

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**“Implementación de un conjunto de objetos de aprendizaje y una aplicación Web de cuestionarios dinámicos integrados en una herramienta de soporte para el autoaprendizaje de los conceptos de programación básica dirigido a estudiantes de la Facultad de Ingeniería en Electricidad y Computación.”**

**TESIS DE GRADO**

Previo a la obtención del Título de:

**INGENIERO EN COMPUTACIÓN  
ESPECIALIZACIÓN EN SISTEMAS TECNOLÓGICOS**

Presentada por:

**VERÓNICA JANNETH CARRASCO IDROVO**

**JORGE HENRY GUEVARA SANTAMARÍA**

**GUAYAQUIL – ECUADOR**

2009

# AGRADECIMIENTO

A Dios por permitirnos llegar a culminar esta nueva etapa de nuestras vidas.

A nuestras familias, por su cariño, y motivación proporcionados durante  
nuestra vida universitaria.

A nuestra Directora de Tesis, Carmita, por su tiempo invertido, en el  
desarrollo de este trabajo de tesis y por inculcarnos el deseo de conocer más  
acerca de las tecnologías Web.

A los profesores que colaboraron con sus conocimientos y sugerencias para  
la elaboración de esta tesis.

# DEDICATORIA

A Dios por ser el pilar fundamental de nuestras vidas.

A nuestras familias por siempre apoyarnos en todas las metas que nos  
hemos propuesto.

A Carmita por todos sus conocimientos transmitidos.

A nuestros amigos.

# TRIBUNAL DE GRADUACIÓN

  
Ing. Jorge Aragundi  
SUBDECANO DE LA FACULTAD

  
Msc. Carmen Vaca R.  
DIRECTOR DE TESIS

  
Dr. Xavier Ochoa Ch.  
MIEMBRO DEL TRIBUNAL

  
Ing. Cristina Abad R.  
MIEMBRO DEL TRIBUNAL

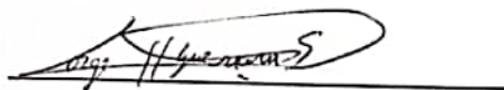
# DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(Reglamento de Graduación de la ESPOL).



Verónica Janneth Carrasco Idrovo



Jorge Henry Guevara Santamaría

# RESUMEN

El presente proyecto muestra el desarrollo de una aplicación que será de utilidad, como herramienta de autoestudio, a los estudiantes de los cursos introductorios de programación ofertados por la FIEC, ya que como valor agregado utiliza la metodología de Objetos de Aprendizaje para su implementación y estándares relacionados con estos.

Esta herramienta provee al estudiante la mayor cantidad ejemplos posibles-y su retroalimentación respectiva- usando la modalidad de cuestionarios y ejecución (compilación) de código para que sus habilidades en la solución de problemas mejoren. También el sistema cuenta con un módulo de teoría, que facilitará al estudiante el recordar o reforzar el conocimiento adquirido en clases.

Por otro lado la solución permite a los profesores la administración de los ejemplos y la teoría que se desea publicar y además permite la visualización de las estadísticas de las prácticas realizadas por los estudiantes, lo que les permitirá tomar decisiones sobre la necesidad de reforzar determinados temas referentes al currículum de la materia.

En el Capítulo 1 se expone el planteamiento del problema que el presente proyecto de tesis busca solucionar.

En el Capítulo 2 se establecen los objetivos y alcance propuestos para este proyecto y además se revisan los fundamentos teóricos respecto a los Objetos de Aprendizaje puesto que van a ser utilizados para el desarrollo de esta herramienta.

En el capítulo 3 se detalla la implementación de los Objetos de Aprendizaje, es decir la arquitectura que se aplica para el desarrollo de la herramienta; la plataforma sobre la que va a trabajar, y la forma en que se maneja la administración del aplicativo.

En el capítulo 4 se realizan las mediciones y se analizan los resultados de las pruebas realizadas a estudiantes de la materia de Fundamentos de programación utilizando la herramienta desarrollada.

# ÍNDICE GENERAL

AGRADECIMIENTO

DEDICATORIA

TRIBUNAL DE GRADUACIÓN

DECLARACIÓN EXPRESA

RESUMEN

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

CAPITULO 1

1.	Planteamiento del problema.....	17
1.1	Problemas en cursos introductorios de programación.....	17
1.2	Características de los programadores Novatos.....	20
1.3	Visión general de la solución propuesta.....	22
1.4	Aplicaciones Ricas en Internet.....	22

CAPITULO 2

2.	Especificación del proyecto.....	26
2.1	Objetos de Aprendizaje.....	27
2.1.1	Definición.....	27
2.1.2	Beneficios.....	27
2.1.3	Estándares.....	28

2.1.4	Descripción del Proyecto.....	29
2.2	Objetivos del proyecto .....	30
2.2.1	Objetivo general .....	30
2.2.2	Objetivos Específicos .....	31
2.3	Alcance .....	32

### CAPITULO 3

3.	Implementación de los objetos de aprendizaje.....	39
3.1	Plataforma .....	39
3.1.1	Interfaz de Usuario .....	40
3.1.2	Acceso a datos.....	42
3.1.3	Compilador.....	47
3.2	Arquitectura de la aplicación.....	48
3.3	Objetos de aprendizaje .....	53
3.3.1	Cuestionarios Dinámicos.....	53
3.3.2	Compilación de código .....	59
3.3.3	Revisión de conceptos .....	63
3.3.4	Errores de compilación.....	65
3.4	Administración .....	66

### CAPITULO 4

4.	Pruebas, mediciones y resultados.....	75
4.1	Descripción de plan de pruebas .....	75
4.2	Tiempo de descarga y optimización.....	76
4.3	Evaluación de los objetos de aprendizaje.....	79

4.4	Estadísticas producidas por el sistema.....	87
-----	---	----

CONCLUSIONES Y RECOMENDACIONES

ANEXOS

BIBLIOGRAFÍA

# ÍNDICE DE FIGURAS

Figura 1.1.- Cuadro estadístico de estudiantes registrados en la materia Fundamentos de Programación durante el año 2007 .....	20
Figura 1.2.- Estadísticas de empresas que solicitan profesionales para desarrollar aplicaciones RIA usando Silverlight o Flex .....	24
Figura 1.3.- Visualizador de Horarios de Clases(Schedule Viewer).....	25
Figura 2.1.- Modelo Mental del módulo de Teoría .....	35
Figura 2.2.- Modelo Mental del módulo Cuestionario.....	36
Figura 2.3.- Modelo Mental del módulo Compilación .....	37
Figura 3.1.- Entorno de trabajo de Flex .....	42
Figura 3.2.- RPC - Remote Procedure Call.....	43
Figura 3.3.- Esquema de acceso a datos Cliente - Servidor.....	45
Figura 3.4.- Código que muestra detalle de la etiqueta HttpServices y una función que contacta al Servlet.....	46
Figura 3.5.- Código del LoginServlet.....	47
Figura 3.6.- Arquitectura de la aplicación.....	49
Figura 3.7a.- Componente Tree creado en Flex .....	51
Figura 3.8.- Vista de pantalla con lista de cuestionario.....	54
Figura 3.9.- Vista de pantalla para resolución de cuestionario .....	55
Figura 3.10.- Vista de panel con el tiempo y datos de la resolución del cuestionario.....	57
Figura 3.11.- Vista de panel con la retroalimentación.....	58
Figura 3.12.- Vista de ventana con el cuestionario resuelto .....	59
Figura 3.13.- Vista de ventana con lista de ejercicios.....	60

Figura 3.14.- Vista de ventana con editor de código C. ....	61
Figura 3.15.- Vista de respuesta de compilación de código.....	62
Figura 3.16.- Vista de ventana con temas previamente ingresados .....	63
Figura 3.17.- Vista de ventana de revisión de teoría.....	64
Figura 3.18.- Vista de ventana de temas de teoría .....	67
Figura 3.19.- Vista de editor de texto para ingreso de Teoría .....	68
Figura 3.20.- Vista de mantenimiento de cuestionario .....	69
Figura 3.21.- Ingreso de parámetros para el cuestionario .....	70
Figura 3.22.- Menú de Sentencias de Control Establecidas para el módulo Código .....	71
Figura 3.23.- Pantalla inicial de mantenimiento de ejercicios correspondiente a la opción Condicionales .....	71
Figura 3.24.- Vista del panel con datos del LOM .....	73
Figura 4.1.- Módulos de la aplicación CWDI .....	78
Figura 4.2.- Configuración del Compilador de FLEX.....	79
Figura 4.3.- Evaluación del Módulo de Cuestionarios por parte de Alumnos de Fundamentos de Programación.....	81
Figura 4.4.- Estadísticas correspondientes a la evaluación de las Calidad de los Contenidos del Cuestionario por parte de los estudiantes. ....	82
Figura 4.5.- Estadísticas correspondientes a la evaluación de de la Retroalimentación por parte de los estudiantes. ....	83
Figura 4.6.- Estadísticas correspondientes a la evaluación de Motivación que ofrece el Cuestionario, por parte de los estudiantes. ....	84
Figura 4.7.- Estadísticas correspondientes a la evaluación del Diseño y presentación los Contenidos del Cuestionario por parte de los estudiantes. ....	85
Figura 4.8.- Estadísticas correspondientes a la evaluación de la Usabilidad del Cuestionario por parte de los estudiantes. ....	86

Figura 4.9.- Evaluación General de la herramienta por parte de Alumnos de Fundamentos de Programación.....	87
Figura 4.10.- Reporte de la evaluación realizada a los alumnos de Fundamentos de Programación Semestre II-2008 .....	88
Figura 4.11.- Estadística de la evaluación realizada a los alumnos de Fundamentos de Programación Semestre II-2008 .....	89
Figura A.1.- Esquema del estándar LOM.....	95
Figura D.1.a.- Diagrama de Casos de Uso .....	97
Figura D.1.b.- Continuación del Diagrama de Casos de Uso .....	98

# ÍNDICE DE TABLAS

Tabla C.1.- Tabla de Errores Generales de Compilación con GCC.....	97
--	----

# INTRODUCCIÓN

Durante largo tiempo los estudiantes que han tomado la materia Fundamentos de Programación, se han visto con dificultades para aprobarla, por lo que generalmente cada semestre alrededor de un 50% de alumnos la abandona y/o reprueban.

Actualmente los alumnos de los cursos introductorios de programación que dicta la FIEC, cuentan con herramientas de estudio como material de apoyo al dictado de clases por el profesor, los ejercicios que éste plantea en clases, los deberes que se envían, ayudantías, y recursos que se publican en el METIS. Para el caso de las ayudantías no muchos estudiantes pueden acudir a ellas debido a problemas de horario, pues Fundamentos de Programación no es la única materia que toman durante el semestre, este inconveniente conlleva a que los alumnos no tengan el refuerzo necesario para comprender el contenido de la materia.

La meta de este proyecto de tesis es proveer a los estudiantes de los cursos de programación básica una herramienta que les permita reforzar sus conocimientos de la materia y que además les permita mejorar sus habilidades en la misma. Todo esto a través de la implementación de Objetos de Aprendizaje y del desarrollo de una herramienta Web que permita tener acceso a estos recursos para realizar prácticas relacionadas con el contenido

de la materia en cualquier momento. Los objetos de aprendizaje serán ingresados al sistema por usuarios que dominan los conceptos relacionados a programación asegurando un material de mayor calidad.

