



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

# “Sistema de Apoyo a la Asignatura Estructura de Datos”

---

SAED

**Alumnos:**

Karen Carrasco Jaque.

Anggie Fica Fuentes.

**Profesor Guía:**

Brunny Troncoso Pantoja.

**Fecha:**

01 de Marzo del 2013.

*Este Proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad del Bío-Bío en el proceso de titulación para la carrera Ingeniería Civil Informática.*

## RESUMEN

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad de Bío-Bío en el proceso de titulación para la carrera de Ingeniería Civil en Informática. El proyecto se titula “Sistema de apoyo a la asignatura Estructuras de datos SAED”.

La Universidad del Bío-Bío cuenta con dos carreras del área informática, las que imparten distintas asignaturas, entre ellas Estructura de datos, la cual tiene una gran dificultad entre los alumnos al momento de la resolución de ejercicios, por lo que provoca la necesidad de tener herramientas capaces de apoyar en esta etapa a los estudiantes, permitiendo facilitar la comprensión de la asignatura.

El presente proyecto se desarrolló para realizar la corrección automática de los distintos tipos de ejercicios de la asignatura señalada considerando la individualidad entre los alumnos para evitar la copia entre ellos, generando distintos listados de ejercicios para los estudiantes.

Para la elaboración del proyecto se escogió el lenguaje de programación PHP debido a la compatibilidad con la plataforma con la cual va a interactuar el sistema, y considerando que se adapta a la metodología de desarrollo iterativa, permitiendo realizar cambios en el sistema de manera simple.

Al implementar el sistema permitirá mejorar la tasa de aprobación de esta asignatura, facilitando la entrega y corrección de ejercicios, además ayuda a monitorear 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 "Support system to Data Structures subject SAED".

Universidad del Bio-Bio has two undergraduate programs that impart different subjects; among them Data Structures, which has a great difficulty for the students at the moment of resolving exercises. Because of that, there is a need of having tools able of supporting the students in this stage, allowing a better understanding of the subject.

This project was developed to make the automatic correction of the different types of exercises of the indicated subject, considering the individuality between the students to avoid cheating among them, and generating different exercises listings for the students.

For the development of this project, it was chosen the PHP programming language due to the compatibility with the platform which the system is going to interact with, and considering that it is adapted to the iterative development methodology, permitting making changes in the system in a simple way.

Implementing this system, will allow the improving in the passing rate of this subject, facilitating the delivery and correction of the exercises. Moreover, it helps the monitoring of the performance of every single student.

## Índice General

1	Introducción .....	10
2	Definición de la empresa o institución .....	11
2.1	Descripción de la empresa .....	11
2.2	Descripción del área de estudio .....	12
2.3	Descripción de la problemática.....	14
2.4	Posibles soluciones .....	16
2.4.1	Moodle .....	16
2.4.2	Plataforma ILIAS.....	18
2.4.3	Plataforma FLE3.....	19
2.4.4	Claroline .....	20
2.5	Definir Solución.....	22
3	Definición proyecto.....	26
3.1	Objetivos del proyecto.....	26
3.1.1	Objetivo General.....	26
3.1.2	Objetivos Específicos .....	26
3.2	Ambiente de Ingeniería de Software .....	26
3.3	Definiciones, Siglas y Abreviaciones.....	28
4	Especificación de requerimientos de Software.....	29
4.1	Alcances.....	29
4.2	Objetivo del software .....	30
4.2.1	Objetivos Generales.....	30
4.2.2	Objetivos Específicos .....	30
4.3	Descripción Global del Producto .....	31
4.3.1	Interfaz de usuario .....	31
4.3.2	Interfaz De Hardware .....	32
4.3.3	Interfaz Software.....	33

4.3.4	Interfaces de comunicación .....	33
4.4	Requerimientos Específicos .....	34
4.4.1	Requerimientos Funcionales del sistema .....	34
4.4.2	Interfaces externas de entrada .....	35
4.4.3	Interfaces externas de Salida.....	35
4.4.4	Atributos del producto.....	36
5	Factibilidad.....	37
5.1	Factibilidad técnica. ....	37
5.2	Factibilidad operativa.....	38
5.3	Factibilidad económica.....	39
5.4	Conclusión de la factibilidad.....	41
6	Análisis.....	42
6.1	Diagrama de casos de uso .....	42
6.1.1	Actores .....	42
6.1.2	Casos de Uso y descripción .....	43
6.1.3	Especificación de los Casos de Uso .....	46
6.2	Modelamiento de datos.....	60
7	Diseño.....	61
7.1	Diseño de Físico de la Base de datos .....	61
7.2	Diseño de arquitectura funcional .....	63
7.3	Diseño interfaz y navegación.....	64
7.4	Especificación de módulos .....	67
8	Pruebas.....	69
8.1	Elementos de prueba .....	69
8.2	Especificación de las pruebas. ....	70
8.3	Responsables de las pruebas .....	71
8.4	Calendario de pruebas.....	71

8.5	Detalle de las pruebas.....	71
8.6	Conclusiones de Prueba.....	71
9	Plan de Capacitación y entrenamiento .....	72
10	Plan de Implantación y Puesta en marcha .....	73
11	Resumen Esfuerzo requerido.....	74
12	Conclusiones.....	75
13	Bibliografía.....	76
14	ANEXO: PLANIFICACION INICIAL DEL PROYECTO .....	77
14.1	Estimación inicial de tamaño .....	78
15	ANEXO: RESULTADOS DE ITERACIONES EN EL DESARROLLO.....	82
16	ANEXO: MANUAL DE USUARIO.....	88
16.1	Profesor.....	88
16.2	Administrador.....	94
16.3	Alumno.....	96
17	ANEXO: ESPECIFICACION DE LAS PRUEBAS.....	98
17.1	Pruebas de Unidad.....	98
17.1.1	<Ingreso de Problemas>.....	98
17.1.2	<Ingreso de Casos de Pruebas>.....	99
17.1.3	<Ingreso de Prototipos de Función> .....	101
18	ANEXO: DICCIONARIO DE DATOS DEL MODELO DE DATOS .....	103

## Índice Tablas

TABLA 1: TABLA DECISIÓN PLATAFORMA .....	23
TABLA 2: REQUERIMIENTOS FUNCIONALES.....	35
TABLA 3: INTERFACES EXTERNAS DE ENTRADA.....	35
TABLA 4: INTERFACES EXTERNAS DE SALIDA.....	35
TABLA 5 COSTO ENERGÍA ELÉCTRICA .....	39
TABLA 6 PRESUPUESTO DEL PROYECTO .....	40
TABLA 7: "ESPECIFICACIÓN DE PRUEBAS" .....	70
TABLA 8: "ESFUERZO REQUERIDO" .....	74
TABLA 9 FACTORES DE PESO ACTORES.....	78
TABLA 10 FACTORES DE PESO CASOS DE USO.....	79
TABLA 11 EVALUACIÓN FACTORES TÉCNICOS.....	80
TABLA 12 EVALUACIÓN FACTORES AMBIENTALES.....	81
TABLA 13 CALCULO LOE.....	81
TABLA 14 ESPECIFICACIÓN DE PRUEBA INGRESAR PROBLEMA .....	98
TABLA 15 ESPECIFICACIÓN PRUEBA INGRESO DE CASOS DE PRUEBA.....	100
TABLA 16 ESPECIFICACIÓN PRUEBA INGRESO DE PROTOTIPO DE FUNCIÓN .....	102

## Índice Imágenes

<i>IMAGEN 1: DISEÑO INTERFAZ</i> .....	32
<i>IMAGEN 2: "INICIO MOODLE"</i> .....	82
<i>IMAGEN 3: "INGRESO ALUMNO A MOODLE"</i> .....	83
<i>IMAGEN 4: "CURSO POR TEMAS EN MOODLE"</i> .....	83
<i>IMAGEN 5: "CASOS DE PRUEBA"</i> .....	84
<i>IMAGEN 6: "COMANDO"</i> .....	85
<i>IMAGEN 7: "VISTA DE ACTIVIDAD DE ALUMNO"</i> .....	85
<i>IMAGEN 8: "ACTIVIDAD EN PROCESO DE COMPILACIÓN"</i> .....	86
<i>IMAGEN 9: "COMPILACIÓN FINALIZADA"</i> .....	86
<i>IMAGEN 10 "INICIO DEL PROFESOR"</i> .....	88
<i>IMAGEN 11 "INGRESAR PROBLEMA"</i> .....	89
<i>IMAGEN 16 "SELECCIÓN DEL TIPO DE EJERCICIO A CONSULTAR"</i> .....	90
<i>IMAGEN 17 "CONSULTA DE PROBLEMAS"</i> .....	90
<i>IMAGEN 12 "ACTUALIZAR PROBLEMA"</i> .....	91
<i>IMAGEN 13 "AGREGAR PROTOTIPO"</i> .....	92
<i>IMAGEN 14 "SELECCIÓN DE PROTOTIPO A MODIFICAR"</i> .....	92
<i>IMAGEN 15 "MODIFICAR PROTOTIPO"</i> .....	93
<i>IMAGEN 18 "MENÚ DEL ADMINISTRADOR"</i> .....	94
<i>IMAGEN 19 "EXTRACCIÓN DE ALUMNOS"</i> .....	95
<i>IMAGEN 20 "INICIO ALUMNO"</i> .....	96

## Índice Diagramas

<i>DIAGRAMA 1: ORGANIGRAMA FACULTAD DE CIENCIAS EMPRESARIALES.....</i>	<i>13</i>
<i>DIAGRAMA 2: FUNCIONAMIENTO ACTUAL ASIGNATURA ESTRUCTURA DE DATOS.....</i>	<i>15</i>
<i>DIAGRAMA 3: CASOS DE USO GENERAL.....</i>	<i>44</i>
<i>DIAGRAMA 4: "CASO DE USO DEL SERVICIO WEB".....</i>	<i>45</i>
<i>DIAGRAMA 5: "CASO DE USO DE PVA".....</i>	<i>45</i>
<i>DIAGRAMA 6: MODELO ENTIDAD RELACIÓN.....</i>	<i>60</i>
<i>DIAGRAMA 7: ÁRBOL DE DESCOMPOSICIÓN FUNCIONAL SISTEMA.....</i>	<i>63</i>
<i>DIAGRAMA 8: DISEÑO INTERFAZ.....</i>	<i>64</i>
<i>DIAGRAMA 9: JERARQUÍA MENÚ.....</i>	<i>65</i>
<i>DIAGRAMA 10: ESQUEMA DE NAVEGACIÓN PROFESOR.....</i>	<i>66</i>
<i>DIAGRAMA 11: ESQUEMA DE NAVEGACIÓN ALUMNO.....</i>	<i>66</i>

---

## 1 INTRODUCCIÓN

---

En la actualidad las herramientas computacionales son una base sustentable para el apoyo de la docencia en todas las universidades, más aun el acceso a internet, el cual facilita el aprendizaje e investigación tanto de alumnos como de profesores.

Hoy en día la universidad del Bío-Bío cuenta con diversas plataformas de apoyo a la docencia para facilitar material de las asignaturas y la comunicación entre profesores y alumnos. Una de las plataformas existente en la universidad es PVA plataforma virtual de aprendizaje, la cual está destinada específicamente para los alumnos de la Facultad de Ciencias Empresariales. Dentro de esta facultad se encuentran las carreras del área informática las que tienen diversas asignaturas de programación, las cuales necesitan un apoyo fundamental para la comprensión y aprendizaje de los alumnos.

El proyecto consiste en crear un sistema web que apoye a la docencia de la asignatura estructura de datos para facilitar la ejercitación de los alumnos y detectar los puntos más débiles dentro de la asignatura y de esta forma lograr menos deserción dentro de las carreras que la imparten.

A continuación se presentan las bases teóricas que sustentan el proyecto que se va a desarrollar. Para esto se realiza una investigación cuyo objetivo es conocer las herramientas disponibles para la construcción del sistema.

El sistema se encuentra descrito en distintos capítulos los que se basan en las etapas del desarrollo de un software tales como la especificación de los requerimientos de los usuarios detallando cada aspecto relevante del sistema, los distintos tipos de factibilidad dentro del proyecto, análisis del problema y diseño de la solución, además de pruebas del sistemas y temas posteriores al desarrollo del software.

---

## 2 DEFINICION 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 se dedica a contribuir mediante el cultivo del saber, de la educación superior, de la investigación, de la asistencia técnica y de la capacitación, a la formación de profesionales y el desarrollo de la región.
  
- Competencia directa.

En la octava región se encuentran cuatro universidades del consejo de rectores, todas ellas ubicadas en la ciudad de Concepción o sus alrededores. Por lo que la competencia directa de la Universidad del Bío-Bío, es la Universidad de Concepción , Universidad Católica de la Santísima Concepción y la Universidad Técnica Federico Santa María, además se encuentran algunas universidades privadas como la Universidad San Sebastián, Universidad de las Américas, Santo Tomás, INACAP y algunos institutos profesionales como: AIEP, IPP y Virginio Gómez, todos ellos cuentan con carreras relacionadas al área Informática.

## 2.2 Descripción del área de estudio

El sistema está dirigido a la universidad del Bío-Bío y directamente a la facultad de Ciencias Empresariales hacia el departamento de sistemas de información en el cual se encuentra las carreras de Ingeniería Civil Informática (ICI) e Ingeniería Ejecución en Computación e Informática (IECI) con sus respectivos jefes de carrera, los cuales están encargados de ver que asignaturas se dictan en un semestre y cuáles son los profesores que realizan estas asignaturas. Como el sistema es para la asignatura Estructura de datos está directamente relacionado con los profesores que imparten esta asignatura.

Cabe mencionar que durante el desarrollo de este proyecto, la facultad Ciencias Empresariales ha realizado un cambio a nivel organizacional, por lo que en su nueva estructura las carreras de ICI e IECI pasaron a ser Escuelas teniendo directa relación con el Decano de la Facultad.

### ➤ Objetivos Departamento Sistemas de Información

- 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.

### ➤ Estructura organizativa Departamento Sistemas de Información.

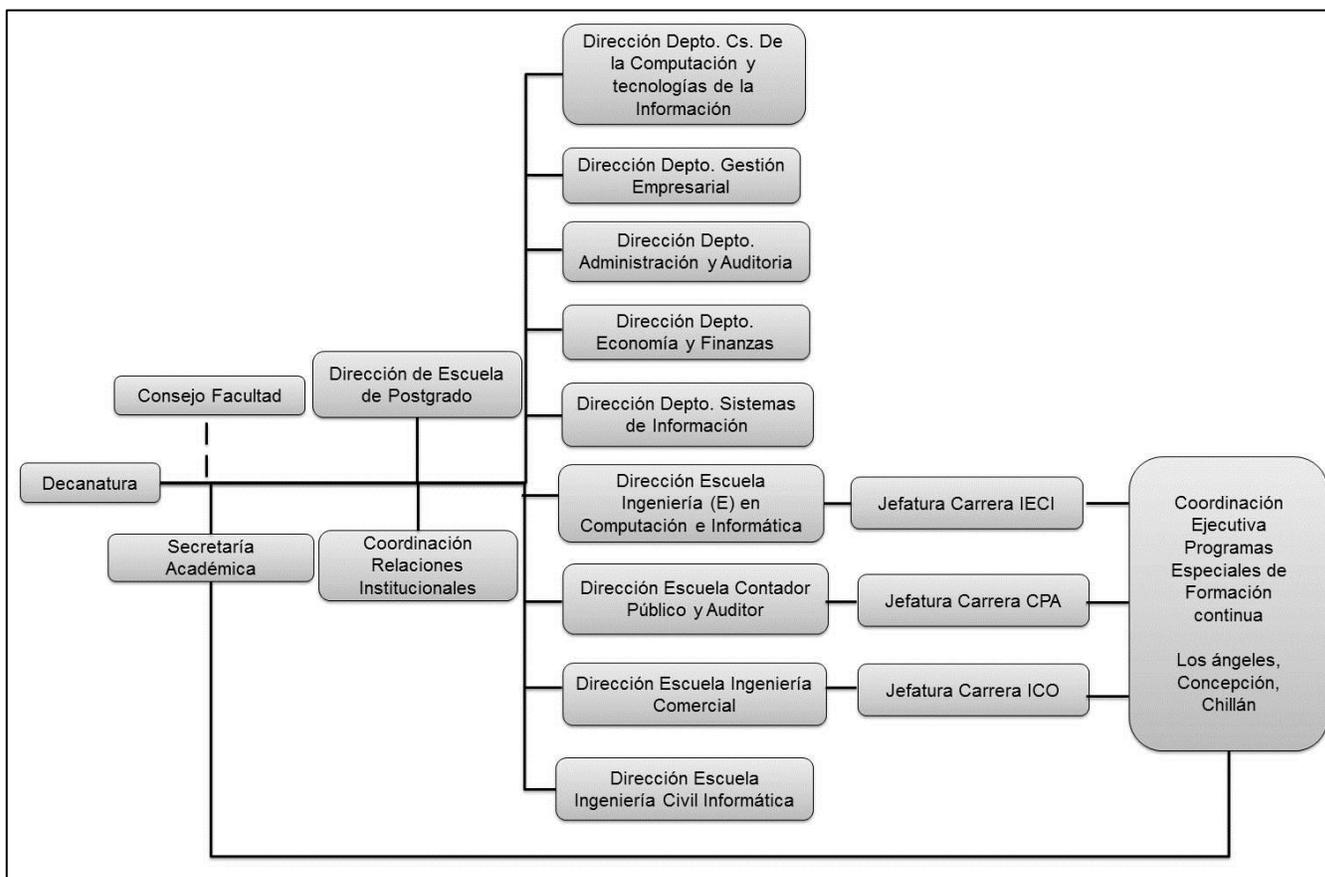


Diagrama : Organigrama Facultad de Ciencias Empresariales

La Dirección del Departamento, según lo señalado en el estatuto de la universidad, artículo 24, es responsable ante el Decano, de organizar la docencia y la investigación. Es la representante de la unidad bajo su dirección en todas las comunicaciones oficiales con el Decano y otros funcionarios superiores de la Universidad y también en todas las comunicaciones de la misma unidad con los estudiantes.

Los jefes de Carrera, se encargan de coordinar actividades de docencia para sus respectivas carreras, como también de desarrollar actividades de extensión.

### 2.3 Descripción de la problemática

La asignatura “Estructuras de Datos” es un curso de tercer semestre tanto para ICI como para IECI, los alumnos que cursan la asignatura conocen las bases de la programación (recorridos, búsquedas, ordenación, acciones/funciones, paso de parámetros, etc.) y aspectos elementales de estructuras de datos (arrays, punteros). En la asignatura se presentan las estructuras dinámicas encadenadas, las estructuras lineales (pilas, colas y listas), árboles binarios y estructuras más complejas (árboles de búsqueda, grafos, etc.) más algoritmos asociados a su tratamiento. Las clases ofrecen sesiones teóricas en las que se exponen los conceptos explicados con ejemplos, que se complementan con prácticas para anclar los conceptos presentados, utilizando C como lenguaje de programación.

Pero gran cantidad de los alumnos no realizan los ejercicios o al momento de practicar desconocen si encontraron la solución correcta y óptima, lo que genera problemas y retrasos en su aprendizaje, situación que los profesores no pueden detectar a tiempo, además la cantidad de alumnos en el curso hace difícil ayudar a cada uno en forma individual.

El funcionamiento actual de la asignatura se puede apreciar de mejor forma en el siguiente diagrama:

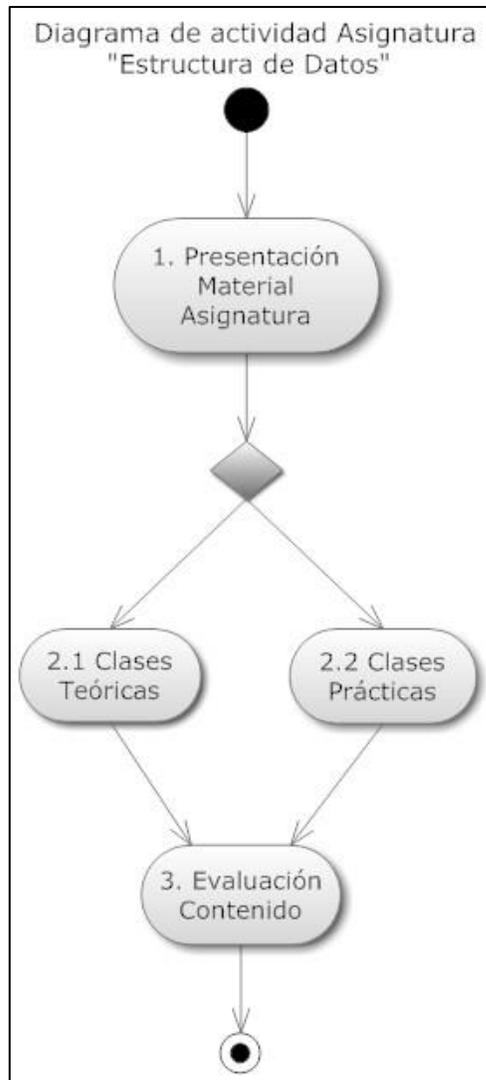


Diagrama : Funcionamiento Actual Asignatura Estructura de Datos.

## 2.4 Posibles soluciones

Para resolver la problemática planteada se ha investigado sobre las diferentes herramientas con las cuales se podría desarrollar un sistema de ambiente educativo virtual.

De las herramientas encontradas, se ha seleccionado a las cuatro más relevantes en términos de popularidad (1) y de la información disponible de cada una. Estas herramientas son:

### 2.4.1 Moodle

Es una aplicación web, este tipo de plataformas se usa para crear un tipo de ambiente educativo virtual, se ejecuta sin problemas en sistemas que soportan PHP.

El objetivo de Moodle es facilitar a los educadores las mejores herramientas para gestionar y promover el aprendizaje. Según lo descrito en (2; 3).

Entre estas herramientas se encuentran:

- La lista de cursos muestra descripciones de cada uno de los cursos que hay en el servidor, incluyendo la posibilidad de acceder como invitado.
- La mayoría de las áreas de introducción de texto (materiales, mensajes de los foros, entradas de los diarios, etc.) pueden ser editadas en forma sencilla, como cualquier editor de texto de Windows.
- En la administración de cursos el profesor tiene control total sobre todas las opciones de un curso.
- Ofrece una serie de actividades flexibles para los cursos: foros, diarios, cuestionarios, materiales, consultas, encuestas y tareas. En la página principal del curso se pueden presentar los cambios ocurridos desde la última vez que el usuario entró en el curso, lo que ayuda a crear una sensación de comunidad.
- Registro y seguimiento completo de los accesos del usuario. Se dispone de informes de actividad de cada alumno, con gráficos y detalles sobre su paso por cada módulo (último acceso, número de veces que lo ha leído) así como también

de una detallada "historia" de la participación de cada alumno, incluyendo mensajes enviados, entradas en el diario, etc. en una sola página.

- Integración del correo. Pueden enviarse por correo electrónico copias de los mensajes enviados a un foro, los comentarios de los profesores, etc. en formato HTML o de texto.
- Puede funcionar en cualquier computador en el que pueda correr PHP, y soporta varios tipos de bases de datos (en especial MySQL). Es fácil de instalar en casi cualquier plataforma que soporte PHP. Sólo requiere que exista una base de datos (que puede compartir).
- Apropiaada para el cien por ciento de las clases en línea, así como también para complementar el aprendizaje presencial.
- Tiene una interfaz de navegador de tecnología sencilla, ligera, eficiente, y compatible.
- La lista de cursos muestra descripciones de cada uno de los cursos que hay en el servidor.
- Se ha puesto énfasis en una seguridad sólida en toda la plataforma. Todos los formularios son revisados, las cookies encriptadas, etc. El sitio es administrado por un usuario administrador, definido durante la instalación.
- Los "temas" permiten al administrador personalizar los colores del sitio, la tipografía, presentación, etc., para ajustarse a sus necesidades.
- Pueden añadirse nuevos módulos de actividades a los ya instalados en Moodle.
- Los paquetes de idiomas permiten una localización completa de cualquier idioma. Estos paquetes pueden editarse usando un editor integrado. Actualmente hay paquetes de idiomas para 34 clases de idiomas: árabe, catalán, chino (simplificado y tradicional), checo, danés, holandés, inglés (versiones de Reino Unido y de los E.E.U.U.), finlandés, francés (versiones de Francia y de Canadá), alemán, griego, húngaro, indonesio, italiano, japonés, noruego, portugués (Portugal y el Brasil), rumano, ruso, eslovaco, español (España, México, la Argentina y versiones del Caribe), sueco, tailandés y turco.
- Soporta un rango de mecanismos de autenticación a través de módulos de autenticación, que permiten una integración sencilla con los sistemas existentes.
- Soporta método estándar de alta por correo electrónico: los alumnos pueden crear sus propias cuentas de acceso. La dirección de correo electrónico se verifica mediante confirmación.

