**Tabla 30: Flujo de Eventos Alternativo CU Consultar Rendimiento individual Alumno**

- Post-Condiciones: Si el caso de uso se ejecuta de forma correcta, el sistema desplegará la información de rendimiento.

**6.1.3.6 Caso de Uso: <Emitir Informe Rendimiento grupal/individual>**

- Descripción: El Profesor o Alumno puede emitir informe de rendimiento, de acuerdo a su perfil podrá visualizar el progreso grupal del curso completo o individual por alumno.
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Profesor o Alumno.

▪ Flujo de Eventos Básicos Profesor:

Al actor	El sistema
1) Selecciona la opción “Administrar Curso”	2) Despliega el menú con las opciones para curso.
3) Selecciona la opción “Ver Curso”	4) Despliega listado de cursos creados por el profesor.
5) Selecciona la opción “Rendimiento”	6) Visualiza una tabla con todos los alumnos pertenecientes al curso y su progreso.
7) Seleccionar la opción PDF	8) Visualiza PDF con información
9) Selecciona un alumno específico	10) Visualiza una tabla con el progreso de este alumno.
11) Seleccionar la opción PDF	12) Visualiza PDF con información

**Tabla 31: Flujo de Eventos Básicos CU Emitir Informe Rendimiento grupal/individual Profesor**

- Flujo de Eventos Alternativo Profesor:

Al actor	El sistema
3(a) Selecciona la opción "Ver Curso"	9) No registra el curso deseado.
5(a) Selecciona la opción "Rendimiento"	10) No se visualiza ningún alumno, ya que no están inscritos.

**Tabla 32: Flujo de Eventos Alternativo CU Emitir Informe Rendimiento grupal/individual Profesor**

- Flujo de Eventos Básicos Alumno:

Al actor	El sistema
1) Selecciona opción "Rendimiento"	2) Visualiza la información de progreso que el alumno ha desarrollado en su cuaderno.
3) Seleccionar la opción PDF	4) Visualiza PDF con información

**Tabla 33: Flujo de Eventos Básicos CU Emitir Informe Rendimiento individual Alumno**

- Flujo de Eventos Alternativo Alumno:

Al actor	El sistema
1(a) Selecciona opción "Rendimiento"	4) No registra cuaderno de ejercicios.

**Tabla 34: Flujo de Eventos Alternativo CU Emitir Informe Rendimiento individual Alumno**

- Post-Condiciones: Si el caso de uso se ejecuta de forma correcta, el sistema desplegará la información de rendimiento.

**6.1.3.7 Caso de Uso: <Gestionar Profesor>**

Este caso de uso implica tres acciones:

**6.1.3.7.1 “Registrar Profesor”**

- Descripción: El Administrador podrá registrar profesor en el sistema.
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Administrador.
- Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona opción “Administrar Docente”	2) Despliega el menú con las opciones para curso.
3) Selecciona opción “Crear docente”	4) El sistema muestra el formulario con los datos que debe completar.
5) El actor completa el formulario y selecciona la opción guardar los datos ingresados.	6) El sistema muestra mensaje de éxito.

**Tabla 35: Flujo de Eventos Básicos CU Registrar Profesor**

- Flujo de Eventos Alternativo:

Al actor	El sistema
5(a) El actor ingresa datos erróneos dentro de los campos del formulario	7) El sistema muestra un mensaje de error.
5(b) El actor no completa todos los datos que son obligatorios.	8) El sistema muestra un mensaje de datos incompletos.

**Tabla 36: Flujo de Eventos Alternativo CU Registrar Profesor**

- Post-Condiciones: La base de datos registra al profesor.

**6.1.3.7.2 “Eliminar/Modificar Profesor”**

- Descripción: El Administrador podrá modificar o anular profesor en el sistema.
  
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Administrador.
  
- Flujo de Eventos Básicos:

Al actor	El sistema
2) Selecciona opción “Administrar Docente”	2) Despliega el menú con las opciones para curso.
5) Selecciona opción “Lista de profesores”	6) El sistema muestra listado de profesores registrados.
7) Selecciona “actualizar”	8) Despliega formulario con los datos registrados del profesor.
9) Selecciona “borrar”	10) Elimina profesor.

**Tabla 37: Flujo de Eventos Básicos CU Eliminar/Modificar Profesor**

- Flujo de Eventos Alternativo:

Al actor	El sistema
7(a) El actor ingresa datos erróneos dentro de los campos del formulario	11) El sistema muestra un mensaje de error.
7(b) El actor no completa todos los datos que son obligatorios para hacer el correcto registro del cuaderno.	12) El sistema muestra un mensaje de datos incompletos.
5(a) Selecciona opción “Lista de profesores”	13) No existe registro de profesor.

**Tabla 38: Flujo de Eventos Alternativo CU Eliminar/Modificar Profesor**

Post-Condiciones: La base de datos modifica o anula al profesor.

**6.1.3.8 Caso de Uso: <Agregar Tipo de Problema>**

- Descripción: Administrador agrega tipo de problema al sistema.
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Administrador.
- Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona opción “Agregar tipo de problema”	2) Visualiza listado existente de tipos de problemas, mas campo para ingresar nuevo tipo de problema.
3) Selecciona botón “guardar”	4) Despliega un mensaje de éxito.

**Tabla 39: Flujo de Eventos Básicos CU Agregar Tipo de Ejercicio**

- Flujo de Eventos Alternativo:

Al actor	El sistema
*No aplica	

**Tabla 40: Flujo de Eventos Alternativo CU Agregar Tipo de Problema**

- Post-Condiciones: La base de datos registra nuevo tipo de problema.

**6.1.3.9 Caso de Uso: <Actualizar Problema>**

- Descripción: Administrador actualiza problema registrado en el sistema.
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario debe iniciar sesión como Administrador.

- Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona opción “Agregar problema”	2) Visualiza listado existente de problemas con la opción “actualizar”
3) Selecciona botón “actualizar”	4) Actualiza los datos del problema.
5) Selecciona botón “guardar”	6) Despliega mensaje de éxito.

**Tabla 41: Flujo de Eventos Básicos CU Actualizar Problema**

- Flujo de Eventos Alternativo:

Al actor	El sistema
*No aplica	

**Tabla 42: Flujo de Eventos Alternativo CU Actualizar Problema**

- Post-Condiciones: La base de datos actualiza problema existente.

### 6.1.3.10 Caso de Uso: <Ver Cuaderno>

- Descripción: Alumno visualiza su cuaderno, encontrando los ejercicios asignados al curso correspondiente.

- Pre-Condiciones:

- El usuario debe estar registrado en el Sistema Web SAAPE.
- El usuario debe iniciar sesión como Alumno.

- Flujo de Eventos Básicos:

Al actor	El sistema
7) Selecciona opción “Ver Cuaderno”	8) Visualiza listado de ejercicios para resolver.

**Tabla 43: Flujo de Eventos Básicos CU Ver Cuaderno**

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a) Selecciona opción “Ver Cuaderno”	9) Visualiza No existen ejercicios asignados al cuaderno.

**Tabla 44: Flujo de Eventos Alternativo CU Ver Cuaderno**

- Post-Condiciones: Si el caso de uso se ejecuta de forma correcta, el sistema desplegará la información de rendimiento.

#### 6.1.3.11 Caso de Uso: <Resolver Ejercicio>

- Descripción: Alumno escoge un problema y plantea una solución propia.
- Pre-Condiciones:
  - El Profesor debe registrar en el sistema al alumno.
  - El usuario debe iniciar sesión como Alumno.
  - En el sistema debe existir al menos un cuaderno y un problema.
  - Debe haber consultado en ver cuaderno.

- Flujo de Eventos Básicos:

Al actor	El sistema
1) El usuario selecciona un ejercicio del listado en su cuaderno, presiona el botón Resolver.	2) Se conecta a la base de datos y retorna enunciado del problema junto a los datos correspondientes y una zona para resolver el problema.
3) El usuario ingresa su propuesta de solución (código).	
4) Presiona el botón Enviar Solución.	

**Tabla 45: Flujo de Eventos Básicos CU Resolver Ejercicio**

- Flujo de Eventos Alternativo:

Al actor	El sistema
3.1) El usuario no ingresa solución propuesta.	5) No realiza ninguna operación, ni considera esto como un intento de solución.

**Tabla 46: Flujo de Eventos Alternativo CU Resolver Ejercicio**

- Post-Condiciones: La base de datos comparara los datos ingresados como solución del ejercicio y retornara la compatibilidad existente en los casos de prueba.

**6.1.3.12 Caso de Uso: <Evaluar solución>**

- Descripción: Alumno ingresa su solución de algún problema y la envía.
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario como Alumno debe iniciar sesión
- Flujo de Eventos Básicos:

Al actor	El sistema
1) Selecciona la opción “Ver Compilador”	2) Abre una pestaña donde se muestra el campo para ingresar código del alumno.
3) Ingresa código de respuesta, selecciona la opción “Compilar” y posteriormente la opción “Enviar Respuesta”.	4) Evalúa la solución enviada, y muestra un mensaje de éxito.

**Tabla 47: Flujo de Eventos Básicos CU Evaluar Solución**

- Flujo de Eventos Alternativo:

Al actor	El sistema
1)(b)Selecciona la opción "Ver Compilador"	2)Abre una pestaña donde se muestra el campo para ingresar código del alumno.
3)Ingresa código de respuesta, selecciona la opción "Compilar" y posteriormente la opción "Enviar Respuesta".	4)Evalúa la solución enviada, y muestra un mensaje de solución incorrecta.

**Tabla 48: Flujo de Eventos Alternativo CU Evaluar Solución**

- Post-Condiciones: Si el caso de uso se ejecuta de forma correcta, el sistema cambia el estado del problema ha completado, en caso contrario se suma un nuevo intento de resolver.

### 6.1.3.13 Caso de Uso: <Validar Usuario>

- Descripción: Acceso al Sistema usuarios registrados en la plataforma.
- Pre-Condiciones:
  - El usuario debe iniciar sesión en el Sistema web SAAPE.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1) Se conecta al sistema para obtener datos de usuario.
	2) Se visualiza el perfil del usuario, según su autenticación.

**Tabla 49: Flujo de Eventos Básicos CU Autenticar Usuario**

- Flujo de Eventos Alternativo:

Al actor	El sistema
	2.1) No tiene acceso al sistema el usuario no autenticado.

**Tabla 50: Flujo de Eventos Alternativo CU Autenticar Usuario**

- Post-Condiciones: usuario autenticado en sistema.

#### 6.1.3.14 Caso de Uso: <Compilar y ejecutar solución>

- Descripción: Compila y ejecuta código fuente enviado por alumno.
- Pre-Condiciones:
  - El usuario debe estar registrado en el Sistema Web SAAPE.
  - El usuario como Alumno debe iniciar sesión
  - El usuario debe poseer su cuaderno ejercicio correspondiente.
  - El alumno debe haber ingresado a resolver problema.

- Flujo de Eventos Básicos:

Al actor	El sistema
5) Selecciona la opción "Ver Compilador"	6) Se conecta al compilador. Abre una pestaña donde se muestra el campo para ingresar código del alumno.
7) Ingresa código de respuesta, selecciona la opción "Compilar" y posteriormente la opción "Enviar Respuesta".	

**Tabla 51: Flujo de Eventos Básicos CU Compilar y Ejecutar Solución**

- Flujo de Eventos Alternativo:

Al actor	El sistema
1)(b) Selecciona la opción "Ver Compilador"	2) Problema de conexión.

**Tabla 52: Flujo de Eventos Alternativo CU Compilar y Ejecutar Solución**

Post-Condiciones: El sistema evalúa solución enviada.

## 6.2 Modelamiento de datos

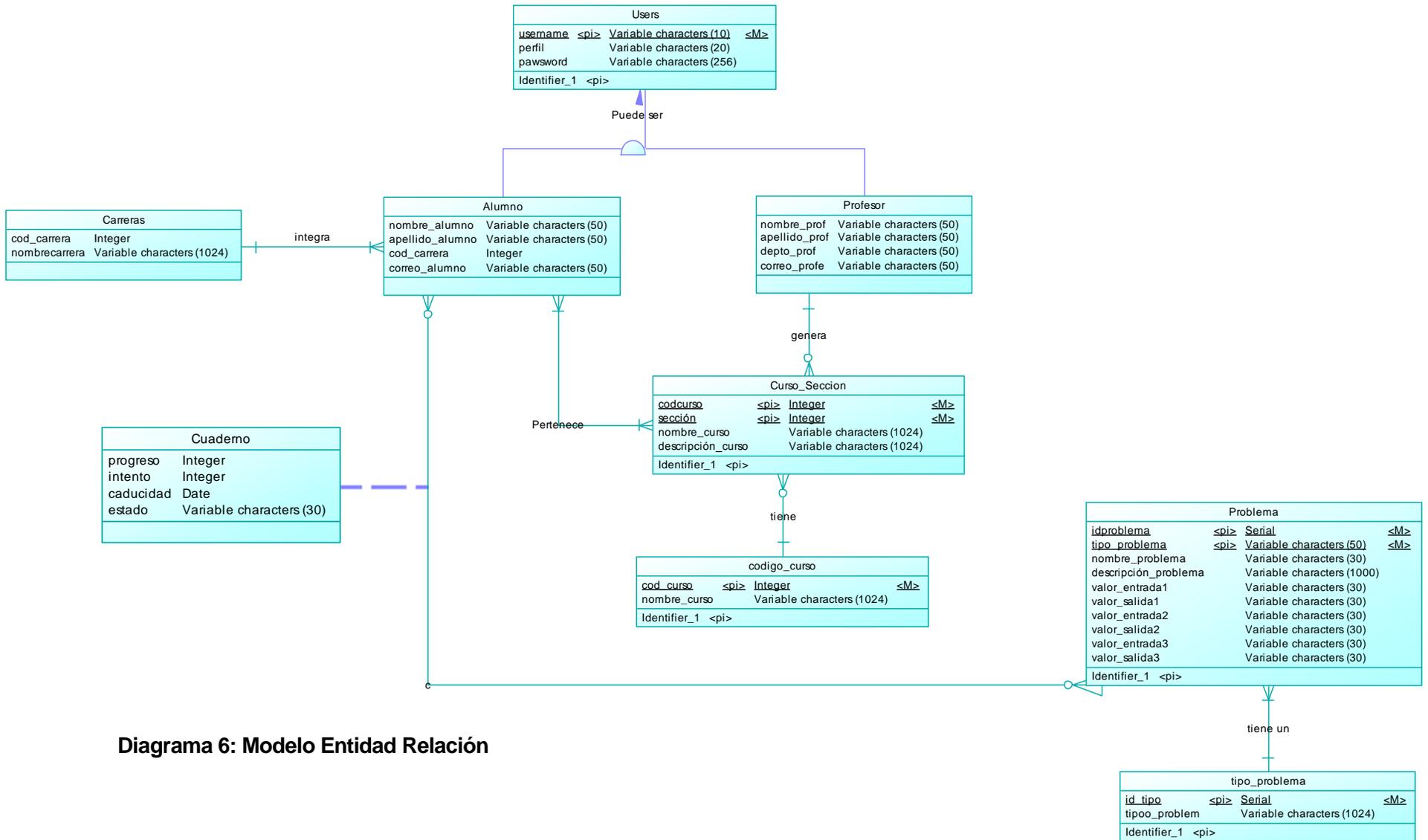
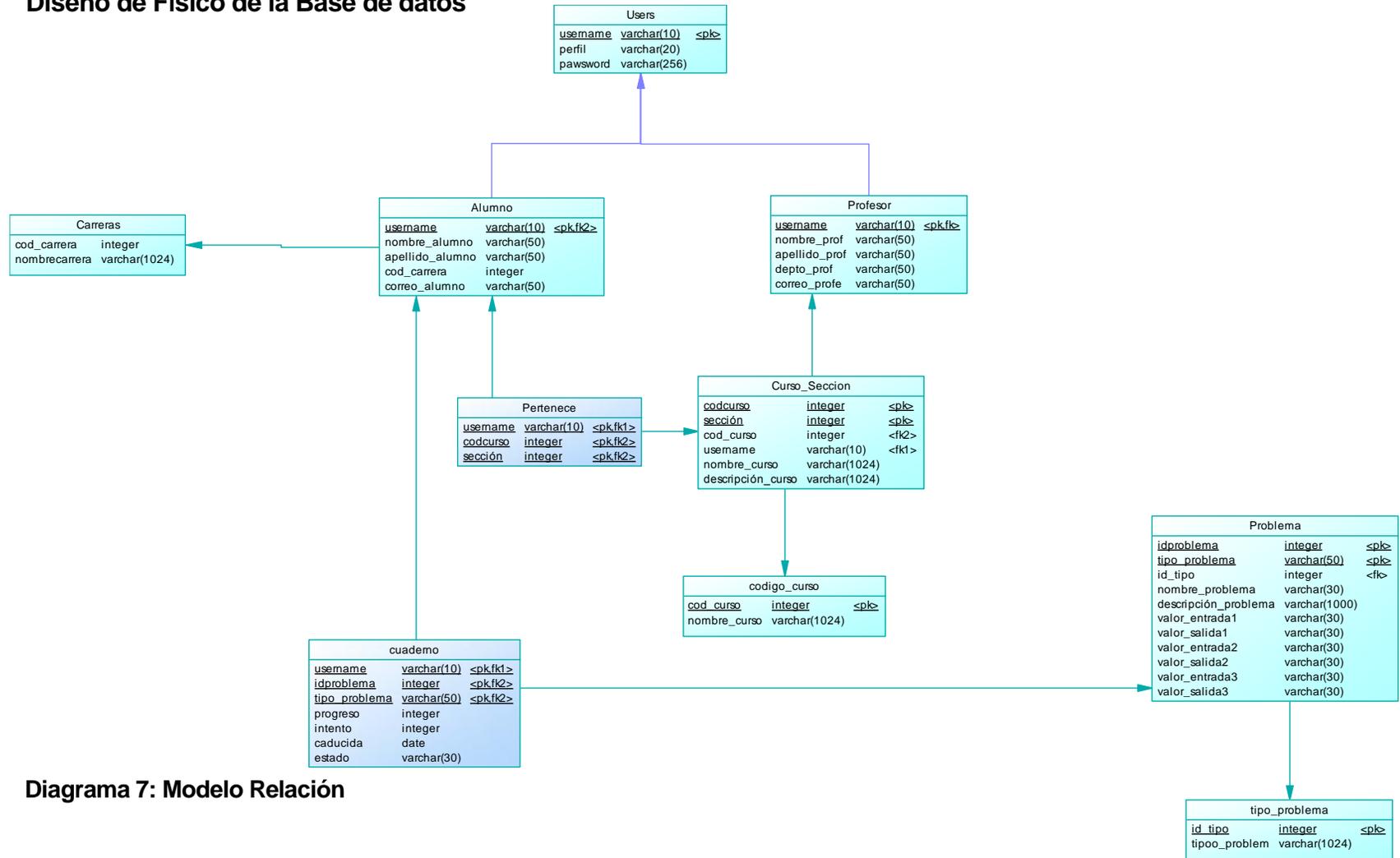


Diagrama 6: Modelo Entidad Relación

## 7 DISEÑO

### 7.1 Diseño de Físico de la Base de datos



El modelo refleja como se maneja la información del sistema SAAPE, y en la estructura de la base de datos se encuentran las siguientes tablas que logran que la base de datos sea consistente.