Adicionalmente, a través de la interface web, los profesores contarán con las estadísticas de cada una de las prácticas que han realizado sus estudiantes, lo cual les facilitará tomar decisiones con respecto a qué punto del contenido de la materia requiere una mayor atención.

# CAPITULO 1

## 1. Planteamiento del problema

### 1.1 Problemas en cursos introductorios de programación

Aprender a programar por lo general se considera difícil, y los cursos de programación básica suelen tener altas tasas de deserción. Incluso se ha dicho, que se requieren unos 10 años para que un novato se convierta en un experto programador (1).

La Universidad de Helsinki en el año 2006 realizó un estudio acerca del porcentaje de deserción de los estudiantes en cursos introductorios de programación. Se concluyó que, de los estudiantes registrados (500 a 600 estudiantes), alrededor del 30% al 50% abandonan o pierden el

curso. Se determinó que los problemas más frecuentes entre los estudiantes eran (2):

- Falta de tiempo, la resolución de ejercicios requería demasiado tiempo, tiempo que no tenían pues debían concentrarse en otras materias,
- Falta de técnicas de autoaprendizaje, muchos estudiantes que toman la materia están empezando la universidad y no han aprendido aún a organizar el tiempo de forma eficiente.
- No sabían cómo resolver los problemas, por falta de motivación o porque no estaban acostumbrados a hacer investigación.

De acuerdo con los datos tomados de los cursos de Fundamentos de Programación dictados por la Facultad de Ingeniería Eléctrica y Computación correspondiente a los términos del 2007 podemos ver que los porcentajes de alumnos reprobados (Figura 1.1) se encuentran dentro de los porcentajes indicados por el estudio expuesto anteriormente.

De acuerdo a la Figura 1.1 en el I Término del año 2007, reprobaron el 54% de estudiantes. En esa cifra se incluyen tanto los alumnos que desertaron como los que la reprobaron, pues es difícil establecer el

indicador correspondiente de los alumnos que han optado por retirarse de la materia después de dar el primer examen parcial, debido a que no anulan el curso.

Podemos ver en la Figura 1.1 que en el III término del 2007 el porcentaje de alumnos que han reprobado y/o desertado es más bajo. Esto probablemente es debido a que en el término vacacional se suele tomar una materia y por lo tanto el tiempo que el estudiante dedica es mayor. Adicionalmente, los alumnos que optan por tomar esta materia en vacaciones generalmente ya tienen ciertos conocimientos previos del contenido de la misma debido a que están repitiendo el curso.

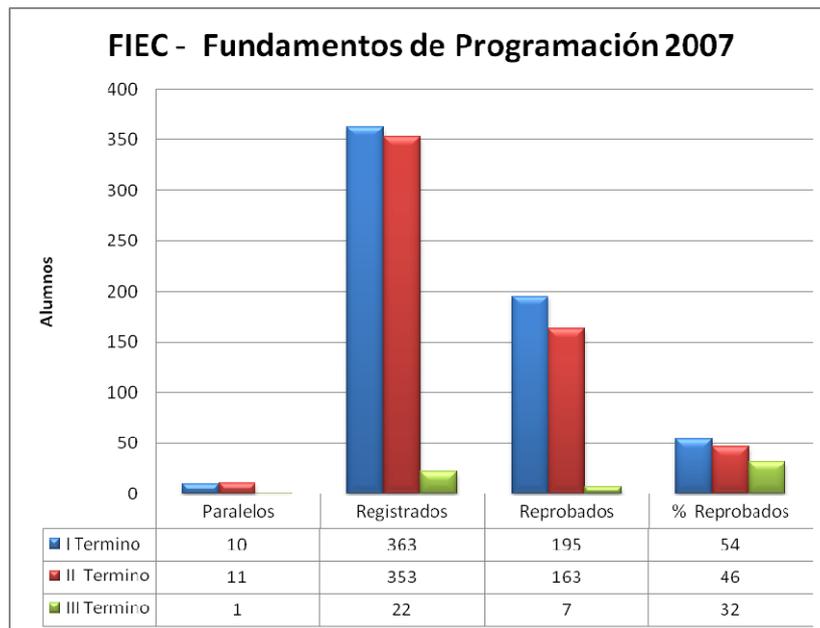


Figura 1.1.- Cuadro estadístico de estudiantes registrados en la materia  
Fundamentos de Programación durante el año 2007

## 1.2 Características de los programadores Novatos

Los programadores novatos tienen ciertas características que han sido identificadas en algunos estudios realizados a estudiantes de cursos introductorios de programación. Por ejemplo, los nuevos programadores invierten poco tiempo en la planificación y pruebas del código, para corregir sus programas. Al corregir el código prueban siempre modificaciones pequeñas en lugar de pensar en una reformulación de la solución implementada. Además, a menudo, no logran aplicar

correctamente los conocimientos que han adquirido en clases. De hecho, un promedio de los estudiantes no suele hacer grandes progresos en un curso introductorio de programación (1).

En un curso de programación, los distintos comportamientos de los estudiantes reflejan modelos que pueden ser reconocidos. En [2] se identifica dos tipos principales de novatos: “bloqueados” y “preactivos”. En una situación problemática los “bloqueados” simplemente se detienen y abandonan todo intento de resolver el problema por sí mismo, mientras que los “preactivos” siguen intentando, modificando su código y recibiendo una retroalimentación de sus errores de manera efectiva.

Con todo, existen novatos eficaces e ineficaces, es decir, estudiantes que aprenden sin esfuerzo excesivo y los que para aprender requieren excesiva atención personalizada. La capacidad de aprendizaje depende de la motivación que tenga el estudiante y de cómo el profesor pueda reconocer esto (1).

Dado que los problemas en los programadores novatos han sido reconocidos por varios estudios cuyos resultados han sido publicados con anterioridad, es necesario crear herramientas que faciliten la experiencia de los estudiantes en esta etapa de su formación profesional. La siguiente sección propone el desarrollo de tal herramienta.

### **1.3 Visión general de la solución propuesta**

Tomando en cuenta los puntos tratados anteriormente, podemos ver que es necesario proveer de herramientas de estudio, que permitan a los estudiantes de los cursos introductorios de programación mejorar sus habilidades en esta materia.

La herramienta que proponemos debe proveer al estudiante la mayor cantidad de ejemplos posibles-y su retroalimentación respectiva- para que el esfuerzo invertido en la solución de problemas tenga resultados más satisfactorios. También el sistema debe contar con un módulo que incluya conceptos sobre el material del curso, asegurando una mejor comprensión de los fundamentos teóricos de la materia.

Por otro lado, la solución deberá facilitar a los profesores la administración de los ejemplos y la teoría que se desea publicar y además debe permitir la visualización de las estadísticas de las prácticas realizadas por los estudiantes, lo que permitirá la toma de decisiones basada en determinados temas referentes al currículum de la materia.

### **1.4 Aplicaciones Ricas en Internet**

Al estudiar las alternativas para escoger la plataforma en la que se desarrollará la mencionada herramienta, observamos que una aplicación

rica en Internet (RIA – Rich Internet Application) sería la mejor elección por la popularidad que tienen este tipo de soluciones entre la gente joven actualmente y por las ventajas de independencia de plataforma y facilidad de acceso desde cualquier lugar que toda aplicación web ofrece. Una aplicación Rica en Internet (RIA) es la combinación entre aplicaciones Web, aplicaciones tradicionales y capacidad multimedia, que busca el mejoramiento de la experiencia visual de la aplicación por parte del usuario -permitiéndole hacer uso de ella de la manera más sencilla- y por otro lado , mejorar la conectividad y el tiempo de despliegue de la aplicación. El desarrollo de RIAs ha venido creciendo durante los últimos años, debido a las nuevas plataformas para el desarrollo de estas, tales como Adobe Flex, Silverlight (Microsoft), JavaFX Script(3). La Figura 1.2 muestra un gráfico comparativo del uso de las plataformas Silverlight y Adobe Flex y cómo ha ido creciendo la demanda de programadores que conozcan estas tecnologías desde Enero del 2005 hasta Septiembre del 2008

## "Adobe Flex", Silverlight Job Trends

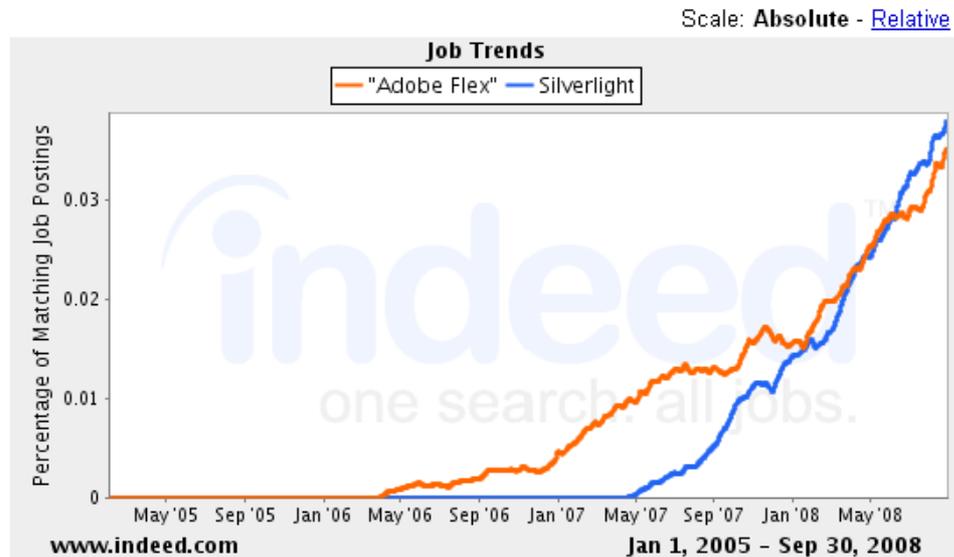


Figura 1.2.- Estadísticas de empresas que solicitan profesionales para desarrollar aplicaciones RIA usando Silverlight o Flex

La ESPOL conocedora de las nuevas tendencias tecnológicas ha empezado a desarrollar Aplicaciones Ricas en Internet. Uno de los más recientes proyectos que implementa un RIA es el generador de Horarios de Clases, pues a través de un diseño sencillo, permite al usuario de manera intuitiva consultar su horario de clases y además ver cierto tipo de información referente a la materia con solo el simple paso del ratón sobre las celdas (Figura 1.3).

*Schedule Viewer*

1	MATEMATICA	PSICOLOGIA	CIENCIAS	1	
2	TECNOLOGIA	TECNOLOGIA	FISICA		
3	LENGUAJE	INGLES	INGLES	LENGUAJE	HISTORIA
4	TECNOLOGIA	LENGUAJE	FRANCES	BIOLOGIA	
5	H. COGNITIVAS	INGLES	LENGUAJE	MATEMATICA	MATEMATICA

Matricula:

Clases  
 Examen

[Ver Horario!](#)

(Schedule Viewer)  
Schedule Viewer es una adecuación del proyecto final

Powered by **taws**

Figura 1.3.- Visualizador de Horarios de Clases(Schedule Viewer)

Dado que la herramienta a desarrollarse es fundamentalmente educativa es necesario tener en cuenta estrategias exitosas que han sido diseñadas por expertos en el área de educación. Una de estas estrategias consiste en modelar la aplicación utilizando módulos que representen Objetos de Aprendizaje(OA). El capítulo siguiente describe conceptos relacionados a OAs.