- Método LDAP (Lightweight Directory Access Protocol): las cuentas de acceso pueden verificarse en un servidor LDAP. El administrador puede especificar qué campos usar.
- IMAP (Internet Message Access Protocol), POP3 (Post Office Protocol 3), NNTP (Network News Transfer Protocol): las cuentas de acceso se verifican contra un servidor de correo o de noticias (news). Soporta los certificados SSL (Secure Socker Layer) y TLS (Transport Layer Security lookups).
- Base de datos externa: cualquier base de datos que contenga al menos dos campos puede usarse como fuente externa de autenticación.
- Cada persona necesita sólo una cuenta para todo el servidor. Por otra parte, cada cuenta puede tener diferentes tipos de acceso.
- Una cuenta de administrador controla la creación de cursos y determina los profesores, asignando usuarios a los cursos.
- Seguridad: los profesores pueden añadir una "clave de acceso" para sus cursos, con el fin de impedir el acceso de quienes no sean sus alumnos. Pueden transmitir esta clave personalmente o a través del correo electrónico personal, etc.
- Los profesores pueden dar de baja a los alumnos manualmente si lo desean, aunque también existe una forma automática de dar de baja a los alumnos que permanezcan inactivos durante un determinado período de tiempo (establecido por el administrador).
- Se permite a los alumnos crear un perfil en línea de sí mismos, incluyendo fotos, descripción, etc. De ser necesario, pueden esconderse las direcciones de correo electrónico.
- Cada usuario puede especificar su propia zona horaria, y todas las fechas marcadas en Moodle se traducirán a esa zona horaria (las fechas de escritura de mensajes, de entrega de tareas, etc.).
- Cada usuario puede elegir el idioma que se usará en la interfaz de Moodle (inglés, francés, alemán, español, portugués, etc.).

#### 2.4.2 Plataforma ILIAS

ILIAS proviene de *Integrated Learning, Information and Cooperation System* (4), es decir, Sistema Integrado de Cooperación, Información y Aprendizaje. Es un tipo de plataformas

educativa la cual facilita la enseñanza. Fue creada para reducir costos de las nuevas tecnologías en la educación, y siempre considerando las ideas de los usuarios de enseñanza. Además ilias es un software libre de código abierto por lo que se puede utilizarse sin restricciones.

Dentro de sus principales ventajas se encuentran:

- Cuenta con un sistema de control de acceso basado en roles, la cual soporta múltiples clientes.
- Se enfoca en el estudiante ya que atiende principalmente sus necesidades.
- Rompe los límites de espacio y tiempo.
- El alumno es quien marca su propio ritmo.
- Ofrece herramientas de comunicación síncrona y asíncrona.
- Permite actualizaciones constantes y económicas.
- Es gratuita y permite trabajar simultáneamente entre sistemas y software homogéneos.
- Permite la creación de test y encuestas.

Así como se mencionaron las ventajas de esta plataforma, también se pueden ver las desventajas que existen de ella y algunas de estas son:

- Presenta problemas de derecho de autor, de seguridad y autenticación.
- Requiere personal técnico de apoyo.
- Problemas con el ancho de banda.
- No cuenta con un lenguaje matemático adecuado, el procesador de texto no cuenta con tiene todas las herramientas necesarias.

### 2.4.3 Plataforma FLE3

FLE3 proviene de Future Learning Environment y el 3 se refiere a la cantidad de veces que se realizó el código desde cero (5). Es un ambiente de aprendizaje basado en la web, es decir es un software de servidor para un equipo colaborativo de aprendizaje. Esta plataforma

de aprendizaje es desarrollado en código abierto. FLE3 es diseñada para apoyar el trabajo en grupo y el aprendizaje centralizado.

Entre las ventajas que se puede encontrar de esta plataforma están:

- Se encuentra disponible en muchos idiomas.
- Tiene herramientas para facilitar la comunicación.
- Ofrece herramientas para administrar a los usuarios y cursos.

Y entre sus desventajas o mejoras se encuentran:

- Al ser un software de servidor es necesario tener un servicio propio lo que involucra mayor gasto de recursos.
- No existe gran interacción con los recursos disponibles como sería corrección automática.

#### 2.4.4 Claroline

Se basa en una comunidad global de usuarios y desarrolladores, Iniciado por UCLouvain (Bélgica) en 2001, el proyecto está dirigido por el Consorcio Claroline que reúne a instituciones de varios países dentro de una organización internacional sin fines de lucro, información proporcionada en su página web (6) .

Es un software de código abierto, en el que se puede implementar una plataforma para el aprendizaje, a través de la web lo que permite a los profesores crear y administrar cursos online, dispone de una administración sencilla, en la que se puede compartir herramientas con un grupo de alumnos, complementando las clases.

Esta herramienta permite:

- Transmitir informaciones específicas sobre un curso.
- Organizar las informaciones por secciones: competencias requeridas, objetivos, soportes.

- Rellenar las secciones predeterminadas o crear sus propias secciones.
- Publicar documentos (en todos los formatos) y archivos accesibles a todos los usuarios.
- Crear directorios y subdirectorios para agrupar archivos.
- Crear hyperlinks.
- Diseñar en línea sus propias páginas html (editor WYSIWYG).
- Crear varios grupos con los estudiantes registrados en el espacio de curso.
- Precisar los criterios de registro.
- Disponer de herramientas propias a cada grupo: espacio de discusión, zona de descarga de documentos, foros, wiki.
- Compartir un espacio público de discusión, partido en temas y subtemas.
- Organizar intercambios de ideas, opiniones y informaciones (conversación asíncrona).
- Consultar asuntos ya evocados.
- Permitir a los usuarios que encuentren rápidamente la información deseada.
- Crear ejercicios en línea en forma de una lista de preguntas
- Elaborar diferentes tipos de preguntas (elección múltiple, verdadero o falso, texto que completar).
- Correspondencia
- Conservar los resultados obtenidos por los usuarios.
- Seguir la progresión en los ejercicios vía las estadísticas del curso.
- Publicar un anuncio que se fijará en la página de inicio del curso
- Enviar un anuncio por correo electrónico a un usuario o a un grupo de usuarios
- Crear un link hacia un otro componente del curso
- Publicar archivos interesantes para los otros miembros del curso
- Someter un trabajo pedido por el instructor
- Entregar trabajos de grupo
- Proponer una corrección automática
- Permitir diferentes usos según la configuración establecida por el instructor
- Elaborar o completar en grupo un trabajo a distancia sobre un solo y mismo soporte
- Seguir el historial de las modificaciones
- Ganar en tiempo y en eficacia en el trabajo de grupo

En Claroline la seguridad no es garantizada, sino que se depende de muchos otros aspectos como la configuración hardware y software de clientes y servidores, el servicio Web, los firewalls, el canal de comunicación, el tipo de información asegurada, la configuración de los motores de bases de datos, e incluso las propias estrategias de administración de los servidores y las plataformas instaladas. Información presentada en el estudio (3).

## 2.5 Definir Solución

Para escoger una plataforma se ha completado la siguiente tabla de decisión de acuerdo a los criterios presentados en (7).

Criterios	Moodle	ILIAS	FLE3	Claroline
<b>Herramientas del Profesor</b>				
1. Herramientas Autor	X	X	X	X
2. Aviso Notas	X	X		X
3. Calendario	X			X
<b>4. Seguimiento de los alumnos:</b>				
Desempeño académico.	X	X	X	X
Desempeño en el Aprendizaje.	X	X		X
<b>5. Armado de grupos Colaborativos</b>				
Elegidos por el docente	X	X		X
Dinámicos a lo largo del curso	X	X		X
Asignación de roles.	X	X		X
<b>Herramientas de Comunicación</b>				
1. Chat	X	X	X	X
2. Email	X	X		X
3. Foros de discusión	X	X		X
4. Whiteboard	X			
5. Videoconferencia	X			X
<b>Herramientas del Alumno</b>				
1. Notas	X	X		X

<b>2. Portafolio</b>	X	X		
<b>3. Trabajo Individual</b>	X	X	X	X
<b>4. Repositorio de documentación</b>	X	X	X	X
<b>5. Consulta de situación académica</b>	X	X		X
<b>6. Ambiente colaborativo de Trabajo en grupo basado en la interacción.</b>	X	X	X	X
<b><i>Especificación Técnica</i></b>				
<b>1. Requerimientos de Cliente Servidor.</b>			X	
<b>2. Código Abierto.</b>	X	X	X	X
<b>3. Licencia</b>				
<b><i>Herramientas de Administración</i></b>				
<b>1. Autenticación (Profesor, alumno, Administrador).</b>	X	X	X	X
<b>2. Administración de la Plataforma.</b>	X	X		X

*Tabla : Tabla decisión Plataforma*

De acuerdo a la investigación realizada la mejor herramienta para apoyar a la asignatura estructuras de datos sería Moodle ya que según antecedentes el departamento de sistemas de información cuenta con una plataforma desarrollada bajo esta aplicación web, conocida como Plataforma Virtual de Aprendizaje (PVA) (8) por lo que los alumnos y profesores ya se encuentran familiarizados con esto. Además Moodle es muy superior en la cantidad de recursos y configuraciones que se pueden realizar.

En el ámbito de la interfaz tiene una mayor variedad de diseño y soporta una gran cantidad de base de datos como en este caso se necesita contar con una gran cantidad de problemas y soluciones almacenados.

La mayoría de las plataformas analizadas dejan la seguridad a cargo del sistema operativo de los servidores de donde se aloja la aplicación, los servidores web que soportan el proceso, los motores de base de datos utilizados y principalmente en los administradores para asegurar la información y soportar los procesos, en cambio Moodle ofrece seguridad, confiable y robusta.

Luego de determinar que la plataforma a utilizar es Moodle, se realiza una extensa investigación sobre las herramientas que permiten la compilación de un código fuente escrito en lenguaje C de manera online. Algunas de las herramientas que permiten realizar esta tarea son:

- CodePad: Es una plataforma web que soporta distintos lenguajes como C, C++, D entre otros, generando una URL que se puede utilizar para compartir por chat o correo electrónico (9).
- IDEone: Es un compilador y depurador en línea que permite compilar y ejecutar código en más de 40 lenguajes de programación, además se encuentra disponible como un servicio web (10).
- OnlineJudge: Es un plugins para moodle, el que está diseñado para cursos relacionados con la programación. Se puede evaluar automáticamente código fuente probándolos contra los casos de prueba personalizables, una descripción completa del plugins se encuentra en (11).

Una vez analizada las herramientas disponible, se optó por utilizar OnlineJudge, debido a que está especialmente diseñado para moodle, lo que permite agregarlo como cualquier otro recurso estándar de este. Además utiliza IDEone al momento de compilar y ejecutar el código ingresado por un alumno.

Una desventaja de moodle es su rigidez, lo que complica el desarrollo de nuevos módulos para la plataforma o el modo de asignación de tareas, lo que causa un problema en cuanto a una diferenciación de tareas por alumnos.

Por esta razón la solución definitiva para desarrollar este proyecto es un sistema externo que se conecte con moodle, específicamente con la Plataforma Virtual de Aprendizaje (PVA), donde se encuentran los recursos relacionados a la asignatura estructura de datos. La herramienta elegida para utilizar en este sistema es IDEone, debido a que se puede utilizar como un servicio web dentro de la aplicación.

Se ha denominado a este proyecto como SAED, sistema de apoyo a la asignatura estructuras de datos.

---

## 3 DEFINICIÓN PROYECTO

---

### 3.1 Objetivos del proyecto

#### 3.1.1 Objetivo General

- Diseñar y construir una plataforma de Apoyo a la Docencia de la Asignatura Estructura de datos.

#### 3.1.2 Objetivos Específicos

- Desarrollar un breve Estudio del arte del desarrollo de plataformas educativas similares, sus aplicaciones y características técnicas con el fin de seleccionar las herramientas de desarrollo y capturar ideas de utilidad para el desarrollo del proyecto.
- Probar el software desarrollado, siguiendo estándares establecidos en la ingeniería del software.
- Desarrollar la documentación asociada al proyecto (Informe del proyecto, manuales de usuarios).

### 3.2 Ambiente de Ingeniería de Software

- La metodología aplicada al desarrollo de este proyecto, será entrega evolutiva, la cual es iterativa y por lo que beneficia al desarrollo, ya que a través de este ciclo de vida de software se puede mantener un mejor control de los riesgos asociados al desarrollo de un sistema, considerando como factor relevante la inexperiencia de los alumnos tesisistas con respecto a este tipo de sistemas.

Otro punto a considerar a favor de la metodología a utilizar es la temprana generación de versiones de software, permitiendo una retroalimentación

constructiva y realizar modificaciones en cualquier nivel en base a las necesidades del cliente, para así alcanzar un mejor desarrollo del sistema.

- Las técnicas que se utilizan en el desarrollo de software son la utilización de la plantilla para la creación del documento, la técnica de Casos de uso, reuniones constantes con profesora Sra. Brunny Troncoso como guía, El modelo relación-entidad, árbol de descomposición funcional, diagrama para representar la jerarquía del menú, esquema de navegación.
- La notación que se utiliza en Casos de Usos es UML 1.0, en el modelo entidad relación se encuentra la notación MPM1999. La notación de la plantilla utilizada se basa en IEEE Software requirements Specifications Std 830-1998, además de una adaptación basada en IEEE Software Test Documentation Std 829-1998.
- La documentación que se realiza en la toma de requerimientos y reuniones con profesora, se mantendrá registros impresos, para evitar un cambio de funcionalidad o especificaciones dadas con anterioridad y así no ocasionar conflicto, además de tener un respaldo de todo lo realizado.
- Para la documentación de la base de datos se utilizan modelos de entidad – relación, ya que es el más apropiado para el desarrollo de la base de datos.
- Las herramientas que se utilizan en el desarrollo de este proyecto en el ámbito de la implementación son Moodle, PHP, MySQL. Para el apoyo de diagramas y diseño SmartDraw 2010.

### 3.3 Definiciones, Siglas y Abreviaciones

Durante este documento se utilizan estas abreviaciones:

- ICI: Ingeniería Civil en Informática
- IECI: Ingeniería Ejecución en Computación e Informática
- PVA: Plataforma Virtual de Aprendizaje
- 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.
- MySQL: Es un sistema de gestión de base de datos relacional.
- Pluggin: Es un modulo que le añade características a un servicio específico o aun sistema.

---

## 4 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

---

### 4.1 Alcances

Como es un recurso exclusivo para la asignatura estructura de datos, el software tiene las siguientes particularidades:

- Mantener el registro un repositorio on-line de distintos problemas, que permitirá la creación de un cuaderno de ejercicios distintos para cada alumno.
- Dar soporte de corrección automática de los problemas, informando a cada estudiante sobre la calidad de la solución propuesta, y guardarla en un “cuaderno resuelto”.
- Permite generar reportes de desempeño académico de los estudiantes, tanto de manera individual como colectiva, que los profesores podrán consultar cuando estimen conveniente.

Además a nivel de funcionalidad se tiene:

- La existencia de una jerarquía dentro del software, lo cual es beneficioso para mantener la seguridad de la información.

## 4.2 Objetivo del software

### 4.2.1 Objetivos Generales

- Desarrollar un repositorio on-line de distintos problemas, que facilite a los profesores de la asignatura ingresar y compartir nuevos problemas, catalogados según los temas tratados.
- Dar soporte a la corrección automática de los problemas. Para ello, cada problema debe llevar asociado las pautas que hagan posible su corrección automática (varios test de pruebas con entradas y salidas esperadas para cada una de ellas).
- Ofrecer en lo posible, a cada alumno, un cuaderno de problemas distinto para su desarrollo; el cuaderno estará organizado por temas en los que se incorporarán diferentes problemas al momento de su generación.
- Registrar el número total de las soluciones enviadas por el alumno, además de otra información relevante para el curso, pudiendo esta ser consultada por los profesores de la asignatura.

### 4.2.2 Objetivos Específicos

- Recibir, desde cada estudiante, las soluciones propuestas para los problemas asignados. Estas respuestas deben ir formando parte del cuaderno resuelto de cada estudiante.
- Informar a cada estudiante de la evaluación (automática) de sus soluciones, brindándole la oportunidad de volver a ingresar una respuesta hasta obtener una que sea correcta. En caso de ser soluciones erróneas se facilitarán las indicaciones oportunas para su corrección.

- Registrar el número total de las soluciones enviadas por el alumno, además de otra información relevante para el curso, pudiendo esta ser consultada por los profesores de la asignatura.

### 4.3 Descripción Global del Producto

#### 4.3.1 Interfaz de usuario

La interfaz de usuario debe ser similar a la actual plataforma utilizada en la Facultad de Ciencias Empresariales por lo que tendrá:

- Título.
- Banner.
- Menú de navegación.
- Un espacio libre.
- La información se debe desplegar dentro de la misma página cuando sea posible.
- Sidebar.
- Pie de página.
- La interfaz tiene un aspecto como se muestra en la imagen (Ver Imagen 1)



Imagen : Diseño Interfaz

#### 4.3.2 Interfaz De Hardware

Para la Universidad no es necesario comprar nuevos dispositivos para implementar el software ya que en la Facultad de Ciencias Empresariales ya se trabaja con Moodle el cual se encuentra ubicado en un servidor X, por esta razón no hay

inconvenientes en cuanto a servidores o a la configuración de los equipos de los profesores.

Para los alumnos solo se debe contar con el hardware básico, mouse, teclado, monitor, ningún hardware que necesite ser configurado para utilizar el sistema.

### 4.3.3 Interfaz Software

Para la implementación del software no es necesario agregar un software específico esto tanto en servidores de la universidad como en los computadores de los usuarios, tan solo que cuente con el software básico tales como Windows y algún navegador para acceder a internet.

El sistema se relaciona con otros paquetes de software los que se detallan a continuación:

- Plataforma Virtual de aprendizaje.
  - Abreviación: PVA.
  - Versión: Moodle 2.3
  
- IDEone
  - Abreviación: IDE
  - Versión: 1.1.2

### 4.3.4 Interfaces de comunicación

Las interfaces para desarrollar el proceso de comunicación es el protocolo:

HTTP: Es el protocolo más utilizado en internet, ya que su principal función es la transferencia de archivos.

SOAP: (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML (13).

HTML: Es la base para la creación de páginas web. Es usado para describir la estructura y el contenido en forma de texto.

#### 4.4 Requerimientos Específicos

##### 4.4.1 Requerimientos Funcionales del sistema

Id	Nombre	Descripción
1	Autenticar usuario	El sistema solo puede ser accedido por usuarios registrados en el curso estructura de datos de la plataforma moodle PVA.
2	Ingresar Problemas.	Registrar los datos de un problema en el sistema, junto a sus casos de prueba asociado.
3	Consultar Problema.	Realizar una consulta a la base de datos por el problema en cuestión.
4	Modificar Problema.	Realizar cambios en un algún dato de un problema.
5	Crear cuaderno de ejercicios.	Asocia a un alumno con diferentes problemas de diferentes tópicos.
6	Modificar Cuaderno.	Realizar cambios en el sistema a un determinado cuaderno.
7	Eliminar Cuaderno	Borrar un cuaderno indicado.
8	Consultar Cuaderno.	Ejecutar consulta al sistema por un cuaderno específico.
9	Resolver problema.	El alumno selecciona un determinado problema de su cuaderno y propone una solución a dicho problema.
10	Evaluar calidad de solución	El Sistema realiza una consulta a la base de datos con los parámetros ingresados por el alumno y devuelve el nivel de compatibilidad con los parámetros almacenados en la base de datos.
11	Emitir informes	El sistema debe generar 3 tipos de informes.
12	Extraer Usuarios	Obtener los datos de los usuarios de moodle, tanto alumnos como profesores, y almacenarlos en la base de datos del sistema.

13	Eliminar Usuarios	Quitar del sistema los usuarios almacenados en la base de datos.
----	----------------------	--

*Tabla : Requerimientos Funcionales.*

#### 4.4.2 Interfaces externas de entrada

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem
DE_01	Datos de Usuario	NOMBRE, ID, CURSO
DE_02	Datos de Problema	TIPO DE EJERCICIO, ENUNCIADO, DESCRIPCION, PROTOTIPO DE FUNCION
DE_03	Datos Cuaderno	INTENTOS, SOLUCION, ALUMNO, PROBLEMA
DE_04	Datos Solución	DATOS

*Tabla : Interfaces externas de entrada.*

#### 4.4.3 Interfaces externas de Salida

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem	Medio Salida
IDA_01	Informe Desempeño Alumno	NOMBRE, NUMERO DE EJERCICIOS RESUELTOS, INTENTOS POR CADA EJERCICIO, CALIDAD DE SOLUCION FINAL POR EJERCICIO.	Archivo PDF Pantalla
IDC_01	Informe de Desempeño Curso	NOMBRE DE LOS ALUMNOS, PORCENTAJE DE EJERCICIOS RESUELTOS, PROMEDIO DE CALIDAD DE SOLUCIONES, TOPICO DE ASIGNATURA CON MEJOR Y PEOR DESEMPEÑO POR EL CURSO.	Archivo PDF Pantalla

*Tabla : Interfaces externas de Salida.*

#### 4.4.4 Atributos del producto

- **USABILIDAD- OPERABILIDAD:** el sistema tiene una interfaz amigable para el usuario con un diseño similar a PVA, lo que facilita el uso mediante la existencia de link, campos de textos y opciones de menú que están bien identificadas en este. El sistema muestra mensajes de errores en forma clara, mediante una ventana de alerta sobre la pantalla en la cual se trabaja, cuando el usuario realiza una operación de forma incorrecta.
- **EFICIENCIA- TIEMPO DE EJECUCIÓN/RESPUESTA:** El tiempo de respuesta del sistema dependerá de la conexión a internet disponible, así como de características propias de cada equipo, pero aún así no debería presentar un tiempo de respuesta superior a 30 segundos, con menos de 15 usuarios conectados realizando la misma petición.
- **FUNCIONALIDAD-SEGURIDAD:** El control de acceso lo maneja PVA con quién se conecta el sistema, por lo que el control de acceso queda en manos de está.
- **DESEMPEÑO Y RENDIMIENTO:** Para ver el rendimiento del Software se ingresan códigos fuente y se imprimen los informes para los profesores. Al lograr resultados se comprueba que el rendimiento es óptimo. Al obtener dicho objetivo se puede afirmar que el sistema tiene un excelente desempeño como apoyo a la Asignatura Estructura de Datos.
- **PORTABILIDAD:** El Software es compatible tanto a nivel de servidor como de clientes, con cualquier Sistema Operativo, pero teniendo una menor calidad de visualización con el navegador Internet Explorer.
- **SEGURIDAD:** Los datos ingresados al sistema son controlados por medio del software, ya que el acceso es solo a través de las cuentas de usuario moodle, sin una cuenta no se podrá acceder a ningún tipo de información. Cada usuario registrado tiene distintos privilegios para mantener mayor confiabilidad de la información.