- **Users:**

Se encuentran los tipos de perfil de usuarios existentes en el sistema.

Sus atributos son:

- Username.

- Perfil.

- Password.

- **Alumno:**

Es el usuario que resuelve los ejercicios planteados en el sistema.

Sus atributos son:

- Username.

- Nombre\_alumno.

- Apellido\_alumno.

- Cod\_carrera.

- Correo\_alumno.

- **Profesor:**

Es el usuario que registra a los alumnos, genera cursos y los problemas a resolver en el sistema.

Sus atributos son:

- Username.

- Nombre\_prof.

- Apellido\_prof.

- Depto\_prof.

- Correo\_prof.

- **Problema:**

Ejercicios propuestos por los profesores para los alumnos.

Sus atributos son:

- Idproblema.

- Tipo problema.
- Nombre\_problema.
- Descripción\_problema.
- Valor\_entrada1.
- Valor\_salida1.
- Valor\_entrada2.
- Valor\_salida2.
- Valor\_entrada3.
- Valor\_salida3.

- **Curso\_Sección:**

Es donde están los datos del curso que se va impartir.

Sus atributos son:

- Codcurso.
- Sección.
- Nombre\_curso.
- Descripción\_curso.
- Username.

- **Cuaderno:**

Es el lugar físico donde se visualizan los ejercicios asignados a cada alumno.

Sus atributos son:

- Username.
- Idproblema.
- Tipo problema.
- Progreso.
- Intento.
- Caducidad.
- Estado.

- **Carreras:**  
Están los datos de las carreras informáticas que manipularan el sistema.  
Sus atributos son:
  - Codcarrera.
  - Nombrecarrera.
  
- **Codigo\_curso:**  
Son los datos de los cursos existentes en las carreras informáticas.  
Sus atributos son:
  - Codcurso.
  - Nombrecurso.
  
- **Pertenece:**  
Son los datos del curso al que pertenece cada alumno en el sistema.  
Sus atributos son:
  - Username.
  - Codcurso.
  - Sección.
  
- **Tipo\_problema:**  
Son los tópicos que se dictan en las distintas unidades de cada asignatura.  
Sus atributos son:
  - Id\_tipo.
  - Tipo\_problema.

## 7.2 Diseño de arquitectura funcional

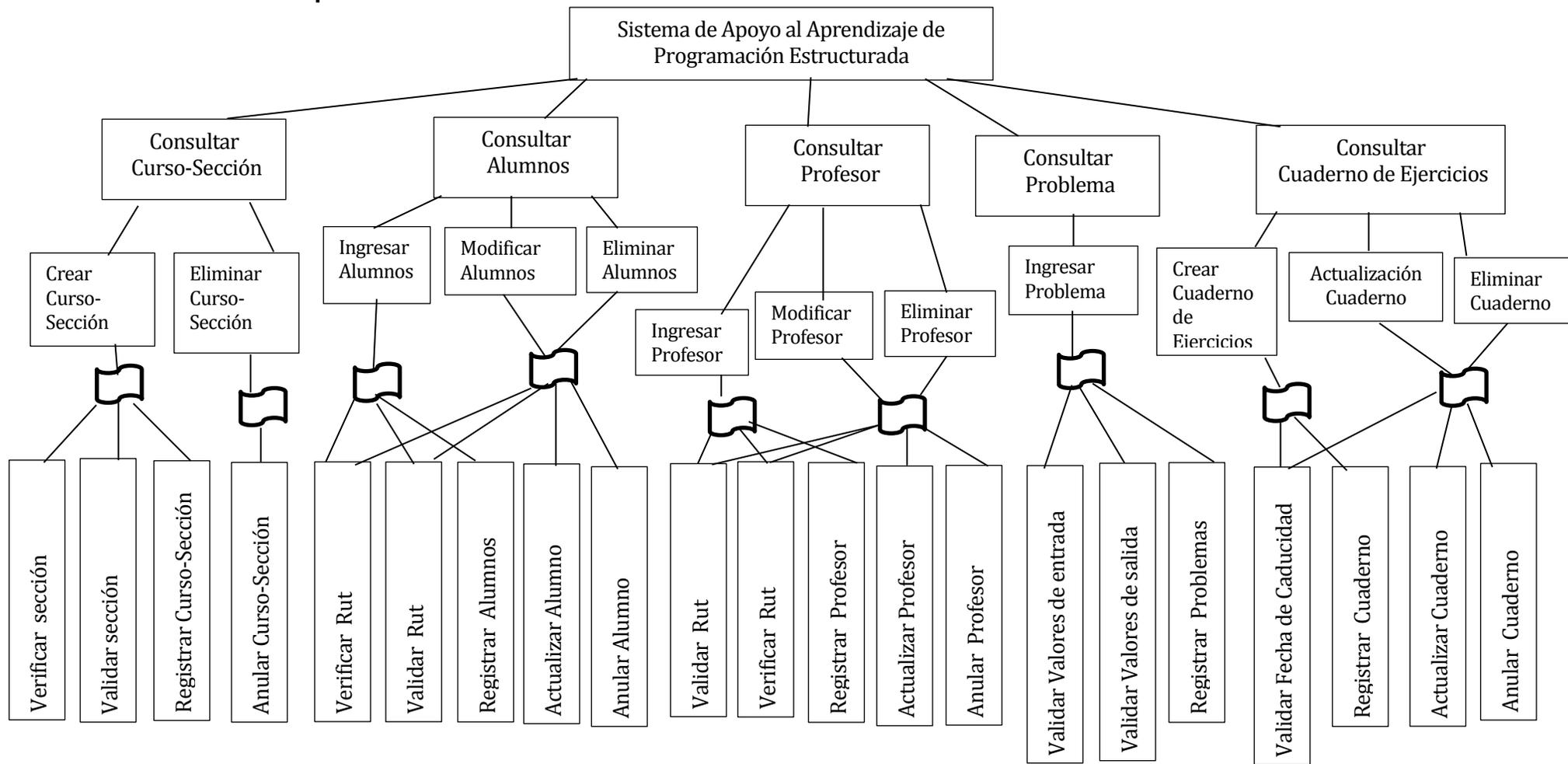
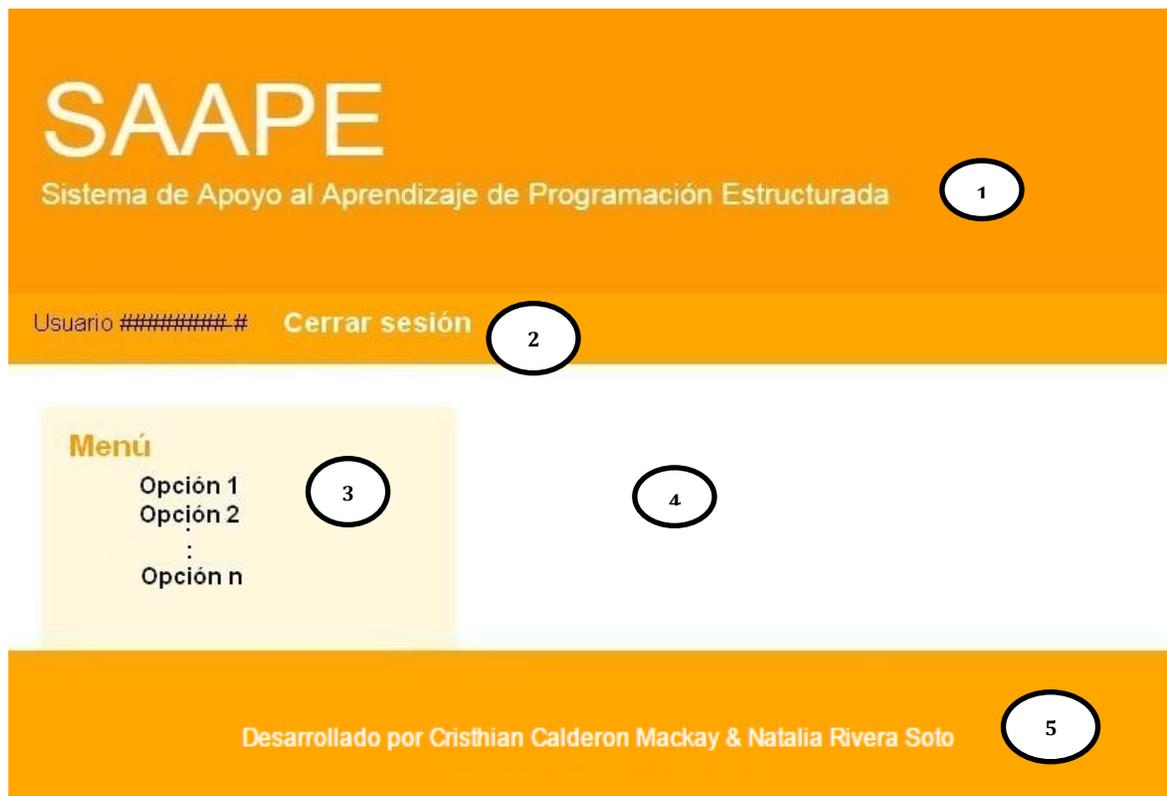


Diagrama 8: Árbol de Descomposición funcional

### 7.3 Diseño interfaz y navegación

El diseño de la interfaz de usuario considera un diseño estándar que será respetado en todas las pantallas.



#### Ilustración 1: Diseño de interfaz de usuario

La organización de la interfaz considera las siguientes áreas:

- (1)Cabecera1: Es utilizada en todas las páginas del sistema, con un fondo de color naranja y letras blancas. Aquí se encuentra el título del sistema “Sistema de Apoyo al Aprendizaje de Programación Estructurada SAAPE”.
- (2)Cabecera2: Es utilizada en todas las páginas del sistema, con fondo de color naranja y letras negras y blancas. Aquí se indica el usuario que está usando el sistema y cerrar sesión.
- (3)Menú: Se ubica al lado izquierdo de la pantalla, contiene las opciones disponibles para el usuario, cada una de estas opciones corresponde a un hipervínculo.
- (4)Espacio Libre: En esta área se visualiza el contenido seleccionado por el usuario.
- (5)Pie de Página: Muestra información del Framework utilizado para la elaboración de este sistema web.

Algunas imágenes del sistema junto a sus módulos y los distintos perfiles de usuarios:

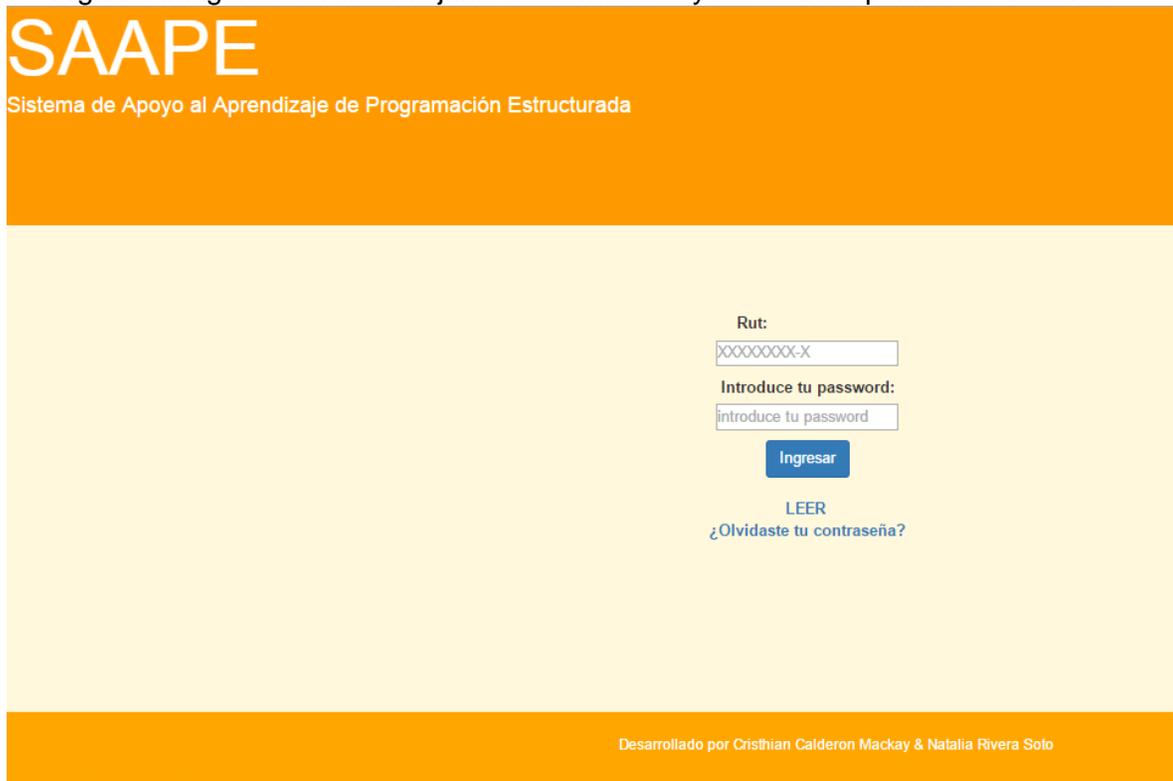


Ilustración 2: Página de Acceso



Ilustración 3: Perfil Administrador

**SAAPE**  
Sistema de Apoyo al Aprendizaje de Programacion Estructurada

Usuario 17896613-8    Cerrar sesion    Cambiar password

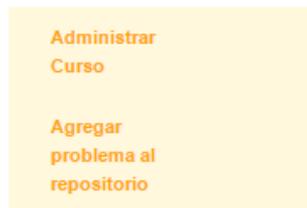
- Administrar docente
- Administrar tipo de problema
- Administrar Asignaturas
- Lista de profesores
- Crear docente

### Datos de los profesores

Numero	Rut	Nombre	Apellidos	Borrar	Actualizar
1	17539593-8	ines	Rivera Soto	<a href="#">Borrar</a>	<a href="#">Actualizar</a>
2	9728990-5	Brunny	Troncoso	<a href="#">Borrar</a>	<a href="#">Actualizar</a>
3	9191729-7	Susanne	Höhm	<a href="#">Borrar</a>	<a href="#">Actualizar</a>
4	11111111-1	Mario	sanchez	<a href="#">Borrar</a>	<a href="#">Actualizar</a>
5	55555555-5	pedro	perez	<a href="#">Borrar</a>	<a href="#">Actualizar</a>

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

**Ilustración 4: Vista Profesores (Administrador)**



**Ilustración 5: Perfil Profesor**



## lista de cursos

Practica profesional I (620171) Seccion 1 [Editar lista del curso](#)



**Ilustración 6: Vista Cursos (Profesor)**

## Bienvenido de nuevo alumno

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

**Ilustración 7: Perfil Alumno**

### Datos del cuaderno

Numero del problema	Nombre del problema	Tipo de problema	Descripcion del problema	Estado	Intentos
15	Imprime 4	secuencia simple	imprima la division de dos numeros enteros	Pendiente	0
16	Imprime 5	secuencia simple	imprime el resto de dos numeros enteros	Pendiente	0
17	division1	secuencia simple	divi	Pendiente	0
21	Suma de dos matrices	arrays bidimensionales (matrices)	Cree un programa que sume dos matrices de 2 x 2	Pendiente	0
12	Imprime 1	secuencia simple	imprima la multiplicacion de dos numeros enteros	Pendiente	0
13	Imprime 2	secuencia simple	imprima la suma de dos numeros enteros	Pendiente	0

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

**Ilustración 8: Vista Cuaderno (Alumno)**

- Jerarquía de Menú:

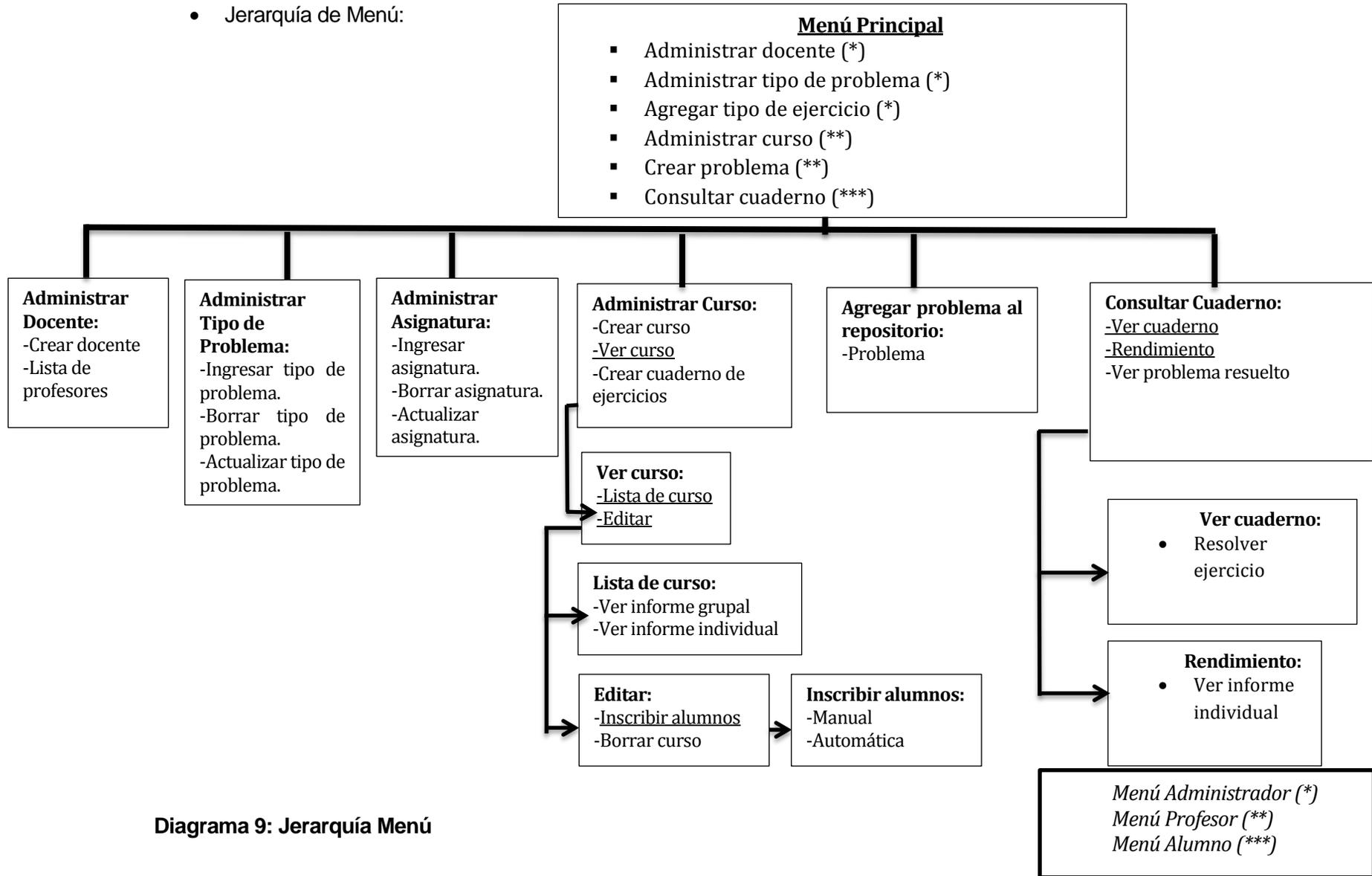
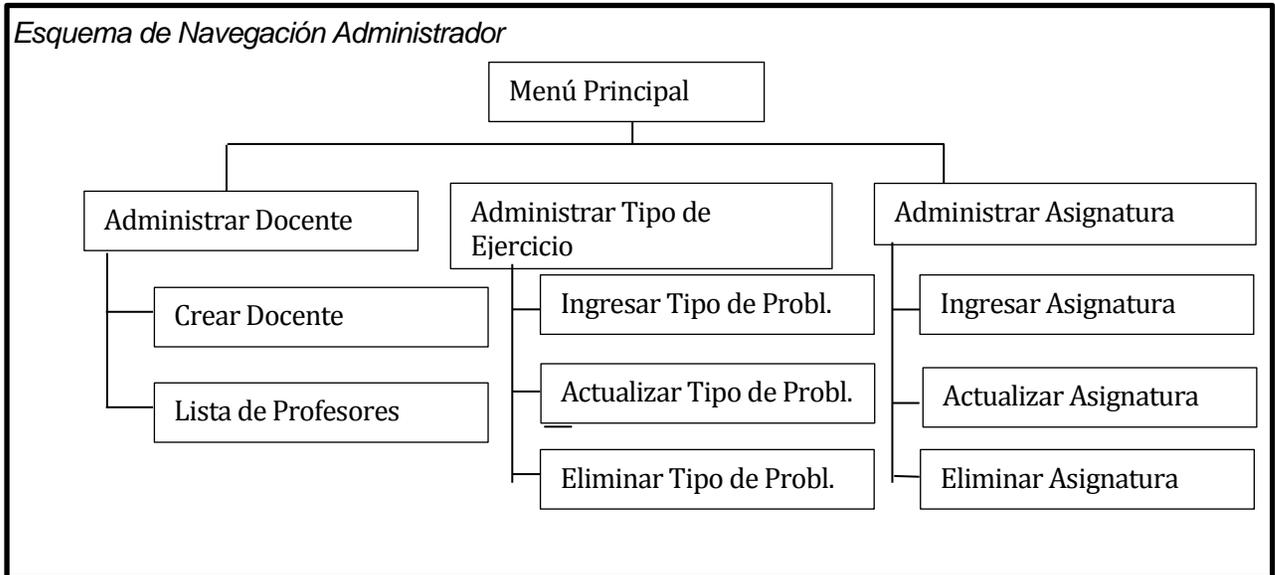
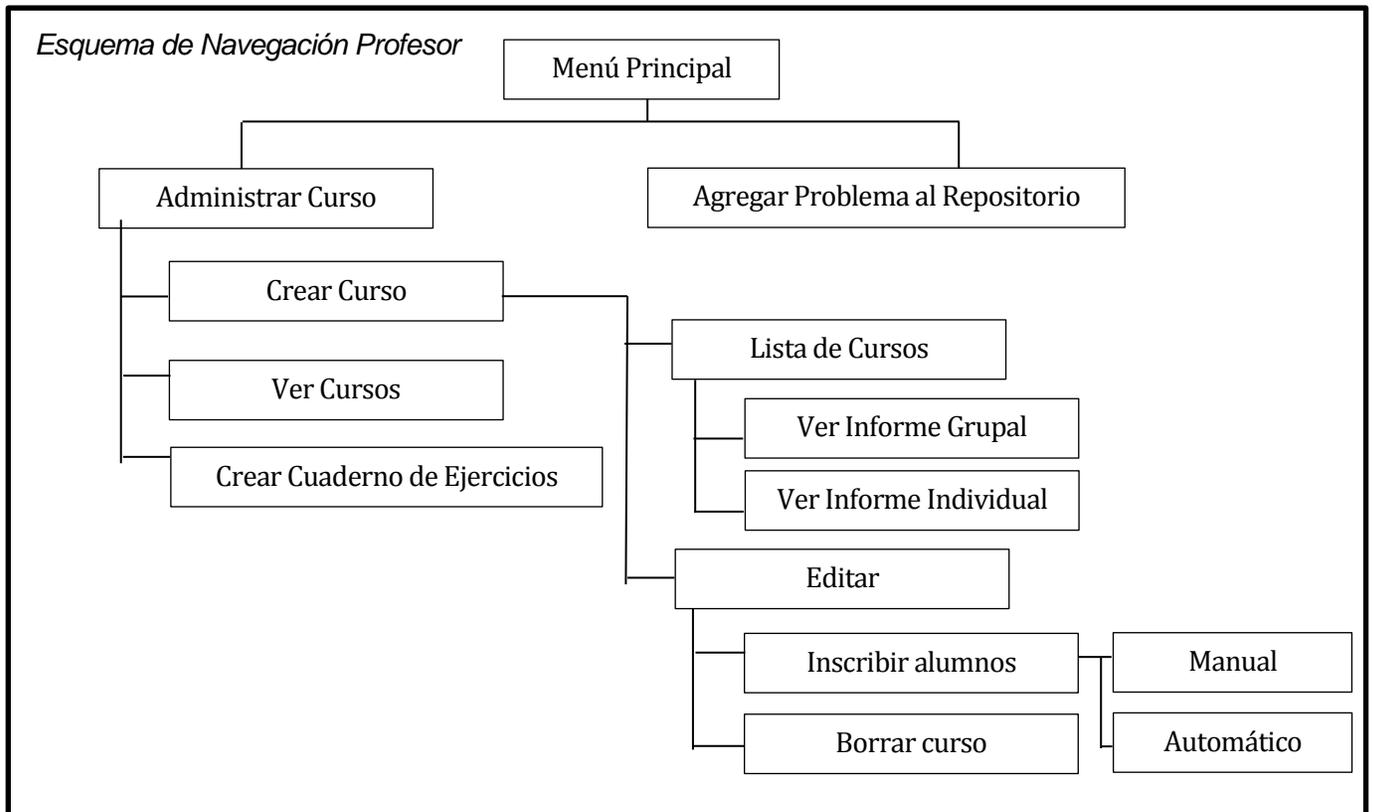


Diagrama 9: Jerarquía Menú

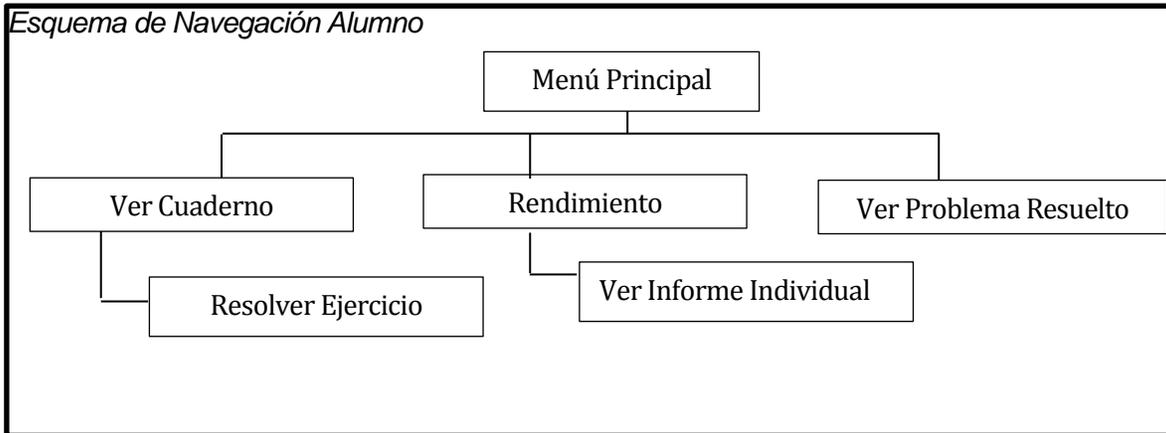
- Esquema de Navegación:



**Diagrama 10: Esquema de Navegación Administrador**



**Diagrama 11: Esquema de Navegación Profesor**



**Diagrama 12: Esquema de Navegación Alumno**

## 7.4 Especificación de módulos

Nombre Módulo: Validar Sección			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Sección	Integer	True/False	Boolean

Tabla 53: Módulo Validar Sección

Nombre Módulo: Verificar Sección			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Sección	Integer	True/False	Boolean

Tabla 54: Módulo Verificar Sección

Nombre Módulo: Registrar Curso-Sección			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Codcurso	Integer	True/False	Boolean
Sección	Integer		
Nombre_curso	Varchar		
Descripción_curso	Varchar		
Username	Varchar		

Tabla 55: Módulo Registrar Curso-Sección

Nombre Módulo: Anular Curso-Sección			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Codcurso	Integer	Indicador de anulación	Boolean
Sección	Integer		
Nombre_curso	Varchar		
Descripción_curso	Varchar		
Username	Varchar		

Tabla 56: Módulo Anular Curso-Sección

Nombre Módulo: Verificar Rut (Alumnos)			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Valida_rut	String	True/False	Boolean

**Tabla 57: Módulo Verificar Rut (Alumnos)**

Nombre Módulo: Validar Rut (Alumnos)			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Valida_rut	String	True/False	Boolean

**Tabla 58: Módulo Validar Rut (Alumnos)**

Nombre Módulo: Registrar Alumnos			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Username	Varchar	True/False	Boolean
Nombre_alumno	Varchar		
Apellido_alumno	Varchar		
Cod_carrera	Integer		
Correo_alumno	Varchar		

**Tabla 59: Módulo Registrar Alumnos**

Nombre Módulo: Actualizar Alumnos			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Username	Varchar	Username	Varchar
Nombre_alumno	Varchar	Nombre_alumno	Varchar
Apellido_alumno	Varchar	Apellido_alumno	Varchar
Cod_carrera	Integer	Cod_carrera	Integer
Correo_alumno	Varchar	Correo_alumno	Varchar

**Tabla 60: Módulo Actualizar Alumnos**

Nombre Módulo: Anular Alumnos			
-------------------------------	--	--	--

Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Username	Varchar	Indicador de anulación	Boolean
Nombre_alumno	Varchar		
Apellido_alumno	Varchar		
Cod_carrera	Integer		
Correo_alumno	Varchar		

**Tabla 61: Módulo Anular Alumnos**

Nombre Módulo: Verificar Rut (Profesor)			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Valida_rut	String	True/False	Boolean

**Tabla 62: Módulo Verificar Rut (Profesor)**

Nombre Módulo: Validar Rut (Profesor)			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Valida_rut	String	True/False	Boolean

**Tabla 63: Módulo Validar Rut (Profesor)**

Nombre Módulo: Registrar Profesor			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Username	Varchar	True/False	Boolean
Nombre_prof	Varchar		
Apellido_prof	Varchar		
Depto_prof	Varchar		
Correo_prof	Varchar		

**Tabla 64: Módulo Registrar Profesor**

Nombre Módulo: Actualizar Profesor	
Parámetros de entrada	Parámetros de Salida

Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Username	Varchar	Username	Varchar
Nombre_prof	Varchar	Nombre_prof	Varchar
Apellido_prof	Varchar	Apellido_prof	Varchar
Depto_prof	Varchar	Depto_prof	Varchar
Correo_prof	Varchar	Correo_prof	Varchar

**Tabla 65: Módulo Actualizar Profesor**

Nombre Módulo: Anular Profesor			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Username	Varchar	Indicador de anulación	Boolean
Nombre_prof	Varchar		
Apellido_prof	Varchar		
Depto_prof	Varchar		
Correo_prof	Varchar		

**Tabla 66: Módulo Anular Profesor**

Nombre Módulo: Validar Valores de Entrada (Problema)			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Valor_entrada1	Varchar	True/False	Boolean
Valor_entrada2	Varchar		
Valor_entrada3	Varchar		

**Tabla 67: Módulo Validar Valores de Entrada (Problema)**

Nombre Módulo: Validar Valores de Salida (Problema)			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:

Valor_salida1	Varchar	True/False	Boolean
Valor_salida2	Varchar		
Valor_salida3	Varchar		

**Tabla 68: Módulo Validar Valores de Salida (Problema)**

Nombre Módulo: Registrar Problemas			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Idproblema	Serial	True/False	Boolean
Tipo_problema	Varchar		
Nombre_problema	Varchar		
Descripción_problema	Varchar		
Valor_entrada1	Varchar		
Valor_entrada2	Varchar		
Valor_entrada3	Varchar		
Valor_salida1	Varchar		
Valor_salida2	Varchar		
Valor_salida3	Varchar		

**Tabla 69: Módulo Registrar Problemas**

Nombre Módulo: Validar Fecha de Caducidad			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Caducidad	Date	True/False	Boolean

**Tabla 70: Módulo Validar Fecha de Caducidad**

Nombre Módulo: Registrar Cuaderno			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Username	Varchar	True/False	Boolean

Idproblema	Integer		
Tipo_problema	Varchar		
Progreso	Integer		
Intento	Integer		
Caducidad	Date		
Estado	Varchar		

**Tabla 71: Módulo Registrar Cuaderno**

Nombre Módulo: Actualización Cuaderno			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Username	Varchar	Username	Varchar
Idproblema	Integer	Idproblema	Integer
Tipo_problema	Varchar	Tipo_problema	Varchar
Progreso	Integer	Progreso	Integer
Intento	Integer	Intento	Integer
Caducidad	Date	Caducidad	Date
Estado	Varchar	Estado	Varchar

**Tabla 72: Módulo Actualización Cuaderno**

Nombre Módulo: Eliminar Cuaderno			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Username	Varchar	Indicador de anulación	Boolean
Idproblema	Integer		

Tipo_problema	Varchar		
Progreso	Integer		
Intento	Integer		
Caducidad	Date		
Estado	Varchar		

**Tabla 73: Módulo Eliminar Cuaderno**

---

## 8 PRUEBAS

---

### 8.1 Elementos de prueba

Dentro de los principales módulos que se van a probar están la creación de usuarios como profesor y/o alumno, el ingreso de problemas al sistema, generación de cuadernos para los alumnos, el correcto funcionamiento del compilador con los ejercicios, además de probar la conexión con la base de datos con la extracción e ingresos de datos al sistema y viceversa.

Se utilizarán las pruebas de caja negra, las que se realizan sobre la interfaz del software siendo indiferente al comportamiento interno y de estructura del software. De esta manera se demostrará que los elementos de entrada sean apropiados, obteniendo así un resultado correcto, y además que las funciones del software son eficaces.

Se definieron algunos módulos internos para efectuar las pruebas:

- Profesor: se realizan pruebas de ingreso, modificación y eliminación del sistema.
- Alumno: se efectúan pruebas de ingreso (manual o automático) y/o eliminación del sistema.
- Cuaderno: se hacen pruebas de ingreso y/o modificar de una solución a un ejercicio, y además de consultar por algún problema resuelto en el sistema.
- Curso-Sección: se llevan a cabo pruebas de ingreso y eliminación del sistema.
- Problemas: se ejecutan pruebas de ingreso en el sistema.

## 8.2 Especificación de las pruebas

Características a probar	Nivel de prueba	Objetivo de la Prueba	Enfoque para la definición de casos de prueba	Técnicas para la definición de casos de prueba	Actividades de prueba	Criterios de cumplimiento
Funcionalidad	Sistema	Probar conexión en el módulo Profesor.	Caja Negra	Particiones	1.- Ingresar al módulo profesor. 2.- Acceder al módulo donde se encuentra Ingresar Profesor. 3.- Probar el ingreso de un profesor. 4.- Acceder a modificar datos de profesor ingresado.	Comprobar la correcta conexión.
Funcionalidad	Sistema	Probar conexión en el módulo Alumno.	Caja Negra	Particiones	1.- Ingresar al módulo alumno. 2.- Acceder al módulo donde se encuentra Ingresar alumno. 3.- Probar el ingreso de un alumno. 4.- Acceder a modificar datos de alumno ingresado.	Comprobar la correcta conexión.
Funcionalidad	Sistema	Probar	Caja Negra	Particiones	1.- Ingresar	Comprobar la

		conexión en el módulo Cuaderno.			al módulo Cuaderno. 2.- Acceder al módulo donde se encuentra Resolver Ejercicio. 3.- Probar el ingreso de una solución al cuaderno. 4.- Ingresar al módulo consultar Ejercicios resueltos.	correcta conexión.
Funcionalidad	Sistemas	Probar conexión en el módulo Curso- Sección.	Caja Negra	Particiones	1.- Ingresar al módulo curso- sección. 2.- Acceder al módulo donde se encuentra Ingresar curso- sección. 3.- Probar el ingreso de un curso- sección.	Comprobar la correcta conexión.
Funcionalidad	Sistema	Probar conexión en el módulo Problemas.	Caja Negra	Particiones	1.- Ingresar al módulo problema. 2.- Acceder al módulo donde se encuentra Ingresar Problema. 3.- Probar el ingreso de un problema.	Comprobar la correcta conexión.

## **Tabla 74: Especificación de las pruebas**

### **8.3 Responsables de las pruebas**

Las personas encargadas de efectuar las pruebas son los desarrolladores del software, además de algunas personas voluntarias adicionales, quienes probaran las distintas funcionalidades y las correctas conexiones correspondientes.

### **8.4 Calendario de pruebas**

Las pruebas realizadas se desarrollaron al término de los módulos para comprobar el funcionamiento correcto del sistema. Para ello se consideró integrar personas voluntarias adicionales para probar el software y así evaluar y corroborar las pruebas ya realizadas por los desarrolladores web.

### **8.5 Detalle de las pruebas**

Los detalles de las pruebas se encuentran en el anexo 17.