# CAPITULO 2

## 2. Especificación del proyecto

Para la implementación de la aplicación web a desarrollar se trabajará bajo la perspectiva dada por la teoría de los Objetos de Aprendizaje, que permite desarrollar herramientas efectivas para mejorar la experiencia educativa en cualquier ámbito. Todos los componentes de la aplicación serían un objeto de aprendizaje, por lo tanto es importante conocer la definición y beneficios que estos nos ofrecen. Las secciones siguientes incluyen esta información.

## 2.1 Objetos de Aprendizaje

### 2.1.1 Definición

Un Objeto de Aprendizaje (OA) es una entidad, digital o no digital, que puede ser usada para aprendizaje, educación o entrenamiento(4). Es decir abarca principalmente a un conjunto de materiales digitales los que como unidad o agrupación permiten o facilitan alcanzar un objetivo educacional por ejemplo pequeñas páginas Web, imágenes, textos(5).

### 2.1.2 Beneficios

Los beneficios que los objetos de aprendizaje pueden tener en un contexto educativo son: (5)

- **Flexibilidad:** ya que el mismo recurso puede utilizarse en distinto contextos.
- **Administración:** del contenido: se facilita por que los recursos están descritos con metadatos que permiten su control.
- **Adaptabilidad:** se facilita al diseñador poder seleccionar y componer recursos según la aplicación.
- **Código abierto:** se eliminan problemas de incompatibilidad entre plataformas

### 2.1.3 Estándares

En el ámbito del aprendizaje electrónico (e-learning), para la descripción de un OA, se han desarrollado dos estándares [5 ]:

- **SCORM** (del inglés Sharable Content Object Reference Model) es una especificación que permite crear objetos pedagógicos estructurados. Los sistemas de gestión de contenidos Web originales usaban formatos propietarios para los contenidos que distribuían. Como resultado, no era posible el intercambio de tales contenidos. Con SCORM se hace posible el crear contenidos que puedan importarse dentro de sistemas de gestión de aprendizaje diferentes, siempre que estos soporten la norma SCORM (6).

1. **IEEE LOM (*Learning Object Metadata*)(IEEE, 2002):** En LOM se especifica la sintaxis y la semántica de los atributos necesarios para describir los objetos de aprendizaje. Este estándar está compuesto de nueve categorías de metadatos, que agrupan elementos con los que se desea proveer una descripción completa de los recursos educativos (6).

Para este proyecto de tesis se ha optado por trabajar con el estándar LOM debido a que está diseñado para guardar

referencias de un Objeto de Aprendizaje, además de poseer un esquema sencillo para generar un OA.

En el Anexo A se muestra el diseño propuesto para el estándar LOM

#### **2.1.4 Descripción del Proyecto**

Basados en los argumentos presentados en el Capítulo 1, acerca del problema que representa para los estudiantes aprender a programar, se plantea el desarrollo de una herramienta que ayudará a los estudiantes de Fundamentos de Programación a mejorar sus habilidades en esta materia.

La herramienta contará con tres componentes, los cuales serán:

- Cuestionario(Evaluación): Proveerá la mayor cantidad de ejemplos posibles con respecto a la materia a través de preguntas objetivas y su retroalimentación respectiva.
- Compilación (Ejecución) de Código: Contará con ejemplos planteados por el profesor con el fin de que el estudiante cree una solución al problema expuesto, para que sus habilidades en la solución de problemas aumenten y sean más precisas.
- Teoría: Facilitará al estudiante el recordar o reforzar algún conocimiento adquirido en clases.

Cada uno de estos componentes-Cuestionarios, Compilación, Teoría- serán implementados para que generen OAs siguiendo uno de los estándares mencionados en la sección 2.1.3, dando lugar a que puedan ser reutilizados o compartidos con alguna otra plataforma que tenga como fin la enseñanza y que utilice el mismo estándar.

El profesor tendrá una interfaz donde podrá ingresar la teoría, cuestionarios y ejercicios para la compilación de código C

## **2.2 Objetivos del proyecto**

### **2.2.1 Objetivo general**

Implementar una herramienta Web de cuestionarios dinámicos y un conjunto de objetos de aprendizaje que serán integrados en una solución que constituya un soporte en el aprendizaje de conceptos presentados en un curso de programación básica para los estudiantes de la FIEC, con el fin de reforzar los conocimientos de los alumnos que toman estos cursos y servir como herramienta complementaria para los profesores.

### **2.2.2 Objetivos Específicos**

- Determinar las sentencias de control que se van a implementar para el desarrollo de la herramienta de aprendizaje.
- Desarrollar una herramienta que genera objetos de aprendizaje basados en las sentencias de control establecidas para el desarrollo de la herramienta.
- Desarrollar una interfaz en la que el profesor puede configurar los cuestionarios que el sistema generará como objetos de aprendizaje ingresando las preguntas y especificando la categoría y subcategoría a la que pertenecen y su nivel de dificultad.
- Implementar opciones para el ingreso de ejemplos en lenguaje C por parte del profesor permitiéndole establecer dos niveles de ayuda para cada ejemplo que ingrese. Estos niveles de ayuda permitirán al estudiante ver la solución propuesta por el profesor por partes, para permitirle pensar en una alternativa antes de visualizar la solución final.
- Generar dinámicamente cuestionarios de acuerdo a los ejemplos ingresados por el profesor. En estos cuestionarios las preguntas no se repetirán para un

mismo estudiante y las posibles soluciones deben auto-generarse por cada pregunta, de tal manera que aparezcan opciones distintas cada vez que el sistema plantea una pregunta.

- Generar gráficos estadísticos de los temas más revisados por los estudiantes, para que puedan ser visualizados por los profesores.
- Generar correos electrónicos para el Profesor y/o Ayudante de cátedra con los resultados obtenidos por cada alumno que ha realizado alguna práctica.

**Los objetivos específicos presentados en esta sección constituirán los requerimientos funcionales del sistema.**

### **2.3 Alcance**

El sistema será una herramienta web que generará objetos de aprendizaje que se pueden clasificar en una de las categorías siguientes:

- Teoría
- Cuestionarios
- Compilación de código

Se brindará funcionalidad a dos tipos generales de usuarios: estudiante y profesor ( o administrador ). El profesor ( o administrador ) alimentará el

sistema con datos que permiten a la aplicación, generar objetos de aprendizaje dinámicamente. A su vez estos objetos de aprendizaje serán utilizados por un usuario de tipo estudiante como material de autoestudio. El sistema permitirá al usuario profesor o administrador ingresar información para la generación de los mencionados objetos de Aprendizaje:

- Teoría, se permitirá el ingreso de nuevos temas y subtemas con sus respectivas definiciones.
- Cuestionarios (Evaluación), se permitirá el ingreso de preguntas, alternativas de respuesta y los correspondientes mensajes de retroalimentación. Estos últimos se mostrarán al estudiante una vez finalizada la resolución de un cuestionario. Además el usuario profesor podrá asignar puntaje y límites de tiempo para los diferentes cuestionarios y visualizar las estadísticas de cada evaluación.
- Los cuestionarios que son ingresados por el usuario profesor permitirán luego generar evaluaciones dinámicas. Estas evaluaciones incluirán preguntas relacionadas a un tema específico pero aparecerán en un orden aleatorio para cada evaluación que el usuario estudiante seleccione.
- Para la sección de Compilación de Código se permitirá el ingreso del enunciado de un problema y el fragmento de código propuesto como solución. Este código podrá luego ser visto por el usuario estudiante al

utilizar este objeto de aprendizaje. Sin embargo, el sistema permitirá asignar niveles de ayuda. De tal manera que el código que resuelve el enunciado propuesto no se muestra en su totalidad sino de forma progresiva dependiendo del nivel escogido por el usuario estudiante.

- Para cada una de las secciones anteriormente mencionadas se adicionará el ingreso de la información que requieren los objetos de aprendizaje de acuerdo al estándar LOM.
- El sistema permite al usuario estudiante utilizar los objetos de aprendizaje como sigue:
- Para la sección de Teoría, se permitirá la visualización de temas y subtemas con sus definiciones previamente ingresadas por el profesor o administrador. La Figura 2.1 nos muestra una vista del funcionamiento del modulo de Teoría.

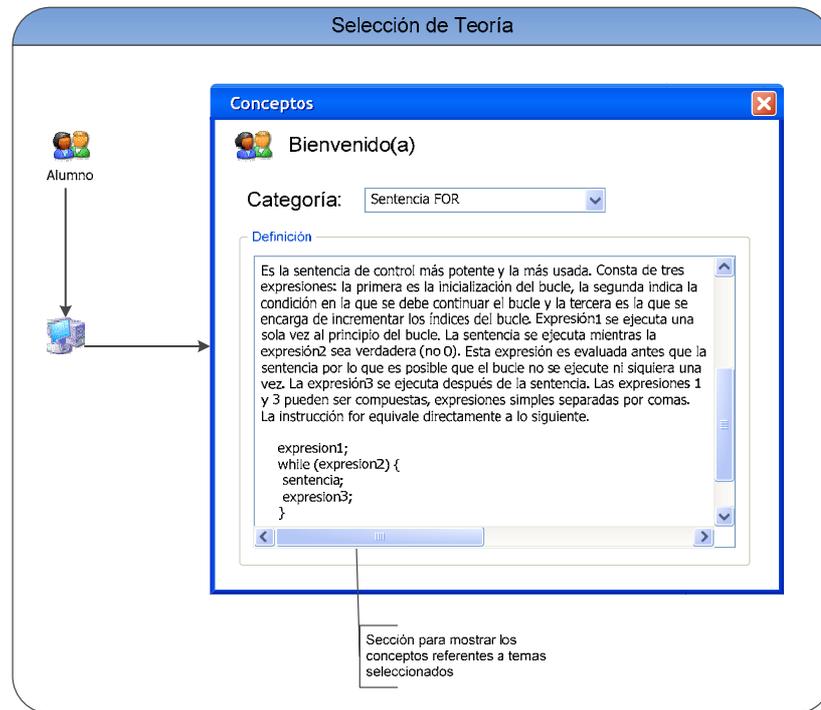


Figura 2.1.- Modelo Mental del módulo de Teoría

- Para la sección de Cuestionario, se permitirá escoger uno y seleccionar el nivel de dificultad para posteriormente resolverlo en el tiempo establecido por el profesor previamente. Una vez finalizada su resolución se podrá visualizar una retroalimentación de cada una de las preguntas, la puntuación obtenida y generar un archivo PDF con el cuestionario resuelto. La Figura 2.2 muestra el modelo que se desea implementar para el módulo de Evaluaciones.