---

## 5 FACTIBILIDAD

---

### 5.1 Factibilidad técnica.

Los dispositivos y el equipamiento con el que cuenta el Departamento de Sistemas de Información permiten el correcto uso del sistema de apoyo a la asignatura Estructuras de Datos, sin necesidad de adquirir nuevas licencias ni software específico para la implementación de este.

El departamento de Sistemas de información cuenta con determinados recursos de software en su servidor:

- Sistema operativo Linux debían 6.
- Moodle 2.3
- PHP 5
- MySQL

De acuerdo a los recursos mencionados con lo que cuenta la Facultad de Ciencias Empresariales en su servidor, es que se considera que el funcionamiento del software desarrollado es óptimo bajo los parámetros dados.

El hardware con el que la Facultad cuenta es el que corresponde a los dispositivos necesarios para el software de apoyo a la asignatura Estructura de datos. Sin la necesidad de adquirir nuevo equipamiento computacional.

La facultad cuenta con los siguientes recursos de hardware en su servidor:

- 4 gigas de RAM
- 2 Terabyte de Disco Duro

Además se cuenta con los conocimientos y habilidades para aplicar las técnicas y herramientas que se utilizan en el desarrollo del software, por ende el equipo de desarrollo no requiere capacitación para la construcción de este proyecto, tan solo el uso de un manual del

lenguaje de programación PHP en el cual se implementa el software y un manual de CSS para un buen diseño de la página.

En conclusión la factibilidad técnica es viable para la Facultad de Ciencias Empresariales, de acuerdo a las características a nivel de software como de hardware ya mencionadas.

## **5.2 Factibilidad operativa.**

Uno de los mayores impactos que se produce al desarrollar este proyecto es facilitar el control sobre el avance de los alumnos en los diferentes tópicos de la asignatura Estructura de Datos, permitiendo a los docentes reforzar los contenidos más complicados para el curso. Además de entregarle información con respecto al tiempo que le dedica cada alumno a la asignatura. Se considera que también se disminuya el tiempo que el profesor le dedica a la corrección de las tareas en clases.

Por otra parte los alumnos pueden ejercitar los contenidos vistos en clases, ya que el sistema le asigna un cuaderno de ejercicios para resolver, donde el alumno obtiene una respuesta inmediata para conocer su desempeño en la solución propuesta, y tienen la posibilidad de intentar de resolver el ejercicio más de un vez y en todos los intentos obtener resultados de la solución propuesta.

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.

Debido a los impactos beneficiosos en la disminución tiempo y en el seguimiento de los alumnos que se producen con el desarrollo de este software, se concluye que la factibilidad operativa es viable para la Facultad.

### 5.3 Factibilidad económica.

Los beneficios esperados al desarrollar el proyecto son automatizar el proceso de ejercitación en los diferentes tópicos de los alumnos y la generación de informes para los profesores. Más aun con las ventajas evidentes que esto significa, resulta cuantificar los beneficios económicos por alumnos y docentes del curso.

Como la Facultad cuenta con todas las herramientas necesarias para el uso del software no se necesitan recursos económicos para la adquisición de nuevas herramientas de implementación ni de desarrollo ya que se utiliza software libre.

Al momento de cuantificar el valor final del software se debe considerar la mano de obra, y los gastos asociado a su desarrollo, principalmente la energía eléctrica la cual se detalla a continuación.

Equipo	Cantidad	Consumo de Energía	Tiempo de Consumo	Total Consumo
Computadores	2	65	6	18.720
Router	1	20	6	5.760
Luz	2	10	6	4.500
Total				28.980
Total KW				28,9
Valor(KWh):				88
Total en pesos				2.543

*Tabla Costo Energía Eléctrica*

Para tener una idea de cómo se podría cuantificar este software se consideran las horas destinadas al desarrollo por el equipo las cuales son 756 horas y el valor promedio de una hora hombre a \$4.000. Por lo tanto el software se podría cuantificar en \$3.024.000 solo en mano de obra, además se deben considerar los gastos asociados a este, pero al ser un proyecto de título la mano de obra del proyecto es cero.

Recurso	Total
Recurso Humano	\$3.024.000
Energía Eléctrica	\$ 2.543
<b>Total</b>	<b>\$3.026.543</b>

*Tabla Presupuesto del Proyecto*

De acuerdo a los beneficios otorgados a la facultad y los dispositivos que esta cuenta, indica que existe un ahorro de tiempo al implementar el software y así reducir gastos de tiempo en los docentes y alumnos.

En conclusión de acuerdo a los beneficios que otorga el desarrollo del proyecto y considerando que este proyecto ayuda a los alumnos a disminuir la tasa de reprobación de la asignatura, lo que conlleva a que no se atrasen en la carrera, lo cual significa un ahorro para ellos lo que produce un beneficio considerable es que se dice que la factibilidad económica de la Facultad es viable.

## 5.4 Conclusión de la factibilidad

Mediante el análisis realizado se demuestra las necesidades que existen en la asignatura de crear un sistema que les facilite la comunicación con los alumnos para poder realizar la asignación de ejercicios y a la vez poder llevar un monitoreo del avance de estos mismo en la asignatura.

Algunos de los beneficios esperados con el desarrollo de este proyecto son:

- Elaboración de un cuaderno de ejercicios por alumno.
- Corrección automática de los ejercicios a los alumnos.
- Monitorear el avance de los alumnos en los ejercicios.
- Disminuir el tiempo destinado por los profesores a la corrección de los ejercicios propuestos.
- Disminuir la tasa de reprobación de los alumnos en la asignatura Estructuras de datos.
- Detectar los tópicos que más dificultades tienen los alumnos para poder reforzarlos.

En base al análisis detallado de cada punto de la factibilidad es que se puede concluir que técnicamente la facultad cuenta con los dispositivos adecuados tanto a nivel de software como de hardware para la implementación del software y sin tener la necesidad de invertir en nuevos equipos.

En el ámbito operativo permite mejorar el manejo de la información y automatizar los procesos de la asignatura facilitando tanto a profesores como a alumnos el trabajo durante el semestre. Y finalmente en el ámbito económico se concluye que la facultad no necesita costear el software, y el cual otorga un beneficio significativo para los estudiantes.

De acuerdo a todos los análisis ya mencionados se concluye que el proyecto es factible de realizar, ya que no tiene costos de implementación y desarrollo, y entrega beneficios importantes a todos los usuarios que utilizan el sistema, colaborando de esta en el desarrollado de las carreras que imparten esta asignatura.

---

## 6 ANÁLISIS

---

### 6.1 Diagrama de casos de uso

#### 6.1.1 Actores

➤ Profesor:

Es el encargado de impartir la asignatura Estructura de Datos para cualquiera de las dos carreras ICI o IECI, dentro del sistema es uno de los usuarios con mayor privilegio ya que puede ingresar, modificar y consultar todos los datos que pueden ser manejados, además de extraer los informes de rendimientos.

Al ser profesionales del área informática cuentan con los conocimientos técnicos para el uso del sistema, los que son básicos.

➤ Alumno:

Son los estudiantes de las carreras ICI o IECI que inscriben la asignatura Estructura de Datos, quienes pueden acceder al sistema solo para visualizar un cuaderno de ejercicios, el cual deben resolver hasta encontrar la mejor solución para cada problema.

Se requiere que para el desarrollo de estos problemas los alumnos poseen conocimientos sobre los distintos tópicos de la asignatura y el lenguaje de programación C.

➤ Web Service:

Es el encargado de compilar y ejecutar el código que los alumnos ingresan y entregar una respuesta al sistema. Utilizando la API proporcionada por IDEone.

➤ PVA:

Es la plataforma utilizada por la facultad de Ciencias Empresariales para entregar diferentes materiales a los alumnos sobre cada una de las asignaturas inscritas por cada uno.

Es la encargada de autenticar a los usuarios del sistema.

➤ **Administrador SAED:**

Es el encargado de mantener el sistema, tanto de extraer a los alumnos y profesores de la plataforma PVA como de eliminarlos del sistema cuando ya no correspondan al semestre. Este usuario es el con mayor privilegio, ya que también tiene acceso a ingresar problemas y de todos los privilegios de un profesor.

Para mantener el sistema el administrador SAED debe contar con los conocimientos del funcionamiento del software, por lo que requiere tener conocimientos técnicos avanzados.

### 6.1.2 Casos de Uso y descripción

Como se aprecia en el diagrama de casos de uso, se visualizan las interacciones de distintos usuarios que tiene el sistema.

Los actores Alumnos y Profesores interactúan con el sistema a través de un computador conectado a internet, en donde deben ingresar a la plataforma PVA e ingresar al curso Estructura de Datos, para tener acceso al sistema. PVA es otro sistema el que está encargado de autenticar a los usuarios que se conectan a él, y de asignarle un rol ya sea profesor, alumno o administrador.

El Actor web service IDEone es un servicio web encargado de compilar y ejecutar códigos fuentes en lenguaje, ingresados por los alumnos como solución a los problemas propuestos por los profesores.

### Diagramas de Caso de Uso

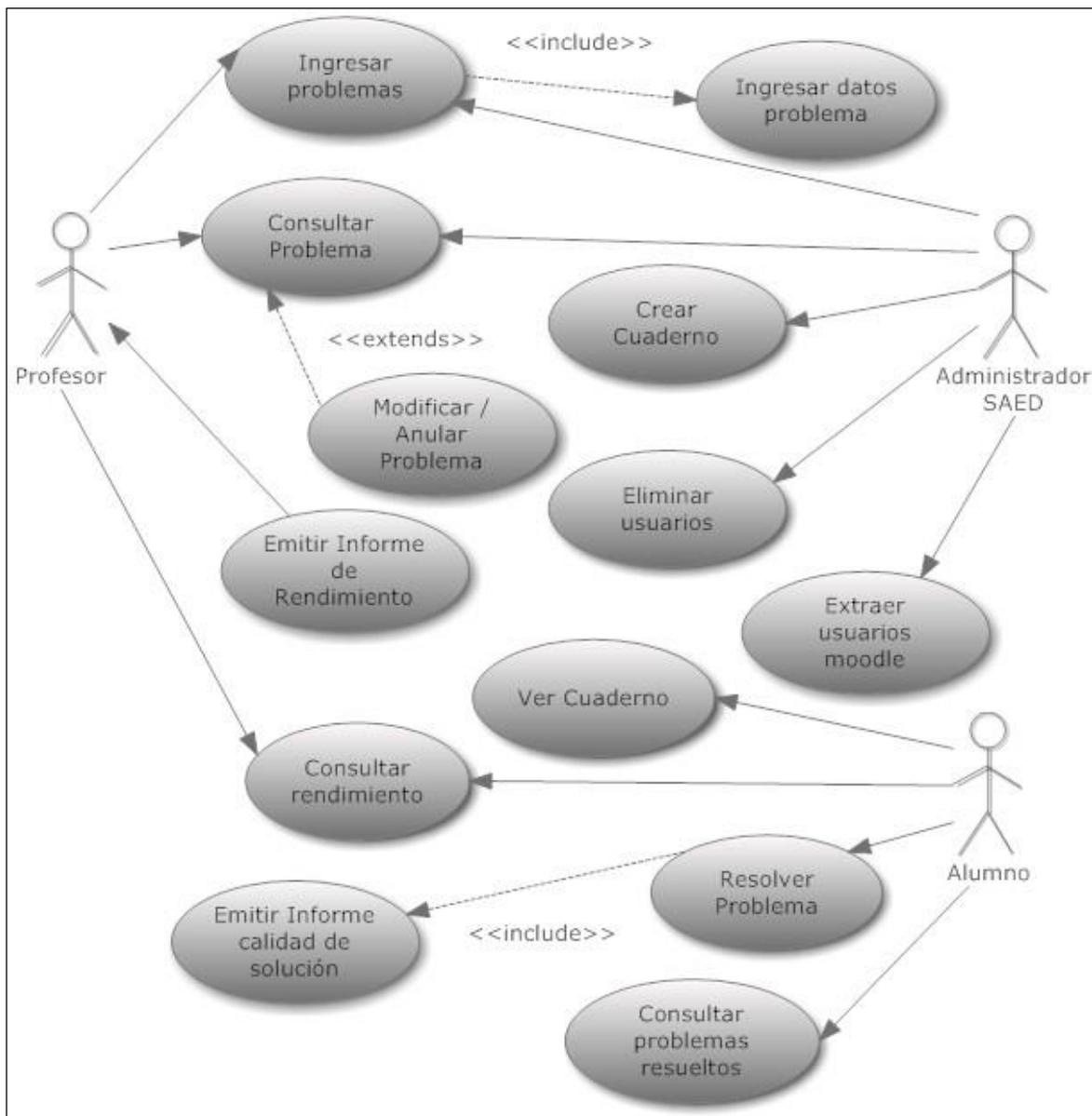


Diagrama : Casos de uso General

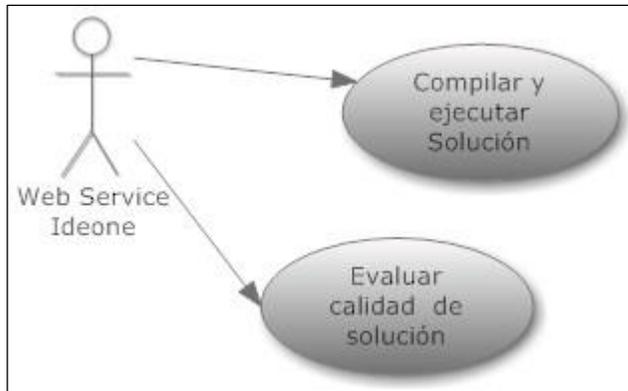


Diagrama : "Caso de Uso del servicio Web"



Diagrama : "Caso de Uso de PVA"

### 6.1.3 Especificación de los Casos de Uso

#### 6.1.3.1 Caso de Uso: <Autenticar usuario>

- Descripción: La plataforma PVA devuelve al sistema el usuario autenticado.
- Pre-Condiciones:
  - El usuario debe iniciar sesión en PVA.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1.- Se conecta a moodle para obtener datos de usuario.
	2.- Muestra perfil según autenticación de usuario.

- Flujo de Eventos Alternativo:

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

- Post-Condiciones:
  - Usuario autenticado en sistema.

#### 6.1.3.2 Caso de uso <Ingresar Problemas>

- Descripción: Para que los alumnos puedan acceder a un cuaderno con ejercicios, se deben ingresar los problemas, los cuales son almacenados en la base de datos y solo los usuarios registrados en moodle como profesores y administrador pueden acceder a esta opción.
- Pre-Condiciones:
  - Estar logeado como Profesor o administrador SAED.

- Debe existir por lo menos un usuario en el sistema ya sea profesor o administrador.

➤ Flujo de Eventos Básicos:

Al actor	El sistema
1.- Ingresar el problema y los atributos pertenecientes a este.	2.- Validar los datos del problema que se están ingresando. 3.- Verificar los datos ingresados del problema. 4.- Guardar en la base de datos los problemas ingresados.

➤ Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1.- Los datos ingresados no son permitidos por el sistema
4(a)	4.1.- Los datos ingresados no son guardados, por no ser permitidos o por una falla en la conexión.

➤ Post-Condiciones:

- Los problemas quedan almacenados en la base de datos.

### 6.1.3.3 Caso de uso <Consultar Problemas>

➤ Descripción:

Los usuarios que ingresen problemas también podrán consultarlos para evitar la duplicación de información de la base de datos, y así conocer que problemas se encuentren ingresados.

➤ Pre-Condiciones:

- Debe estar logeado en el sistema como profesor o administrador SAED.
- Debe existir por lo menos un problema ingresado en el sistema.

➤ Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona el tipo de problema que quiere consultar.	2.- Se conecta con la base de datos y se muestran todos los problemas de acuerdo al tipo seleccionado.

➤ Flujo de Eventos Alternativo:

Al actor	El sistema
	2.1.- Falló en la conexión, por lo que no se realizó la consulta.

➤ Post-Condiciones:

- Se muestra la respuesta a la consulta realizada.

#### 6.1.3.4 Caso de uso <Modificar/Anular Problemas>

➤ Descripción: Para poder realizar la modificación de un problema se debe consultar por la existencia de este, luego de haberlo encontrado se pueden actualizar los datos asociados al problema.

- Modificar Problemas utiliza el caso de uso Consultar Problemas para comprobar la existencia del problema a modificar.

➤ Pre-Condiciones:

- Debe existir por lo menos un problema en el sistema para modificar.
- Debe estar logeado como profesor o administrador.

➤ Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción consultar.	2.- Debe invocar al caso de uso consultar problemas.
3.- El usuario selecciona	4.- Selecciona el campo a

	modificar. 5.- Valida los datos del problema. 6.- Verificar los datos del problema. 7.- Guarda los cambios realizado en el problema en la base de datos.
--	---

➤ Flujo de Eventos Alternativo:

Al actor	El sistema
4(a)	4.1.- El campo seleccionado no se puede modificar.
5(a)	5.1.- Los datos a ingresar no son permitidos.

➤ Post-Condiciones:

- El problema queda actualizado en la base de datos.

### 6.1.3.5 Caso de uso <Ingresar datos del Problema>

➤ Descripción: Para que los alumnos puedan obtener una compilación de él código que ellos enviaron es necesario que al sistema se le ingresen casos de pruebas asociado a un problema, por lo que se requiere almacenarlos en la base de datos junto a sus atributos. Además un problema puede tener asociado a él más de un caso de prueba como también más de un prototipo de función.

- Ingresar datos del problema que va ligado con el caso de uso Ingresar Problema, ya que al ingresar un problema obligatoriamente se debe ingresar un caso de prueba y un prototipo de función obligatorio.

➤ Pre-Condiciones:

- El usuario debe tener los privilegios de profesor o administrador para poder ingresar los casos de pruebas.
- Debe existir un problema al cual esté asociado el caso ingresado.

➤ Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción de ingresar casos de prueba.	2.- Valida los datos del caso de prueba que se quiere ingresar. 3.- Verifica los datos que son ingresados del caso de prueba. 4.- Almacena la información en la base de datos de los casos de prueba ingresada.

➤ Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1.- Los datos ingresados no son permitidos por el sistema
4(a)	4.1.- Los datos ingresados no son guardados, por existencia de estos, ya que internamente consulta y envía un mensaje cuando ya existe.

➤ Post-Condiciones:

- Los casos de pruebas son almacenados en la base de datos.

### 6.1.3.6 Caso de uso <Emitir informe de rendimiento>

➤ Descripción: El sistema debe generar informes de rendimientos tanto a nivel de curso como de manera individual por alumno en el sistema.

➤ Pre-Condiciones:

- El usuario debe tener los privilegios de profesor o administrador para poder recibir los informes que emite el sistema.
- Deben existir alumnos en el sistema.
- Los alumnos deben haber desarrollado por lo menos un ejercicio del cuaderno asignado.

➤ Flujo de Eventos Básicos:

Al actor	El sistema
1.- El usuario selecciona emitir informes	2.- Muestra el menú de informes disponibles en pantalla.
3.- El usuario selecciona el tipo de informe que requiere.	4.- Genera el informe seleccionado por el usuario.
	5.- Emite el informe solicitado en formato PDF.

➤ Flujo de Eventos Alternativo:

Al actor	El sistema

➤ Post-Condiciones:

- Informe de rendimiento entregado a usuario que lo solicito.

**6.1.3.7 Caso de Uso: <Consultar rendimiento>**

- Descripción: Informa a los usuarios el rendimiento del curso o de un alumno en particular, dependiendo quien realice la consulta.
- Pre-Condiciones:
  - El usuario debe estar autenticado.
  - Debe existir un cuaderno en el sistema para el alumno a consultar.

➤ Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción de consultar rendimiento.	2.- Se conecta a la base de datos devolviendo un listado con los alumnos y porcentaje de resolución de problemas por tópicos.

➤ Flujo de Eventos Alternativo:

Al actor	El sistema
1(a).- Selecciona un alumno en particular para consultar su rendimiento.	3.- Se conecta a la base de datos, devolviendo el progreso del alumno y la calidad de las soluciones enviadas por él.

➤ Post-Condiciones:

- El usuario puede imprimir informe de rendimiento.

6.1.3.8 Caso de Uso: <Ver cuaderno>

➤ Descripción: Al resolver problemas el alumno debe seleccionar los que están disponibles para su desarrollo, para acceder al listado de problemas asignados el sistema debe realizar una consulta para obtener el cuaderno personal de ejercicios del alumno que inicio sesión.

➤ Pre-Condiciones:

- El usuario debe estar autenticado como alumno.

➤ Flujo de Eventos Básicos:

Al actor	El sistema
1.- El usuario selecciona la opción ver cuaderno.	2.- Se conecta a la base de datos y devuelve un listado de los problemas asociados al usuario.

➤ Flujo de Eventos Alternativo:

Al actor	El sistema
	2.1.- Falló la conexión por lo que no visualizará ningún problema.

- Post-Condiciones:
  - El usuario alumno puede acceder a visualizar su cuaderno de problemas.

### 6.1.3.9 Caso de Uso: <Resolver problema>

- Descripción: Muestra al alumno el enunciado de un problema con su descripción y el espacio para poder desarrollarlo.
- Pre-Condiciones:
  - El usuario debe estar autenticado como alumno.
  - Debe existir por lo menos un problema en el sistema.
  - Debe existir un cuaderno por lo menos en el sistema.
  - Debe haber consultado cuaderno.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- El usuario selecciona un problema de su cuaderno.	2.- Se conecta a la base de datos y devuelve enunciado del problema.
	3.- Muestra ambiente para resolver problema.
4.- El usuario ingresa el código fuente c.	
5.- Presiona el botón "enviar código".	

- Flujo de Eventos Alternativo:

Al actor	El sistema
4(a).- El usuario no ingresa el código.	6.- No realiza ninguna operación, ni considera esto como un intento de solución del problema.

- Post-Condiciones:
  - El alumno espera conocer calidad de su solución.

#### 6.1.3.10 Caso de Uso: <Emitir Informe Calidad de solución>

- Descripción: Informa al alumno la calidad de la solución enviada para resolver un problema determinado.
  - Este caso de uso está relacionado con resolver problema, no se puede acceder a él sin no se ha resuelto un problema.
  
- Pre-Condiciones:
  - El usuario debe estar autenticado como alumno.
  - Debe existir un cuaderno asociado al usuario.
  - Debe existir un cuaderno por lo menos en el sistema.
  - El usuario debe haber consultado cuaderno.
  - El usuario debió resolver un problema.

- Flujo de Eventos Básicos:

Al actor	El sistema
	1.- Se conecta a la base de datos y busca que existan los prototipos de funciones requeridos, para resolver el problema.
	2.- Se conecta a la base de datos para obtener resultados.
	3.- Informa al usuario la calidad de la solución enviada y registra un intento más por solucionar el problema.

- Flujo de Eventos Alternativo:

Al actor	El sistema

--	--

- Post-Condiciones:
  - El alumno tiene otro problema marcado como resuelto y con una evaluación.

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

- Descripción: Compila y ejecuta código fuente enviado por alumno.
- Pre-Condiciones:
  - El usuario debe estar autenticado como alumno.
  - Debe existir un cuaderno por lo menos en el sistema.
  - El alumno debe haber ingresado a resolver problema.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1.- Se conecta a web service ideone.
2.- Compila y ejecuta código enviado por el sistema con los respectivos casos de prueba enviados.	

- Flujo de Eventos Alternativo:

Al actor	El sistema
	1.1.- Se le informa al usuario problemas de conexión.

- Post-Condiciones:
  - Sistema listo para emitir calidad de la solución entregada.

### 6.1.3.12 Caso de Uso: <Informar calidad de solución>

- Descripción: Compila y ejecuta código fuente enviado por alumno.
- Pre-Condiciones:
  - El usuario debe estar autenticado como alumno.
  - Debe existir un cuaderno por lo menos en el sistema.
  - El alumno debe haber ingresado resolver problema.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1.- Solicita a web service ejecutar problema con distintos casos de prueba.
2.- Ejecuta código enviado por el sistema con los respectivos casos de prueba enviados y entrega salidas.	3.- Compara salidas entregadas por web service con salidas esperadas registradas en la base de datos.
	4.- Registra compatibilidad entre salidas esperadas y salidas entregadas por web service.