### **8.6 Conclusiones de Prueba**

En conclusión se ratificó que las conexiones de los distintos módulos del sistema interactúan de manera correcta con la base de datos y el compilador, al momento de ingresar o actualizar datos, así como la extracción e importación de datos. Aparte de generar satisfactoriamente los cursos-sección y los cuadernos con sus problemas correspondientes.

---

## 9 PLAN DE CAPACITACIÓN Y ENTRENAMIENTO

---

El sistema está diseñado de modo fácil y simple para el uso del usuario, de esta forma será más sencillo su entendimiento para su capacitación y entrenamiento que será de tipo práctico.

Los usuarios a capacitar serán el administrador y los profesores de las asignaturas que utilizarán el sistema y los responsables de su capacitación serán los desarrolladores web.

Las funcionalidades que serán abordadas son la manipulación del sistema por parte de los usuarios con exactitud algunas de ellas:

- Ingreso de usuarios Profesor y Alumno.
- Ingreso de problemas.
- Ingreso de valores de prueba.
- Obtención de informes de rendimiento.
- Creación de curso-sección, cuadernos de ejercicios y tipo de ejercicios.

---

## 10 PLAN DE IMPLANTACIÓN Y PUESTA EN MARCHA

---

El sistema tiene que ser instalado en el servidor que dispone la Facultad de Ciencias Empresariales. Además se debe conectar también el compilador del sistema con el servidor para lograr que el software funcione adecuadamente. De esta manera realizar una etapa de marcha blanca en la que se probará el software con datos de prueba, así los usuarios pueden verificar el correcto funcionamiento y en caso contrario informar de alguna falencia. Al corroborar que el sistema este en óptimas condiciones de actividad se podrá proseguir con el poblamiento de la base datos con antecedentes reales.

Las etapas son:

- **Instalación del sistema en servidor:**  
Instalar el sistema en el servidor de la facultad y realizar pruebas para ver el correcto funcionamiento.
- **Conectar el compilador al servidor:**  
Conectar el compilador local al servidor para lograr el correcto funcionamiento del sistema.

## 11 RESUMEN ESFUERZO REQUERIDO

Al desarrollar un software se debe estimar las etapas que éste implica durante su desarrollo. Para alcanzar este objetivo hay que dedicar varias horas de trabajo, que corresponden a cada una de las horas gastadas por cada integrante del equipo de trabajo. Para ello se ha elaborado una tabla resumen con las horas involucradas en el avance por cada etapa de desarrollo.

Actividades/Fases	N° Horas Cristhian	N° Horas Natalia	N° Horas Equipo
Planeación	35	35	70
Reunión con Profesor guía	2	2	4
Desarrollo etapa de análisis	74	74	148
Reunión con Profesor guía	2	2	4
Desarrollo etapa de diseño	72	72	144
Reunión con Profesor guía	2	2	4
Implementación 1º y 2º iteración	150	150	300
Reunión con Profesor guía	4	4	8
Implementación 3º y 4º	300	300	600
Reunión con profesor guía	2	2	4
Implementación 5º	200	200	400
Reunión con profesor guía	2	2	4
<b>Total Equipo</b>			<b>1690</b>

Tabla 75: Resumen Esfuerzo Requerido

---

## 12 CONCLUSIONES

---

Con el desarrollo de este proyecto se logró dar apoyo a las carreras del área informática que poseen asignaturas de programación, con una herramienta de trabajo adecuada a sus requerimientos que permite mejorar la comprensión y aprendizaje de los alumnos.

Para poder llegar a una solución apropiada en primera instancia se realizó un estudio de las opciones convenientes para el desarrollo de este sistema. De esta manera se optó por un software a medida que se adaptó a los requerimientos establecidos, consiguiendo así favorecer el aprendizaje de los estudiantes mediante ejercicios de programación propuestos en el sistema elaborado. Además de la utilización de un framework y un compilador local, que se adecuan al proyecto logrando que sea más óptimo a su fin.

Durante el diseño del software se escogieron herramientas que se encuentran dentro de las nuevas tecnologías y que se adaptan al funcionamiento del servidor de la Facultad de Ciencias Empresariales para obtener una apropiada compatibilidad al momento de la implementación. La selección de la metodología, lenguaje de programación y herramientas de trabajo se basaron pensando en el modo de desarrollo del sistema y las modificaciones pertinente que podrían surgir de acuerdo a los requerimientos del usuario en el momento y en el futuro.

Para finalizar se puede mencionar que la problemática escogida para construir el sistema, produjo dentro del equipo de trabajo un gran interés por todo lo que abordaba. Ya que a nivel académico permitió conocer las diversas herramientas existentes en el área de la informática como las distintas plataformas, framework, compiladores, entre otras. El funcionamiento y aplicaciones a la situación planteada para encontrar la mejor solución. Por otra parte a nivel personal se adquirió una gran experiencia en cuanto a herramientas de trabajo como a temas de la índole, además de buscar una correcta y eficaz solución a la problemática, todo esto será de gran ayuda como experiencia y conocimiento personal al momento de la inserción laboral.

---

## 13 BIBLIOGRAFÍA

---

1. **Sergio Luján-Mora, Fernando Llopis, Jesús Peral, Antonio Ferrández.** *AWAM: Sistema de Apoyo del Aprendizaje basado en Web.* Alicante, España: Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, 2004.
2. **Mateo Lezcano Brito, Giraldo Valdés Pardo.** *LOGSIM: Sistema de Apoyo al Aprendizaje de la programación lógica.* Santa Clara, Cuba: Facultad de Matemáticas, Física y Computación, Universidad Central de Las Villas.
3. **Kurt Nørmark.** *A Web Support System for Submission and Handling of Programming Assignment.* Aalborg, Dinamarca: Department of Computer Science, Aalborg University.
4. **Jorge López, Cecilia Hernández Rivas, Yussef Farran Leiva.** *Una plataforma de evaluación automática con una metodología efectiva para la enseñanza/aprendizaje en programación de computadores.* Concepción, Chile: Departamento de Ingeniería Informática y Ciencias de Computación, Facultad de Ingeniería, Universidad de Concepción.
5. **Claudio Ariel Clarenc “et al”.** *Analizamos 19 plataformas e-learning: Investigación colaborativa sobre LMS* [en línea]. Congreso Virtual Mundial de E-Learning, 2003. Disponible en < <http://cooperacionib.org/191191138-Analizamos-19-plataformas-de-eLearning-primera-investigacion-academica-colaborativa-mundial.pdf> >
6. Onlinecompiler. [En línea]. Disponible en < <http://www.onlinecompiler.net/> >
7. Ideone. [En línea]. Disponible en < <https://ideone.com/> >
8. Compileonline. [En línea]. Disponible en < <http://compileonline.com/> >
9. Compilr. [En línea]. Disponible en < <https://compilr.com/> >
10. Codetwist. [En línea]. Disponible en < <http://codetwist.com/> >
11. GCC. [En línea]. Disponible en < <https://gcc.gnu.org/> >
12. CodeIgniter. [En línea]. Disponible en < <http://www.codeigniter.com/> >

## 14 ANEXO: PLANIFICACION INICIAL DEL PROYECTO

	<b>Planeación</b>	<b>25 días</b>	<b>lun 01/09/14</b>	<b>vie 03/10/14</b>
	Definición de la empresa	5 días	lun 01/09/14	vie 05/09/14
	Definición de la em	1 día	lun 01/09/14	lun 01/09/14
	Descripción de la empresa	2 días	mar 02/09/14	mié 03/09/14
	Descripción del área de estudio	2 días	mié 03/09/14	jue 04/09/14
	Descripción de la problemática	2 días	jue 04/09/14	vie 05/09/14
	<b>Estado del arte</b>	<b>11 días</b>	<b>vie 05/09/14</b>	<b>vie 19/09/14</b>
	Definición del problema	3 días	vie 05/09/14	mar 09/09/14
	Búsqueda de soluciones	3 días	mié 10/09/14	vie 12/09/14
	Definición de la solución	5 días	lun 15/09/14	vie 19/09/14
	<b>Definición del proyecto</b>	<b>8 días</b>	<b>lun 22/09/14</b>	<b>mié 01/10/14</b>
	Definición del proy	2 días	lun 22/09/14	mar 23/09/14
	Objetivos del proyecto	2 días	mié 24/09/14	jue 25/09/14
	Ambiente de la ingeniería de Sw	2 días	vie 26/09/14	lun 29/09/14
	Siglas y abreviaciones	2 días	mar 30/09/14	mié 01/10/14
	Entrega a Profesor guía: Etapa Planeación	1 día	vie 03/10/14	vie 03/10/14
	<b>Análisis</b>	<b>31 días</b>	<b>lun 06/10/14</b>	<b>lun 17/11/14</b>
	<b>Especificación de requerimientos de SW</b>	<b>15 días</b>	<b>lun 06/10/14</b>	<b>vie 24/10/14</b>
	Alcances	4 días	lun 06/10/14	jue 09/10/14
	Objetivos del SW	3 días	lun 13/10/14	mié 15/10/14
	Descripción del Producto	3 días	jue 16/10/14	lun 20/10/14
	Requerimientos específicos	4 días	mar 21/10/14	vie 24/10/14
	<b>Factibilidad</b>	<b>6 días</b>	<b>lun 27/10/14</b>	<b>lun 03/11/14</b>
	Factibilidad técnica	2 días	lun 27/10/14	mar 28/10/14
	Factibilidad operativa	2 días	mié 29/10/14	jue 30/10/14
	Factibilidad económica	2 días	vie 31/10/14	lun 03/11/14
	Diagrama de flujo de datos	3 días	mar 04/11/14	jue 06/11/14
	Casos de uso	3 días	vie 07/11/14	mar 11/11/14
	Modelamiento de dat	3 días	mié 12/11/14	vie 14/11/14

	<b>Entrega a Profesor guía: Etapa Análisis</b>	<b>1 día</b>	<b>lun 17/11/14</b>	<b>lun 17/11/14</b>
	<b>▢ Diseño</b>	<b>29 días</b>	<b>mié 19/11/14</b>	<b>lun 29/12/14</b>
	Diseño de físico de la base de datos	8 días	mié 19/11/14	vie 28/11/14
	Diseño de arquitectura	5 días	lun 01/12/14	vie 05/12/14
	Diseño interfaz y navegación	5 días	lun 08/12/14	vie 12/12/14
	Especificación de módulos	5 días	lun 15/12/14	vie 19/12/14
	Diseñar pruebas del sistema	5 días	lun 22/12/14	vie 26/12/14
	Entrega a Profesor guía: Etapa Diseño	1 día	lun 29/12/14	lun 29/12/14
	<b>▢ Implementación</b>	<b>192 días</b>	<b>lun 05/01/15</b>	<b>mar 29/09/15</b>
	<b>▢ 1º Iteración</b>	<b>61 días</b>	<b>lun 05/01/15</b>	<b>lun 30/03/15</b>
	Desarrollo versión 1	41 días	lun 05/01/15	lun 02/03/15
	Entrega versión 1	1 día	mar 03/03/15	mar 03/03/15
	Retroalimentación del cliente versión	8 días	mié 04/03/15	vie 13/03/15
	Agregar la retroalimentacion del cliente versión	7 días	lun 16/03/15	mar 24/03/15
	Desarrollar pruebas versión 1	3 días	jue 26/03/15	lun 30/03/15
	Mostrar avance a Profesor guía: version 1	1 día	mié 01/04/15	mié 01/04/15
	<b>▢ 2º Iteración</b>	<b>40 días</b>	<b>lun 06/04/15</b>	<b>vie 29/05/15</b>
	Desarrollo versión 2	21 días	lun 06/04/15	lun 04/05/15
	Entrega versión 2	1 día	vie 08/05/15	vie 08/05/15
	Retroalimentación del cliente versión	5 días	lun 11/05/15	vie 15/05/15
	Agregar la retroalimentacion del cliente versión	5 días	lun 18/05/15	vie 22/05/15
	Desarrollar pruebas versión 2	3 días	mié 27/05/15	vie 29/05/15
	Mostrar avance a Profesor guía: version 2	1 día	lun 01/06/15	lun 01/06/15
	<b>▢ 3º Iteración</b>	<b>45 días</b>	<b>mar 02/06/15</b>	<b>lun 03/08/15</b>
	Desarrollo versión 3	27 días	mar 02/06/15	mié 08/07/15
	Entrega versión 3	1 día	vie 10/07/15	vie 10/07/15
	Retroalimentación del cliente versión 3	7 días	lun 13/07/15	mar 21/07/15

	Agregar la retroalimentacion del cliente versión 3	6 días	mié 22/07/15	mié 29/07/15
	Desarrollar pruebas versión 3	3 días	jue 30/07/15	lun 03/08/15
	Mostrar avance a Profesor guía: version 3	1 día	mar 04/08/15	mar 04/08/15
	<b>4º Iteración</b>	<b>38 días</b>	<b>mié 05/08/15</b>	<b>vie 25/09/15</b>
	Desarrollo versión 4	20 días	mié 05/08/15	mar 01/09/15
	entrega versión 4	1 día	vie 04/09/15	vie 04/09/15
	retroalimentación del cliente versión	7 días	lun 07/09/15	mar 15/09/15
	agregar la retroalimentacion del cliente versión	6 días	mié 16/09/15	mié 23/09/15
	desarrollar pruebas versión 4	2 días	jue 24/09/15	vie 25/09/15
	Mostrar avance a Profesor guía: version 4	2 días	lun 28/09/15	mar 29/09/15
	Entrega Proyecto Sistema de Apoyo al Aprendizaje de Programación	1 día	jue 02/10/14	jue 02/10/14

	<b>5º Iteración</b>	<b>38 días</b>	<b>lun 05/10/15</b>	<b>mié 25/11/15</b>
	Desarrollo versión 5	20 días	lun 05/10/15	vie 30/10/15
	entrega versión 5	1 día	lun 02/11/15	lun 02/11/15
	retroalimentación del cliente versión 5	5 días	mar 03/11/15	lun 09/11/15
	agregar la retroalimentacion del cliente versión 5	6 días	jue 12/11/15	jue 19/11/15
	desarrollar pruebas versión 5	4 días	vie 20/11/15	mié 25/11/15
	Mostrar avance a Profesor guía: version 5	2 días	jue 26/11/15	vie 27/11/15
	Entrega Proyecto Sistema de Apoyo al Aprendizaje de Programación Estructurada	1 día	mié 09/12/15	mié 09/12/15

**Ilustración 9: Carta Gantt**

### 14.1.1 Estimación inicial de tamaño

Estimación de Tamaño del software aplicando técnicas basadas en el método Puntos de Casos de Uso. Este utiliza los actores y casos de uso reconocidos para medir el esfuerzo que equivale el desarrollarlos. A continuación las etapas que lo componen:

- Puntos de caso de uso sin ajustar (UUCP)
  - Factor de peso de los actores sin ajustar (UAW)
 

Es la evaluación de la complejidad de los actores con los que tendrá que interactuar el sistema. Se calcula decretando si cada actor es una persona o sistema.

Actor	Tipo Actor	Peso
Administrador	Complejo	3
Profesor	Complejo	3
Alumno	Complejo	3
GCC	Simple	1
<b>Factor de peso de los actores sin ajustar (UAW)</b>		10

**Tabla 76: Factor de Peso de los Actores sin Ajustar**

- Factor de peso de los casos de uso sin ajustar(UUCW)
 

Se determina el nivel de complejidad mediante dos métodos, uno basado en transacciones y otro basado en clases de análisis.

<b>Caso de uso</b>	<b>Transacciones</b>	<b>Peso</b>
<b>Gestionar Curso-Sección</b>	3	5
<b>Gestionar Alumnos</b>	3	5
<b>Ingresar Problema</b>	3	5
<b>Gestionar Cuaderno de Ejercicios</b>	3	5
<b>Consultar Rendimiento Grupal/Individual</b>	2	5
<b>Emitir Informe de Rendimiento Grupal/Individual</b>	2	5
<b>Gestionar Profesor</b>	3	5
<b>Agregar Tipo de Ejercicio</b>	3	5
<b>Actualizar Problema</b>	2	5
<b>Ver Cuaderno</b>	2	5
<b>Resolver Ejercicios</b>	3	5
<b>Evaluar Solución</b>	2	5
<b>Validar Usuario</b>	3	5
<b>Compilar y ejecutar solución</b>	3	5
<b>Factor de peso de los casos de uso sin ajustar(UUCW)</b>		70

**Tabla 77: Factor de Peso de los Casos de Uso Sin Ajustar**

La fórmula para calcular los puntos de caso de uso sin ajustar es la siguiente:

$$\text{UUCP (Puntos de casos de Uso sin Ajustar)} = \text{UAW} + \text{UUCW}$$

$$\text{UUCP} = 10 + 70 = 80$$

- Puntos de caso de uso ajustados (UCP)

- Factores de complejidad técnica

Se integran 13 factores que calculan la complejidad de los módulos del sistema que se desarrolla, cada uno posee un peso definido con los cuales se lograrán puntos ponderados por cada uno de ellos, según la valoración que se le asigne. Para una mejor comprensión, a continuación se mostrará una tabla con los ítems:

Factor	Descripción	Peso	Valor	Total
F1	Sistema distribuido.	2	1	2
F2	Objetivos de performance o tiempo de respuesta.	1	5	5
F3	Eficiencia del usuario final.	1	3	3
F4	Procesamiento interno complejo.	1	3	3
F5	El código debe ser usuario final.	1	2	2
F6	Facilidad de instalación	0.5	3	1.5
F7	Facilidad de	0.5	5	2.5

	cambio.			
<b>F8</b>	Portabilidad.	2	3	6
<b>F9</b>	Facilidad de cambio.	1	3	3
<b>F10</b>	Concurrencia.	1	5	5
<b>F11</b>	Incluye objetivos especiales de seguridad.	1	3	3
<b>F12</b>	Provee acceso directo a terceras partes.	1	1	1
<b>F13</b>	Se requiere facilidades especiales de entrenamiento a usuario.	1	3	3
<b>TFactor = Sum</b>				40
<b>(Valor*Peso)</b>				

**Tabla 78: Factores de Complejidad Técnica**

La fórmula para obtener los factores técnicos se detalla a continuación:

$$\text{Factores técnicos (TCF)} = 0.6 + (0.01 * \text{TFactor})$$

$$\text{TCF} = 0.6 + (0.01 * 40) = 1$$

- Factores ambientales

Se evalúan 8 factores, que están vinculados con las habilidades y experiencia del grupo de personas involucradas con el desarrollo del proyecto. Estos factores son:

<b>Factor</b>	<b>Descripción</b>	<b>Peso</b>	<b>Valor</b>	<b>Total</b>
<b>E1</b>	Familiaridad con el modelo de proyecto utilizado.	1.5	2	3
<b>E2</b>	Experiencia en la aplicación.	0.5	2	1
<b>E3</b>	Experiencia en orientación a objetos.	1	4	4
<b>E4</b>	Capacidad de analista líder.	0.5	4	2
<b>E5</b>	Motivación.	1	5	5
<b>E6</b>	Estabilidad de los requerimientos.	2	4	8
<b>E7</b>	Personal part-time.	-1	1	-1
<b>E8</b>	Dificultad del lenguaje de programación.	-1	3	-3
<b>EFactor = Sum</b>				<b>19</b>
<b>(Valor*Peso)</b>				

**Tabla 79: Factores Ambientales**

Para calcular los factores ambientales se utiliza la siguiente formula:

$$\text{Factores Ambientales (EF)} = 1.4 + (-0.03 * \text{EFactor})$$

$$\text{EF} = 1.4 + (-0.03 * 19) = 0.83$$

- Esfuerzo Horas-Hombre (E)

Se calcula una aproximación del esfuerzo, pensando en el desarrollo según las funcionalidades de los casos de uso.