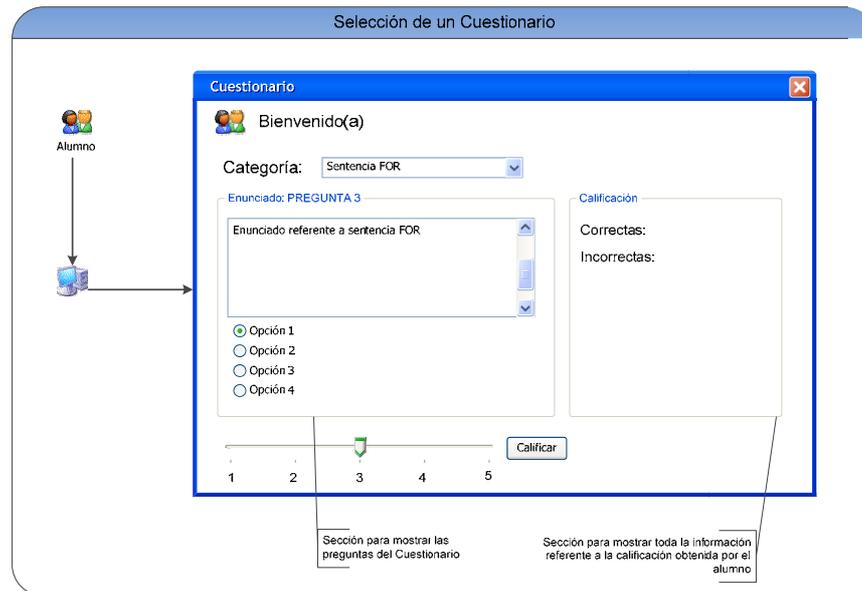


Figura 2.2.- Modelo Mental del módulo Cuestionario

- Para la sección de Compilación de Código, se podrá seleccionar un ejemplo propuesto para solucionarlo. Una vez escogido el problema el estudiante podrá ingresar su propuesta de solución y visualizar la propuesta por el profesor, para lo cual se contará con niveles de ayuda que han sido previamente configurados. Dependiendo del nivel de ayuda seleccionado se mostrará la solución del profesor de forma parcial o total. Una vez solucionado el problema planteado -no debe tener errores de compilación- se le enviará un correo al profesor indicándole que debe calificar la solución propuesta. La Figura 2.3 muestra una vista del funcionamiento del modulo de compilación.

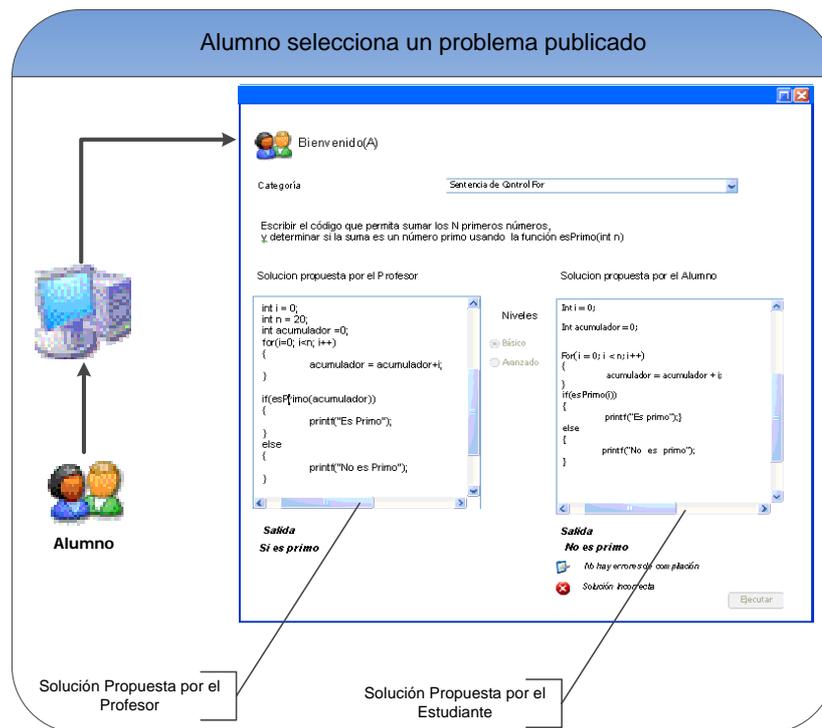


Figura 2.3.- Modelo Mental del módulo Compilación

Para la sección de Compilación de Código tanto para el estudiante como para el profesor o administrador, no se deberán ingresar soluciones y ejemplos que permitan el ingreso de variables desde teclado, es decir que no se permite utilizar las funciones tales como scanf() o getch().

Además se utilizarán librerías estándar de Lenguaje C, es decir, no se podrá utilizar la librería <conio.h> ya que no es una librería estándar por lo tanto el compilador generará un error al momento de compilar y no permitirá el ingreso de la solución o el ejemplo propuesto. Siguiendo esta

característica de incluir sólo librerías estándar, el compilador no admitirá la modularización en los ejemplos y soluciones propuestas.

Antes de implementar la aplicación que generará los OAs descritos anteriormente es necesario escoger las sentencias de control sobre las cuales se trabajará. Las sentencias de control escogidas son aquellas que se incluyen en el programa de la materia de Fundamentos de Programación de la FIEC y que además se incluyen en el documento publicado por la ACM (7). Estas sentencias son:

- Variables: Definición y tipos de datos
- Condicionales
- Sentencias de Repetición
- Arreglos Unidimensionales
- Funciones

# CAPITULO 3

## 3. Implementación de los objetos de aprendizaje

En este capítulo se presentan las herramientas a utilizar para la implementación de los objetos de aprendizaje descritos en el capítulo anterior, así como el mecanismo de comunicación entre la implementación del modelo del sistema (Java ) y la interfaz de usuario (Flex ) de la aplicación.

### 3.1 Plataforma

La tecnología que se usó para la capa del servidor es J2EE la misma que se fundamenta en Java como solución multiplataforma, además de ser código libre podemos describir otras ventajas:

- Reutilización de Código.

- Simplificación de los procesos de desarrollo.
- Un mantenimiento más rápido debido a que son pequeñas unidades de código.
- Alta escalabilidad de la aplicación.

### **3.1.1 Interfaz de Usuario**

Para la implementación del sistema de cuestionarios dinámicos se escogió una herramienta en la cual se pueda desarrollar formas iterativas y atractivas para el usuario final ya que se trata de que la herramienta sea un apoyo de estudio fácil y agradable de usar. Además, las herramientas analizadas debían permitir la implementación de una Aplicación Rica en Internet, tal como se describió en el capítulo 1. La herramienta escogida para implementar la interfaz de usuario es Adobe Flex.

Adobe Flex es un framework o conjunto de librerías open source para desarrollo de IU (Interfaz de Usuario). Este marco de trabajo es altamente productivo para la creación y el mantenimiento de aplicaciones web de alta calidad y además permite crear efectos que hacen más agradable la aplicación. Los efectos son parte de la experiencia que disfruta el usuario y pueden definir el punto de calidad que distingue una solución de las demás, dándole un uso correcto mejorará la usabilidad de las aplicaciones.

Flex no reemplaza a ninguna herramienta existente, más bien es un complemento ya que normalmente se puede conectar la interfaz del usuario, basada en Flex, a los componentes de lógica empresarial existente o a los servicios Web, sin necesidad de hacer ninguna modificación (8).

Flex pone en relieve el desarrollo de Interfaces gráficas de usuario usando un lenguaje XML llamado MXML. Así mismo, tiene varios componentes y características que aportan funcionalidades tales como Servicios Web, objetos remotos, arrastrar y soltar, columnas ordenables, gráficas, efectos de animación y otras interacciones simples. El cliente solo carga la aplicación una vez, mejorando así el flujo de datos frente a aplicaciones basadas en HTML(eg.PHP, ASP, JSP, CFMX).El lenguaje y la estructura de archivos de Flex buscan el desacoplamiento de la lógica y el diseño (9).

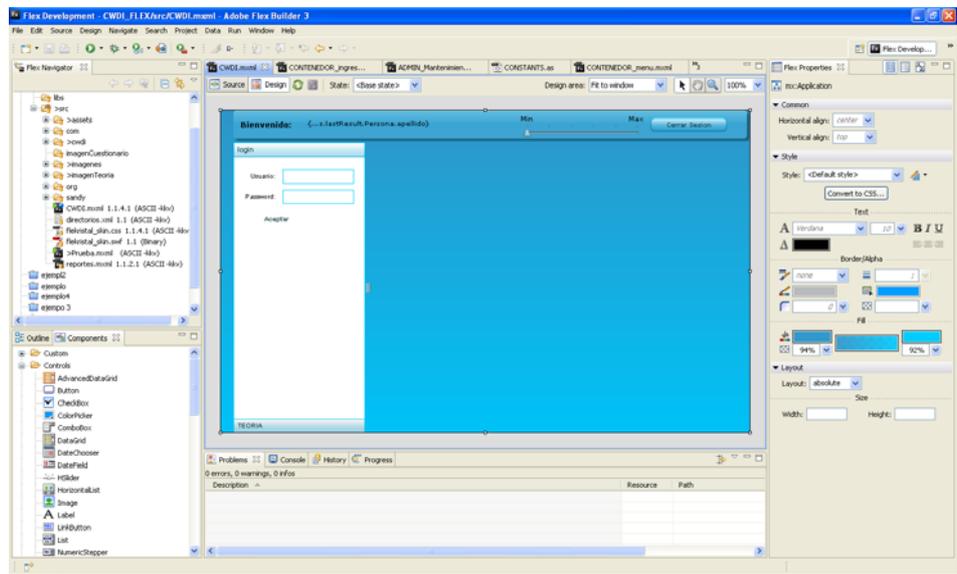


Figura 3.1.- Entorno de trabajo de Flex

Flex está contenido en un SDK y Adobe ofrece los comandos para compilar proyectos, con lo que se puede usar cualquier editor de texto para programar. Sin embargo, existe el entorno de desarrollo FlexBuilder, una herramienta comercial para crear aplicaciones basadas en Flex, con una vista de diseño que permite crear un IU en poco tiempo, optimizando el tiempo invertido por el desarrollador en la implementación de la aplicación.

### 3.1.2 Acceso a datos

Flex soporta dos protocolos de comunicación: HTTP y RTMP pero utilizando Sockets puede llegar a soportar cualquier protocolo como

por ejemplo el FTP, o se puede hacer que se conecte a servicios como POP3, IMAP y SMTP.

Para tener el acceso a componentes que procesan la salida al usuario se puede crear componentes en MXML para trabajar con tres tipos de servicios: HTTPServices, WebServices y Remote Object Services (figura 3.2.).

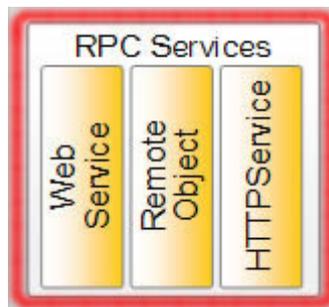


Figura 3.2.- RPC - Remote Procedure Call

Estos servicios se describen a continuación:

**WebService:**

Estos servicios Web exponen métodos específicos a los cuales se puede acceder enviando parámetros ya definidos, para esto se necesita que el Cliente y el Sevidor manejen un protocolo como el SOAP (Simple Object Access Protocol). La respuesta de un WebService es un XML formateado bajo estándares ya establecidos.