- Flujo de Eventos Alternativo:

Al actor	El sistema
2(a).- Informa de error al sistema.	3.1 Informa al usuario el error ocurrido.

- Post-Condiciones:
  - Registra en sistema calidad de solución de un problema determinado.

### 6.1.3.13 Caso de uso <Consultar Problemas Resueltos>

- Descripción: ayuda a los alumnos para poder visualizar los problemas que ya tienen una solución.

- Pre-Condiciones:
  - El alumno debe existir en el sistema.
  - En el sistema deben existir problemas asignado a alumnos.
  - El alumno debe haber resuelto a lo menos un problema.
  
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción consultar problemas resueltos.	2.- Se conecta a la base de datos devolviendo un listado con los problemas resueltos para el alumno que realiza la consulta.

- Flujo de Eventos Alternativo:

Al actor	El sistema
	2.1.- Falló en la conexión con la base de datos, por lo que no se muestra ningún problema.

- Post-Condiciones:
  - El alumno obtiene un listado de los problemas resueltos.

#### 6.1.3.14 Caso de uso <Extraer Usuarios moodle>

- Descripción:
 

Obtiene los usuarios de la plataforma PVA que se encuentren registrados de un determinado curso, el cual el usuario debe ingresar.
  
- Pre-Condiciones:
  - El usuario debe contar con los permisos de administrador.
  
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción extraer alumnos e ingresa el código del curso.	2.- Se realiza la conexión con moodle.
	3.- Extrae los alumnos correspondientes al curso ingresado por el usuario.
	4.- Registra en la base de datos los alumnos extraídos.

➤ Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1 Informa que el curso seleccionado no existe en el sistema.

➤ Post-Condiciones:

- Se realiza una correcta extracción de los alumnos del cursos al sistema.

### 6.1.3.15 Caso de uso <Eliminar Usuarios>

➤ Descripción:

Es una opción de mantención del sistema en la cual realiza una limpieza de los usuarios registrados en el sistema.

➤ Pre-Condiciones:

- El usuario debe contar con los permisos de administrador.
- Deben existir usuarios que ya fueron extraídos en el sistema.

➤ Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción de eliminar alumnos.	2.- Elimina los alumnos existentes en el sistema.

➤ Flujo de Eventos Alternativo:

Al actor	El sistema

➤ Post-Condiciones:

- Se eliminan de forma exitosa los usuarios del sistema.

6.1.3.16 Caso de uso <Crear Cuaderno>

➤ Descripción:

El sistema debe contar con cuadernos para cada alumno, por lo que es necesario que este la opción de crearlos, de esta forma se le asigna automáticamente a los alumnos registrados en el sistema, asignándole ejercicios de distintos tipos.

➤ Pre-Condiciones:

- El usuario debe contar con los permisos de administrador.
- Deben existir por lo menos un alumno en el sistema.
- Deben existir problemas en el sistema

➤ Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción de crear cuaderno.	2.- Crea los cuadernos automáticamente con ejercicios de cada tipo.

➤ Flujo de Eventos Alternativo:

Al actor	El sistema

➤ Post-Condiciones:

- Los cuadernos de ejercicios se encuentran creados.

## 6.2 Modelamiento de datos

Los ejercicios son registrados en el sistema por los profesores, para que sean los alumnos los que deban resolver un conjunto de problemas, registrando como el cuaderno la asignación de problema a alumno.

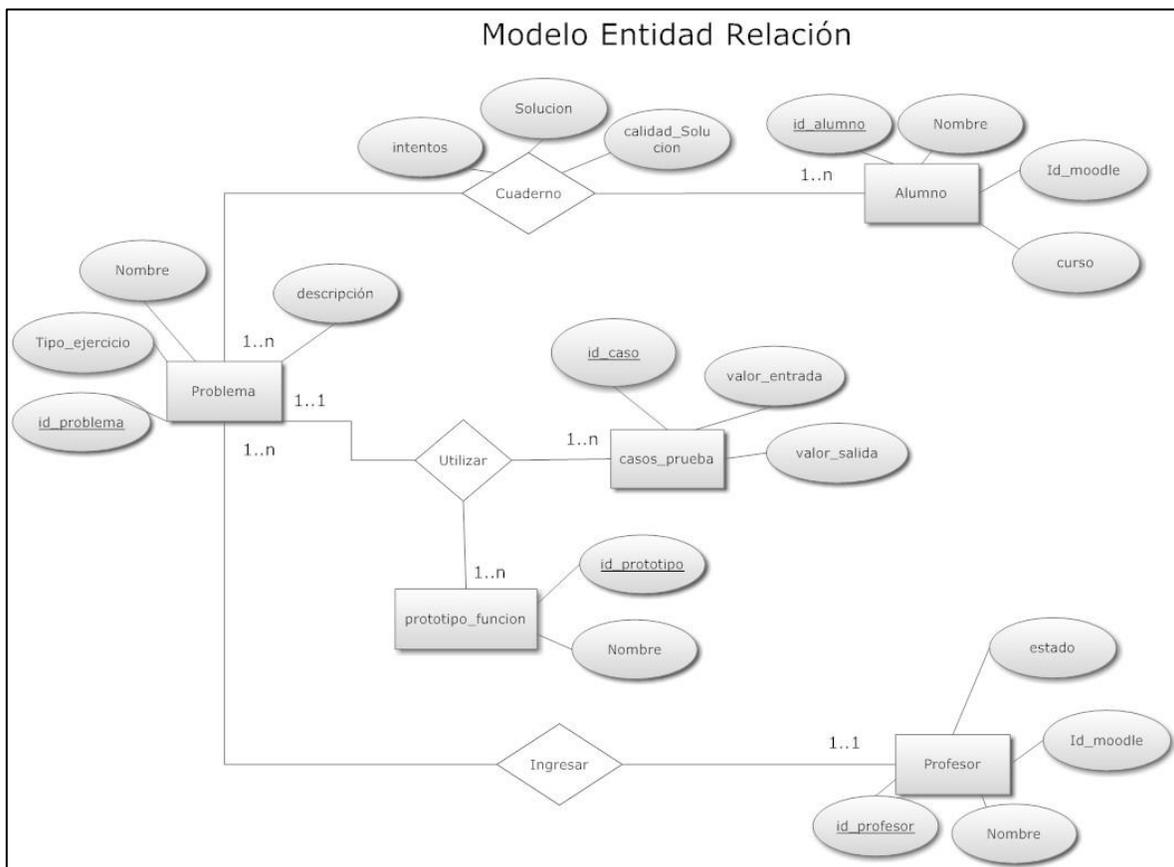


Diagrama : Modelo Entidad Relación

---

## 7 DISEÑO

---

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

Este modelo representa la forma en que se administra la información del sistema SAED. La estructura de la base de datos está constituida por las siguientes tablas las cuales hacen que la base de datos creada sea consistente.

➤ **Problema:**

Se refiere a los ejercicios proporcionados por los profesores.

Id_problema	Tipo_ejercicio	Nombre	Descripción	Id_profesor

Donde sus atributos son:

- Id\_problema
- Tipo\_ejercicio
- Nombre
- Descripción
- Id\_profesor

➤ **Cuaderno:**

Es el lugar físico donde se muestran los ejercicios asignados a un alumno.

Id_problema	Id_alumno	Intentos	Solución	Calidad_solucion

Donde sus atributos son:

- Id\_problema
- Id\_alumno
- Intentos
- Solución
- Calidad\_solución

➤ **Alumno:**

Son los usuarios que utilizan el sistema.

Id_alumno	Nombre	id_moodle	curso

Donde sus atributos son:

- Id\_alumno
- Nombre
- Id\_moodle
- curso

➤ **Profesor:**

Es quien imparte la asignatura.

Id_profesor	Nombre	id_moodle	estado

Donde sus atributos son:

- Id\_profesor
- Nombre
- Id\_moodle
- estado

➤ **Casos de Prueba:**

Son los valores con los que se prueba el código del alumno.

Id_caso	Id_problema	Valor_entrada	Valor_salida

Donde los atributos son:

- Id\_caso
- Id\_problema
- Valor\_entrada
- Valor\_salida

➤ **Prototipo\_funcion:**

Se refiere al encabezado de la función que debe llevar el código del alumno.

Id_prototipo	Id_problema	nombre

Donde los atributos son:

- Id\_prototipo
- Id\_problema
- Nombre

**7.2 Diseño de arquitectura funcional**

En el siguiente diagrama se observa el diseño de la arquitectura funcional del sistema.

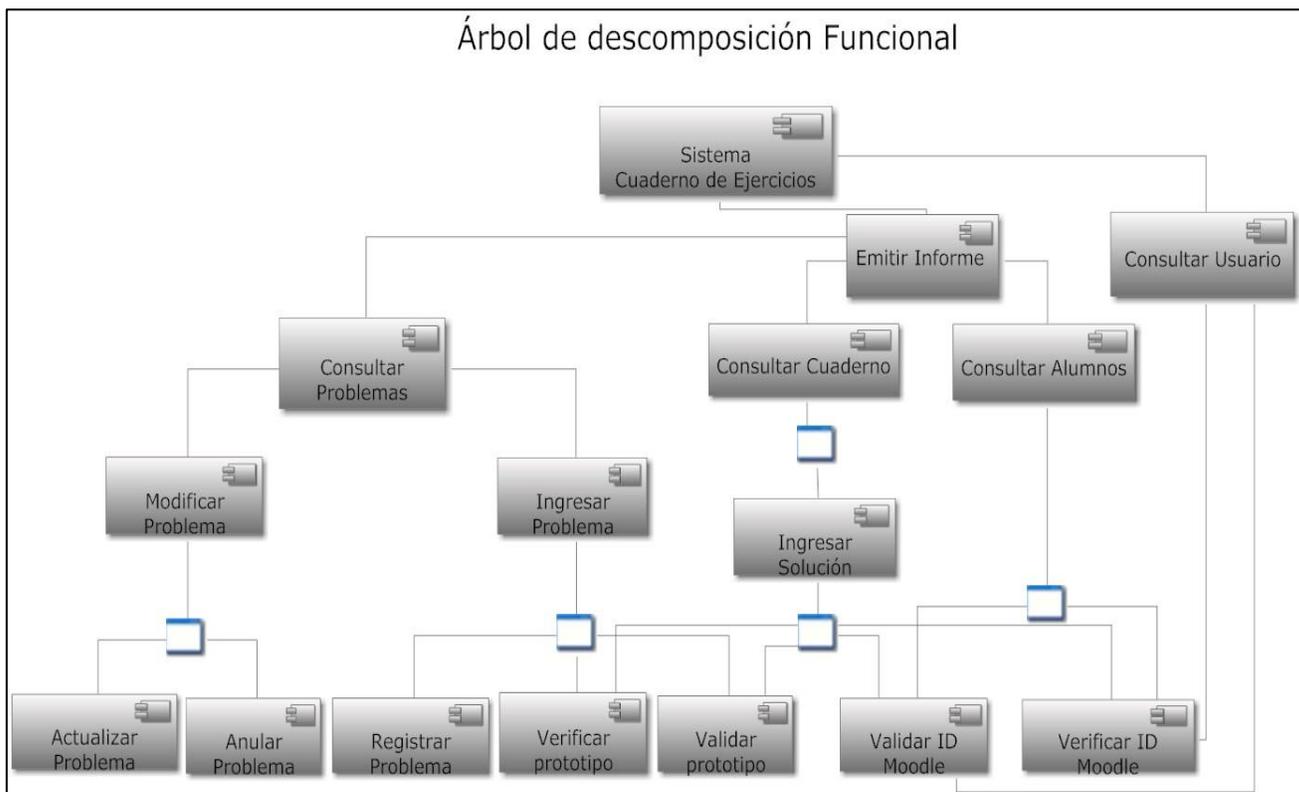


Diagrama : Árbol de descomposición Funcional Sistema

### 7.3 Diseño interfaz y navegación



Diagrama : Diseño Interfaz

El diseño de la interfaz de usuario considera un diseño estándar que será respetado en todas las pantallas. Como se describe a continuación según los números del diagrama 6.

- 1) Cabecera: Es utilizada en todas las páginas del sitio, con un fondo en degradado de color naranja, en ella se encuentra el título "Cuaderno de ejercicios", con una imagen animada de unos libros.

- 2) Menú: El menú se encuentra al lado izquierdo de la pantalla, el que contiene las opciones disponibles para el usuario, cada una de estas opciones serán un link.
- 3) Espacio Libre: En este espacio se muestra la opción seleccionada por el usuario.
- 4) Pie de página: Se muestra información relevante sobre IDEone, la condición para poder utilizar este compilador online es indicar claramente que se ocupa como parte del proyecto por este motivo se incluye en este espacio.
- 5) Botón: Se muestra cada vez que se requiere enviar un formulario de la opción seleccionada por el usuario.

➤ **Jerarquía Menú**

La jerarquía de menú representa los anidamientos y agrupaciones de las opciones de menú.

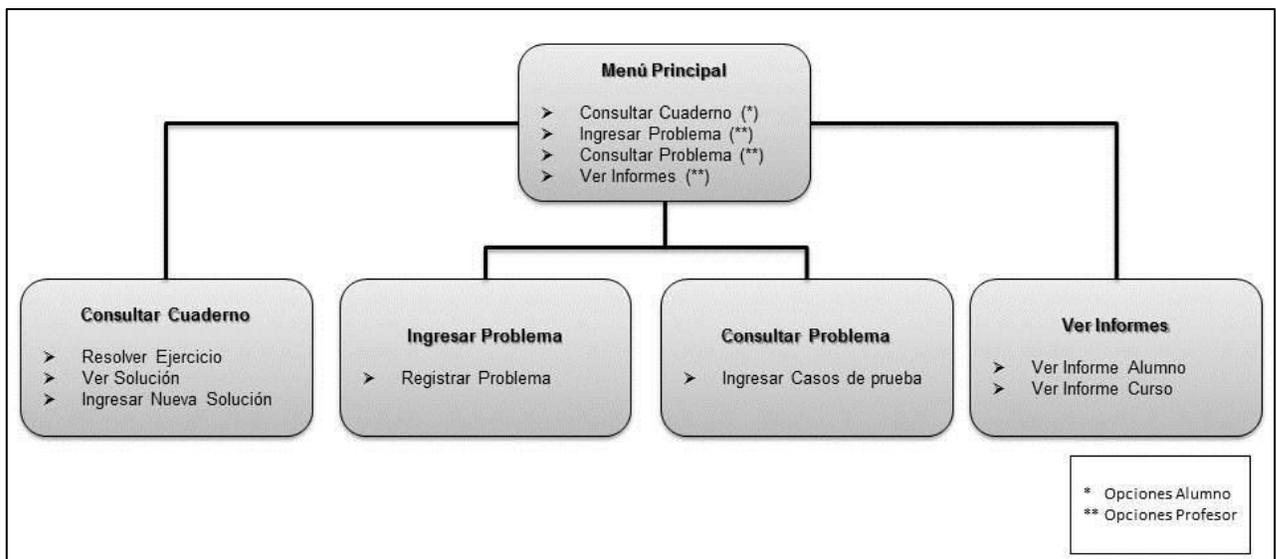
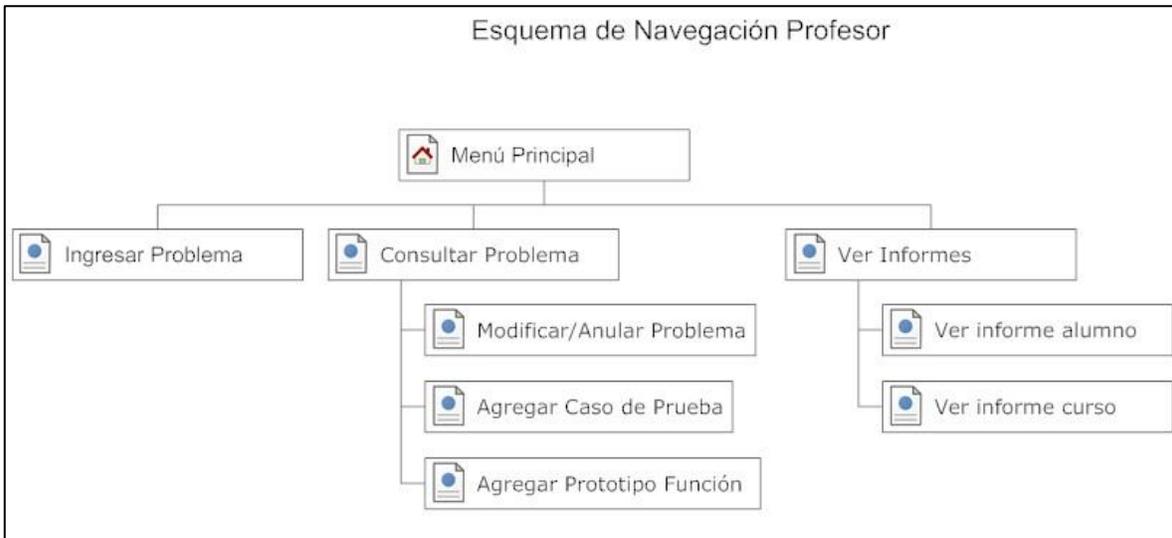


Diagrama : Jerarquía Menú

➤ **Esquema de Navegación**

El esquema de navegación representa las opciones que tiene el usuario para navegar y/o recorrer dentro de las distintas opciones del sistema, en este caso existen dos tipos de esquema de navegación, ya que los privilegios de los usuarios son distintos.



*Diagrama : Esquema de Navegación Profesor*



*Diagrama : Esquema de Navegación Alumno*

## 7.4 Especificación de módulos

Nombre Módulo: Validar prototipo			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
prototipo_funcion	String	flag	booleano

Nombre Módulo: Verificar prototipo			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
prototipo_funcion	String	Flag	booleano

Nombre Módulo: Registrar Problema			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Tipo_ejercicio	String	Indicador de registro	booleano
enunciado	String		
descripcion	String		
Prototipo_función	String		

Nombre Módulo: Actualizar Problema			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Tipo_ejercicio	String	Tipo_ejercicio	String
enunciado	String	enunciado	String
descripcion	String	descripcion	String
Prototipo_función	String	Prototipo_función	String

Nombre Módulo: Anular Problema			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
id_problema	int	Indicador borrado	booleano

Nombre Módulo: Validar ID_Moodle			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Id_moodle	String	flag	booleano

Nombre Módulo: Verificar ID_Moodle			
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Id_moodle	String	Flag	booleano

---

## 8 PRUEBAS

---

### 8.1 Elementos de prueba

Uno de los principales módulos que se va probar es el de la conexión tanto a la base de datos con extracción e ingreso de datos al sistema como a la plataforma moodle extrayendo los datos de los usuarios de esta plataforma.

Las pruebas a utilizar son de caja negra y se llevan a cabo sobre la interfaz del software, y es completamente indiferente al comportamiento interno y la estructura del programa, con esto se pretende demostrar que las funciones del software son operativas, los elementos de entrada sean verificados y adecuados y que se produzca una salida correcta y esperada.

Al momento de realizar las pruebas al sistema se determinaron algunos módulos internos que son:

- Problemas, donde se realizan pruebas de ingreso, de consultas, modificación y eliminación de los problemas en el sistema.
- Casos de prueba, los cuales están cien por ciento relacionados con el ingreso de los problemas, por lo tanto se debe probar la conexión entre ambas entidades.
- Prototipo de función los que están cien por ciento relacionados con el ingreso de los problemas, por lo tanto se debe probar la conexión entre ambas entidades.
- Cuademo, donde se realizan pruebas de ingreso de solución de un ejercicio, consultar problemas resueltos que se encuentran 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 entre el modulo Problema.	Caja Negra	Particiones	1.- Ingresar al modulo problema. 2.- Acceder al modulo donde se encuentra Ingresar Problema. 3.- Probar el ingreso de un problema. 4.- Ingresar al modulo consultar problema. 5.- Acceder a modificar datos de problemas ingresados.	Verificar el correcto funcionamiento de las conexiones.
Funcionalidad	sistema	Probar conexión entre el modulo Cuaderno.	Caja negra	Particiones	1.- Ingresar al modulo Cuaderno. 2.- Acceder al modulo donde se encuentra Resolver Ejercicio. 3.- Probar el ingreso de una solución al cuaderno. 4.- Ingresar al modulo consultar Ejercicios resueltos.	Verificar el correcto funcionamiento de las conexiones.
Funcionalidad	sistema	Probar conexión entre el modulo Casos de Prueba.	Caja negra	Particiones	1.- Ingresar al modulo Problemas. 2.- Acceder al modulo donde se encuentra Ingresar problema. 3.- Ingresar al modulo modificar problema. 4.- Acceder a ingresar caso de prueba.	Verificar el correcto funcionamiento de las conexiones.
Funcionalidad	sistema	Probar conexión entre el modulo Prototipo.	Caja negra	Particiones	1.- Ingresar al modulo Problemas. 2.- Acceder al modulo donde se encuentra Ingresar problema. 3.- Ingresar al modulo modificar problema. 4.- Acceder a ingresar prototipo.	Verificar el correcto funcionamiento de las conexiones.

Tabla : "Especificación de Pruebas"

### 8.3 Responsables de las pruebas

Las personas encargadas y responsables de realizar las pruebas en el sistema de forma correcta son los desarrolladores del software, para probar las funcionalidades y las conexiones a los distintos sistemas con el que interactúa.

### 8.4 Calendario de pruebas

Las pruebas realizadas se hicieron a medida que se fue terminando la implementación de los módulos, y así verificar que el sistema está trabajando de manera correcta.

Es importante integrar en esta sección personas que interactúen con el sistema es por esto que se tiene pronosticado evaluar el sistema con la ayuda de los usuarios finales, para verificar las pruebas ya realizadas hasta el momento con el fin de poder brindar la mejor experiencia de usuario posible.

### 8.5 Detalle de las pruebas

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

### 8.6 Conclusiones de Prueba

Las conexiones del sistema están bien realizadas, ya que interactúan de buena forma con la base de datos al ingresar, actualizar datos del sistema.

La conexión del sistema SAED con PVA también se considera realizada de forma satisfactoria ya que se obtienen los datos requeridos desde PVA hacia el sistema sin dificultades, siendo esto probado desde forma local.

---

## 9 PLAN DE CAPACITACIÓN Y ENTRENAMIENTO

---

El plan de capacitación y entrenamiento será de tipo práctica, esto se debe a que el sistema está diseñado de una manera simple y amigable al usuario lo cual lo hace de fácil entendimiento.

Los temas a tratar consisten en el manejo del sistema por parte de los usuarios, con supervisión y guía del desarrollador, que será el encargado de la capacitación con la finalidad de que si ocurre una consulta se pueda clarificar de manera rápida.

Los usuarios a quienes se capacitan son a los profesores de la asignatura los cuales están encargados de ingresar problemas al sistema, consultar por el rendimiento de los alumnos y explicarles a los alumnos el uso de este sistema.

Los temas que se consideran son:

- Ingreso de Problemas.
- Ingreso múltiple de casos de prueba asociados a un problema.
- Ingreso de múltiples prototipos para un problema.
- Obtención de informes de rendimiento.
- Actualización de datos de un problema.
- Ingreso de múltiples prototipos asociados a un problema.

## 10 PLAN DE IMPLANTACIÓN Y PUESTA EN MARCHA

Se debe instalar el sistema desarrollado en el servidor de la universidad, éste pasa por una etapa de “Marcha Blanca”, con datos de prueba, para que los usuarios comprueben su correcto funcionamiento, e informar de algún error existente. Una vez realizadas las pruebas correspondientes y todo este en un correcto orden, se procede a poblar la base de datos con información real para el sistema.

Al realizar el correcto funcionamiento del sistema con el servidor de la facultad se procede a conectar el sistema con la plataforma PVA de la universidad y probar su correcto funcionamiento con este.