Factor	Filtro	Total
De F1 a F6	Factor < 3	2
De F7 a F8	Factor > 3	0
<b>CF</b>		20

**Tabla 80: Esfuerzo Horas-Hombre**

Puntos de caso de uso Ajustados (UCP)=UUCP \* TCF \*EF

$$UCP = 80 * 1 * 0.83 = 66.4$$

El esfuerzo en horas-persona viene dado por:

$$E = UCP * CF$$

$$E = 66.4 * 20 = 1328 \text{ Puntos de Caso de Uso.}$$

## 15 ANEXO: RESULTADOS DE ITERACIONES EN EL DESARROLLO

### 15.1 1º Iteración:

Se comienza instalando y configurando un servidor local, en este caso se trabaja con WampServer.

The screenshot shows the WampServer control panel interface. At the top left is the WampServer logo. Below it, the text 'WampServer' is displayed. On the right side, the version 'Version 2.5' and a link for 'Version Française' are visible. The main section is titled 'Server Configuration' and includes the following details:

- Apache Version:** 2.4.9 - [Documentation](#)
- PHP Version:** 5.5.12 - [Documentation](#)
- Server Software:** Apache/2.4.9 (Win64) PHP/5.5.12
- Loaded Extensions:** A grid of 40 extensions including apache2handler, Core, ereg, gd, imap, mhash, openssl, pdo\_sqlite, shmop, sqlite3, xml, zip, bcmath, ctype, exif, gettext, json, mysql, pcre, pgsql, SimpleXML, standard, xmlreader, zlib, bz2, curl, fileinfo, gmp, libxml, mysqli, PDO, Phar, soap, tokenizer, xmlrpc, calendar, date, filter, hash, mbstring, mysqlnd, pdo\_mysql, Reflection, sockets, wddx, xmlwriter, com\_dotnet, dom, ftp, iconv, mcrypt, odbc, pdo\_pgsql, session, SPL, xdebug, and xsl.
- MySQL Version:** 5.6.17 - [Documentation](#)

Below the configuration section, there are three columns:

- Tools:** phpinfo(), phpmyadmin
- Your Projects:** chamilo, CodeIgniter-3.0.3, codeigniter\_excel\_library-master
- Your Aliases:** phpmyadmin, phpsysinfo, sqlbuddy

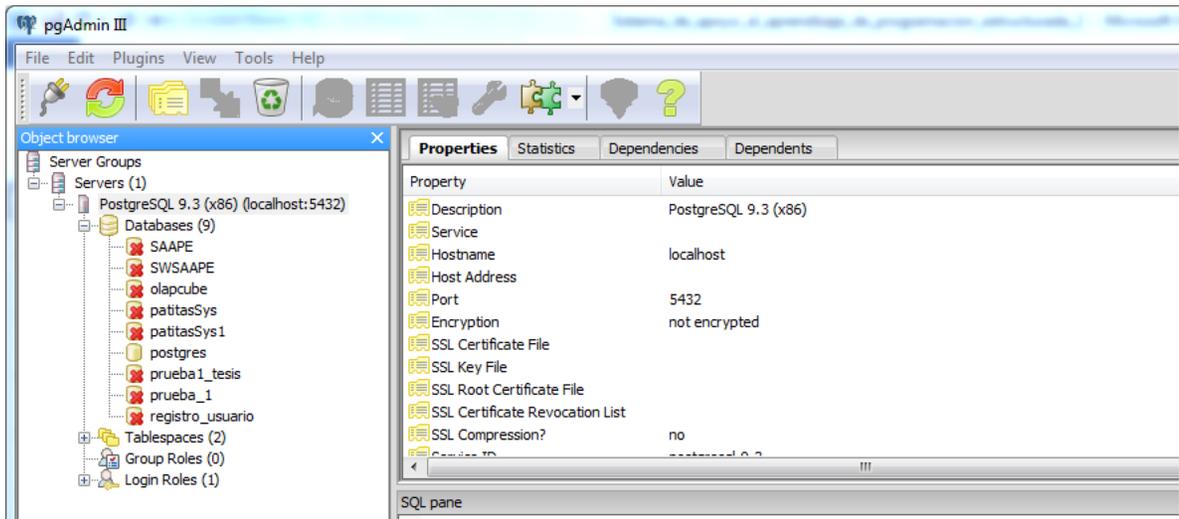
Ilustración 10: WampServer

Al desarrollar esta primera iteración se descarga el Framework CodeIgniter, para realizar posteriormente su instalación de manera local. La que se visualiza de la siguiente manera:

The screenshot shows the CodeIgniter welcome page. It starts with the text 'Welcome to CodeIgniter!'. Below this, it states: 'The page you are looking at is being generated dynamically by CodeIgniter.' It then provides the file path: 'If you would like to edit this page you'll find it located at: application/views/welcome\_message.php'. The corresponding controller is also shown: 'The corresponding controller for this page is found at: application/controllers/Welcome.php'. At the bottom, there is a link to the 'User Guide' and a footer indicating 'Page rendered in 0.1390 seconds. CodeIgniter Version 3.0.3'.

Ilustración 11: Inicio CodeIgniter

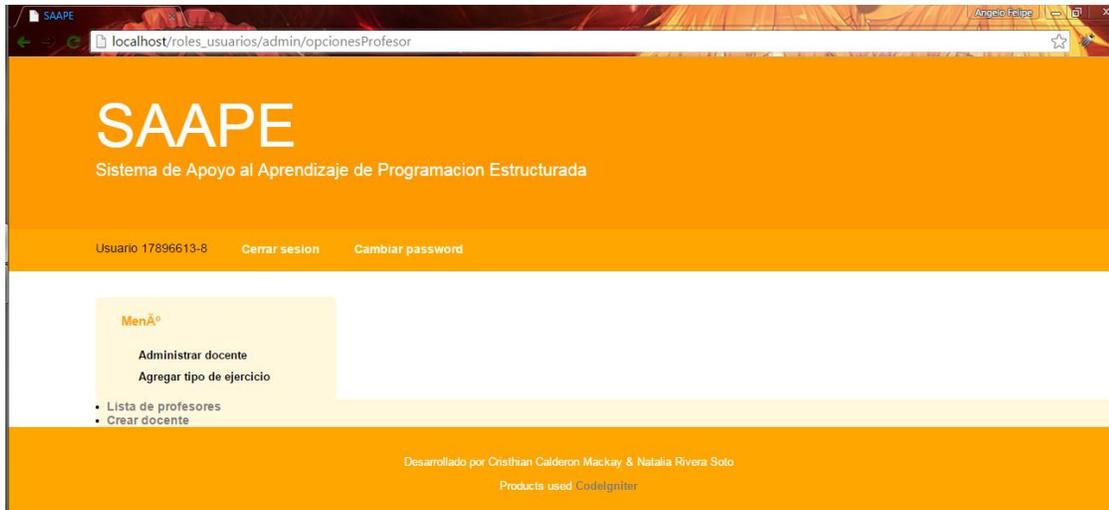
Además se configura la base de datos donde en este caso utilizamos PostgreSQL, junto a la creación de tablas y poblamientos de las mismas.



**Ilustración 12: Configuración BD**

## 15.2 2º Iteración:

Posteriormente se comienza a trabajar en el sistema web utilizando el lenguaje de programación PHP, para implementar las funcionalidades del sistema desde crear un usuario bajo el perfil de “Administrador, Profesor o Alumno” hasta los distintos módulos como de “Crear Problemas”, “Crear Cuaderno de ejercicios”, etc.

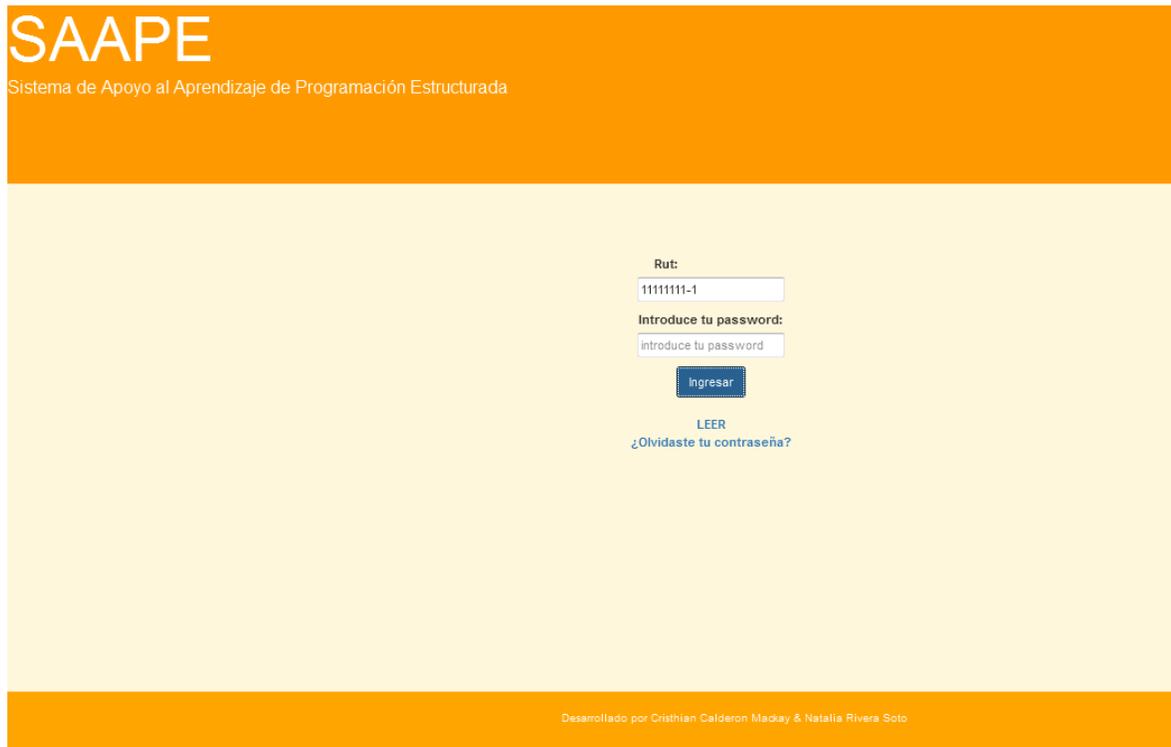


**Ilustración 13: Menú Administrador Localhost**

## 16 ANEXO: MANUAL DE USUARIO

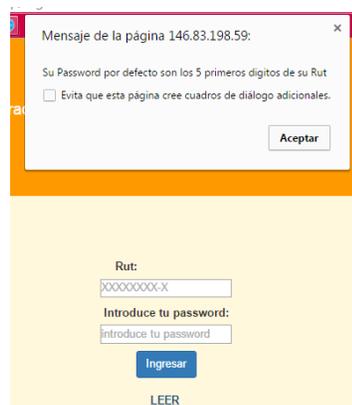
Para un correcto manejo del sistema SAAPE es fundamental contar con la documentación necesaria para comprender el funcionamiento de cada parte de este.

Al iniciar el sistema se encuentra la pantalla de acceso, donde debe ingresar como nombre de usuario su Rut con guión y dígito verificador y como contraseña los 5 primeros dígitos de éste.



**Ilustración 14: Pantalla de Acceso**

Al pasar el cursor sobre la opción “LEER”, se visualizará la información sobre la contraseña por defecto.



**Ilustración 15: Mensaje LEER**

En el caso de Olvidar contraseña existe la opción para recuperarla como se muestra a continuación:



Rut:  
XXXXXXXX-X

Introduce tu password:  
introduce tu password

Ingresar

LEER  
¿Olvidaste tu contraseña?

**Ilustración 16: Mensaje Olvidaste tu Contraseña**

El sistema cuenta con distintos tipos de usuarios los que tienen diferentes privilegios y opciones de visualización, por lo que el manual se dividirá de acuerdo a ellos. Los tipos de usuarios que existen en el sistema son: Administrador, Profesor y Alumno.

### 16.1 Administrador:

Al iniciar sesión como administrador se le presenta el siguiente menú:



SAAPE  
Sistema de Apoyo al Aprendizaje de Programacion Estructurada

Usuario 17896613-8    Cerrar sesion    Cambiar password

- Administrar docente
- Administrar tipo de problema
- Administrar Asignaturas
- Actualizar problema

Bienvenido de nuevo administrador

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

**Ilustración 17: Menú Administrador Final**

Dentro de las opciones, se encuentra:

- **Administrar Docente:**  
*Permite manejar la información de los profesores, tanto como su creación, modificación, eliminación y visualización.*
  - **Lista de Profesores:**  
 Se muestran los profesores que actualmente se encuentran registrados en el sistema. Esta opción además permite la actualización y eliminación de cualquiera de estos.

**SAAPE**  
 Sistema de Apoyo al Aprendizaje de Programacion Estructurada

Usuario 17896613-8    Cerrar sesion    Cambiar password

- Administrar docente
- Administrar tipo de problema
- Administrar Asignaturas
- Actualizar problema

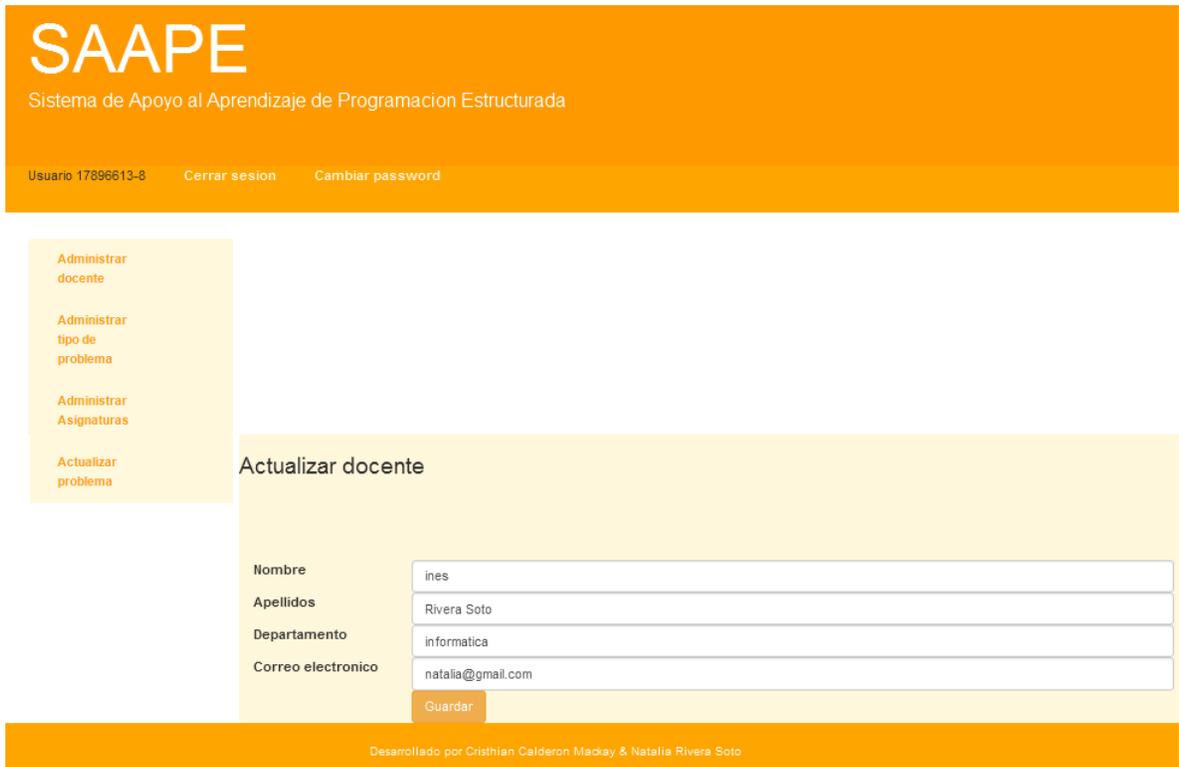
### Datos de los profesores

Numero	Rut	Nombre	Apellidos	Borrar	Actualizar
1	17539593-8	Ines	Rivera Soto	Borrar	Actualizar
2	9728990-5	Brunny	Troncoso	Borrar	Actualizar
3	9191729-7	Susanne	Höhm	Borrar	Actualizar
4	11111111-1	Mario	sanchez	Borrar	Actualizar
5	55555555-5	pedro	perez	Borrar	Actualizar

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

**Ilustración 18: Lista de Profesores**

Al presionar “Actualizar” puede modificar cualquiera de los campos disponibles.



### Ilustración 19: Actualizar Docente

Al seleccionar “Borrar” se solicita una confirmación de la petición.



## Ilustración 20: Eliminar Docente

- Crear Docente:**  
 Solo el Administrador tiene la facultad de ingresar profesores al sistema con sus respectivos datos personales, de esta manera se generan como usuarios.

The screenshot displays the SAAPE (Sistema de Apoyo al Aprendizaje de Programación Estructurada) interface. At the top, the header includes the SAAPE logo and the text 'Sistema de Apoyo al Aprendizaje de Programación Estructurada'. Below the header, there are navigation links for 'Usuario 17896613-8', 'Cerrar sesión', and 'Cambiar password'. A sidebar on the left contains menu items: 'Administrar docente', 'Administrar tipo de problema', 'Administrar Asignaturas', and 'Actualizar problema'. The main content area is titled 'Crear Docente' and contains a form with the following fields: 'Rut' (with a placeholder 'Rut con formato xxxxxxxx-x'), 'Nombre' (placeholder 'Ingrese nombre'), 'Apellidos' (placeholder 'Ingrese Apellidos'), 'Departamento' (placeholder 'Ingrese el departamento'), and 'Correo electronico' (placeholder 'Ingrese el email'). A 'Guardar' button is located at the bottom of the form. At the very bottom of the page, a footer note reads 'Desarrollado por Christian Calderon Maday & Natalia Rivera Soto'.