Esta etiqueta (<mx:WebService> </mx:WebService>) permite invocar desde la aplicación FLEX a servicios Web externos a la misma.

#### **Remote Object:**

Permite acceder directamente a métodos incluidos en objetos Java instanciados en el servidor mediante AMF(un protocolo binario y ligero basado en SOAP). También se puede acceder a servicios en otros lenguajes ya que actualmente existen diversas implementaciones open source.

#### **HTTPServices:**

Permite hacer llamadas a servicios desarrollados con ASP, PHP, JSP, ColdFusion, etc o simplemente hacer una llamada a archivos XML. Todo esto utilizando el protocolo HTTP con el respectivo método de envío (POST o GET), esto equivaldría al objeto XmlHttpRequest de Javascript que permite hacer llamadas asincrónicas utilizando HTTP.

En la Figura 3.3 se muestra el esquema de acceso a datos Cliente – Servidor.

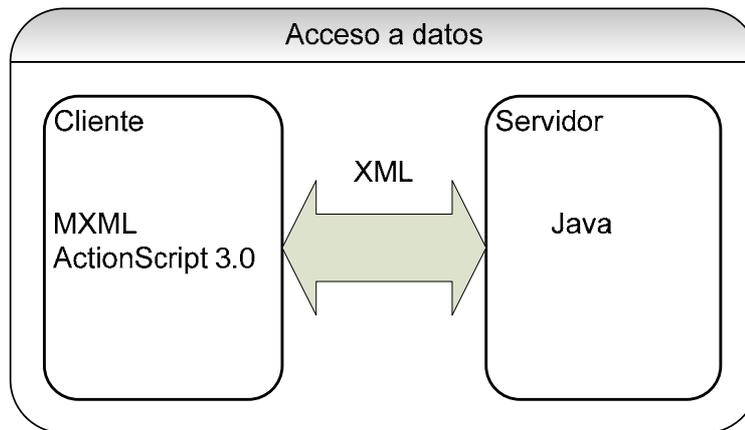


Figura 3.3.- Esquema de acceso a datos Cliente - Servidor

Dado que el protocolo utilizado por defecto en las aplicaciones web para el transporte de datos es HTTP (Hypertext Transfer Protocol), se decidió utilizar el método `HttpService`. De esta forma se puede aprovechar el hecho de que los datos transferidos utilizarán el mismo puerto que la aplicación ( puerto 80 por defecto ) y no se tendrían problemas de bloqueo debido a firewalls o filtros. Además los datos que se van a transmitir tienen formato de texto plano, ideal para ser transferido usando HTTP.

La figura 3.4 contiene el código que muestra cómo establecer una conexión vía HTTP en Flex. Para la explicación se usará como ejemplo el proceso de login del sistema, en el cual es necesario

contactar al servidor para validar el usuario y contraseña suministrados.

```
Código Flex

<mx:HTTPService uri="{CONSTANTS.SERVLET_LOGIN}" id="loginServices" resultFormat="e4x">
    <mx:request>
        <usuario>{usuario}</usuario>
        <clave>{clave}</clave>
    </mx:request>
</mx:HTTPService>

<mx:Script>
<![CDATA[

private function inicio():void
{
    this.soSesion = SharedObject.getLocal("mydata");

    if(this.soSesion.data.usuario!=null )
    {
        this.usuario=this.soSesion.data.usuario.toString();
        this.clave=this.soSesion.data.clave.toString();
    }
    this.soSesion.data.bandEstaExam="0";
    loginServices.send();
}
]]>

</mx:Script>
```

Figura 3.4.- Código que muestra detalle de la etiqueta HttpServices y una función que contacta al Servlet

Un usuario se logonea y envía la petición al servidor. En la interfaz se accede a los datos usando la etiqueta HTTPService que requiere como atributo el URL del recurso que se solicitará al servidor y los

parámetros que éste recibe, en caso de haber alguno. La petición del request es recibida y despachada por el servlet en el servidor.

El servlet, utilizando el código mostrado en la figura 3.5, genera una respuesta de texto en formato XML y la envía al cliente, el cual recibe la respuesta XML, la procesa y la muestra al usuario.

```
LoginServlet

public void doConsultarLogin(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    UsuarioDTO usuarioDTO=new UsuarioDTO();
    usuarioDTO.setUsuario(request.getParameter("usuario"));
    usuarioDTO.setClave(request.getParameter("clave"));

    LoginDAO loginDAO = new LoginDAO();
    String respuesta = loginDAO.consultarLogin(usuarioDTO);

    out.println(respuesta);
    out.flush();
    out.close();

}
```

Figura 3.5.- Código del LoginServlet

### 3.1.3 Compilador

Para implementar la herramienta de Compilación de código C vía web se utiliza un compilador en C en el servidor. En este proyecto se utilizó el GCC (GNU Compiler Collection), compilador integrado del proyecto GNU para C. El GCC recibe

como entrada un programa fuente en C y genera un programa ejecutable binario.

Para servidores Windows se tiene como opción utilizar el **MinGW** o **MinGW32** (Minimalist [GNU](#) for [Windows](#)), implementación de los compiladores GCC para la plataforma Win32, que permite migrar la capacidad de este compilador a entornos Windows.

Tomando como referencia lo indicado en la sección 2.4, para hacer uso de estos compiladores correctamente, se deben utilizar las librerías estándares de C, puesto que de no ser así se generarán errores durante la compilación del código.

### **3.2 Arquitectura de la aplicación**

En el sistema se utiliza la arquitectura cliente – servidor, si bien es cierto que esta arquitectura es una de la más antiguas, también es cierto que es una de la más utilizadas y aún en la actualidad muchas empresas de gran prestigio utilizan este esquema (10).

La figura 3.6. muestra una diagrama de la arquitectura de la aplicación donde el servidor contiene los Servlets que instanciarán a los objetos que se comunicarán con el cliente enviando texto con formato XML. El

cliente descarga la aplicación Flex por una sola vez puesto que a partir de esto sólo requerirá hacer peticiones de acceso a datos. Estos datos serán reflejados en el lado del cliente sin necesidad de refrescar la página.

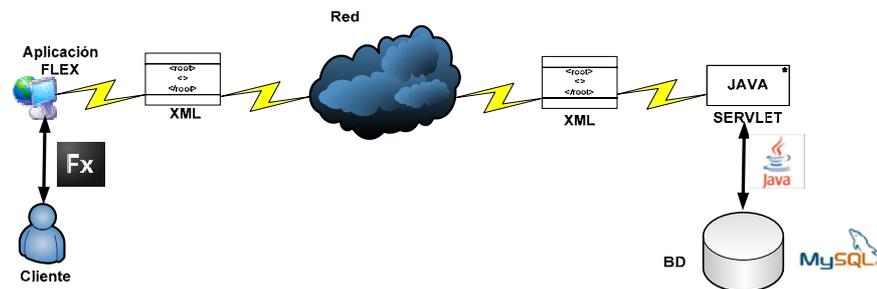


Figura 3.6.- Arquitectura de la aplicación

Para ilustrar la arquitectura de la aplicación y el intercambio de datos en la misma, en la figura 3.7a se muestra la creación de un árbol (TREE) el cual obtiene los datos dinámicamente al conectarse via HTTP a través de la etiqueta `HttpService` con un Servlet, el cual retorna el XML correspondiente al menú solicitado. Adicionalmente, dentro de la etiqueta `Tree` se encuentra embebido código `ActionScript` con el cual se manejan los eventos del componente. La Fig. 3.7b muestra un extracto de los datos XML que envía el Servlet cuando se hace la petición desde el cliente. Estos datos son tomados como fuente para llenar el componente `Tree` mencionado anteriormente.

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Tree xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="submenuServices.send();"
width="100%"
height="100%"
labelField="@label">
<mx:Script>
<![CDATA[
import cwdi.actionscript.constants.CONSTANTS;
[Bindable] public var menu_id:String;
[Bindable] public var sub_Padre:String;

public function refrescar()
{
    this.submenuServices.send();

this.dataProvider=submenuServices.lastResult.NodoSubMenu;
}

public function llenarArbol(){
    if(submenuServices.lastResult != null)
        this.dataProvider =
submenuServices.lastResult.NodoSubMenu
}

public function ejecutarConsulta():void{
    submenuServices.send();
}

]]>
</mx:Script>
<mx:HTTPService url="{CONSTANTS.SERVLET_SUBMENU}"
id="submenuServices"
resultFormat="e4x"
result="llenarArbol()">
<mx:request>
<menu_id>{menu_id}</menu_id>
<sub_Padre>{sub_Padre}</sub_Padre>
<action>{CONSTANTS.ACTION_CONSULTAR}</action>
</mx:request>
</mx:HTTPService>
</mx:Tree>

```

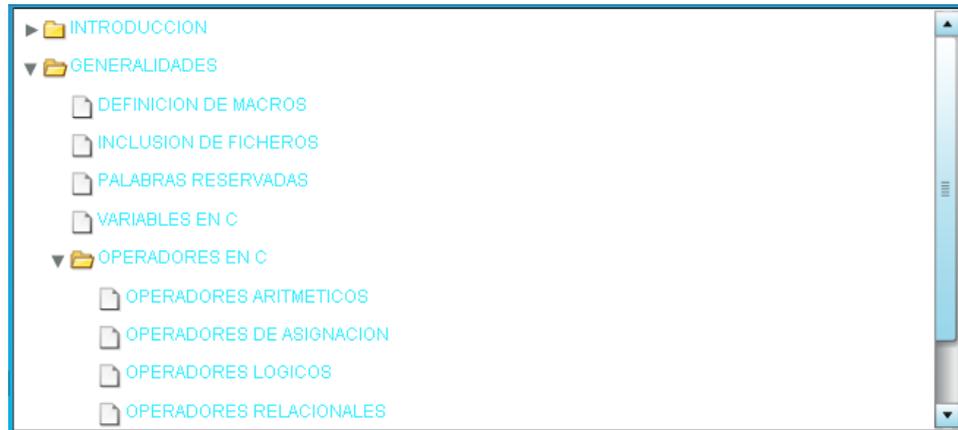


Figura 3.7a.- Componente Tree creado en Flex