El plan de implantación, el cual consta de dos etapas, se realizará durante el mes de marzo del presente año. A continuación se presenta el calendario de estas:

Id	Nombre de tarea	Duración	Comienzo	10 mar '13							17 mar '13				
				D	L	M	X	J	V	S	D	L	M	X	J
1	Etapa 1	3,5 días	lun 11-03-13		■										
2	Etapa 2	4 días	lun 18-03-13									■			

Las etapas del plan de implantación son:

*Etapa 1: Instalación del sistema en servidor*

Instalar el sistema en el servidor de la facultad y realizar pruebas para ver el correcto funcionamiento del web-service utilizado.

*Etapa 2: Conexión con PVA*

Luego de tener el sistema funcionando en el servidor de la facultad se debe realizar la conexión con la plataforma PVA que se encuentra en este mismo servidor y probar su correcto funcionamiento con este.

## 11 RESUMEN ESFUERZO REQUERIDO

Al momento de desarrollar un software se deben considerar todas las etapas que este conlleva; para lograr esto es necesario destinar horas de trabajo, es por esto que para el desarrollo del sistema SAED se ha confeccionado una tabla resumen de las horas dedicadas por el equipo de desarrollo a cada etapa.

Actividades/Fases	N° horas	N° horas	N° horas Equipo
	Karen	Anggie	
Planeación	20	20	40
Reunión con Profesora guía	2	2	4
Desarrollo etapa de análisis	60	60	120
Reunión con Profesora guía	2	2	4
Desarrollo etapa de diseño	25	25	50
Reunión con Profesora guía	2	2	4
Implementación 1° iteración	60	60	120
Reunión con Profesora guía	5	5	10
Implementación final	200	200	400
Reunión con Profesora guía	2	2	4
Total Equipo			756

*Tabla : "Esfuerzo requerido"*

---

## 12 CONCLUSIONES

---

A través del desarrollo de este proyecto se logró detectar los componentes con los que la universidad cuenta para dar apoyo a la docencia dentro de la Facultad de Ciencias Empresariales, y de esta forma utilizar estos elementos ya conocidos tanto por alumnos y profesores, como lo es PVA. Mediante la investigación realizada se comprendió que la plataforma que se utiliza actualmente en la facultad es la que cuenta con más herramientas y opciones de configuración de acuerdo a las necesidades requeridas.

En primera instancia se generó una solución en base a la plataforma existente, sin considerar la individualidad que el problema necesita satisfacer, por lo que se optó por una solución compatible con las herramientas de desarrollo que cuenta la plataforma. Esto se logró gracias a la metodología utilizada durante el desarrollo de software la cual permitió realizar las modificaciones pertinentes de acuerdo a los requerimientos del usuario y en el tiempo correspondiente.

Durante el diseño del software se seleccionaron herramientas actualizadas que se encuentran al nivel óptimo de las nuevas tecnologías y además considerando que estén de acuerdo con las tecnologías y herramientas con las que cuenta la universidad para lograr una compatibilidad al momento de la implementación. En la elección del lenguaje y herramientas se consideró la posibilidad de que ofrezcan a futuro una expansión en el software.

Para concluir cabe mencionar que la problemática seleccionada generó en el equipo de desarrollo un gran interés tanto a nivel académico como a nivel personal: a nivel académico permitió conocer las distintas herramientas que existen para solucionar la problemática planteada ya sean plataformas, como herramientas para ayudar a la corrección automática y en las diversas aplicaciones existentes; a nivel personal se adquirió una experiencia que será de gran ayuda para la inserción como profesionales al mundo laboral.

---

## 13 BIBLIOGRAFÍA

---

1. **Pontificia Universidad Católica de Valparaíso.** Biblioteca ágora. *agora*. [En línea] [Citado el: 2 de Octubre de 2012.] <http://agora.ucv.cl/manual/plataformas%5Cplataformas.html>.
2. **Trust Moodle.** moodle. [En línea] [Citado el: 6 de Octubre de 2012.] <https://moodle.org/>.
3. **Carlos Agustín Álvarez Fuentes, Mauricio Andrés Parada Henríquez.** *Análisis Comparativo De Plataformas E-Learning "Moodle V/S Claroline"*. Chillán : Facultad de Ciencias Empresariales, Universidad del Bío-Bío, 2005.
4. **Ogayar Fernández Blas, Martínez Bazan Carlos.** *PLATAFORMA ILIAS COMO HERRAMIENTA PARA DOCENCIA ECTS. UTILIZACIÓN EN LA ESCUELA POLITÉCNICA SUPERIOR DE JAEN.* Jaén, Andalucía (España). : Escuela Politécnica Superior. Universidad de Jaén.
5. **Aalto University.** uiah. [En línea] [Citado el: 8 de Octubre de 2012.] <http://fle3.uiah.fi/>.
6. **Consortium Claroline aisbl.** claroline. [En línea] [Citado el: 5 de Octubre de 2012.] <http://www.claroline.net/>.
7. **Mg. Marcela Chiarani, Lic. Irma Pianucci, Mg. Margarita Lucero.** *Criterios de Evaluación de Plataformas Virtuales de Código Abierto para Ambientes de Aprendizajes Colaborativos.* San Luis, Argentina : Facultad de Ciencias Físico-Matemáticas y Naturales, Universidad Nacional de San Luis.
8. Plataforma Virtual de Aprendizaje. *PVA*. [En línea] [Citado el: 22 de 10 de 2012.] [pva.face.ubiobio.cl](http://pva.face.ubiobio.cl).
9. Codepad. [En línea] 3 de 10 de 2012. <http://codepad.org/>.
10. Ideone. [En línea] 03 de 10 de 2012. <http://ideone.com/>.
11. github. [En línea] [Citado el: 10 de 11 de 2012.] [https://github.com/hit-moodle/moodle-local\\_onlinejudge](https://github.com/hit-moodle/moodle-local_onlinejudge).
12. Wikipedia. [En línea] 30 de 01 de 2013. [http://es.wikipedia.org/wiki/Localizador\\_uniforme\\_de\\_recursos](http://es.wikipedia.org/wiki/Localizador_uniforme_de_recursos).
13. Wikipedia. [En línea] 30 de 01 de 2013. [http://es.wikipedia.org/wiki/Simple\\_Object\\_Access\\_Protocol](http://es.wikipedia.org/wiki/Simple_Object_Access_Protocol).

## 14 ANEXO: PLANIFICACION INICIAL DEL PROYECTO

Number	Task	Start	End	Duration
1	Planeación	11/9/2012	30/9/2012	14
1.1	Definir empresa	11/9/2012	12/9/2012	2
1.2	Estudio del arte	13/9/2012	26/9/2012	10
1.2.1	Definir problema	13/9/2012	13/9/2012	1
1.2.2	Buscar Soluciones al problema	14/9/2012	25/9/2012	8
1.2.3	Definir Solución	26/9/2012	26/9/2012	1
1.3	Definir Proyecto	27/9/2012	28/9/2012	2
1.4	Entrega Etapa Planeación a profesora guía	1/10/2012	1/10/2012	
2	Análisis	1/10/2012	4/11/2012	25
2.1	Captura de Requisitos	1/10/2012	11/10/2012	9
2.2	Definir Producto	8/10/2012	11/10/2012	4
2.3	Requisitos Específicos	15/10/2012	17/10/2012	3
2.4	Estudio de Factibilidad	18/10/2012	23/10/2012	4
2.5	Diagrama Casos de Uso	24/10/2012	29/10/2012	4
2.6	Modelamiento de Datos	30/10/2012	31/10/2012	2
2.7	Entrega etapa análisis a profesora guía	5/11/2012	5/11/2012	
3	Diseño	5/11/2012	16/11/2012	10
3.1	Diseño Físico de la Base de Datos	5/11/2012	5/11/2012	1
3.2	Diseño de la Arquitectura Funcional	6/11/2012	7/11/2012	2
3.3	Diseño Interfaz y Navegación	8/11/2012	12/11/2012	3
3.4	Especificación de Módulos	13/11/2012	14/11/2012	2
3.5	Diseñar Pruebas de sistema	15/11/2012	15/11/2012	1
3.6	Entrega etapa Diseño a profesora guía	16/11/2012	16/11/2012	1
4	Implementación	17/11/2012	9/1/2013	38
4.1	Implementar 1 Iteración	17/11/2012	30/11/2012	10
4.1.1	Desarrollar Interfaces	17/11/2012	23/11/2012	5
4.1.2	Entregar Versión	24/11/2012	24/11/2012	
4.1.3	Retroalimentación Cliente	24/11/2012	27/11/2012	2
4.1.4	Agregar Retroalimentación	28/11/2012	30/11/2012	3
4.1.5	Desarrollar Pruebas	30/11/2012	30/11/2012	1
4.2	Mostrar Avance a Profesora guía	1/12/2012	1/12/2012	
4.3	Implementar 2° Iteración	1/12/2012	14/12/2012	10
4.3.1	Desarrollar Curso	1/12/2012	7/12/2012	5
4.3.2	Entregar 2 Versión	8/12/2012	8/12/2012	
4.3.3	2 Retroalimentación Cliente	10/12/2012	11/12/2012	2
4.3.4	Agregar 2 Retroalimentación	12/12/2012	14/12/2012	3
4.3.5	Desarrollar Pruebas 2 Versión	14/12/2012	14/12/2012	1
4.4	Mostrar Avance a Profesora guía	15/12/2012	15/12/2012	
4.5	Implementar 3 Iteración	17/12/2012	9/1/2013	18
4.5.1	Desarrollar Navegación e Informes emitidos	17/12/2012	21/12/2012	5
4.5.2	Entrega 3 Versión	22/12/2012	22/12/2012	
4.5.3	3 Retroalimentación Cliente	22/12/2012	28/12/2012	5
4.5.4	Agregar 3 Retroalimentación	29/12/2012	8/1/2013	7
4.5.5	Desarrollar Pruebas 3 Versión	9/1/2013	9/1/2013	1
4.6	Mostrar Avance a Profesora guía	10/1/2013	10/1/2013	
5	Entrega Proyecto "Plataforma de Apoyo a la Docencia de la Asignatura Estructura de Datos"	11/1/2013	11/1/2013	

### 14.1 Estimación inicial de tamaño

Para calcular el tamaño del software se utiliza el método Puntos de casos de uso, utiliza los actores y casos de uso identificados para calcular el esfuerzo que significará desarrollarlos.

- Factor de peso de los actores sin ajustar (UAW)

Se comienza evaluando la complejidad de los actores con los que tendrá que interactuar el sistema. Este puntaje se calcula determinando si cada actor es una persona u otro sistema.

Actor	Tipo Actor	Peso
Profesor	Complejo	3
Alumno	Complejo	3
Administrador	Complejo	3
PVA	Medio	2
Web Service IDEone	Simple	1
<b>UAW (Factor de peso de los actores sin ajustar)</b>		<b>12</b>

*Tabla Factores de peso actores*

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

En esta etapa, para determinar el nivel de complejidad toma en cuenta el número de transacciones que se pueden realizar en un caso de uso y lo evalúa como se especifica en el método.

Caso de Uso	Transacciones	Peso
Ingresar Problema	3	5
Consultar Problema	2	5
Ingresar casos de prueba	3	5
Modificar/ Anular Problema	2	5
Emitir Informe de rendimiento	2	5

Consultar Rendimiento	2	5
Ver Cuaderno	2	5
Resolver Problema	3	5
Emitir Informe Calidad de solución	2	5
Consultar Problemas Resueltos	2	5
Compilar y Ejecutar Solución	1	5
Autenticar Usuario	3	5
Evaluar Calidad de solución	3	5
Extraer usuarios moodle	3	5
Eliminar Usuarios	2	5
<b>UUCW (Factor de peso de los casos de uso sin ajustar)</b>		<b>75</b>

*Tabla Factores de Peso Casos de Uso*

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

UUCP (Puntos de casos de Uso sin Ajustar) = UAW + UUCW

UUCP= 12 + 75 = 87

➤ Factores de complejidad técnica

Son 13 factores que evalúan la complejidad técnica de los módulos del sistema, cada uno de estos factores tienen un peso definido, los cuales se multiplican por un valor asignado de acuerdo a la relevancia de cada factor.

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	4	4
F5	El código debe ser reutilizable	1	1	1

F6	Facilidad de instalación	0,5	5	5
F7	Facilidad de uso	0,5	5	5
F8	Portabilidad	2	5	10
F9	Facilidad de cambio	1	3	3
F10	Concurrencia	1	5	5
F11	Incluye objetivos especiales de seguridad	1	4	4
F12	Provee acceso directo a terceras partes	1	1	1
F13	Se requiere facilidades especiales de entrenamiento a usuario	1	1	1
<b>TFactor (Sum (Valor*Peso))</b>				<b>44</b>

*Tabla Evaluación Factores Técnicos*

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

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

$$\text{TCF} = 0.6 + 0.01 * 44 = 1.04$$

➤ Factores ambientales

Los factores son 8, que están relacionados con las habilidades y experiencia del equipo de desarrollo del proyecto. Evaluados de la misma forma que los factores técnicos.

Factor	Descripción	Peso	Valor	Total
F1	Familiaridad con el modelo de proyecto utilizado	1,5	1	1,5
F2	Experiencia en la aplicación	0,5	3	1,5
F3	Experiencia en orientación a objetos	1	0	0
F4	Capacidad del analista líder	0,5	5	2,5
F5	Motivación	1	5	5
F6	Estabilidad de los	2	5	10

	requerimientos			
F7	Personal part-time	-1	1	-1
F8	Dificultad del lenguaje de programación	-1	3	-3
<b>EFactor (Sum(Valor * Peso))</b>				<b>15,5</b>

*Tabla Evaluación Factores Ambientales*

Para calcular los factores ambientales se utiliza la siguiente formula:

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

$$\text{EF} = 1.4 + (-0.03 * 15,5) = 0,935$$

- Nivel de Esfuerzo (LOE)

Las horas de esfuerzo requerido se estiman de acuerdo a los factores ambientales según lo indica la siguiente tabla.

Factores Ambientales	Condición	Total
F1 a F6	Valor menor a 3	2
F7 y F8	Valor mayor a 3	0
<b>LOE</b>		<b>20</b>

*Tabla Calculo LOE*

- Puntos de Caso de Uso Ajustados (UCP)

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

$$\text{UCP} = 87 * 1,04 * 0,935 = 84,5988 \text{ Puntos de caso de uso.}$$

- Esfuerzo horas-hombre (E)

El esfuerzo en horas-persona viene dado por:

$$E = \text{UCP} * \text{LOE}$$

$$E = 84,5988 * 20 = 1.691,976$$

---

## 15 ANEXO: RESULTADOS DE ITERACIONES EN EL DESARROLLO

---

➤ 1° iteración

Durante el desarrollo de la primera iteración se lleva a cabo la instalación de moodle en forma local y la creación de una cuenta de administrador, para poder crear y asignar roles, actividades y cursos dentro de la plataforma.

Se visualiza de la siguiente manera:



The screenshot shows the Moodle login interface for a site named "Estructuras de Datos". At the top right, it says "Usted no se ha identificado." and "Español - Internacional (es)". Below this is a navigation bar with "Página Principal" and "Entrar al sitio". The main content area is titled "Usuarios registrados" and contains the following text: "Entre aquí usando su nombre de usuario y contraseña (Las 'Cookies' deben estar habilitadas en su navegador)". There are two input fields: "Nombre de usuario" and "Contraseña", followed by an "Entrar" button. Below the fields is a checkbox for "Recordar nombre de usuario" and a link "¿Olvidó su nombre de usuario o contraseña?". A horizontal line separates this from the "Algunos cursos permiten el acceso de invitados" section, which has an "Entrar como invitado" button. At the bottom, it says "Usted no se ha identificado." and has a "Página Principal" button.

*Imagen : "Inicio Moodle"*

Una vez con la cuenta de administrador se procede a crear un curso dentro de la plataforma el cual es denominado "datos", además de crear una cuenta de alumno y la instalación del plugin encontrado para moodle el cual es "online-jugde".

Con lo anterior ya creado, el alumno inicia sesión dentro de la plataforma, lo cual tiene una apariencia de la siguiente forma:

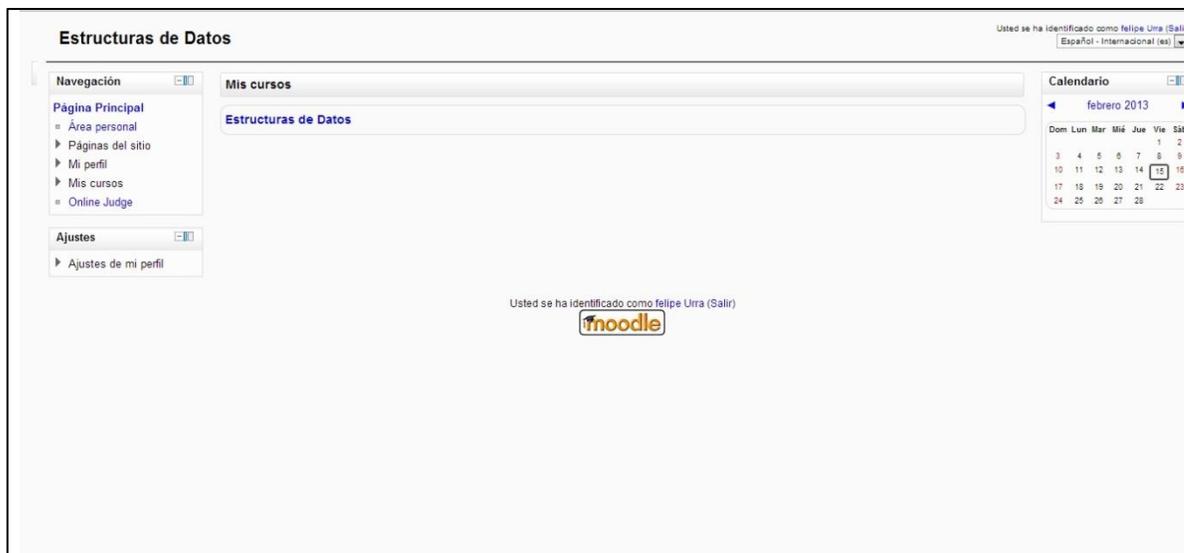


Imagen :” Ingreso alumno a moodle”

Cuando se crea el curso se le da la estructura que se divide por temas, lo cual a cada tema se le asigna el nombre de cada tipo de ejercicio que se ve durante la asignatura, y de esta manera se introducen los ejercicios por cada tema y a medida que el curso va avanzando. De esta forma:

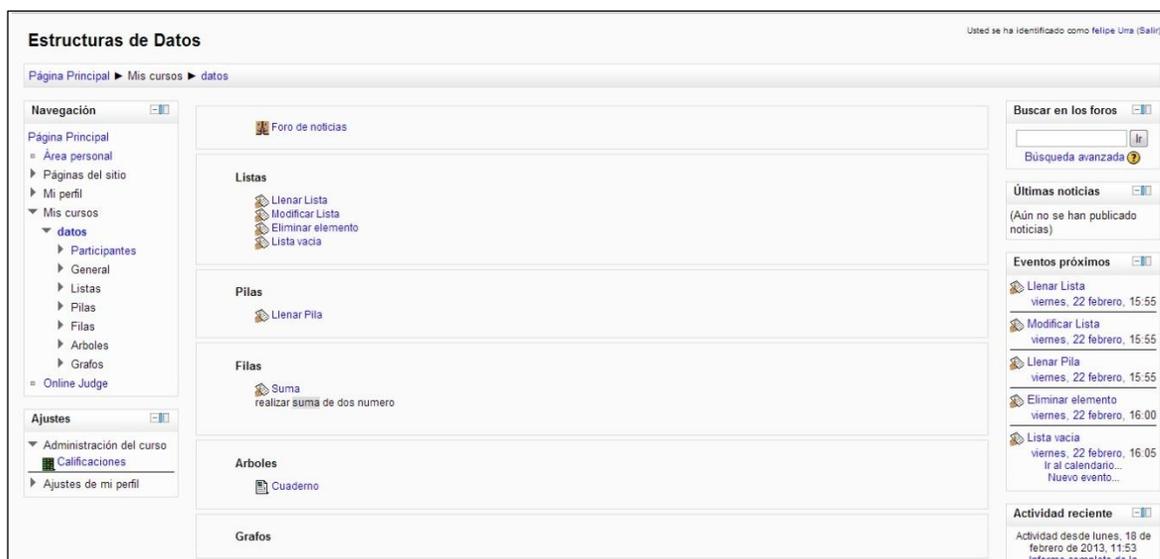


Imagen :”Curso por temas en moodle”

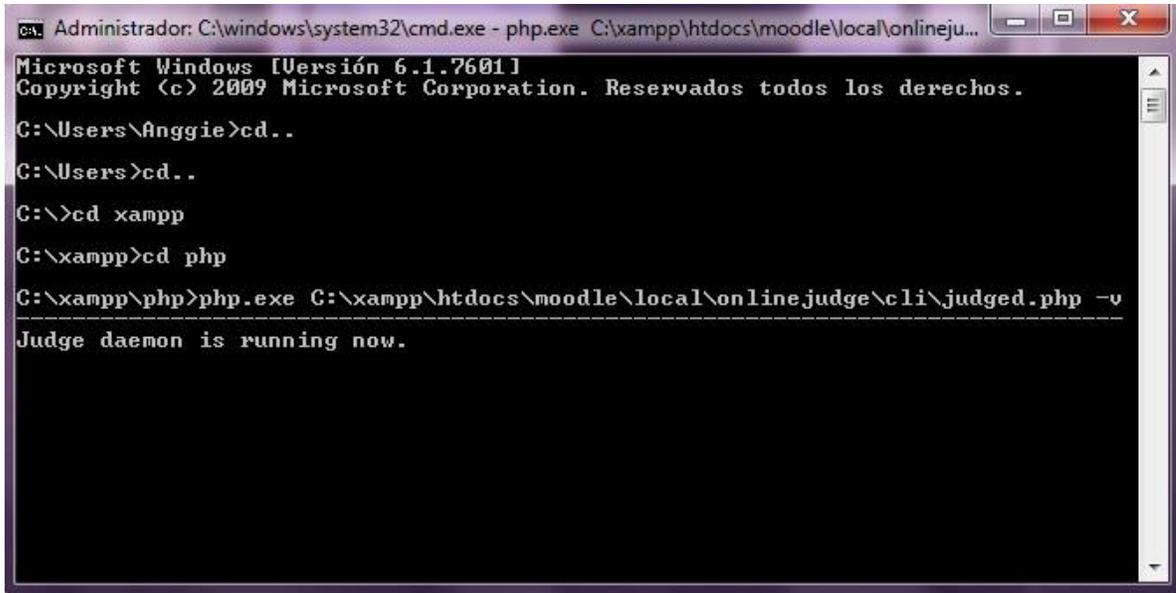
Las actividades que se crean dentro de cada tema son de tipo online-judge, que es el plugin que se instala en moodle para realizar la corrección automática del código que el

alumno sube, también se deben ingresar los casos de pruebas asociados a dicha actividad para que el código pueda ser calificado, asignándole un porcentaje de validez al momento de ejecutar e ingresando valores de entrada y valores de salida esperados. Lo que se puede visualizar en la siguiente imagen:

The image shows a web interface for configuring test cases. It is divided into two main sections: 'Test Cases1' and 'Test Cases2'.  
 In the 'Test Cases1' section:  
 - 'Grade' is set to 75% (dropdown menu).  
 - 'Testcase from files' is an unchecked checkbox.  
 - 'Input' is a text box containing '2 3'.  
 - 'Output' is a text box containing '5'.  
 - 'Input file' and 'Output file' are large empty text areas.  
 - 'Feedback for Wrong Answer' is an empty text box.  
 In the 'Test Cases2' section:  
 - 'Grade' is set to 25% (dropdown menu).  
 - 'Testcase from files' is an unchecked checkbox.  
 - 'Input' is a text box containing '8 8'.