## Ilustración 21: Crear Documento

- Administrar Tipo de Problema:**  
 Permite manejar la información de los tipos de problemas existentes, tanto como su creación, modificación, eliminación y visualización.

**SAAPE**  
Sistema de Apoyo al Aprendizaje de Programacion Estructurada

Usuario 17896613-8    Cerrar sesion    Cambiar password

- Administrar docente
- Administrar tipo de problema
- Administrar Asignaturas
- Actualizar problema

### Crear tipo de ejercicio

Numero	Tipo de problema	Borrar	Actualizar
1	secuencia simple	Borrar	Actualizar
2	arrays unidimensionales (vectores)	Borrar	Actualizar
3	cadena (strings)	Borrar	Actualizar
4	arrays bidimensionales (matrices)	Borrar	Actualizar
5	funciones No recursivas	Borrar	Actualizar
6	funciones recursivas	Borrar	Actualizar
7	listas lineales	Borrar	Actualizar
8	condicion compuesta (if-then-else)	Borrar	Actualizar
9	repeticion simple	Borrar	Actualizar
10	condicion simple (if-then)	Borrar	Actualizar
11	Arbol Binario de Busqueda	Borrar	Actualizar
12	filas	Borrar	Actualizar
13	pilas	Borrar	Actualizar
14	punteros	Borrar	Actualizar

Tipo de problema

Guardar

**Ilustración 22: Administrar Tipo de Problema**

- Administrar Asignatura:  
Permite manejar la información de las distintas asignaturas, tanto como su creación, modificación, eliminación y visualización.

# SAAPE

Sistema de Apoyo al Aprendizaje de Programacion Estructurada

Usuario 17896613-8    Cerrar sesion    Cambiar password

Administrar docente  
 Administrar tipo de problema  
 Administrar Asignaturas  
 Actualizar problema

### Administrar Asignatura

Codigo	Nombre	Borrar	Actualizar
620108	introduccion a la programacion (620108)	Borrar	Actualizar
620137	Diseño y contruccion de algoritmos (620137)	Borrar	Actualizar
620435	Estructura de datos (620435)	Borrar	Actualizar
620168	Teoria de Sistemas	Borrar	Actualizar
620186	Practica profesional 2 (620186)	Borrar	Actualizar
610166	paradigma de la programacion (610166)	Borrar	Actualizar
620171	Practica profesional I (620171)	Borrar	Actualizar

Codigo:

Nombre:

Guardar

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

### Ilustración 23: Administrar Asignaturas

- Actualizar Problema:**  
 Permite actualizar el nombre, descripción y valores de entrada y salida de los distintos problemas registrados en el sistema.

# SAAPE

Sistema de Apoyo al Aprendizaje de Programacion Estructurada

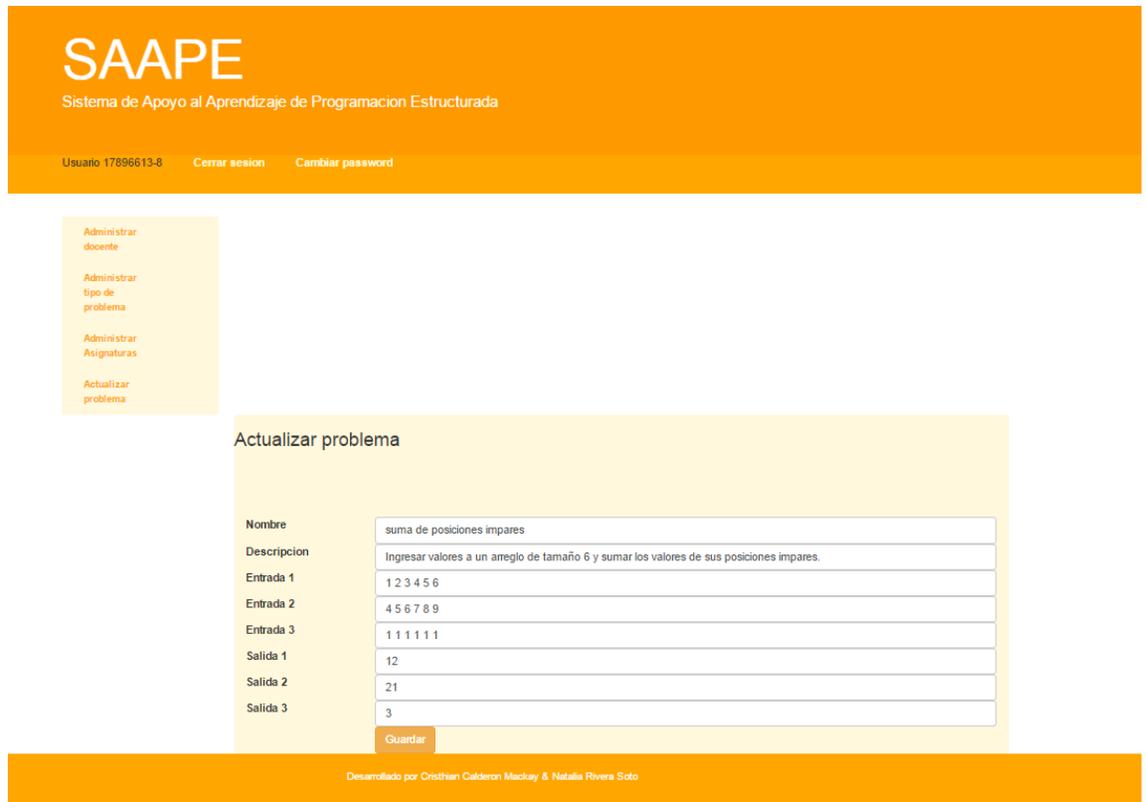
Usuario 17896613-8    Cerrar sesion    Cambiar password

Administrar docente  
 Administrar tipo de problema  
 Administrar Asignaturas  
 Actualizar problema

### Lista de ejercicios

Numero	Tipo de problema	Nombre	Descripcion	Actualizar
1	arrays	suma de unidimensionales (vecciones) impares	Ingresar valores a un arreglo de tamaño 6 y sumar los valores de sus posiciones impares.	Actualizar
2	Arbol Binario de Busqueda	Creacion y Recorrido AB	Crear un arbol binario de 3 nodos con recorrido inorden, es decir, Si visitamos primero hijo izquierdo, luego el padre y finalmente el hijo derecho.	Actualizar
3	Arbol Binario de Busqueda	Creacion y Recorrido AB PREO	Crear un arbol binario de 3 nodos con recorrido preorden, es decir, Primero el padre, luego el hijo izquierdo y finalmente el hijo derecho.	Actualizar
4	Arbol Binario de Busqueda	Creacion y Recorrido AB POSTO	Crear un arbol binario de 3 nodos con recorrido postorden, es decir, Primero hijo izquierdo, luego el hijo derecho y finalmente el padre.	Actualizar
5	funciones recursivas	Funcion Potencia	Calcular potencias mediante una funcion recursiva.	Actualizar
6	funciones recursivas	Retorno digitos pares	Construir una funcion recursiva que permita, dado un numero entero 'n', retornar el numero que se forma usando solo los digitos pares contenidos en 'n'. Por ejemplo, se tienen los siguientes casos de prueba: RECIBE RETORNA 2934 => 243 => 0 8 => 8 3489 => 488	Actualizar
7	funciones recursivas	Sumatoria	Construir una funcion recursiva EFICIENTE que permita RETORNAR el valor de la siguiente sumatoria: S= 1*n + 2*n-1 + 3*n-2 + ... + n*1 OBSERVACIONES: Recuerde que una solucion recursiva EFICIENTE no contempla el uso de ciclos. La potencia DEBERA ser calculada de manera RECURSIVA.	Actualizar

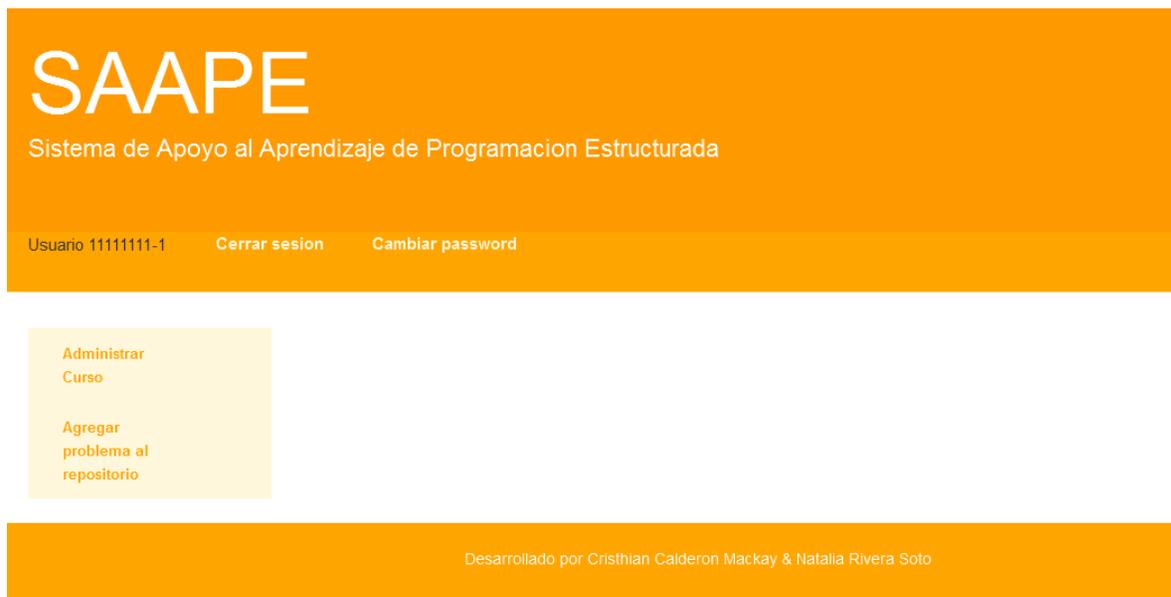
### Ilustración 24: Actualizar Problema



**Ilustración 25: Actualizar Problema 2**

## 16.2 Profesor:

Al iniciar sesión como profesor se le presenta el siguiente menú:



**Ilustración 26: Interfaz Docente**

Dentro de las opciones, se encuentra:

- **Administrar Curso:**  
Permite manejar la información de los cursos tanto como su registro, modificación, eliminación y visualización, además de la creación de un cuaderno de ejercicios para los alumnos.
  - **Crear Curso:**  
Permita la creación de un curso (seleccionando de la lista disponible), asignando además la sección correspondiente y una breve descripción si es que lo desea.

**Ilustración 27: Crear Curso**

- **Ver Curso:**  
Se visualizan los cursos creados por el docente que se encuentra en la sesión. Existen dos opciones por curso.



**Ilustración 28: Ver Curso**

1. **Editar:**  
Se muestra información de la asignatura seleccionada, además se pueden inscribir alumnos y eliminar el curso (al momento de eliminar curso, se borran los alumnos inscritos en él junto a sus cuadernos de ejercicios).

# SAAPE

Sistema de Apoyo al Aprendizaje de Programacion Estructurada

Usuario 11111111-1   Cerrar sesion   Cambiar password

Administrar  
Curso

Agregar  
problema al  
repositorio

## Datos del curso

Nombre del curso : introduccion a la programacion (620108)  
Codigo del curso : 620108  
Seccion : 2  
Descripcion del curso :

[Inscribir alumno](#)

[Borrar curso\(Alumnos,cuadernos\)](#)

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

### Ilustración 29: Editar Curso

# SAAPE

Sistema de Apoyo al Aprendizaje de Programacion Estructurada

Usuario 11111111-1   Cerrar sesion   Cambiar password

Administrar  
Curso

Agregar  
problema al  
repositorio

## Opciones

[Inscribir alumno forma manual](#)  
[Inscribir alumno forma automatica](#)

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

### Ilustración 30: Inscribir Alumnos

**1.1 Inscribir Alumnos Manual:**  
Se registra el alumno de forma individual junto a sus datos personales.

**SAAPE**  
Sistema de Apoyo al Aprendizaje de Programacion Estructurada

Usuario 11111111-1    Cerrar sesion    Cambiar password

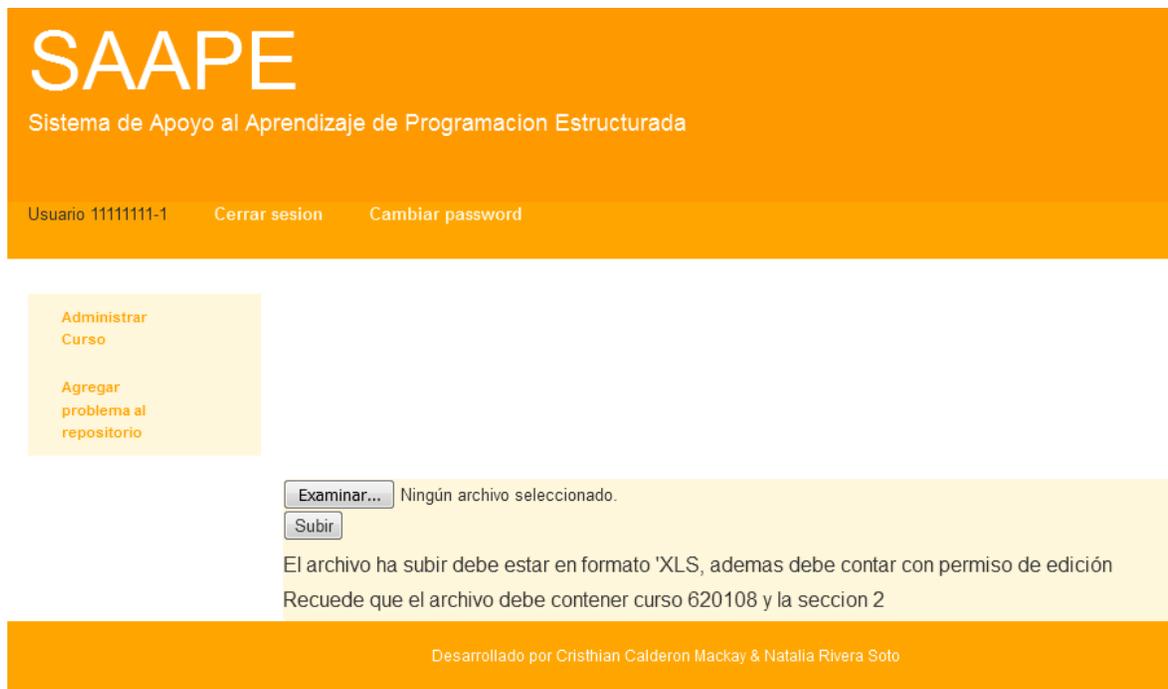
Administrar Curso  
Agregar problema al repositorio

Rut: Rut con formato xxxxxxxx-x  
Nombre: Nombre  
Apellidos: Apellido  
Correo del alumno: correo del alumno  
Nombre de carrera: ingenieria de ejecucion en computacion e informatica  
Nombre del curso: introduccion a la programacion (620108)  
Seccion: Seccion  
Guardar

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

**Ilustración 31: Inscribir Alumno Manual**

**1.2 Inscribir Alumnos Automático:** Se registra el alumno de forma grupal mediante un archivo Excel, éste debe ser formato “XLS”, en el caso de poseer restricciones de edición se debe “Guardar como archivo editable”, además la lista del documento debe coincidir con el código y sección del curso creado.



### Ilustración 32: Inscribir Alumno Automático

- Lista de Curso: Se muestra el listado de alumnos pertenecientes al curso-sección junto a sus datos personales y el progreso que llevan desarrollado de su cuaderno de ejercicios. Por otra parte se pueden modificar y eliminar del curso y exportar esta información grupal a un documento de formato PDF.

# SAAPE

Sistema de Apoyo al Aprendizaje de Programacion Estructurada

Usuario 11111111-1    Cerrar sesion    Cambiar password

Administrar  
Curso

Agregar  
problema al  
repositorio

## Lista del curso

EXPORTAR PDF

Rut	Nombre	Apellidos	Progreso	Borrar	Actualizar
19087691-8	Fernando Matias	Linco Poblete	0%	Borrar	Actualizar
18412209-K	Ignacio Sebastián	Maldonado Mardones	0%	Borrar	Actualizar
17898160-9	Rodrigo Francisco	Oliva Troncoso	0%	Borrar	Actualizar
18359548-2	Luis Alfredo	PÉrez Aguilera	0%	Borrar	Actualizar
18683938-2	Gabriel Antonio	Pinto Encina	0%	Borrar	Actualizar
18505083-1	Hugo Esteban	RÍos Fuentes	0%	Borrar	Actualizar
18420182-8	HÉctor Andrés	Riquelme Cerda	0%	Borrar	Actualizar
18652271-0	Ernesto Alejandro	Roa Ortega	25%	Borrar	Actualizar
17574783-4	Egar Mauricio	Rojas Martínez	0%	Borrar	Actualizar
18405013-7	Camilo José	Sanhueza San Martín	0%	Borrar	Actualizar
18412539-0	Pablo Nicolás	Soto BÓrquez	0%	Borrar	Actualizar
18143361-2	Jorge Lucas Maximiliano	Torres FaÚndez	0%	Borrar	Actualizar
18536104-7	Felipe Ignacio	Valenzuela Cornejo	0%	Borrar	Actualizar
17898990-1	Robert Eduardo	Valenzuela Navarrete	0%	Borrar	Actualizar
18147006-2	Marcos Angelo	Vega Rifo	0%	Borrar	Actualizar
18433854-8	Andrés Ignacio	Villa Bello	0%	Borrar	Actualizar

Desarrollado por Christian Calderon Mackay & Natalia Rivers Soto

### Ilustración 33: Lista del Curso Rendimiento

Al seleccionar el Rut de algún alumno se podrá visualizar el rendimiento de éste, en cuanto a los ejercicios que posee su cuaderno, sus intentos y estados en los que se encuentran. Además esta información se puede **exportar a un documento en formato PDF**.

# SAAPE

Sistema de Apoyo al Aprendizaje de Programacion Estructurada

[Usuario 11111111-1](#)   
 [Cerrar sesion](#)   
 [Cambiar password](#)

- Administrar Curso
- Agregar problema al repositorio

## Rendimiento del Alumno 18359548-2

[EXPORTAR PDF](#)

Nombre	Tipo	Descripcion	intentos	Estado
Imprime 5	secuencia simple	imprime el resto de dos numeros enteros	0	Pendiente
Imprime 3	secuencia simple	imprima la resta de dos numeros enteros	0	Pendiente
Imprime 1	secuencia simple	imprima la multiplicacion de dos numeros enteros	0	Pendiente
Imprime 2	secuencia simple	imprima la suma de dos numeros enteros	0	Pendiente

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

**Ilustración 34: Alumno Rendimiento**

- Crear Cuaderno de Ejercicio: Esta opción le permite al profesor crear un cuaderno con una cantidad de ejercicios (limitados entre 1 y 10) para cada alumno de un curso sección en específico. Primero se debe seleccionar el tópico que se desea trabajar, luego escoger la cantidad de ejercicios, posteriormente escoger a que curso sección se le designará el cuaderno y finalmente escoger una fecha de caducidad, lo que permite al docente presionar a sus alumnos para que realicen los ejercicios en un tiempo límite. Cabe mencionar que el docente puede realizar esta operación las veces que quiera (mientras la cantidad de ejercicios registrados en el repositorio lo permita), es decir se puede volver a crear un cuaderno, ya sea del mismo u otro tópico.