```

- <root>
- <SubMenu>
  <value>LANGUAGE C</value>
  <idMenu>2</idMenu>
  <idSubMenu>16</idSubMenu>
  <ubicacion>/</ubicacion>
</SubMenu>
- <SubMenu>
  <value>CONDICIONALES</value>
  <idMenu>3</idMenu>
  <idSubMenu>88</idSubMenu>
  <ubicacion>/</ubicacion>
</SubMenu>
- <SubMenu>
  <value>LAZOS</value>
  <idMenu>3</idMenu>
  <idSubMenu>89</idSubMenu>
  <ubicacion>/</ubicacion>
</SubMenu>
- <SubMenu>
  <value>TIPOS DE DATOS</value>
  <idMenu>3</idMenu>
  <idSubMenu>90</idSubMenu>
  <ubicacion>/</ubicacion>
</SubMenu>
</root>

```

Figura 3.7b.- Fragmento XML enviado desde el servidor

Para implementar el servidor se utilizó software de distribución libre, que se detalla a continuación:

- JRE 5.0
- Contenedor Web Apache Tomcat 6.0
- Motor de base de datos MySQL Server 5.0
- MinGW32

**JRE 5.0:** El JRE 5.0 es un entorno de ejecución necesario para poder ejecutar la aplicación, además es necesario para el Tomcat 6.0 (11).

**Apache Tomcat 6.0:** Es un contenedor web con soportes de Servlet/JSP, que se escogió debido a su alto rendimiento. Además existen distribuciones para cualquier sistema operativo que disponga de la máquina virtual de Java (12).

**MySQL Server 5.0:** Es un sistema de gestión de base de datos, que se escogió porque además de ser gratuita, es confiable y de buen rendimiento. A partir de la versión 5 incluye funcionalidades como procedimientos almacenados los mismos que se utilizan para mejorar el funcionamiento del sistema.

**MinGW32:** Se utiliza para compilar las secciones de código C, que son escritas por el usuario en la sección del sistema donde le permite practicar con los diferentes ejercicios propuestos.

**Flex:** La herramienta que se escogió para implementar el RIA propuesto es Flex 3.0. Flex tiene un amplio conjunto de componentes predefinidos que permiten la implementación de una aplicación web con funcionalidad muy cercana a las aplicaciones de escritorio.

### **3.3 Objetos de aprendizaje**

El Sistema de Cuestionarios Web Dinámicos (CWDI) ha desarrollado cuatro módulos que permiten generar objetos de aprendizaje los cuales se detallarán a continuación:

#### **3.3.1 Cuestionarios Dinámicos**

Los cuestionarios dinámicos son una herramienta de apoyo con la cual el estudiante puede poner en práctica todos los conocimientos adquiridos y a su vez el catedrático podrá ver las falencias del estudiante y tomar las medidas respectivas.

El estudiante ingresa al sistema con un usuario y clave, luego del cual tiene acceso a una lista de cuestionarios para escoger, como se muestra en la figura 3.8.

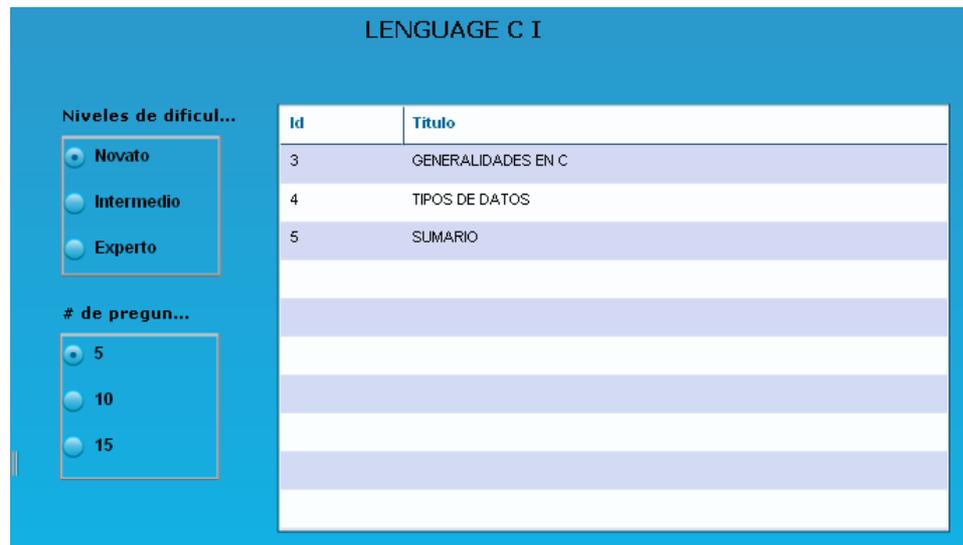


Figura 3.8.- Vista de pantalla con lista de cuestionario

En esta herramienta el usuario puede seleccionar un cuestionario para resolver de una lista de temas que previamente han sido ingresados por el profesor. Esta herramienta permite escoger los niveles de dificultad y el número de pregunta que desee resolver. Existen tres niveles de dificultad para los cuestionarios, los mismos que son asignados desde la administración del sistema en el momento de ingresar los cuestionarios.

Una vez que el estudiante ha escogido el tema, la cantidad de preguntas a resolver y la dificultad del mismo, puede comenzar la solución del cuestionario como se muestra en la figura 3.9

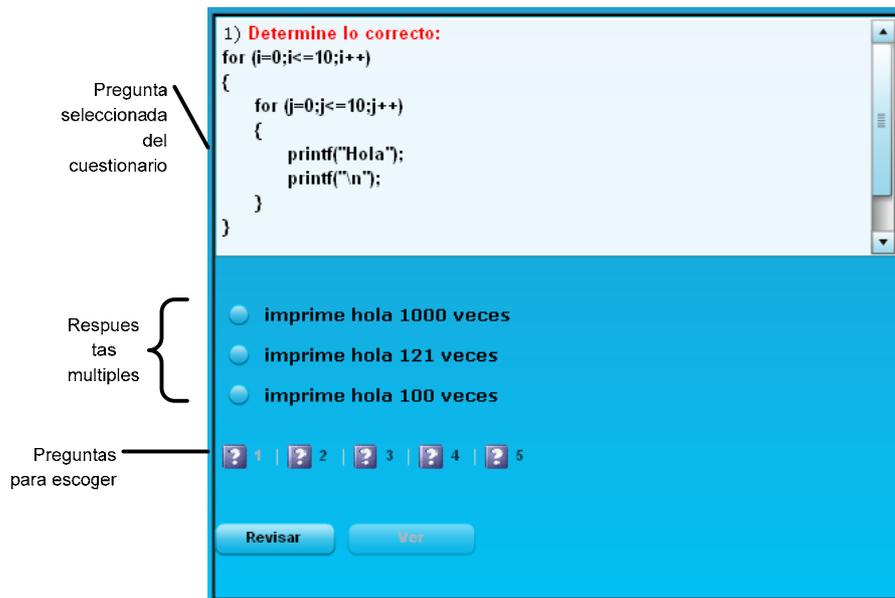


Figura 3.9.- Vista de pantalla para resolución de cuestionario

El sistema muestra preguntas del cuestionario de una manera aleatoria evitando así que se repitan. La cantidad de alternativas de respuesta para cada pregunta de las que dispone el sistema también son previamente ingresadas por el administrador. Estas respuestas se muestran en un orden aleatorio dándole así múltiples opciones de respuestas para las diferentes preguntas, el cuestionario es objetivo, ya que sólo deberá escoger la opción que crea correcta.

El cuestionario es controlado en su tiempo de respuesta utilizando el tiempo máximo definido al momento de crearlo, así que el estudiante sólo dispondrá de ese tiempo para resolverlo.

Una vez finalizada la resolución del cuestionario automáticamente se procederá a autocorregirse, quedando registrados datos como: preguntas que le fueron asignadas al estudiante, respuestas que él escogió y si fueron o no las correctas. Esta información es presentada tal como se muestra en la figura 3.10. Si el estudiante termina con la resolución del cuestionario antes de que se le termine el tiempo asignado para el respectivo cuestionario, entonces dispone de la opción para autocorrección.

El sistema dispone de un envío automático de correo a la persona que registró el cuestionario (el profesor). Éste correo contiene información especificando la puntuación obtenida por el estudiante.

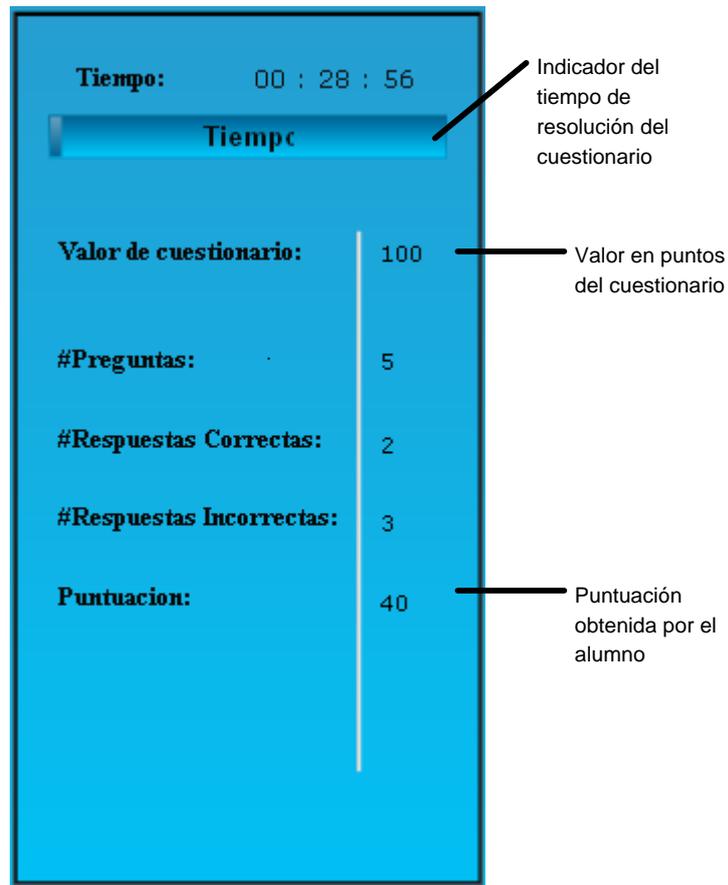


Figura 3.10.- Vista de panel con el tiempo y datos de la resolución del cuestionario

El sistema muestra al usuario retroalimentación referente a cada pregunta resuelta, donde se le indicará si la respuesta que escogió fue la correcta o no, y la debida explicación de la misma, esto se lo puede apreciar en la figura 3.11. En el desarrollo de un OA es importante incluir mecanismos de retroalimentación. De hecho en (13) se establece esta característica como un parámetro para evaluar la calidad del OA utilizado.



Figura 3.11.- Vista de panel con la retroalimentación

La aplicación le da la opción al usuario de observar en una ventana todas las preguntas que le fueron asignadas al cuestionario con sus respectivas respuestas tal como se muestra en la figura 3.12, además el contenido del cuestionario ya resuelto puede ser exportado a un archivo PDF.

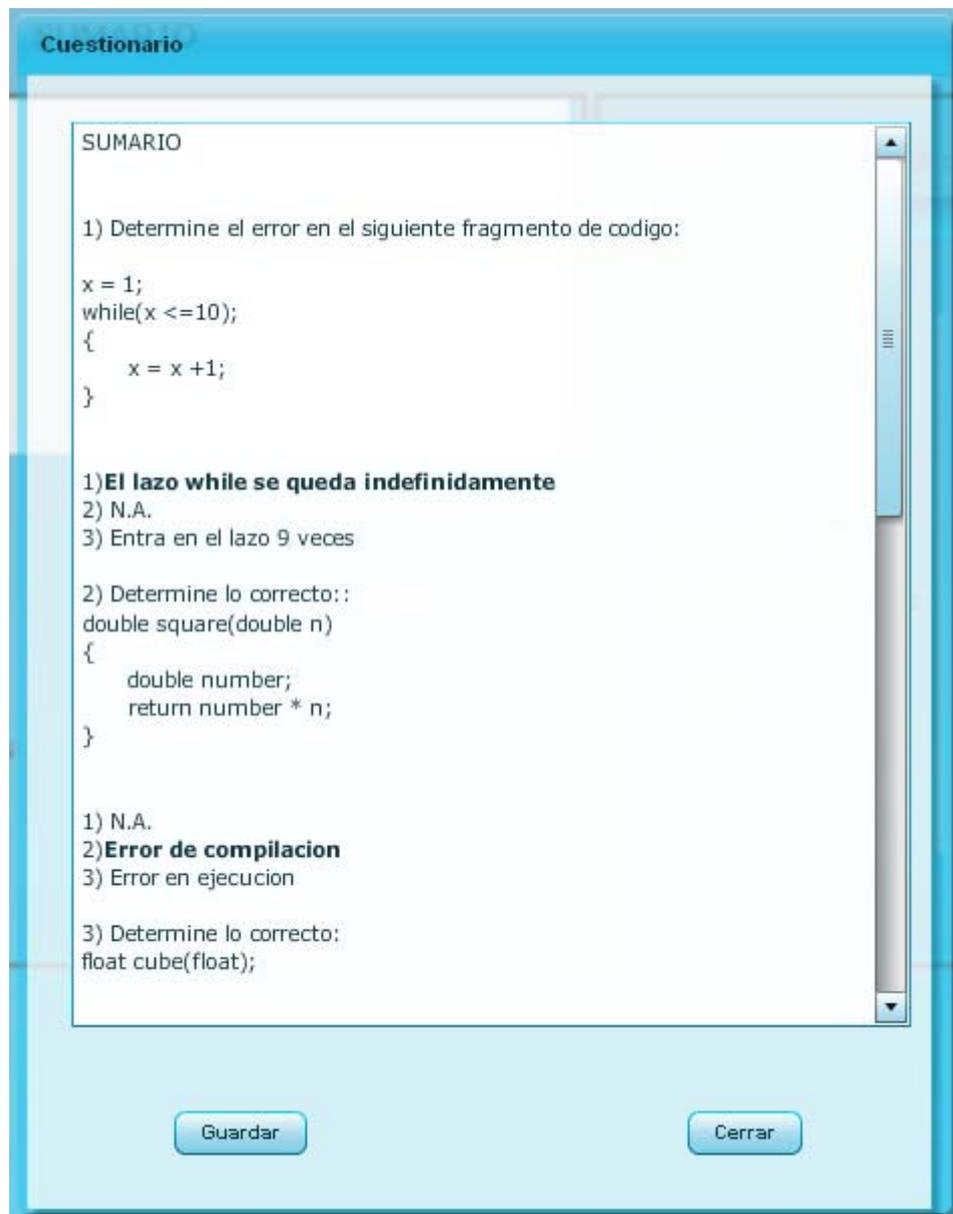


Figura 3.12.- Vista de ventana con el cuestionario resuelto

### 3.3.2 Compilación de código

El sistema le permite al usuario codificar y compilar código C de una serie de ejercicios que han sido ingresados por el administrador, al

escoger uno de los ejercicios propuestos como lo muestra la figura 3.13.