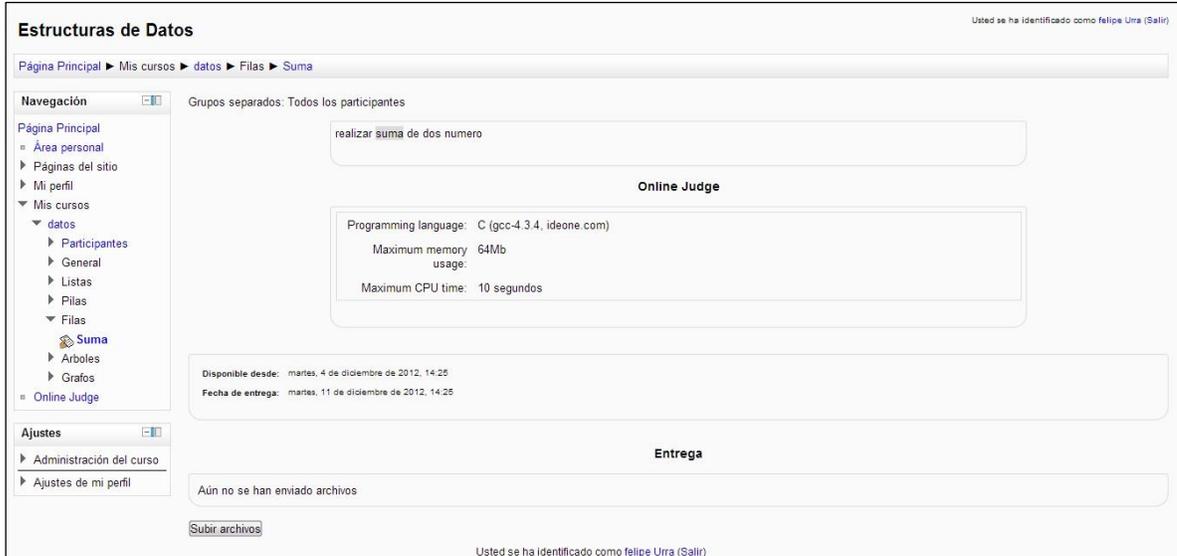
*Imagen : "Casos de Prueba"*

Para que el código del alumno sea compilado se debe ejecutar un comando como se muestra en la siguiente imagen:



*Imagen :” Comando”*

Al intentar desarrollar un ejercicio asignado del tipo online-judge el alumno lo visualizara de la siguiente forma:



*Imagen :”Vista de actividad de alumno”*

El alumno debe subir un archivo donde se encuentre su código de acuerdo al ejercicio. Mientras espera que se ejecute su tarea, visualiza una pantalla como esta:

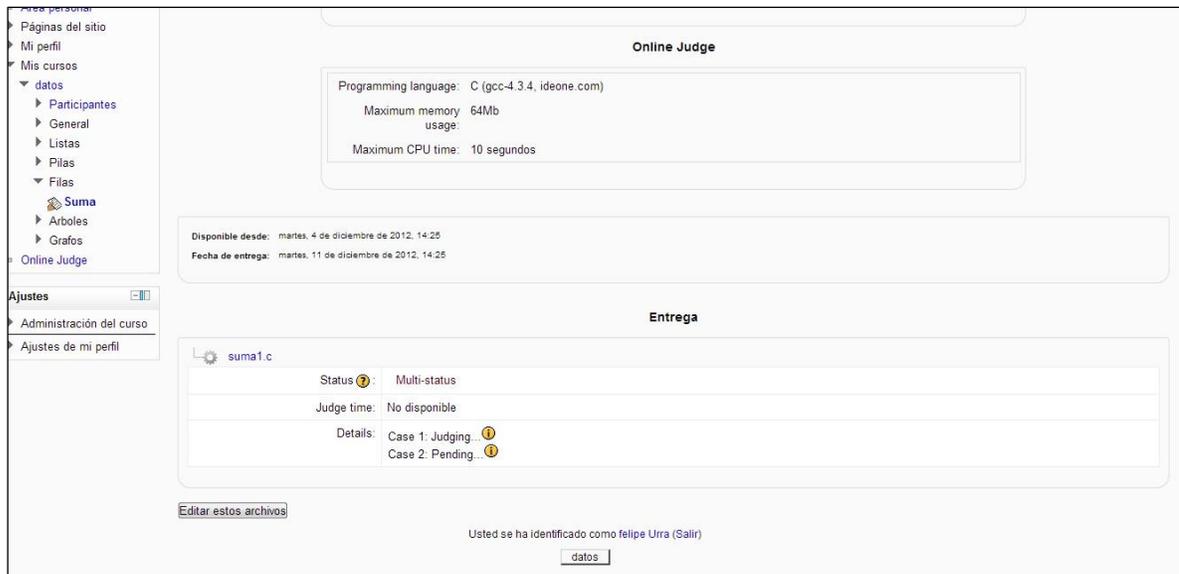


Imagen :” Actividad en proceso de compilación”

Una vez compilada la tarea aparece en pantalla los resultados obtenidos de acuerdo al código ingresado, y la nota asignada a esta, de acuerdo a los casos de prueba que se deben ingresar previamente a que el alumno desarrolle el ejercicio.

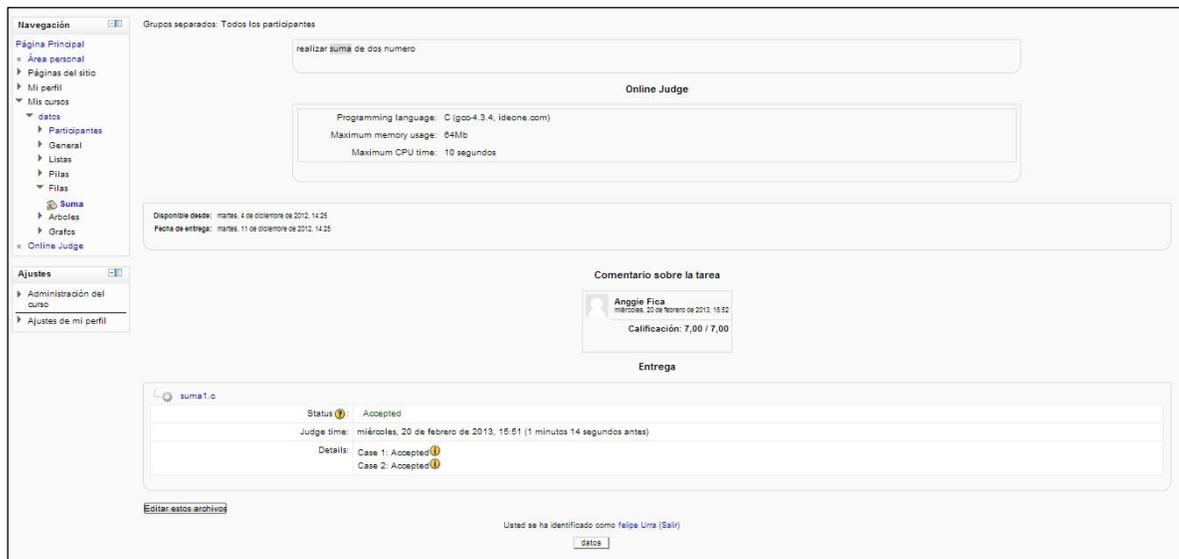


Imagen :”Compilación finalizada”

Una vez finalizada la primera iteración se concreto una revisión con el usuario final, el cual considero realizar modificaciones a lo desarrollado en su totalidad.

➤ 2° Iteración:

Se inicia con la creación de un sistema web desarrollado bajo el lenguaje de programación PHP, el que se conecta con una base de datos Mysql, siguiendo la arquitectura MVC para facilitar las modificaciones y mantener de forma ordenada y estructurada todo el código del sistema desarrollado. El sistema interactúa con el servicio web IDEone el cual realiza la corrección automática de los códigos que los alumnos ingresan para obtener la solución de un problema asignado en el cuaderno.

---

## 16 ANEXO: MANUAL DE USUARIO

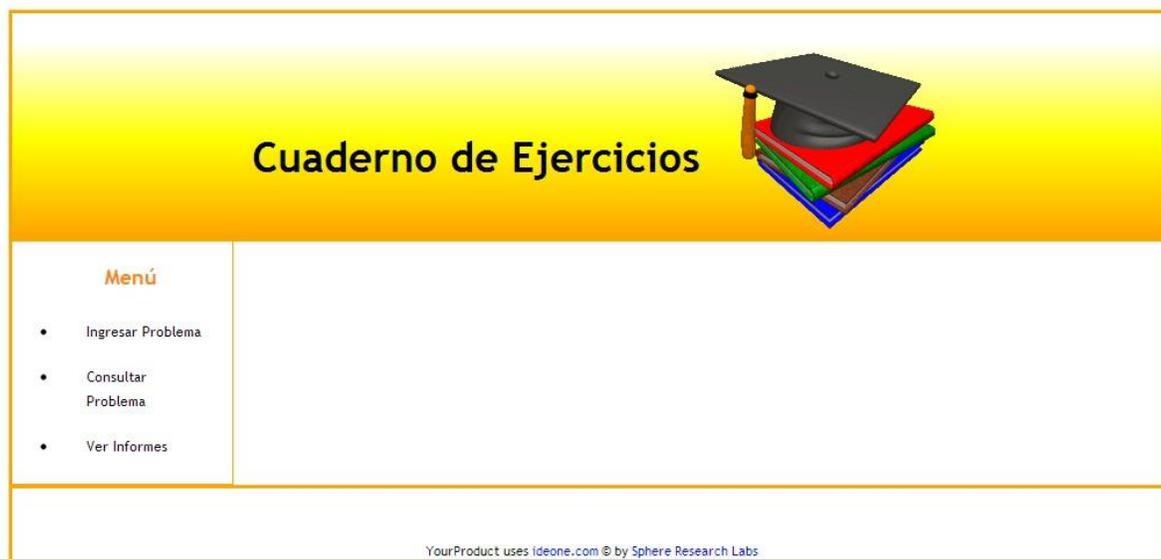
---

Para utilizar de manera adecuada el sistema SAED es fundamental contar con la documentación necesaria para el correcto funcionamiento de cada parte de este.

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: profesor, alumno y administrador.

### 16.1 Profesor

El sistema trabaja de acuerdo a la autenticación de moodle, por lo que es necesario que este logeado en la plataforma como profesor para reconocer su cuenta como tal. Una vez autenticado y teniendo acceso a la plataforma PVA puede ingresar al sistema SAED y visualizará algo similar a la siguiente imagen.



*Imagen "Inicio del profesor"*

En esta pantalla se encuentra el menú de opciones que tiene el profesor entre las cuales están Ingresar Problemas, Consultar Problemas y Ver informes.

➤ Ingresar Problemas

En esta opción del menú al profesor le aparecerán los campos que tiene el problema para poder almacenarlos en la base de datos, estos campos son tipo de problema, el cual se encuentran predeterminado y solo debe seleccionarlo, nombre del ejercicio, descripción del problema, prototipo de la función, valores de prueba tanto de entrada como de salida. La pantalla que visualiza es como la siguiente:

*Imagen "Ingresar Problema"*

➤ Consultar Problemas

En esta opción el profesor puede consultar por los problemas ingresados, la opción que tiene de consultar es mediante el tipo de ejercicio, el debe seleccionar uno y luego visualizará un listado de todos los ejercicios pertenecientes a ese tipo de problema.

Esto se visualiza de la siguiente forma:



Imagen "Selección del tipo de ejercicio a consultar"

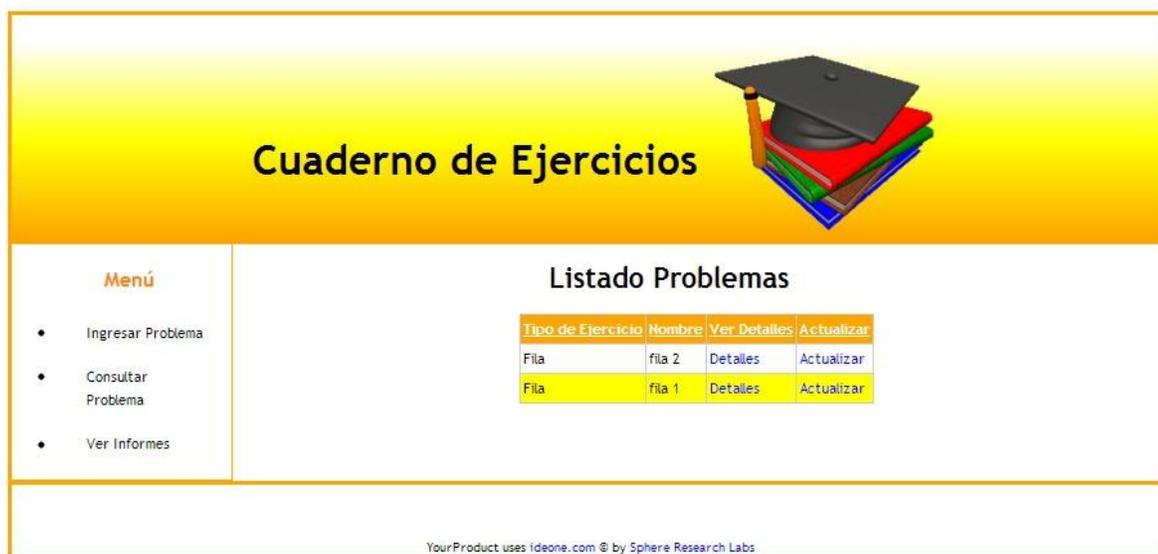


Imagen "Consulta de problemas"

Una vez que el sistema muestra todos los problemas que existen del tipo de ejercicio que selecciono se puede ver que existe la opción de actualizar el problema, al hacer click en esta opción se visualiza de la siguiente forma:

**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes

## Actualizar Problema

Tipo de ejercicio:

Nombre Problema:

Descripción Problema:

Actualizar Problema

Funciones Requeridas:

[Agregar Funciones](#)   [Modificar Funciones](#)

Casos de Prueba:

[Agregar Casos de Pruebas](#)   [Modificar Casos de Pruebas](#)

*Imagen "Actualizar Problema"*

Al tener el modo de visualización de actualizar un problema se encuentran las opciones de agregar funciones y agregar casos de pruebas, ya que un problema puede tener más de un valor de estos campos, y al seleccionar se ve así:



*Imagen "Agregar Prototipo"*

Así como se pueden agregar más prototipos de la función también se pueden modificar, aunque primero se debe seleccionar el prototipo a modificar y luego realizar las modificaciones pertinentes, lo que se visualiza en la siguiente imagen:



*Imagen "Selección de Prototipo a modificar"*



*Imagen "Modificar Prototipo"*

De la misma forma que se hace para agregar y modificar las funciones de prototipo como se visualizó en Imagen 14 e Imagen 15 se hace con los casos de prueba.

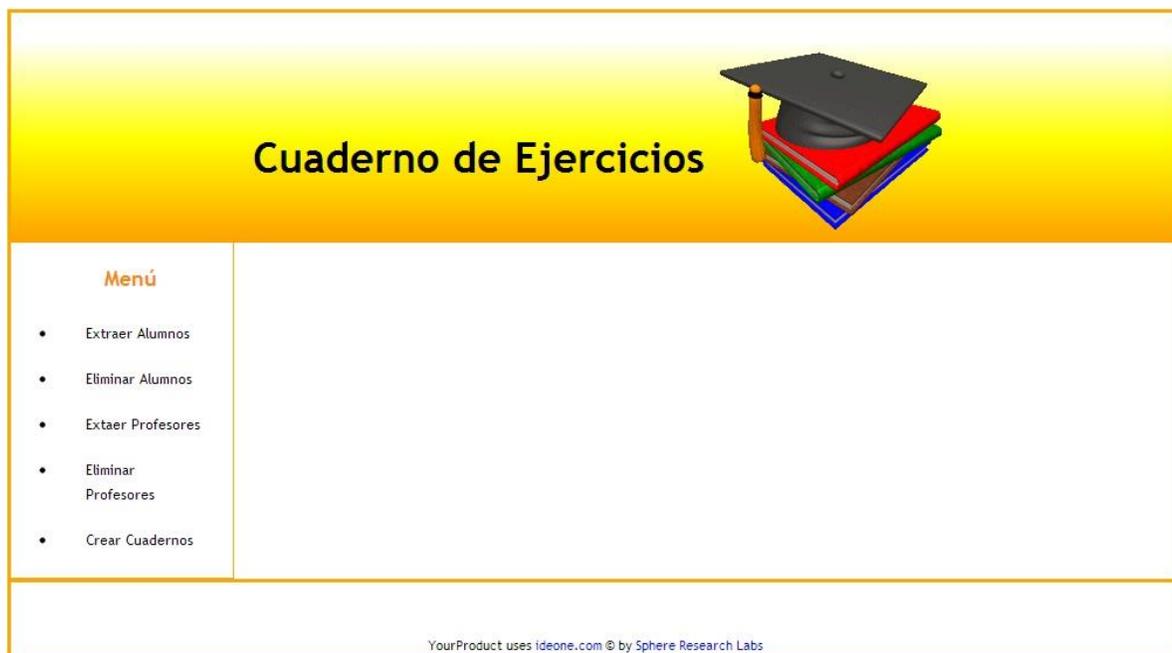
➤ Ver informes

En esta opción el profesor podrá visualizar los informes de rendimientos del curso como por alumno, de acuerdo a como él lo requiera.

## 16.2 Administrador

Este usuario además de tener las opciones ya vistas del profesor cuenta con privilegios más avanzados los cuales son la extracción y eliminación de los usuarios de moodle hacia el sistema.

La pantalla de inicio del administrador es la siguiente:



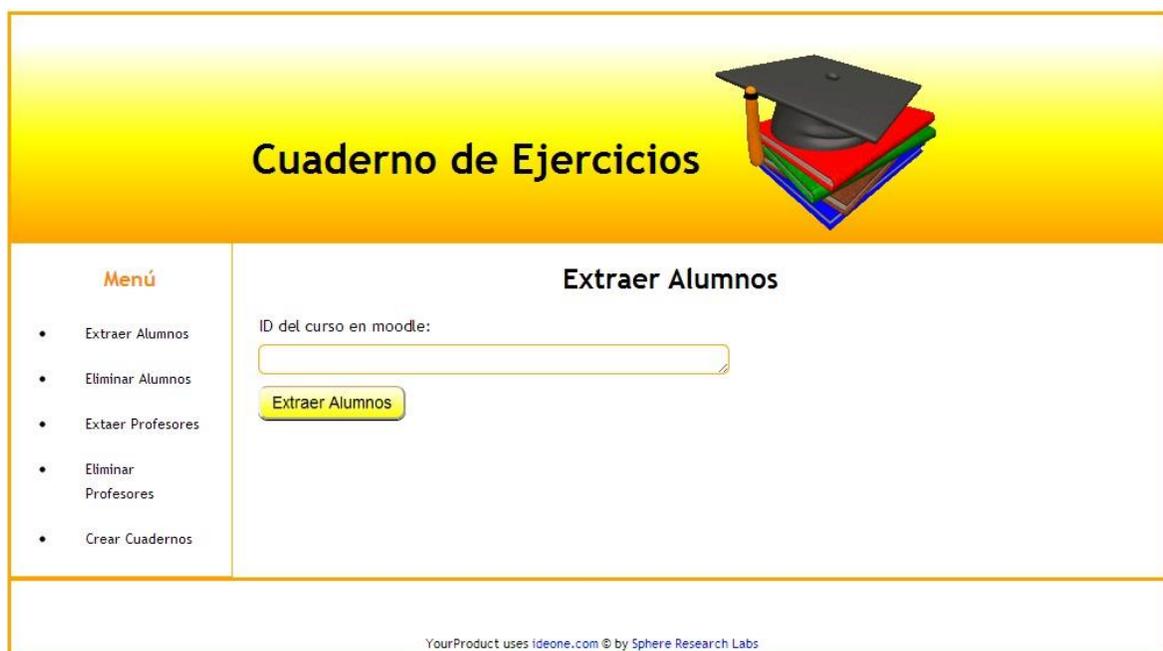
*Imagen "Menú del administrador"*

Las opciones que aparecen en el menú del administrador son las siguientes:

➤ **Extraer alumnos**

La forma en que se extraen los alumnos de la plataforma con la interactúa el sistema es ingresando el código con el cual se identifica la asignatura en PVA y de esta forma los usuarios quedan almacenados en la base de datos del sistema SAED.

La pantalla que visualizará al momento de extraer los alumnos es la siguiente:



*Imagen "Extracción de alumnos"*

➤ Eliminar alumnos

Esta opción permite quitar del sistema SAED a los alumnos cuando ya finalizo el semestre. Esta opción del menú le permite al administrador poder realizar una mantención del sistema semestre a semestre y así solo poder tener a los usuarios que correspondan.

➤ Extraer profesores

Como el sistema desarrollado se encuentra conectado con PVA es que es necesaria esta opción para poder extraer desde la plataforma a los profesores del curso. La forma en que se extraen es ingresando el código de la asignatura y luego presionando el botón extraer.

➤ Eliminar profesores

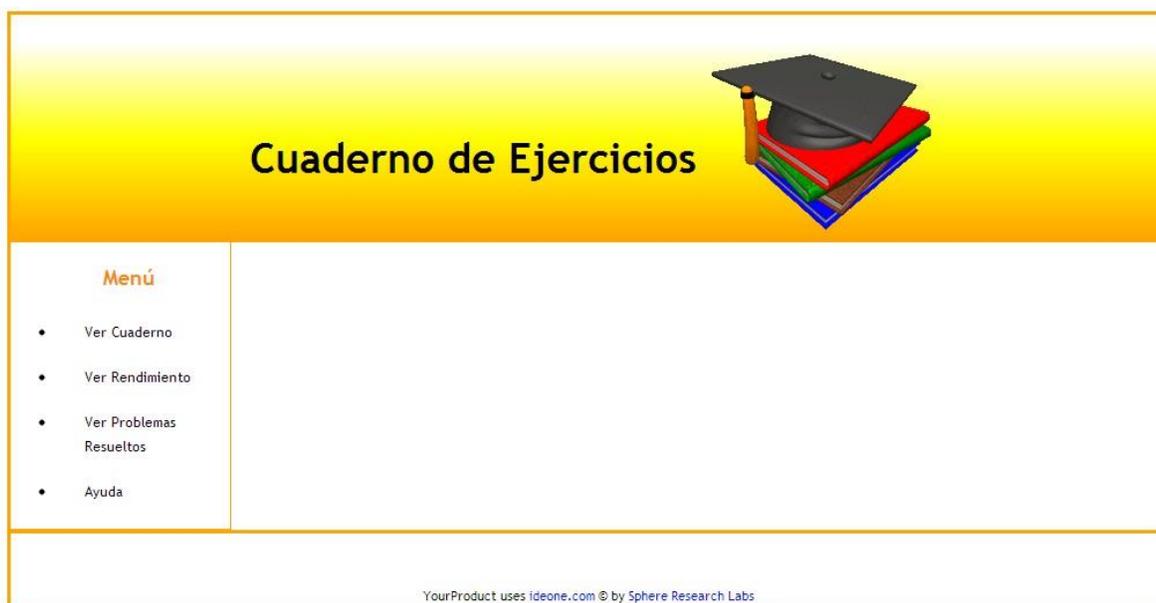
Al igual que sucede con los alumnos es necesario ir haciendo limpieza al sistema semestre a semestre, ya que no siempre son los mismos profesores que imparten la asignatura por lo que se deben quitar del sistema SAED.

➤ Crear cuaderno

El administrador tiene la opción de crear los cuadernos para todos los alumnos que se encuentren registrados en el curso de PVA, facilitando el trabajo a los profesores de acordarse de asignar los ejercicios a los alumnos.