**SAAPE**  
Sistema de Apoyo al Aprendizaje de Programacion Estructurada

Usuario 11111111-1    Cerrar sesion    Cambiar password

Administrar Curso

Agregar problema al repositorio

**Crear cuaderno**

Tipo de problema: sequencia simple

Cantidad de ejercicios: [input field]

Curso: introduccion a la programacion (620108)

Seccion: [input field]

Fecha de caducidad: [input field]

Guardar

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

**Ilustración 35: Crear Cuaderno de Ejercicios**

- Agregar Problema al Repositorio:

**SAAPE**  
Sistema de Apoyo al Aprendizaje de Programacion Estructurada

Usuario 11111111-1   Cerrar sesion   Cambiar password

Administrar Curso

Agregar problema al repositorio

Tipo de problema: secuencia simple

Nombre: nombre del problema

Descripcion: Descripcion del problema

Entrada 1: Valor entrada 1

Salida 1: Valor salida 1

Entrada 2: Valor entrada 2

Salida 2: Valor salida 2

Entrada 3: Valor entrada 3

Salida 3: Valor salida 3

Guardar

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

### Ilustración 36: Agregar Problema al Repositorio

Esta opción le permite a cada docente aportar ejercicios de un tipo en específico al repositorio. Está compuesto por un “Tipo de Problema”, “Nombre”, “Descripción”, tres valores de entrada y tres valores de salida. La descripción corresponde al enunciado del ejercicio. En este punto siempre se debe especificar la cantidad y tamaño de las variables, ya que es esencial para que el alumno lo pueda realizar de manera correcta. Para dejar esto claro se presenta un par de ejemplos.

**Enunciado INCORRECTO:**

“Cree un programa que sume N números”

**Enunciado CORRECTO:**

“Cree un programa que sume 3 números”

**Enunciado INCORRECTO:**

“Cree un programa que sume dos matrices de N x N”

**Enunciado CORRECTO:**

“Cree un programa que sume dos matrices de 2 x 2”

Posterior al enunciado, se deben ingresar los datos de prueba (los que son siempre 3 para cada ejercicio), que son los que permitirán luego corroborar si el ejercicio realizado por el alumno es correcto o no. Cada entrada y/o salida debe tener la cantidad de variables especificadas en el enunciado. Para dejar esto claro se presentan ejemplos para los enunciados anteriormente expuestos.

<b>Tipo de problema</b>	secuencia simple
<b>Nombre</b>	Suma
<b>Descripcion</b>	Cree un programa que sume 3 números
<b>Entrada 1</b>	1 3 4
<b>Salida 1</b>	8
<b>Entrada 2</b>	7 9 2
<b>Salida 2</b>	18
<b>Entrada 3</b>	0 -1 1
<b>Salida 3</b>	0

[Guardar](#)

En el caso de ser una matriz, se deben ingresar de manera lineal, por ejemplo:

$$\begin{pmatrix} 2 & 2 \\ 3 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 2 \\ 8 & 5 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 11 & 6 \end{pmatrix}$$

Si queremos ingresar como caso de prueba la siguiente matriz, ésta es la manera correcta de hacerlo (mirar valor de entrada 1 y valor de salida 1)

<b>Tipo de problema</b>	arrays bidimensionales (matrices)
<b>Nombre</b>	Suma de dos matrices
<b>Descripcion</b>	Cree un programa que sume dos matrices de 2 x 2
<b>Entrada 1</b>	2 2 3 1 1 2 8 5
<b>Salida 1</b>	3 4 11 6
<b>Entrada 2</b>	1 1 1 1 1 1 1 2
<b>Salida 2</b>	2 2 2 3
<b>Entrada 3</b>	0 1 2 1 1 0 1 1
<b>Salida 3</b>	1 1 3 2

[Guardar](#)

Finalmente al presionar en “Guardar” el ejercicio será almacenado en el repositorio, quedando a disposición de cualquier docente (no solo de quién lo ha creado).

### 16.3 Alumno:

Al iniciar sesión como alumno se le presenta el siguiente menú:

# SAAPE

Sistema de Apoyo al Aprendizaje de Programacion Estructurada

[Usuario 18359548-2](#)   
 [Cerrar sesion](#)   
 [Cambiar password](#)

- ver cuaderno
- Rendimiento
- ver problema resuelto

## Bienvenido de nuevo alumno

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

**Ilustración 37: Menú Alumno**

- [Ver Cuaderno:](#)

Numero del problema	Nombre del problema	Tipo de problema	Descripción del problema	Estado	Intentos
22	Suma	secuencia simple	Cree un programa que sume 3 numeros	Pendiente	0
12	Imprime 1	secuencia simple	imprima la multiplicacion de dos numeros enteros	Pendiente	0
14	Imprime 3	secuencia simple	imprima la resta de dos numeros enteros	Pendiente	0
21	Suma de dos matrices	arrays bidimensionales (matrices)	Cree un programa que sume dos matrices de 2 x 2	Pendiente	0
16	Imprime 5	secuencia simple	imprime el resto de dos numeros enteros	Pendiente	0
15	Imprime 4	secuencia simple	imprima la division de dos numeros enteros	Pendiente	0

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

### Ilustración 38: Ver Cuaderno

Esta opción muestra los ejercicios disponibles asignados por el profesor para que el alumno resuelva en un tiempo determinado. Aquí se especificará el estado de los problemas e intentos realizados.

# SAAPE

Sistema de Apoyo al Aprendizaje de Programacion Estructurada

[Usuario 18342451-3](#)
[Cerrar sesion](#)
[Cambiar password](#)

- [ver cuaderno](#)
- [Rendimiento](#)
- [ver problema resuelto](#)

## Datos del problema

Numero del problema: 22

Nombre del problema: Suma

Tipo de problema: secuencia simple

Descripcion del problema:

Cree un programa que sume 3 numeros

[Ver compilador](#)

**Ilustración 39: Descripción del Problema**

Al estar en estado pendiente el alumno puede intentar realizarlo yendo al número del problema correspondiente y posteriormente en la opción ver compilador en donde se abrirá una pestaña donde podrá colocar su código.

<pre> 1 //Coloca tu código aquí, luego compila y finalmente envia tu respuesta 2 3 #include&lt;stdio.h&gt; 4 int main(){ 5 6 int x,y,z,suma; 7 suma=0; 8 scanf("%d",&amp;x); 9 scanf("%d",&amp;y); 10 scanf("%d",&amp;z); 11 suma=x+y+z; 12 printf("%d",suma); 13 getchar();getchar(); 14 return 0; 15 }</pre>	<div style="text-align: right;"> <span>C</span> </div> <div style="text-align: right; margin-top: 10px;"> <span>EnviarRespuesta</span> </div> <div style="text-align: right; margin-top: 10px;"> <span>Compile</span> </div>
--	--

**Ilustración 40: Compilador**

Finalmente el alumno debe primero seleccionar la opción “compilar”, para que se compile su código y luego seleccionar la opción “enviar la respuesta”, para que se envíe su respuesta, donde el sistema le dirá si este está correcto o no. De estar correcto se cambia a estado completado, en caso contrario se suma uno a la cantidad de intentos.

En este punto el alumno debe tener claro que la única salida que debe mostrar (printf) es el resultado final (sin ningún tipo de mensaje adicional. Ej: “el resultado es x”) y se deben omitir los mensajes de ingreso de datos. Para mayor claridad se da un ejemplo a continuación:

### **Manera INCORRECTA**

```
#include<stdio.h>
int main(){

int x,y,suma;
suma=0;
printf("Ingrese el primer valor");
scanf("%d",&x);
printf("Ingrese el segundo valor");
scanf("%d",&y);
suma=x+y;
printf("El resultado es %d",suma);
getchar();getchar();
return 0;
}
```

### **Manera CORRECTA**

```
#include<stdio.h>
int main(){

int x,y,suma;
suma=0;
scanf("%d",&x);
scanf("%d",&y);
suma=x+y;
printf("%d",suma);
getchar();getchar();
return 0;
}
```

En el caso de ser una matriz, el alumno deberá imprimirla sin ningún tipo de espacio, como se muestra a continuación:

```
for(i=0; i<2; i++)
{
  for (j=0; j<2; j++)
  {
    printf("%d", elemento[i][j]);
  }
}
```

- Rendimiento:

En esta opción el alumno podrá ver su rendimiento, los ejercicios que realizó, sus intentos y estados en los que se encuentran. Además esta información se puede exportar a un documento en formato PDF.

**SAAPE**  
Sistema de Apoyo al Aprendizaje de Programación Estructurada

Usuario 18359548-2    Cerrar sesion    Cambiar password

- ver cuaderno
- Rendimiento**
- ver problema resuelto

## Rendimiento del Alumno

EXPORTAR PDF

Nombre	Tipo	Descripción	intentos	Estado
Imprime 5	secuencia simple	imprime el resto de dos numeros enteros	0	Pendiente
Imprime 3	secuencia simple	imprima la resta de dos numeros enteros	0	Pendiente
Imprime 1	secuencia simple	imprima la multiplicacion de dos numeros enteros	0	Pendiente
Imprime 2	secuencia simple	imprima la suma de dos numeros enteros	0	Pendiente

Desarrollado por Cristhian Calderon Mackay & Natalia Rivera Soto

**Ilustración 41: Rendimiento del Alumno**

- Ver problema resuelto:

En esta opción el alumno podrá ver el enunciado de los problemas que ha realizado en el transcurso del semestre.

**SAAPE**  
Sistema de Apoyo al Aprendizaje de Programación Estructurada

Usuario 18342451-3   Cerrar sesión   Cambiar password

ver cuaderno  
Rendimiento  
ver problema resuelto

### Datos del cuaderno

Numero del problema	Nombre del problema	Tipo de problema	Descripción del problema	Estado	Intentos
22	Suma	secuencia simple	Cree un programa que sume 3 números	Completado	1
21	Suma de dos matrices	arrays bidimensionales (matrices)	Cree un programa que sume dos matrices de 2 x 2	Completado	1

Desarrollado por Cristhian Calderon Mackay & Nafalia Rivera Soto

**Ilustración 42: Ver problema resuelto**

---

## **17 ANEXO: ESPECIFICACION DE LAS PRUEBAS**

---

### **17.1 Pruebas de Unidad**

#### **17.1.1 <Crear Docente>**

- Para realizar esta prueba se necesita la configuración estándar del computador en términos de hardware y software.
- Pre condiciones de las pruebas: Ingresar al sistema como perfil administrador.

Id	Descripción Requerimiento Funcional	Entrada					Salida esperada	Salida Obtenida	Evaluación	OBS.
		Rut: String (10)	Nombre: String (50)	Apellidos : String(50)	Depto: String(50)	Correo:String(50 )			Éxito / Fracaso	
1	Ingresando RUT en blanco	""	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: "Campo obligatorio"	Mensaje: "El Rut es un campo obligatorio"	Éxito	
2	Ingresando RUT con caracteres	17asd234-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: "Rut invalido".	Rut invalido".	Éxito	
3	Ingresando RUT correcto sintácticamente pero invalido	17714204-3	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Rut invalido".	Mensaje interfaz: "Rut invalido".	Éxito	
4	Ingresando Rut registrado	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "usuario registrado".	Mensaje interfaz: "El usuario ya está registrado en el sistema".	Éxito	
5	Ingresando RUT válido	55555555-5	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "usuario registrado".	Mensaje interfaz: "Profesor ingresado exitosamente"	Éxito	
6	Ingresando Nombre con números	11111111-1	Pedro1	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Nombre invalido".	Mensaje interfaz: "Profesor ingresado exitosamente"	Fracaso	Se permite números en el nombre.
7	Ingresando Nombre con símbolos	11111111-1	Pedro#@	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Nombre invalido".	Mensaje interfaz: "Profesor ingresado exitosamente"	Fracaso	Se permite símbolos en el nombre.

8	Ingresando Nombre con un carácter	11111111-1	A	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Longitud invalida".	Mensaje interfaz: "El campo nombre debe contener al menos 2 caracteres"	Éxito	
9	Ingresando Nombre en blanco	11111111-1	""	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: "Campo obligatorio"	Mensaje: "El nombre es un campo obligatorio"	Éxito	
10	Ingresando Nombre valido	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: "usuario registrado"	Mensaje interfaz: "Profesor ingresado exitosamente"	Éxito	
11	Ingresando Apellidos con números	11111111-1	Pedro	Pérez1	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Apellidos invalido".	Mensaje interfaz: "Profesor ingresado exitosamente"	Fracaso	Se permite números en apellidos.
12	Ingresando Apellidos con símbolos	11111111-1	Pedro	Pérez#@	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Apellidos invalido".	Mensaje interfaz: "Profesor ingresado exitosamente"	Fracaso	Se permite símbolos en apellidos.
13	Ingresando Apellidos con un carácter	11111111-1	Pedro	P	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Longitud invalida".	Mensaje interfaz: "El campo apellido debe contener al menos 2 caracteres"	Éxito	
14	Ingresando Apellido en blanco	11111111-1	Pedro	""	Sistema de Información	pperez@ubiobio.cl	Mensaje: "Campo obligatorio"	Mensaje: "Apellido es un campo obligatorio"	Éxito	
15	Ingresando	11111111-1	Pedro	Pérez	Sistema de	pperez@ubiobio.cl	Mensaje:	Mensaje	Éxito	

	Apellido valido				Información		"usuario registrado"	interfaz: "Profesor ingresado exitosamente"		
16	Ingresando Depto con números	11111111-1	Pedro	Pérez	Sistema de Información 1	pperez@ubiobio.cl	Mensaje interfaz: "depto. invalido".	Mensaje interfaz: "Profesor ingresado exitosamente"	Fracaso	Se permite números en depto.
17	Ingresando Depto con símbolos	11111111-1	Pedro	Pérez	Sistema de Información #@	pperez@ubiobio.cl	Mensaje interfaz: "Depto invalido".	Mensaje interfaz: "Profesor ingresado exitosamente"	Fracaso	Se permite símbolos en depto.
18	Ingresando Depto con un carácter	11111111-1	Pedro	Pérez	S	pperez@ubiobio.cl	Mensaje interfaz: "Longitud invalida".	Mensaje interfaz: "El campo depto debe contener al menos 2 caracteres"	Éxito	
19	Ingresando Depto en blanco	11111111-1	Pedro	Pérez	""	pperez@ubiobio.cl	Mensaje: "Campo obligatorio"	Mensaje: "Depto es un campo obligatorio"	Éxito	
20	Ingresando Depto valido	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: "usuario registrado"	Mensaje interfaz: "Profesor ingresado exitosamente"	Éxito	
21	Ingresando correo en blanco	11111111-1	Pedro	Pérez	Sistema de Información	""	Mensaje interfaz: "campo obligatorio".	Mensaje interfaz: "El campo es obligatorio"	Éxito	
22	Ingresando correo sin @	11111111-1	Pedro	Pérez	Sistema de Información	pperezubiobio.cl	Mensaje interfaz: "correo invalido".	Mensaje interfaz: "Ingrese una	Éxito	

								dirección de correo electrónico		
23	Ingresando correo sin dominio	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio	Mensaje interfaz: "correo invalida".	Mensaje interfaz: "Ingrese una dirección de correo electrónico"	Éxito	
24	Ingresando correo registrado	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: "correo registrado"	Mensaje: "Profesor ingresado exitosamente"	Fracaso	Se permite correo registrados anteriormente.
25	Ingresando correo valido	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: "usuario registrado"	Mensaje interfaz: "Profesor ingresado exitosamente"	Éxito	

**Tabla 81: Especificación Prueba Crear Docente**

### 17.1.2 <Administrar Asignatura>

- Para realizar esta prueba se necesita la configuración estándar del computador en términos de hardware y software.
- Pre condiciones de las pruebas: Ingresar al sistema como perfil administrador.

Id	Descripción Requerimiento o Funcional	Entrada		Salida esperada	Salida Obtenida	Evaluación	OBS.
		Código: integer	Nombre: String(1024)			Éxito / Fracaso	

1	Ingresar Código en blanco	""	Introducción a la Programación	Mensaje: "Campo obligatorio"	Mensaje: "El código es un campo obligatorio"	Éxito	
2	Ingresar Código con caracteres	123as	Introducción a la Programación	Mensaje: "código invalido".	Mensaje: "El campo código debe contener solo números"	Éxito	
3	Ingresar Nombre en blanco	123	""	Mensaje: "Campo obligatorio"	Mensaje: "El nombre es un campo obligatorio"	Éxito	
4	Ingresar Nombre con números	123	Introducción a la Programación <sup>12</sup>	Mensaje interfaz: "nombre invalido".	Mensaje interfaz: "Asignatura agregada exitosamente".	Fracaso	Se permite números en el nombre.
5	Ingresando Código y Nombre válido	123	Introducción a la Programación	Mensaje interfaz: "asignatura registrado".	Mensaje interfaz: "Asignatura agregada exitosamente".	Éxito	

**Tabla 82: Especificación Prueba Administrar Asignatura**

### 17.1.3 <Crear Curso>

- Para realizar esta prueba se necesita la configuración estándar del computador en términos de hardware y software.
- Pre condiciones de las pruebas: Ingresar al sistema como perfil profesor.

Id	Descripción Requerimiento o Funcional	Entrada		Salida esperada	Salida Obtenida	Evaluación	OBS.
		Sección: Integer	Descripción: String(1024)			Éxito / Fracaso	

1	Ingresando sección en blanco	""	En la siguiente asignatura...	Mensaje: "Campo obligatorio"	Mensaje: "Sección invalida"	Éxito	
2	Ingresando sección con caracteres	aja	En la siguiente asignatura...	Mensaje: "sección invalido".	Mensaje: "Sección invalida"	Éxito	
3	Ingresando sección con símbolo	1#	En la siguiente asignatura...	Mensaje interfaz: "sección invalido".	Mensaje interfaz: "Por favor ingrese un número".	Éxito	
4	Ingresando sección valida	1	En la siguiente asignatura...	Mensaje interfaz: "sección registrado".	Mensaje interfaz: "El curso se ha creado correctamente"	Éxito	
5	Ingresando descripción en blanco	1	""	Mensaje interfaz: "sección registrado".	Mensaje interfaz: "El curso se ha creado correctamente"	Éxito	

**Tabla 83: Especificación Prueba Crear Curso**

#### 17.1.4 <Crear Cuaderno>

- Para realizar esta prueba se necesita la configuración estándar del computador en términos de hardware y software.
- Pre condiciones de las pruebas: Ingresar al sistema como perfil profesor.