Titulo	Enunciado
Promedio de un Alumno	Programa que al recibir el promedio de un alumno en un curso... *Para los valores que se deben ingresar por teclado, se sugiere...
Incremento de Precio	El programa, al recibir como dato el precio de un producto ingresado... *Para los valores que se deben ingresar por teclado, se sugiere...
Funcion Matemática	El programa obtiene el resultado de una función matemática... El programa recibe como dato el nivel del profesor, incrementa su salario en función de la tabla mostrada a continuación.
Incremento de Salario	Nivel 1 -> %3.5 incremento Nivel 2 -> %4.1 incremento Nivel 3 -> %4.8 incremento Nivel 4 -> %5.3 incremento
Ventas Descendentes	*Para los valores que se deben ingresar por teclado, se sugiere asignarles valores por defecto para efectos de pruebas *Para los valores que se deben ingresar por teclado, se sugiere...

Figura 3.13.- Vista de ventana con lista de ejercicios

El ejercicio propuesto se lo debe resolver en lenguaje C, el sistema le permite tres niveles de ayuda (Básico, Intermedio y Avanzado), estos niveles de ayuda le ofrecen la vista de una parte del código que resuelve dicho ejercicio. El código que se muestra como ayuda ha sido previamente ingresado por el creador del OA, se lo puede apreciar en la figura 3.14.



Figura 3.14.- Vista de ventana con editor de código C.

Luego de ejecutar la respuesta ingresada en el editor, el sistema muestra si el ejercicio fue realizado exitosamente, añadiendo además la respuesta propuesta por el administrador, esto se puede observar en la figura 3.15.

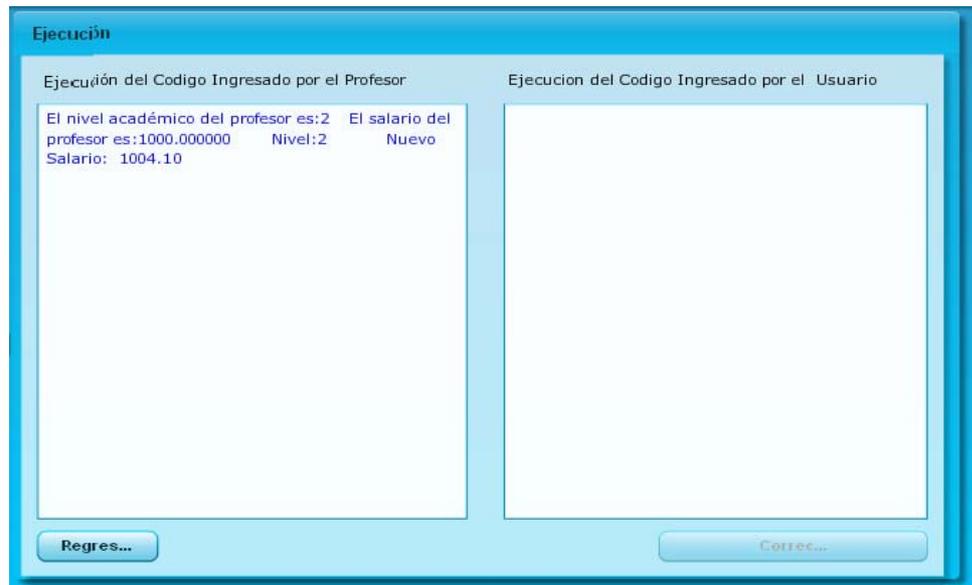


Figura 3.15.- Vista de respuesta de compilación de código.

El OA de Compilación de código (Ejecución) contiene ejemplos que fueron tomados de libros de programación básica y exámenes de cursos de universidades extranjeras dirigidos a estudiantes de cursos introductorios de programación. De acuerdo a (13) el contenido incluido en el OA no sólo debe estar relacionado al tema sino también debe de ser de calidad suficiente para lograr desarrollar las habilidades buscadas en el estudiante.

### 3.3.3 Revisión de conceptos

El sistema presenta la opción de revisar conceptos de teoría que previamente fueron registrados. En la opción de teoría se despliega un conjunto de temas tal como lo muestra la figura 3.16.

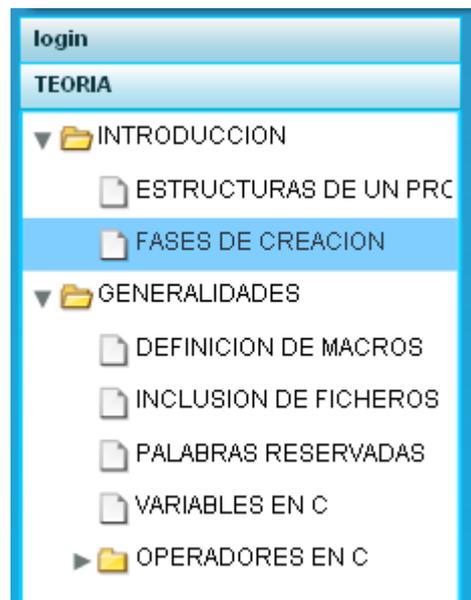


Figura 3.16.- Vista de ventana con temas previamente ingresados

Al escoger uno de los temas se observa la teoría referente al tema escogido. El sistema muestra en pantalla una especie de libro en el cual el usuario podrá desplegar las diferentes páginas y observar todo su contenido.

El estudiante tiene la opción de revisar cada tema en una vista agradable, con la que podrá navegar en las diferentes páginas de forma muy similar a la revisión del contenido de un libro tal como muestra la figura 3.17.

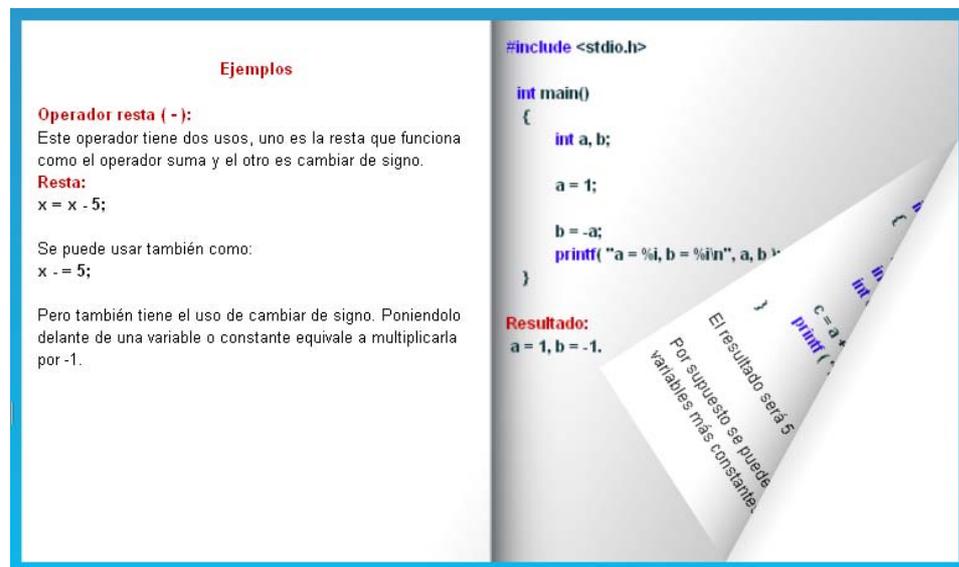


Figura 3.17.- Vista de ventana de revisión de teoría

El contenido publicado en el OA Teoría ha sido seleccionado, tomando en consideración los temas tratados en la materia Fundamentos de programación, dictados durante el primer parcial del semestre I-2008. Es decir, que está acorde con el pensum actual de la asignatura. Adicionalmente, los temas seleccionados

han sido revisados por un profesor de la materia, lo que incrementa la calidad del mismo.

El estudio mencionado anteriormente (13) incluye como otro parámetro de calidad del OA el requisito de que el material incluido en éste, esté actualizado y que haya sido revisado por un experto.

El contenido presentado en el OA teoría fue desarrollado teniendo en cuenta este criterio.

#### **3.3.4 Errores de compilación**

Los errores al programar son muy comunes, y a una persona que está aprendiendo a programar se le complica interpretar los mensajes de error que el compilador muestra. Debido a esto, la aplicación contiene un diccionario de errores comunes que se cometen a la hora de programar en código C, con esta opción el usuario puede buscar el significado del error que se ha generado.

Para la búsqueda el usuario puede ingresar un parte del mensaje de error que él observa en su entorno de desarrollo. La búsqueda se realizará a partir de las palabras claves ingresadas en la interfaz, los que comparan con el catálogo de mensajes de error del sistema que contiene información tanto en inglés como español.

En el Anexo B se incluye la lista de errores de compilación desarrollados para este objeto de aprendizaje.

### **3.4 Administración**

En el módulo de administración encontramos opciones para el mantenimiento de los objetos de aprendizaje desarrollados. Aquí el usuario profesor o administrador del sistema tiene la opción de ingresar nuevos objetos de aprendizaje o actualizar los ya existentes.

#### **Mantenimiento de teoría**

Para el ingreso de un nuevo concepto se utiliza una ventana en el que se lista el temario que se tiene configurado actualmente. Los títulos incluidos en el mismo pueden ser modificados o eliminados, y se pueden crear temas nuevos . La figura 3.18 se muestra el área de Mantenimiento de Teoría.



Figura 3.18.- Vista de ventana de temas de teoría

A los temas que se ingresan se pueden añadir subtemas y estos pueden ser modificados.

El ingreso del concepto para los temas y subtemas se los realiza en un editor de texto que permite darle formato y añadirle gráficos como complemento. Esta pantalla se puede apreciar en la figura 3.19.

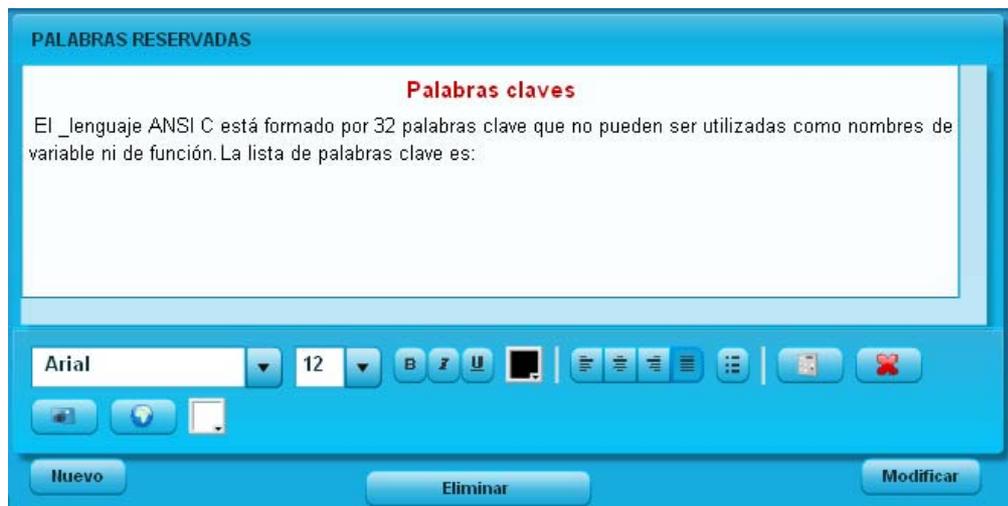


Figura 3.19.- Vista de editor de texto para ingreso de Teoría

De esta forma el sistema permite redactar conceptos de una manera amigable y clara para que pueda ser apreciada por los estudiantes.

### **Mantenimiento de cuestionario**

El sistema permite al administrador el ingreso y modificación de nuevos cuestionarios, en la figura 3.20 muestra la ventana de mantenimiento de los cuestionarios.

Tema: LEIQUAGE C I				