### 16.3 Alumno

Para ser identificado como alumno dentro del sistema, este debe haber iniciado sesión desde PVA con dicha autenticación, ya que el sistema SAED trabaja con los usuarios de la plataforma. Al ingresar al sistema como alumno visualiza el sistema de la siguiente forma:



*Imagen "Inicio alumno"*

Las opciones que se encuentran en el menú del alumno son Ver cuaderno, Ver rendimiento, Ver problemas resueltos y Ayuda.

➤ Ver Cuaderno

Al ir a la opción Ver cuaderno del menú el alumno visualizará el cuaderno completo de ejercicios, el que contiene a lo menos un ejercicio de cada tópico que se ve a lo largo de la asignatura los que podrá resolver cuando así lo desee.

➤ Ver rendimiento

En la opción Ver rendimiento el alumno podrá conocer el porcentaje de los ejercicios resueltos y así conocer cuánto es lo que le falta por resolver.

➤ Ver Problemas Resueltos

Otra opción que es de gran ayuda para el alumno es Ver problemas Resueltos, en esta opción el alumno podrá visualizar todos los ejercicios que ha desarrollado, facilitando el acceso a estos.

➤ Ayuda

En esta opción se le proporciona indicaciones para resolver los problemas.

## 17 ANEXO: ESPECIFICACION DE LAS PRUEBAS

### 17.1 Pruebas de Unidad

#### 17.1.1 <Ingreso de Problemas>

Para probar esta unidad solo se necesita la configuración estándar del computador en términos de hardware y software.

id	Característica a probar							
	tipo_ejercicio	nombre	descripcion	prototipo funcion	valores entrada	valores salida	Datos de entrada	
1	Funcionalidad	Fila	Fila 1	Probar funcionalidad	prototipo de prueba	1 2 3	3 2 1	salida esperada
2	Funcionalidad	Pila	Pila 1					salida obtenida
								Éxito Fracaso
								Observaciones
	Faltan Datos	Datos Ingresados Satisfactoriamente	Datos Ingresados Satisfactoriamente					
	Faltan Datos	Datos Ingresados Satisfactoriamente	Datos Ingresados Satisfactoriamente					
	Éxito		Éxito					
	Sin observaciones		Sin observaciones					

Tabla Especificación de prueba ingresar problema

### 17.1.2 <Ingreso de Casos de Pruebas>

Para probar esta unidad solo se necesita la configuración estándar del computador en términos de hardware y software.

Precondición:

- Debe existir por lo menos un problema para añadir casos de pruebas.

2	1	id	
Funcionalidad	Funcionalidad	Característica a probar	
1 2 3	1 2 3	Valores Entrada	Datos de entrada
	3 2 1	Valores Salida	
Faltan Datos	Datos Ingresados Satisfactoriamente	salida esperada	
Faltan Datos	Datos Ingresados Satisfactoriamente	salida obtenida	
Éxito	Éxito	Éxito Fracaso	
Sin observaciones	Sin observaciones	Observaciones	

4	3
Funcionalidad	Funcionalidad
1 2 3	
	3 2 1
Faltan Datos	Faltan Datos
Datos Ingresados Satisfactoriamente	Faltan Datos
Fracaso	Éxito
Se deben especificar ambos parametros	Sin observaciones

Tabla Especificación prueba ingreso de casos de prueba

### 17.1.3 <Ingreso de Prototipos de Función>

Para probar esta unidad solo se necesita la configuración estándar del computador en términos de hardware y software.

Precondición:

- Debe existir por lo menos un problema para añadir prototipos de función.

	id		
1	Característica a probar		
Funcionalidad	<table border="1"> <tr> <td data-bbox="607 810 782 1058">Prototipo Funcion</td> <td data-bbox="607 564 782 810">Datos de entrada</td> </tr> </table>	Prototipo Funcion	Datos de entrada
Prototipo Funcion	Datos de entrada		
Prototipo de Prueba	salida esperada		
Datos Ingresados Satisfactoriamente	salida obtenida		
Datos Ingresados Satisfactoriamente	Éxito Fracaso		
Éxito	Observaciones		
Sin observaciones			

2
Funcionalidad
Tipo nombreFuncion (parametro 1, parametro 2)
Datos Ingresados Satisfactoriamente
Datos Ingresados Satisfactoriamente
Éxito
Sin observaciones

Tabla Especificación prueba ingreso de prototipo de función

---

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

---

De acuerdo al modelo de datos desarrollado en el proyecto, su diccionario de datos es el siguiente:

- Alumno: Es una entidad del modelo entidad relación. Son los usuarios a los cuales está enfocado el proyecto, ya que son ellos quienes acceden al sistema para ver y resolver el cuaderno de ejercicios que se le asigna.

Los datos que se registran de los alumnos son los siguientes:

- Id\_alumno: es un índice que le asigna el sistema a cada alumno que se registra, por lo que es un valor incremental que es único, lo que significa que no existen los índices iguales.

Está compuesto por números iniciados en 1.

- Nombre: corresponde al nombre alumno.

Está compuesta por caracteres.

- Id\_moodle: Es un identificador que le asigna la plataforma moodle a un usuario registrado.

Está compuesto por un número entero de 10 dígitos.

- Problema: Son los ejercicios que se almacenan en la base de datos para luego ser asignados a cada alumno, sin los problemas no existiría un cuaderno de ejercicios.

Los datos que se requieren del problema para registrar son los siguientes.

- Id\_problema: Es un índice que le asigna el sistema a cada problema que se registra, es un valor incremental y único, por lo que no existen índices iguales.

Está compuesto por números iniciados en 1.

- Tipo\_ejercicio: Se refiere al tópico al cual está relacionado el problema ingresado, estos tópicos pueden ser Listas, Filas, Pilas, Árboles o Grafos.  
Está compuesto con caracteres ya mencionados.
  - Nombre: Se refiere al nombre que se le asigna a un problema, esto depende del profesor o administrador que lo ingrese.  
Está compuesto por caracteres, la cantidad asignada depende de quien la ingrese.
  - Descripción: Se refiere a la explicación detallada del problema, para que el alumno pueda entender lo que se le pide resolver y así desarrollar una buena solución.  
Está compuesta por una gran cantidad de caracteres que varía de acuerdo al problema
  - Id\_profesor: Es un índice que le asigna el sistema a cada profesor que se registra, es un valor incremental y único, por lo que no existen índices iguales.  
Está compuesto por números iniciados en 1.
- Cuaderno: se refiere al sitio en donde el alumno puede visualizar como responder los problemas asignados para él. Este cuaderno es distinto para cada alumno, por lo que no todos pueden desarrollar los mismos ejercicios.

Los datos que se registran de un cuaderno son los siguientes:

- Id\_problema: Es un índice que le asigna el sistema a cada problema que se registra, es un valor incremental y único, por lo que no existen índices iguales.  
Está compuesto por números iniciados en 1.
- Id\_alumno: es un índice que le asigna el sistema a cada alumno que se registra, por lo que es un valor incremental que es único, lo que significa que no existen los índices iguales.

Está compuesto por números iniciados en 1.

- Intentos: se refiere a la cantidad de veces que un alumno trata de resolver el problema para obtener la mejor solución.  
Se almacena el número de veces que este realizo.
- Solución: se refiere al código desarrollado por el alumno en el cual obtiene un menor índice de error.  
Se almacena como archivo en la base de datos.
- Calidad\_solucion: Es el porcentaje de cumplimiento de acuerdo a los casos de prueba y al prototipo de la función.  
Se almacena como porcentaje.

- Profesor: Son las personas que pueden acceder al sistema para ingresar nuevos problemas y casos de prueba, además de poder obtener informes de rendimientos del curso.

Los datos que se registran de los profesores son los siguientes:

- Id\_profesor: Es un índice que le asigna el sistema a cada profesor que se registra, es un valor incremental y único, por lo que no existen índices iguales.  
Está compuesto por números iniciados en 1.
- Nombre: corresponde al nombre del profesor que se ingresa.  
Se ingresa en forma de carácter.
- Id\_moodle: Es un identificador que le asigna la plataforma moodle a un usuario registrado.  
Está compuesto por un número entero de 10 dígitos.

- Caso de prueba: Son valores con los cuales se prueba si la solución que el alumno ingreso es correcta o no, estos van ligados con el problema, ya que para cada

problema deben existir casos de prueba asociados a este, para que el alumno pueda obtener una respuesta con respecto a la calidad de su solución.

Los datos que se registran de los casos de prueba son:

- **Id\_caso:** Es un índice que le asigna el sistema a cada caso de prueba que se registra, es un valor incremental y único, por lo que no existen índices iguales. Está compuesto por números iniciados en 1.
  - **Id\_problema:** Es un índice que le asigna el sistema a cada problema que se registra, es un valor incremental y único, por lo que no existen índices iguales. Está compuesto por números iniciados en 1.
  - **Valor\_entrada:** son valores que se ingresan al sistema para probar el código que el alumno quiere compilar.

Estos valores dependerán del tipo de ejercicio que se desarrolla, ya que el cuaderno contiene ejercicios de distintos tópicos de la asignatura.
  - **Valor\_salida:** son los valores que debe mostrar el código del alumno, correspondiente con los valores de entrada ingresados, de esta forma se realiza una comparación entre los valores de salida reales del código con los que debieron resultar, que son los que el profesor ingresa en este campo.

Estos valores al igual que los valores de entrada dependen del tipo de ejercicio que se ingresa.
- **Prototipo\_funcion:** Es un tipo de ayuda para resolver el ejercicio, ya que el prototipo de función le entrega información importante.

Los datos que se registran de prototipo de función son:

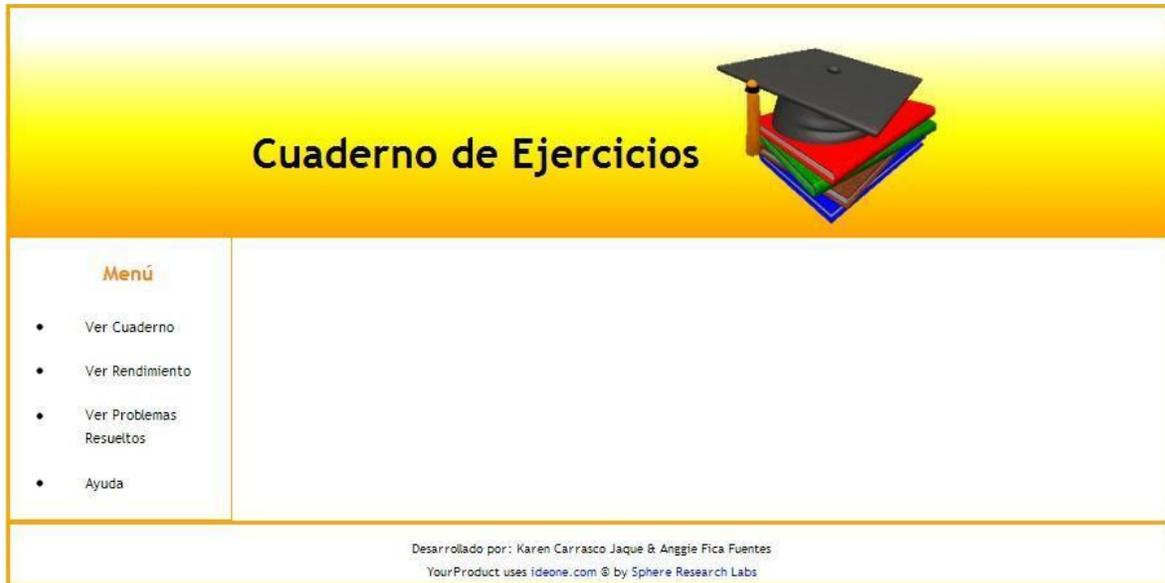
- **Id\_prototipo:** Es un índice que le asigna el sistema a cada prototipo que se registra, es un valor incremental y único, por lo que no existen índices iguales. Está compuesto por números iniciados en 1.

- Nombre: Aquí se especifica el prototipo y es en donde se le indica el tipo de dato que retorna la función, el número, tipo y orden de parámetros que recibe la misma.

Un ejemplo de cómo se almacena nombre es: "int maximo (int x, int y, int z)"

## Ayuda al alumno

Al momento de ingresar al cuaderno de ejercicios los estudiantes de la asignatura Estructura de datos visualizarán un menú de opciones en los cuales podrán interactuar con ellas y las que serán descritas a continuación en mayor profundidad.



Las opciones de este menú son:

- 1 Ver Cuaderno
- 2 Ver Rendimiento
- 3 Ver Problemas Resueltos
- 4 Ayuda

A continuación se describirán de forma detallada cada opción de este menú

- 1 Ver Cuaderno: En esta opción se visualiza un listado de los ejercicios asignados al alumno; estos ejercicios se encuentran separados por cada tópico que se ven en la asignatura, se muestra la descripción del problema y además de la opción de resolver el ejercicio, tal como que se muestra en la siguiente imagen.

**Menú**

- Ver Cuaderno
- Ver Rendimiento
- Ver Problemas Resueltos
- Ayuda

## Listado de Problemas

**• Listas Ordenadas**

Construir Lista Ordenada:  
 Dado un n (n>0), ingresado desde teclado. Construir una lista ordenada e  
 Imprimirla.  
 Se solicita:  
 Utilizar la Estructura:  

```
struct nodo{
```

[Resolver](#)

**• Pila**

Fibonacci:  
 Utilizando la estructura PILA DINÁMICA calcular el N-esimo termino de la  
 sucesión de fibonacci.  
 Se pide utilizar la estructura

[Resolver](#)

**• Fila**

Fila circular:  
 Se pide crear una fila Estática Circular, leyendo un n desde teclado el cual  
 indicará el número de elementos de la fila a crear.  
 Se pide que para imprimir la fila se utilice la siguiente función:  

```
void imprimirfila(int *frente, int *atras, int *cont, int fila[]){
```

[Resolver](#)

**• Árboles Generales**

Recorrido Inorden:  
 Dado un árbol ingresado por teclado mostrar su recorrido en inorden.  
 La estructura del Árbol será  

```
struct arbol{
```

```
    int dato;
```

```
    struct arbol *izq;
```

[Resolver](#)

Una vez que se presiona la opción de resolver el problema, el alumno visualiza una pantalla en la cual se encuentra el tipo de ejercicio, el nombre, el enunciado, la función que debe utilizar al momento de resolver, y el espacio en donde debe ingresar el código completo del programa.

Si en el ejercicio se le pide realizar una función, y necesita utilizar otras funciones que no sean requeridas, debe escribir todas las funciones que se utilizaran en el desarrollo del ejercicio, es decir, debe ingresar el código completo. Además las funciones requeridas para la solución deben ser escritas en el código tal cual se presenta, ya que aunque solo sea con un espacio de diferencia se considera que la función requerida no está presente en el código enviado, por esta razón tampoco se puede cambiar el nombre de las variables definidas en el parámetro, aunque sean del mismo tipo y realicen la misma función.

Por otro lado para una mejor calificación, si el ejercicio es desarrollado previamente en algún compilador, deben fijarse que compile correctamente el archivo guardado en .c, porque el método utilizado para compilar es estricto.

**Menú**

- [Ver Cuaderno](#)
- [Ver Rendimiento](#)
- [Ver Problemas Resueltos](#)
- [Ayuda](#)

## Resolver Problema

Tipo de Problema a resolver:

Nombre Problema:

Enunciado Problema:

Dado un n (n>0), ingresado desde teclado. Construir una lista ordenada e Imprimirla.  
 Se solicita:  
 Utilizar la Estructura:

```

                    struct nodo{
                      int valor;
                      struct nodo *siguiente;
                    };
                    Al momento de imprimir la lista se solicita utilizar la función:
                    void imprimeLista(Nodo x)
                    {
                      Nodo p=NULL;
                    }
                    
```

Funciones Requeridas para su solución:

Nodo Ingresar(Nodo cabeza, Nodo nodo\_nuevo)

Ingrese su solución:

Otra acotación que es relevante cuando presione el botón “Enviar Código” y ya haya ingresado su solución esto quedará registrada como un intento de resolución del ejercicio.

- 2 Ver Rendimiento: En esta opción se podrá visualizar todos los ejercicios asignados, en donde se indica el tipo de ejercicio con el nombre de este, además de las veces que lo ha intentado realizar y la nota más alta que ha obtenido por resolverlo.



## Cuaderno de Ejercicios

	Rendimiento Alumno			
Menú	Tipo de Ejercicio	Nombre	Intentos	Nota
<ul style="list-style-type: none"> <li>• <a href="#">Ver Cuaderno</a></li> <li>• <a href="#">Ver Rendimiento</a></li> <li>• <a href="#">Ver Problemas Resueltos</a></li> <li>• <a href="#">Ayuda</a></li> </ul>	Listas Ordenadas	Construir Lista Ordenada	2	5
	Pila	Fibonacci	0	0
	Fila	Fila circular	0	0
	Árboles Generales	Recorrido Inorden	0	0
	Árboles Binarios	AB	0	0
	ABB	abb	0	0
	Grafo	grafitp	0	0
	Lista	Suma lista dinamica	0	0

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
YourProduct uses ideone.com © by Sphere Research Labs

- 3 Ver Problemas Resueltos: En esta opción verá un listado de todos los ejercicios que ya ha resuelto, en el listado aparecen el tipo de ejercicio, el nombre y además una opción para ver en detalle del ejercicio.



## Cuaderno de Ejercicios

	Listado Problemas		
Menú	Tipo de Ejercicio	Nombre	
<ul style="list-style-type: none"> <li>• <a href="#">Ver Cuaderno</a></li> <li>• <a href="#">Ver Rendimiento</a></li> <li>• <a href="#">Ver Problemas Resueltos</a></li> <li>• <a href="#">Ayuda</a></li> </ul>	Listas Ordenadas	Construir Lista Ordenada	Detalles

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
YourProduct uses ideone.com © by Sphere Research Labs

Al hacer click sobre “Detalles” se visualiza el nombre, la descripción y la solución que se ha evaluado del problema que se ha seleccionado.

**Menú**

- Ver Cuaderno
- Ver Rendimiento
- Ver Problemas Resueltos
- Ayuda

## Ver Solución

**Nombre:**

Construir Lista Ordenada

**Descripción Problema:**

Dado un n (n>0), ingresado desde teclado. Construir una lista ordenada e Imprimirla. Se solicita:

Utilizar la Estructura:

```
struct nodo{
    int valor;
    struct nodo *siguiente;
};
```

Al momento de imprimir la lista se solicita utilizar la función:

```
void imprimeLista(Nodo r)
{
    Nodo p=NULL;
    p=r;
```

**Solución Evaluada:**

```
#include "stdio.h"
#include "stdlib.h"

typedef struct nodo{
    int valor;
    struct nodo *siguiente;
}*Nodo;

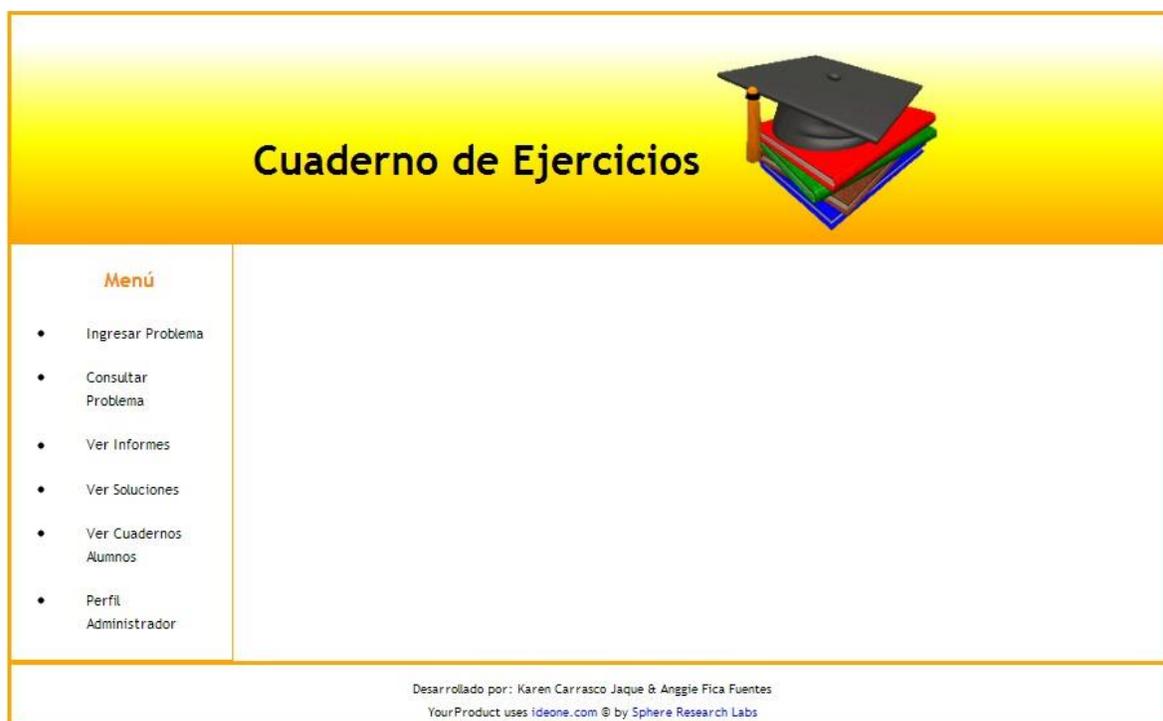
Nodo New()
{
    return((struct nodo*)malloc(sizeof(struct nodo)));
}

void imprimeLista(Nodo r)
{
    Nodo p=NULL;
    p=r;
    while (p != NULL) {
        printf("%d ", p->valor);
        p = p->siguiente;
    }
    printf("NULL");
}
```

- 4 Ayuda: En esta opción del menú se encuentra el detalle de todas las demás pantallas y sus descripciones para facilitar el uso y la comprensión del cuaderno de ejercicios.

## Ayuda al profesor

Una vez que se accede al perfil del profesor se visualiza un nuevo menú de opciones las cuales son descritas en profundidad en el presente documento, la pantalla que se visualiza al iniciar en el perfil profesor es como la que se muestra a continuación:



Las opciones de este menú son:

- 1) Ingresar Problema
- 2) Consultar Problema
- 3) Ver Informes
- 4) Ver Soluciones
- 5) Ver Cuadernos Alumnos
- 6) Perfil Administrador

A continuación se describirán de forma detallada cada opción de este menú.

- 1) **Ingresar Problema:** En esta opción aparecen todos los campos necesarios para almacenar un problema en la base de datos y que luego es asignado por medio de un cuaderno a un alumno. Los campos que se deben completar al momento de ingresar el problema son: el tipo de problema, el nombre del ejercicio, la descripción, el prototipo de función, valores de prueba, tanto de entrada como de salida del ejercicio.

Es importante que al momento de entregarle el enunciado al alumno deba estar claro el orden en que se le proporcionaran y pidan los datos, ya que en este mismo orden se debe ingresar los valores de los datos de prueba (casos de prueba de entrada), cada valor de entrada ingresado debe estar separado entre sí solo por un espacio.

**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

### Ingreso de Problema

Seleccione tipo de problema:  
Listas

Nombre ejercicio:

Descripción problema:

Prototipo de función obligatoria:

Valores de Prueba:

Valores de entrada:

Valores de salida:

**Guardar Problema**

- 2) Consultar Problema: En esta opción se pueden consultar por los problemas que existen en el sistema, para poder realizar esto debe seleccionar el tipo de problema por el que se va a consultar y presionar el botón.



## Cuaderno de Ejercicios

**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

Seleccione tipo de problema:

Listas ▾

Consultar Problema

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
 YourProduct uses ideone.com © by Sphere Research Labs

Una vez realizada la consulta por un tipo de problema se visualiza por pantalla un listado de todos los problemas almacenados indicando el tipo de ejercicio por el cual realizo la consulta, el nombre del problema, una opción para acceder a los detalles del ejercicio y otra opción para actualizar el ejercicio.