Id	Descripción Requerimiento o Funcional	Entrada		Salida esperada	Salida Obtenida	Evaluación	OBS.
		Cant. De ejercicios: Integer	Sección: Integer			Éxito /	

						<b>Fracaso</b>	
1	Ingresando Cant. De ejercicios en blanco	""	1	Mensaje: "Campo obligatorio"	Mensaje: "Campo obligatorio"	Éxito	
2	Ingresando Cant. De ejercicios con caracteres	as	1	Mensaje: "Cant. De ejercicios invalido".	Mensaje: "Por favor ingrese números"	Éxito	
3	Ingresando Cant. De ejercicios con símbolo	#@	1	Mensaje interfaz: "Cant. De ejercicios invalido".	Mensaje: "Por favor ingrese número"	Éxito	
4	Ingresando Cant. De ejercicios con número negativo	-1	1	Mensaje interfaz: "Cant. De ejercicios invalido".	Mensaje interfaz: "Por favor seleccione un valor no menor a cero".	Éxito	
5	Ingresando Cant. De ejercicios con coma	1,1	1	Mensaje interfaz: "Cant. De ejercicios invalido".	Mensaje: "Por favor ingrese número"	Éxito	
6	Ingresando sección en blanco	1	""	Mensaje: "Campo obligatorio"	Mensaje: "Sección invalida"	Éxito	
7	Ingresando sección con caracteres	1	aja	Mensaje: "sección invalido".	Mensaje: "Sección invalida"	Éxito	
8	Ingresando sección con símbolo	1	1#	Mensaje interfaz: "sección invalido".	Mensaje interfaz: "Por favor ingrese un número".	Éxito	

9	Ingresando sección valida	1	1	Mensaje interfaz: "sección registrado".	Mensaje interfaz: "Cuaderno creado correctamente"	Éxito	
---	---------------------------	---	---	---	---	-------	--

**Tabla 84: Especificación Prueba Crear Cuaderno**

### 17.1.5 <Crear Alumno de Forma Manual>

- Para realizar esta prueba se necesita la configuración estándar del computador en términos de hardware y software.
- Pre condiciones de las pruebas: Ingresar al sistema como perfil profesor y haber creado un curso-sección.

Id	Descripción Requerimiento o Funcional	Entrada					Salida esperada	Salida Obtenida	Evaluación	OBS.
		Rut: String (10)	Nombre: String (50)	Apellidos : String(50)	Depto: String(50)	Correo:String(50 )			Éxito / Fracaso	
1	Ingresando RUT en blanco	""	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: "Campo obligatorio"	Mensaje: "El Rut es un campo obligatorio"	Éxito	
2	Ingresando RUT con caracteres	17asd234-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: "Rut invalido".	Rut invalido".	Éxito	
3	Ingresando RUT correcto sintácticament e pero invalido	17714204-3	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Rut invalido".	Mensaje interfaz: "Rut invalido".	Éxito	
4	Ingresando Rut registrado	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "usuario	Mensaje interfaz: "El usuario ya	Éxito	

							registrado".	está registrado en el sistema".		
5	Ingresando RUT válido	55555555-5	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "usuario registrado".	Mensaje interfaz: "Alumno ingresado exitosamente"	Éxito	
6	Ingresando Nombre con números	11111111-1	Pedro1	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Nombre invalido".	Mensaje interfaz: "Alumno ingresado exitosamente"	Fracaso	Se permite números en el nombre.
7	Ingresando Nombre con símbolos	11111111-1	Pedro#@	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Nombre invalido".	Mensaje interfaz: "Alumno ingresado exitosamente"	Fracaso	Se permite símbolos en el nombre.
8	Ingresando Nombre con un carácter	11111111-1	A	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Longitud invalida".	Mensaje interfaz: "El campo nombre debe contener al menos 2 caracteres"	Éxito	
9	Ingresando Nombre en blanco	11111111-1	""	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: "Campo obligatorio"	Mensaje: "El nombre es un campo obligatorio"	Éxito	
10	Ingresando Nombre valido	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: "usuario registrado"	Mensaje interfaz: "Alumno ingresado exitosamente"	Éxito	
11	Ingresando Apellidos con números	11111111-1	Pedro	Pérez1	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: "Apellidos invalido".	Mensaje interfaz: "Alumno ingresado"	Fracaso	Se permite números en apellidos.

								exitosamente”		
12	Ingresando Apellidos con símbolos	11111111-1	Pedro	Pérez#@	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: “Apellidos invalido”.	Mensaje interfaz: “Alumno ingresado exitosamente”	Fracaso	Se permite símbolos en apellidos.
13	Ingresando Apellidos con un carácter	11111111-1	Pedro	P	Sistema de Información	pperez@ubiobio.cl	Mensaje interfaz: “Longitud invalida”.	Mensaje interfaz: “El campo apellido debe contener al menos 2 caracteres”	Éxito	
14	Ingresando Apellido en blanco	11111111-1	Pedro	“	Sistema de Información	pperez@ubiobio.cl	Mensaje: “Campo obligatorio”	Mensaje: “Apellido es un campo obligatorio”	Éxito	
15	Ingresando Apellido valido	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: “usuario registrado”	Mensaje interfaz: “Alumno ingresado exitosamente”	Éxito	
16	Ingresando Depto con números	11111111-1	Pedro	Pérez	Sistema de Información 1	pperez@ubiobio.cl	Mensaje interfaz: “depto. invalido”.	Mensaje interfaz: “Alumno ingresado exitosamente”	Fracaso	Se permite números en depto.
17	Ingresando Depto con símbolos	11111111-1	Pedro	Pérez	Sistema de Información #@	pperez@ubiobio.cl	Mensaje interfaz: “Depto invalido”.	Mensaje interfaz: “Alumno ingresado exitosamente”	Fracaso	Se permite símbolos en depto.
18	Ingresando Depto con un carácter	11111111-1	Pedro	Pérez	S	pperez@ubiobio.cl	Mensaje interfaz: “Longitud invalida”.	Mensaje interfaz: “El campo depto debe contener al menos 2	Éxito	

								caracteres”		
19	Ingresando Depto en blanco	11111111-1	Pedro	Pérez	“”	pperez@ubiobio.cl	Mensaje: “Campo obligatorio”	Mensaje: “Depto es un campo obligatorio”	Éxito	
20	Ingresando Depto valido	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: “usuario registrado”	Mensaje interfaz: “Profesor ingresado exitosamente”	Éxito	
21	Ingresando correo en blanco	11111111-1	Pedro	Pérez	Sistema de Información	“”	Mensaje interfaz: “campo obligatorio”.	Mensaje interfaz: “El campo es obligatorio”	Éxito	
22	Ingresando correo sin @	11111111-1	Pedro	Pérez	Sistema de Información	pperezubiobio.cl	Mensaje interfaz: “correo invalido”.	Mensaje interfaz: “Ingrese una dirección de correo electrónico”	Éxito	
23	Ingresando correo sin dominio	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio	Mensaje interfaz: “correo invalida”.	Mensaje interfaz: “Ingrese una dirección de correo electrónico”	Éxito	
24	Ingresando correo registrado	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: “correo registrado”	Mensaje: “Alumno ingresado exitosamente”	Fracaso	Se permite correo registrados anteriormente.
25	Ingresando correo valido	11111111-1	Pedro	Pérez	Sistema de Información	pperez@ubiobio.cl	Mensaje: “usuario registrado”	Mensaje interfaz: “Alumno ingresado exitosamente”	Éxito	

**Tabla 85: Especificación Prueba Crear Alumno de Forma Manual**

**17.1.6 <Agregar Problema al Repositorio>**

- Para realizar esta prueba se necesita la configuración estándar del computador en términos de hardware y software.
- Pre condiciones de las pruebas: Ingresar al sistema como perfil profesor.

Id	Descripción Requerimiento o Funcional	Entrada				Salida esperada	Salida Obtenida	Evaluación	OBS.
		Nombre: String (30)	Descripción: String (10000)	Valor entrada1: String(30)	Valor salida1: String(30)			Éxito / Fracaso	
1	Ingresando Nombre con números	Sume 3 números	Se pide ingresar ...	1 2 3	6	Mensaje: "Problema registrado"	Mensaje: "Problema registrado correctamente"	Éxito	
2	Ingresando Nombre con símbolos	Sume 3 números #	Se pide ingresar ...	1 2 3	6	Mensaje: "Problema registrado"	Mensaje: "Problema registrado correctamente"	Éxito	
3	Ingresando Nombre con un carácter	S	Se pide ingresar ...	1 2 3	6	Mensaje interfaz: "nombre	Mensaje interfaz: "El campo nombre	Éxito	

						invalido".	debe contener al menos 5 caracteres".		
4	Ingresando Nombre en blanco	""	Se pide ingresar ...	1 2 3	6	Mensaje interfaz: "Campo obligatorio"	Mensaje: "El campo nombre es obligatorio"	Éxito	
5	Ingresando Nombre valido	Sume 3 números	Se pide ingresar ...	1 2 3	6	Mensaje: "Problema registrado"	Mensaje: "Problema registrado correctamente"	Éxito	
6	Ingresando descripción en blanco	Sume 3 números	""	1 2 3	6	Mensaje: "Campo obligatorio"	Mensaje: "El campo descripción es obligatorio"	Éxito	
7	Ingresando Valor de entrada1 en blanco	Sume 3 números	Se pide ingresar ...	""	6	Mensaje: "Campo obligatorio"	Mensaje: "El campo valor entrada1 es obligatorio"	Éxito	
8	Ingresando Valor de salida1 en blanco	Sume 3 números	Se pide ingresar ...	1 2 3	""	Mensaje: "Campo obligatorio"	Mensaje: "El campo valor salida1 es obligatorio"	Éxito	

**Tabla 86: Especificación Prueba Agregar Problema al Repositorio**

---

## 18 ANEXO: DICCIONARIO DE DATOS DEL MODELO DE DATOS

---

A continuación el diccionario de datos del proyecto:

- Users:

Se encuentran los tipos de usuarios existentes en el sistema, junto a un nombre de usuario y su contraseña.

La información del usuario que se registra es la siguiente:

-Username: es el nombre de usuario que se le asigna a cada persona que se registre en el sistema, en este caso corresponde al rut de cada persona. Este es único y está compuesto por caracteres.

-Perfil: son los tipos de usuarios existentes en el sistema para identificar a cada persona que se registre según su categoría. Hay tres tipos el administrador, alumnos y profesor. Está compuesto por caracteres.

-Password: es la contraseña del usuario. El sistema por defecto asigna los primeros 5 dígitos del rut del usuario, una vez que el usuario ingrese a su cuenta tiene la opción de cambiar su contraseña. Está compuesto por caracteres.

- Alumno:

Son los usuarios a quienes está dirigido el sistema, debido a que ellos son los que ingresan al sistema para resolver el cuaderno de ejercicios asignados para cada uno de ellos.

La información del alumno que se registra es la siguiente:

-Username: es el nombre de usuario que se le asigna a cada alumno que se registre en el sistema, en este caso corresponde al rut de cada uno. Este es único y está compuesto por caracteres.

-Nombre\_alumno: es el nombre del alumno. Está compuesto por caracteres.

-Apellido\_alumno: es el apellido del alumno. Está compuesto por caracteres.

-Cod\_carrera: es el código de carrera al que corresponde el alumno. Está compuesto por un número entero.

-Correo\_alumno: es el correo que posee el alumno. Está compuesto por caracteres.

- Profesor: son los usuarios que acceden al sistema para incorporar los problemas de ejercicios a los cuadernos de los alumnos. Además de obtener informes de rendimientos colectivos o individuales.

La información del profesor que se registra es la siguiente:

-Username: es el nombre de usuario que se le asigna a cada profesor que se registre en el sistema, en este caso corresponde al rut de cada uno. Este es único y está compuesto por caracteres.

-Nombre\_prof: es el nombre del profesor. Está compuesto por caracteres.

-Apellido\_prof: es el nombre del profesor. Está compuesto por caracteres.

-Depto\_prof: es el departamento del que forma parte el profesor en la universidad. Está compuesto por caracteres.

-Correo\_prof: es el correo que posee el profesor. Está compuesto por caracteres.

- Curso\_Sección: es el curso y sección a la que pertenece el alumno de la asignatura correspondiente, registrada en el sistema.

La información del curso\_sección que se registra es la siguiente:

-Codcurso: es el código con el cual se identifica el curso (o código de la asignatura). Está compuesto por un número entero.

-Sección: un curso se divide en diversos grupos de alumnos, es decir, secciones. Está compuesto por un número entero.

-Nombre\_curso: es el nombre de la asignatura registrada en el sistema. Está compuesta por caracteres.

-Descripción\_curso: se detalla de que trata el curso registrado. Está compuesto por caracteres.

-Username: es el nombre de usuario que se le asigna a cada alumno que se registre en el sistema, en este caso corresponde al rut de cada uno. Este es único y está compuesto por caracteres.

- Cuaderno: es el sitio donde se le asignan los diversos ejercicios a resolver. Este se crea de manera aleatoria, por lo tanto es diferente para cada alumno.

La información del cuaderno que se registra es la siguiente:

-Username: es el nombre de usuario que se le asigna a cada alumno que se registre en el sistema, en este caso corresponde al rut de cada uno. Este es único y está compuesto por caracteres.

-Idproblema: es el índice que designa el sistema a cada ejercicio registrado. Está compuesto por un valor serial, es decir, que se autoincrementa y es único.

-Tipo\_problema: se visualizan los distintos tópicos existentes por unidad de cada asignatura. Está compuesto por caracteres.

-Progreso: Muestra la evolución del alumno a medida que resuelve los ejercicios del cuaderno, reflejando en porcentaje este progreso. Está compuesto por un valor entero.

-Intento: son las veces que el alumno intentó resolver un ejercicio determinado de su cuaderno hasta llegar a la respuesta correcta. Está compuesto por un valor entero.

-Caducidad: es la fecha límite que establece el profesor para realizar ciertos ejercicios que él ingresa al cuaderno de ejercicios de cada alumno. Está compuesto por valor date.

-Estado: es el estado de los ejercicios que componen el cuaderno. En primera instancia están fijos como “pendiente” y a medida que el alumno resuelva sus ejercicios de manera correcta estos cambiarán de estado. Está compuesto por caracteres.

- Problema: son los ejercicios registrados en el sistema de acuerdo a cada asignatura, que posteriormente son asignados a un cuaderno para que un alumno determinado lo resuelva.

La información del problema que se registra es la siguiente:

-Idproblema: es el índice que designa el sistema a cada ejercicio registrado. Está compuesto por un valor serial, es decir, que se autoincrementa y es único.

-Tipo\_problema: se visualizan los distintos tópicos existentes por unidad de cada asignatura. Está compuesto por caracteres.

-Nombre\_problema: es el nombre que recibe un ejercicio determinado de acuerdo a su contenido. Está compuesto por caracteres.

-Descripción\_problema: se detalla de que trata el ejercicio registrado. Está compuesto por caracteres.

-Valor\_entrada1: son los valores que se ingresan en el caso de prueba 1 para probar el código que el alumno ingreso como su solución a un ejercicio determinado.

Estos valores dependerán del tipo de ejercicio ya que el cuaderno posee diversos tópicos de la asignatura.

-Valor\_salida1: son los valores de respuesta que debe visualizar el código del alumno de acuerdo a los valores de entrada ingresados. De esta manera se efectúa una comparación entre los valores de salida reales (ingresados por el profesor) del código con los que deberían resultar.

Estos valores dependen del tipo de ejercicio que se evalúa.

-Valor\_entrada2: son los valores que se ingresan en el caso de prueba 2 para probar el código que el alumno ingreso como su solución a un ejercicio determinado.

Estos valores dependerán del tipo de ejercicio ya que el cuaderno posee diversos tópicos de la asignatura.

-Valor\_salida2: son los valores de respuesta que debe visualizar el código del alumno de acuerdo a los valores de entrada ingresados. De esta manera se efectúa una comparación entre los valores de salida reales (ingresados por el profesor) del código con los que deberían resultar.

Estos valores dependen del tipo de ejercicio que se evalúa.

-Valor\_entrada3: son los valores que se ingresan en el caso de prueba 3 para probar el código que el alumno ingreso como su solución a un ejercicio determinado.

Estos valores dependerán del tipo de ejercicio ya que el cuaderno posee diversos tópicos de la asignatura.

-Valor\_salida3: son los valores de respuesta que debe visualizar el código del alumno de acuerdo a los valores de entrada ingresados. De esta manera se

efectúa una comparación entre los valores de salida reales (ingresados por el profesor) del código con los que deberían resultar.

Estos valores dependen del tipo de ejercicio que se evalúa.

- Carreras: son las carreras existentes en la universidad, específicamente aquellas del área informática que utilizarán el software.

La información de las carreras que se registran es la siguiente:

-Codcarrera: es el código de carrera al que corresponde el alumno. Está compuesto por un número entero.

-Nombrecarrera: es el nombre de la carrera correspondiente.  
Está compuesto por caracteres.

- Codigo\_curso: son los códigos y nombres de las asignaturas que se impartirán en el sistema.

La información del Codigo\_curso que se registra es la siguiente:

-Codcurso: es el código con el cual se identifica el curso (o código de la asignatura). Está compuesto por un número entero.

-Nombre\_curso: es el nombre de la asignatura correspondiente.  
Está compuesto por caracteres.

- Pertenece: se registran los datos del nombre de usuario, código del curso y sección correspondiente al alumno que pertenece a un cierto curso.

La información de pertenece que se registra es la siguiente:

-Username: es el nombre de usuario que se le asigna a cada alumno que se registre en el sistema, en este caso corresponde al rut de cada uno. Este es único y está compuesto por caracteres.

-Codcurso: es el código con el cual se identifica el curso (o código de la asignatura). Está compuesto por un número entero.

-Sección: un curso se divide en diversos grupos de alumnos, es decir, secciones. Está compuesto por un número entero.

- Tipo\_problema: son los tópicos que se dictan en las distintas unidades de cada asignatura.

La información del Tipo\_problema que se registra es la siguiente:

-Id\_tipo: es el índice que designa el sistema a cada tipo de problema registrado. Está compuesto por un valor serial, es decir, que se autoincrementa y es único.

-Tipo\_problema: se visualizan los distintos tópicos existentes por unidad de cada asignatura. Está compuesto por caracteres.