## Cuaderno de Ejercicios

**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

### Listado Problemas

Tipo de Ejercicio	Nombre	Ver Detalles	Actualizar
Lista	Suma lista dinamica	Detalles	Actualizar

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
 YourProduct uses ideone.com © by Sphere Research Labs

Si presiona la opción de “Detalles” podrá ver por pantalla el tipo de ejercicio, el nombre del problema, la descripción de este, las funciones requeridas y los casos de pruebas asociados al problema, cabe mencionar que si se intenta modificar algún campo de los que se visualiza no se podrá realizar ya que para eso está la opción actualizar, y esta opción solo permite visualizar un detalle completo del ejercicio seleccionado.

**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

## Ver Problema

Tipo de ejercicio:

Nombre Problema:

Descripción Problema:

Dada una lista dinámica de tamaño n (leído desde teclado), sumar todos los valores presentes en esta, para resolver este problema se pide:

1.- Leer el n desde teclado, mayor a 2.

2.- Utilizar la estructura:

```

struct nodo{
    int dato;
    struct nodo *next;
};
typedef struct nodo NODO;
                    
```

3.- Utilizar la función:

Funciones Requeridas:

```

void creaLista(NODO **L1)
int sumaLista(NODO *r)
void imprimeLista(NODO *r)
                    
```

Casos de Prueba:

Caso de Prueba N° 1  
Valores Entrada: 5 1 1 1 1 1  
Valores Salida: 5 1 1 1 1 1 NULL

Caso de Prueba N° 2  
Valores Entrada: 2 3 4

Si presiona la opción de “Actualizar” visualiza por pantalla los campos tipo de ejercicio, nombre del problema y descripción del problema, además de un botón de actualizar, estos campos si pueden ser modificados y se almacenara sus modificaciones al momento de presionar el botón actualizar que se encuentra ubica bajo estos campos.

En esta pantalla también se encuentran los campos de funciones requeridas y casos de prueba, pero estos a su vez presentan más opciones que se encuentran bajo su cuadro respectivo, estas opciones son las de agregar y modificar, para cada campo aparecen estas opciones de forma similar.

**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

## Actualizar Problema

Tipo de ejercicio:

Nombre Problema:

Descripción Problema:  

Dada una lista dinámica de tamaño n (leído desde teclado), sumar todos los valores presentes en esta, para resolver este problema se pide:

1.- Leer el n desde teclado, mayor a 2.

2.- Utilizar la estructura:

```
struct nodo{
    int dato;
    struct nodo *next;
};
typedef struct nodo NODO;
```

3.- Utilizar la función:

Actualizar Problema

Funciones Requeridas:  

```
void creaLista(NODO **L1)
int sumaLista(NODO *r)
void imprimeLista(NODO *r)
```

[Agregar Funciones](#)   [Modificar Funciones](#)

Casos de Prueba:  

```
Caso de Prueba N° 1
Valores Entrada: 5 1 1 1 1
Valores Salida: 5 1 1 1 1 NULL

Caso de Prueba N° 2
Valores Entrada: 2 3 4
```

[Agregar Casos de Pruebas](#)   [Modificar Casos de Pruebas](#)

Si requiere agregar más de un prototipo de función o caso de prueba al ejercicio se debe presionar “agregar”, si presiono en agregar funciones visualizara todas las funciones que existen para el problema y el espacio para escribir la nueva función que se pedirá en el ejercicio, esto mismo ocurre si la opción que se presiono fue la de agregar casos de prueba, y finalmente se encuentra un botón guardar con el cual almacena en el sistema el nuevo prototipo de función.

Al momento de ingresar un caso de prueba nuevo se recomienda que se considere el orden en el que se le entrego los datos al alumno para que al momento del ingreso de los valores de prueba este fuese en el mismo orden en que lo realizará el alumno, para obtener los resultados adecuados al ejecutar el problema.

## Cuaderno de Ejercicios



**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

### Agregar Funciones Requeridas para Problema: Suma lista dinámica

Funciones Requeridas:

```
void creaLista(NODO **L1)
int sumaLista(NODO *x)
void imprimeLista(NODO *x)
```

Ingrese Nuevo Prototipo:

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
YourProduct uses ideone.com © by Sphere Research Labs

Si la opción que se presiona es modificar un caso de prueba se visualiza una pantalla en donde se muestra el nombre del ejercicio al que se le quiere modificar y se debe seleccionar el caso de prueba al cual le realizara alguna operación.

Las opciones que se encuentran dentro de modificar está actualizar o eliminar, estas se visualizan como botones.

Esto mismo ocurre si la opción que se presiona fue modificar funciones, ya que como se mencionó anteriormente funcionan de forma similar.



## Cuaderno de Ejercicios

**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

### Casos de Prueba para Problema: Suma lista dinamica

Casos de Prueba:

- Caso de Prueba N° 1 Valores de Entrada: 5 1 1 1 1 1 Valores de Salida: 5 1 1 1 1 1 NULL
- Caso de Prueba N° 2 Valores de Entrada: 2 3 4 Valores de Salida: 7 4 3 NULL

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
YourProduct uses ideone.com © by Sphere Research Labs

Al presionar sobre modificar un caso de prueba se visualiza los valores de entrada y de salida que fueron ingresados y un botón actualizar con el cual los cambios realizados serán almacenados en el sistema.



## Cuaderno de Ejercicios

**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

### Modificar Caso de prueba

Valores de Prueba

Valores de entrada:

Valores de Salida:

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
YourProduct uses ideone.com © by Sphere Research Labs

- 3) Ver Informes: En esta opción el profesor puede obtener informes de rendimientos de los avances de los alumnos ya sea de manera individual como a nivel de curso.



The screenshot displays a web interface for 'Cuaderno de Ejercicios'. At the top, there is a yellow header with the title 'Cuaderno de Ejercicios' and an icon of a graduation cap on a stack of books. Below the header, the interface is divided into two main sections. On the left is a 'Menú' (Menu) with a list of options: 'Ingresar Problema', 'Consultar Problema', 'Ver Informes', 'Ver Soluciones', 'Ver Cuadernos Alumnos', and 'Perfil Administrador'. On the right is a section titled 'Seleccione tipo de Informe' (Select report type) with two radio button options: 'Individual' and 'Curso'. The 'Curso' option is selected, and a yellow button labeled 'Ver Informe' is positioned below it. At the bottom of the interface, there is a footer with the text: 'Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes' and 'YourProduct uses ideone.com © by Sphere Research Labs'.

Si la opción que se presiona es la de individual se mostrara en pantalla el listado de alumnos que pertenecen a al curso en donde se debe seleccionar uno y presionar el botón para ver el informe de dicho alumno.



## Cuaderno de Ejercicios

Menú

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

### Seleccionar Alumno

Listado de Alumnos:

paola  
 alumno2  
 alumno3

[Ver Informe](#)

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
 YourProduct uses ideone.com © by Sphere Research Labs

El informe por alumno presenta el nombre del alumno junto a un listado de los ejercicios que ha desarrollado en donde se indica el tipo de ejercicio, el nombre del problema, los intentos y la calidad de solución que ha obtenido por cada ejercicio, este informe se encuentra en formato PDF.

### Informe Rendimiento Individual

Alumno: paola

Tipo Ejercicio	Nombre Problema	Intentos	Calidad Solucion
Listas Ordenadas	Construir Lista Ordenada	2	5

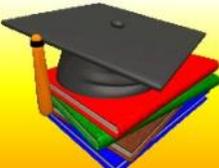
Si el informe que desea ver es el informe por curso visualizara por pantalla un listado con los nombres de todos los alumnos, la cantidad de ejercicios que ha resueltos cada uno de ellos y el promedio por curso. Además de indicar cuál ha sido el mejor y peor tópico dentro de la resolución de los ejercicios por los alumnos y un promedio de la cantidad de ejercicio que realizan los alumnos y de la calidad de solución de ellos. Este informe también se encuentra en formato PDF.

### Informe Rendimiento Curso

Alumno	Cantidad Ejercicios Resueltos	Promedio
alumno2	0	0.0000
alumno3	0	0.0000
paola	1	0.6250

Mejor Topico Curso: Listas Ordenadas  
 Peor Topico Curso: Pila  
 Num. Promedio de Ejercicios Resueltos: 0.3333  
 Promedio Calidad de soluciones: 0.2083

- 4) Ver Soluciones: Mediante esta opción se puede visualizar las soluciones de los ejercicios por el tipo de problema que han resuelto los alumnos, al momento de acceder a esta se debe seleccionar el tipo de problema por el que se desea consultar.



## Cuaderno de Ejercicios

**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

Seleccione tipo de problema:

Listas Ordenadas ▾

Consultar Soluciones

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
 YourProduct uses ideone.com © by Sphere Research Labs

Una vez seleccionada el tipo de problema se muestra un listado de los problemas resueltos de ese tipo en donde se indica el nombre del ejercicio, el nombre del alumno que lo resolvió, la nota que el alumno obtuvo y la opción de ver la solución propuesta por dicho alumno.



## Cuaderno de Ejercicios

**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

### Listado Problemas

Problema	Alumno	Nota	Ver Solución
Construir Lista Ordenada	paola	5	<a href="#">Ver</a>

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
YourProduct uses ideone.com © by Sphere Research Labs

- 5) Ver Cuadernos Alumnos: Mediante esta opción se puede monitorear la construcción de los cuadernos asignados a cada alumno, para esto debe seleccionar al alumno por el cual quiere visualizar su cuaderno y luego presionar para verlo.



## Cuaderno de Ejercicios

**Menú**

- Ingresar Problema
- Consultar Problema
- Ver Informes
- Ver Soluciones
- Ver Cuadernos Alumnos
- Perfil Administrador

### Seleccionar Alumno

Listado de Alumnos:

paola

alumno2

alumno3

[Ver Cuaderno](#)

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
YourProduct uses ideone.com © by Sphere Research Labs

Una vez seleccionado el alumno visualizara una listado de todos los ejercicios que fueron asignados a un alumno en donde aparecen el tipo de ejercicio, el nombre del ejercicio y la opción de ver de forma detallada el problema



## Cuaderno de Ejercicios

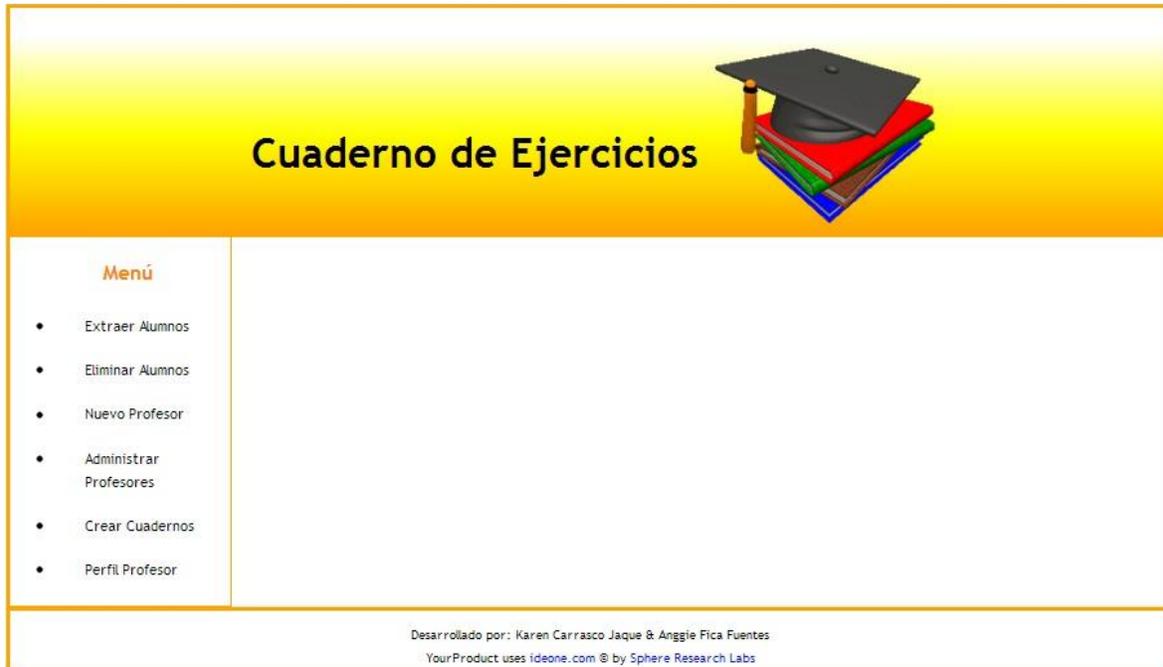
Menú	Cuaderno Alumno: paola																											
<ul style="list-style-type: none"> <li>• Ingresar Problema</li> <li>• Consultar Problema</li> <li>• Ver Informes</li> <li>• Ver Soluciones</li> <li>• Ver Cuadernos Alumnos</li> <li>• Perfil Administrador</li> </ul>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Tipo de Ejercicio</th> <th style="text-align: left;">Nombre</th> <th style="text-align: left;">Detalles Problema</th> </tr> </thead> <tbody> <tr> <td>Listas Ordenadas</td> <td>Construir Lista Ordenada</td> <td><a href="#">Ver Problema</a></td> </tr> <tr> <td>Pila</td> <td>Fibonacci</td> <td><a href="#">Ver Problema</a></td> </tr> <tr> <td>Fila</td> <td>Fila circular</td> <td><a href="#">Ver Problema</a></td> </tr> <tr> <td>Árboles Generales</td> <td>Recorrido Inorden</td> <td><a href="#">Ver Problema</a></td> </tr> <tr> <td>Árboles Binarios</td> <td>AB</td> <td><a href="#">Ver Problema</a></td> </tr> <tr> <td>ABB</td> <td>abb</td> <td><a href="#">Ver Problema</a></td> </tr> <tr> <td>Grafo</td> <td>grafitp</td> <td><a href="#">Ver Problema</a></td> </tr> <tr> <td>Lista</td> <td>Suma lista dinamica</td> <td><a href="#">Ver Problema</a></td> </tr> </tbody> </table>	Tipo de Ejercicio	Nombre	Detalles Problema	Listas Ordenadas	Construir Lista Ordenada	<a href="#">Ver Problema</a>	Pila	Fibonacci	<a href="#">Ver Problema</a>	Fila	Fila circular	<a href="#">Ver Problema</a>	Árboles Generales	Recorrido Inorden	<a href="#">Ver Problema</a>	Árboles Binarios	AB	<a href="#">Ver Problema</a>	ABB	abb	<a href="#">Ver Problema</a>	Grafo	grafitp	<a href="#">Ver Problema</a>	Lista	Suma lista dinamica	<a href="#">Ver Problema</a>
Tipo de Ejercicio	Nombre	Detalles Problema																										
Listas Ordenadas	Construir Lista Ordenada	<a href="#">Ver Problema</a>																										
Pila	Fibonacci	<a href="#">Ver Problema</a>																										
Fila	Fila circular	<a href="#">Ver Problema</a>																										
Árboles Generales	Recorrido Inorden	<a href="#">Ver Problema</a>																										
Árboles Binarios	AB	<a href="#">Ver Problema</a>																										
ABB	abb	<a href="#">Ver Problema</a>																										
Grafo	grafitp	<a href="#">Ver Problema</a>																										
Lista	Suma lista dinamica	<a href="#">Ver Problema</a>																										

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
 YourProduct uses ideone.com © by Sphere Research Labs

6) Perfil Administrador: Esta opción está dada para poder volver a tener el menú de administrador en el sistema.

## Ayuda al administrador

Al acceder al sistema SAED con el perfil de profesor lo primero que se visualiza son las opciones del administrador, ya que todo profesor puede necesitar de estas opciones en algún momento del curso. Las cuales se visualizan en la siguiente imagen



Las opciones de este menú son:

- 1) Extraer Alumnos
- 2) Eliminar Alumnos
- 3) Nuevo Profesor
- 4) Administrar Profesores
- 5) Crear Cuadernos
- 6) Perfil Profesor

A continuación se describirán de forma detallada cada opción de este menú.

- 1) Extraer Alumnos: En esta opción del menú el profesor podrá extraer todos los alumnos matriculados en el curso Estructura de datos. En la pantalla se muestra el código de la asignatura el cual es extraído automáticamente de la PVA.



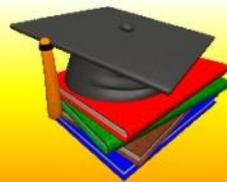
## Cuaderno de Ejercicios

<b>Menú</b> <ul style="list-style-type: none"><li>• Extraer Alumnos</li><li>• Eliminar Alumnos</li><li>• Nuevo Profesor</li><li>• Administrar Profesores</li><li>• Crear Cuadernos</li><li>• Perfil Profesor</li></ul>	<h3>Extraer Alumnos</h3> <p>ID del curso en moodle:</p> <input type="text" value="620138"/> <input type="button" value="Extraer Alumnos"/>
--	---

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
YourProduct uses ideone.com © by Sphere Research Labs

- 2) Eliminar Alumnos: El profesor puede eliminar todos los alumnos que se encuentren en el sistema. Se visualiza con un botón el cual al presionarlo eliminará todos los usuarios existentes.

Es recomendable utilizar esta opción al finalizar el semestre como modo de limpieza del sistema.

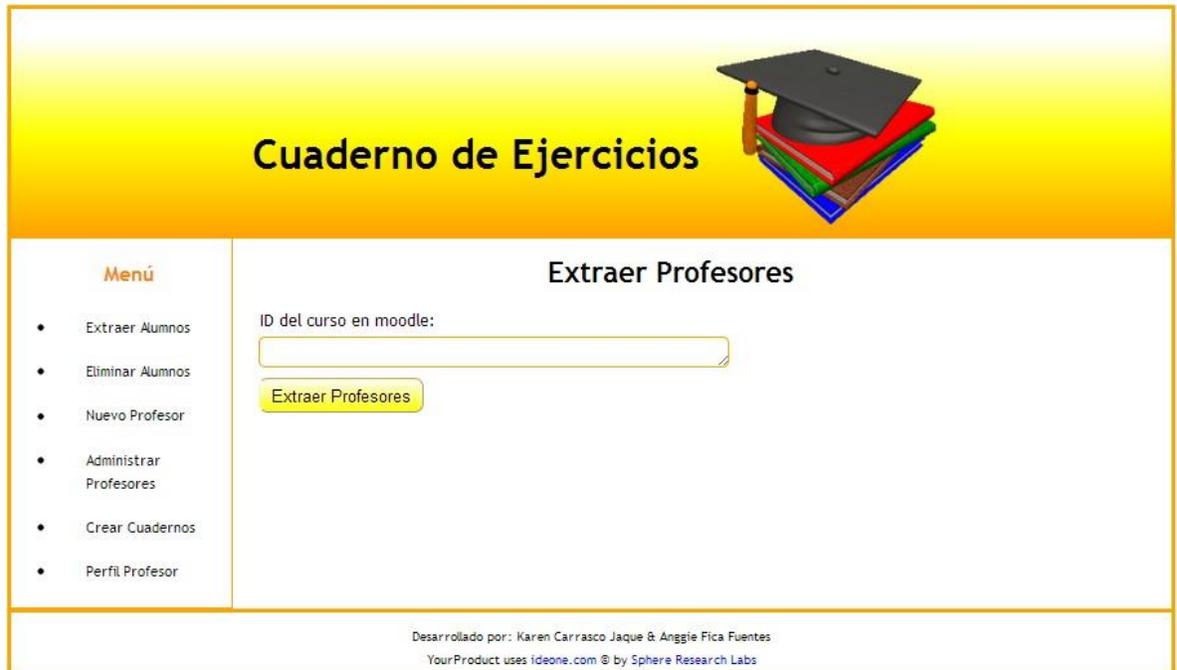


## Cuaderno de Ejercicios

<b>Menú</b> <ul style="list-style-type: none"><li>• Extraer Alumnos</li><li>• Eliminar Alumnos</li><li>• Nuevo Profesor</li><li>• Administrar Profesores</li><li>• Crear Cuadernos</li><li>• Perfil Profesor</li></ul>	<h3>Eliminar Todos los Alumnos</h3> <input type="button" value="Eliminar Alumnos"/>
--	---

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
YourProduct uses ideone.com © by Sphere Research Labs

- 3) Nuevo Profesor: Esta opción será utilizada si en el curso Estructura de datos existen más de un profesor que dicten la asignatura, en este caso el profesor que se encuentra vigente será el encargado de extraer a los demás profesores, ingresando el código del curso del profesor que quiere extraer.



**Cuaderno de Ejercicios**

**Menú**

- Extraer Alumnos
- Eliminar Alumnos
- Nuevo Profesor
- Administrar Profesores
- Crear Cuadernos
- Perfil Profesor

**Extraer Profesores**

ID del curso en moodle:

Extraer Profesores

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes  
YourProduct uses ideone.com © by Sphere Research Labs

- 4) Administrar Profesores: En esta opción se visualiza un listado con todos los profesores que se encuentran en el sistema, además muestra el estado en el que se encuentran y da la opción de modificar el estado de los profesores.



## Cuaderno de Ejercicios

<p style="text-align: center;"><b>Menú</b></p> <ul style="list-style-type: none"> <li>• Extraer Alumnos</li> <li>• Eliminar Alumnos</li> <li>• Nuevo Profesor</li> <li>• Administrar Profesores</li> <li>• Crear Cuadernos</li> <li>• Perfil Profesor</li> </ul>	<h3 style="text-align: center;">Listado Profesores</h3> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #f4a460;">Nombre</th> <th style="background-color: #f4a460;">Estado</th> <th style="background-color: #f4a460;">Actualizar</th> </tr> </thead> <tbody> <tr> <td>anggie</td> <td>Vigente</td> <td>Cambiar Estado</td> </tr> </tbody> </table>	Nombre	Estado	Actualizar	anggie	Vigente	Cambiar Estado
Nombre	Estado	Actualizar					
anggie	Vigente	Cambiar Estado					

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes.  
YourProduct uses ideone.com © by Sphere Research Labs

- 5) Crear Cuadernos: En esta opción se mostrara en pantalla todos los tipos de ejercicios que se encuentran en el sistema y la cantidad existente por cada uno. Además aparece un botón el cual al presionarlo dará inicio a la creación de los distintos cuadernos a los alumnos del curso.



## Cuaderno de Ejercicios

<p style="text-align: center;"><b>Menú</b></p> <ul style="list-style-type: none"> <li>• Extraer Alumnos</li> <li>• Eliminar Alumnos</li> <li>• Nuevo Profesor</li> <li>• Administrar Profesores</li> <li>• Crear Cuadernos</li> <li>• Perfil Profesor</li> </ul>	<h3 style="text-align: center;">Cantidad de Ejercicios por Tipo</h3> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #f4a460;">Tipo de Ejercicio</th> <th style="background-color: #f4a460;">Cantidad</th> </tr> </thead> <tbody> <tr><td>ABB</td><td>2</td></tr> <tr><td>Árboles Binarios</td><td>1</td></tr> <tr><td>Árboles Generales</td><td>1</td></tr> <tr><td>Fila</td><td>1</td></tr> <tr><td>Grafo</td><td>2</td></tr> <tr><td>Lista</td><td>1</td></tr> <tr><td>Listas Ordenadas</td><td>1</td></tr> <tr><td>Pila</td><td>1</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">Se asignará un ejercicio por tipo a cada cuaderno</p> <div style="text-align: center;"> <input type="button" value="Asignar Ejercicios a Cuadernos"/> </div>	Tipo de Ejercicio	Cantidad	ABB	2	Árboles Binarios	1	Árboles Generales	1	Fila	1	Grafo	2	Lista	1	Listas Ordenadas	1	Pila	1
Tipo de Ejercicio	Cantidad																		
ABB	2																		
Árboles Binarios	1																		
Árboles Generales	1																		
Fila	1																		
Grafo	2																		
Lista	1																		
Listas Ordenadas	1																		
Pila	1																		

Desarrollado por: Karen Carrasco Jaque & Anggie Fica Fuentes.  
YourProduct uses ideone.com © by Sphere Research Labs

- 6) Perfil Profesor: Esta opción nos re direcciona al perfil propiamente tal del profesor el cual cuenta con un menú distinto al descrito, el cual se describirá a continuación