Titulo:		Buscar		
Título	Tiempo	Autor	Puntaje	# Alt. Respu
GENERALIDADES EN C	00:01:00	CARMEN VAC	100	3
TIPOS DE DATOS	00:01:00	CARMEN VAC	100	3
SUMARIO	00:30:00	CARMEN VAC	100	3

Figura 3.20.- Vista de mantenimiento de cuestionario

Para el ingreso de nuevos cuestionarios el administrador tiene que ingresar parámetros como tiempo, puntaje y número de alternativas que el estudiante visualizará para responder como muestra la figura. Se pueden añadir nuevos temas para el sistema o escoger un tema ya definido e ingresar un nuevo cuestionario.

The image shows a configuration window for a questionnaire. It has a blue background and contains the following elements:

- Tema:** A dropdown menu with the text "LENGUAGE C I" and a small downward arrow on the right.
- Título:** A large, empty white text input field.
- Tiempo:** Three spinners for time units: "h..", "m...", and "s..", each with a value of "0".
- Puntaje:** A spinner with a value of "0".
- # Alternativ...:** A spinner with a value of "0".
- grabar:** A blue button with the text "grabar" in white.

Figura 3.21.- Ingreso de parámetros para el cuestionario

### **Mantenimiento de código**

El mantenimiento de código se realiza de manera similar a los otros OAs mencionados, la figura 3.22 muestra el menú de las sentencias de control que han sido establecidas en el sistema.



Figura 3.22.- Menú de Sentencias de Control Establecidas para el módulo Código

Para ingresar un nuevo código en una de las sentencias de control se utiliza la ventana inicial de mantenimiento como lo muestra la figura 3.23.

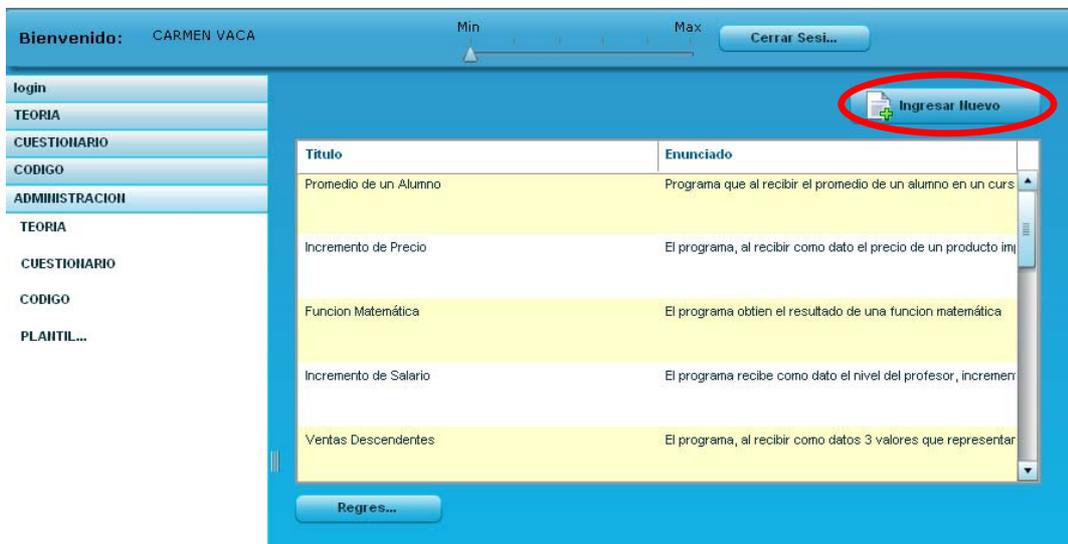


Figura 3.23.- Pantalla inicial de mantenimiento de ejercicios correspondiente a la opción Condicionales

La ventana de mantenimiento permite modificar ejercicios que ya estén en el sistema, así como también permite ingresar nuevos ejercicios.

Para el ingreso de un nuevo ejercicio es necesario establecer las librerías que se van a utilizar, así como el enunciado del ejercicio que se va a plantear. Además, de ingresar la resolución del ejercicio propuesto, el sistema también permite ingresar niveles de ayuda para que el estudiante pueda resolver el ejercicio, estos niveles de ayuda se los ingresa de acuerdo al criterio del administrador o profesor que ingresa el código.

### **Datos LOM**

Un nuevo cuestionario, ejercicio de código o concepto de teoría, representa un nuevo objeto de aprendizaje para lo cual se debe añadir información adicional con el objetivo de seguir un estándar como se mencionaba en la sección 2.1.3. El sistema tiene un panel que permite el ingreso de los datos requeridos para el estándar LOM como se muestra en la figura 3.24.

**LOM v1.0**

General

Ciclo de vida

Meta-Metadatos

técnica

Formato:

Tamaño:

Localización:

Tipo de requerimiento:

Nombre de requerimiento:

Version Mínima:

Version Máxima:

Pautas de instalación:

Otros requisitos de plataforma:

Duración:  hh  mm  ss

Uso Educativo

Derechos

Figura 3.24.- Vista del panel con datos del LOM

Para cada uno de los objetos de aprendizaje que se ingrese, se tiene que añadir datos requeridos por LOM, este estándar tiene como objetivo describir las características relevantes de los objetos de aprendizaje.

Estas características se muestran agrupadas en 6 categorías que son:

- General
- Ciclo de vida
- Meta-Metadatos,
- Técnica,
- Uso Educativo
- Derechos.

# CAPITULO 4

## 4. Pruebas, mediciones y resultados

### 4.1 Descripción de plan de pruebas

Dentro de este capítulo se evaluarán las siguientes características de la herramienta:

- Tiempo de descarga y optimización, de la aplicación, pues se busca ofrecer al estudiante una herramienta que sea liviana.
- Evaluar los Objetos de aprendizaje generados por la herramienta, con el fin de conocer la opinión de los estudiantes con respecto a todo cuanto le ofrece la herramienta, utilizando como puntos de evaluación ciertos criterios determinados por LORI, cuyo rango de calificación va de 1 a 5 siendo 1 la calificación más baja y 5 la más alta (14).

- Analizar las estadísticas producidas por el sistema, las cuales son evaluadas en base a la cantidad de preguntas correctas e incorrectas que ha contestado el estudiante durante el uso del módulo de Cuestionarios (Evaluación), en el tiempo establecido para la resolución de la evaluación.
- Los dos últimos puntos de evaluación se aplicarán a 12 alumnos de la materia de Fundamentos de Programación correspondientes al semestre II-2008.

#### **4.2 Tiempo de descarga y optimización**

En las aplicaciones Web es importante que el tiempo de descarga no sea mayor que el tiempo promedio de espera al cual los usuarios están habituados. Si fuera así, el usuario pronto abandonaría la aplicación. En el caso de la herramienta desarrollada esto es aún más importante porque es crucial, mantener la atención del estudiante.

Dado que las aplicaciones de Flex se compilan a un archivo .swf que suele tener un tamaño mucho mayor que una aplicación Web tradicional, se analizaron opciones para reducir dicho tamaño y por lo tanto reducir el tiempo de descarga

Una de las estrategias que se aplicó fue el desarrollar el aplicativo a través de módulos que permiten disminuir considerablemente el tamaño inicial de la aplicación hasta un 50%. De hecho se observó que la aplicación tenía inicialmente un tamaño cercano a 1 MB(933Kb), pero aplicando la modularización se redujo hasta un tamaño de 435KB, como consecuencia de esto el tiempo de descarga también disminuyó de 3 minutos a 1:30 minutos. Este tiempo se ha medido utilizando un entorno real en el cual los requerimientos se enviaron desde una máquina conectada al Internet hacia un servidor con dirección IP pública en el que se desplegó la aplicación. Sin embargo, en un entorno en el cual el Cliente y el Servidor están en una misma red LAN el tiempo de descarga es de unos cuantos segundos. La Figura 4.1 muestra los módulos en los que se dividió al proyecto.

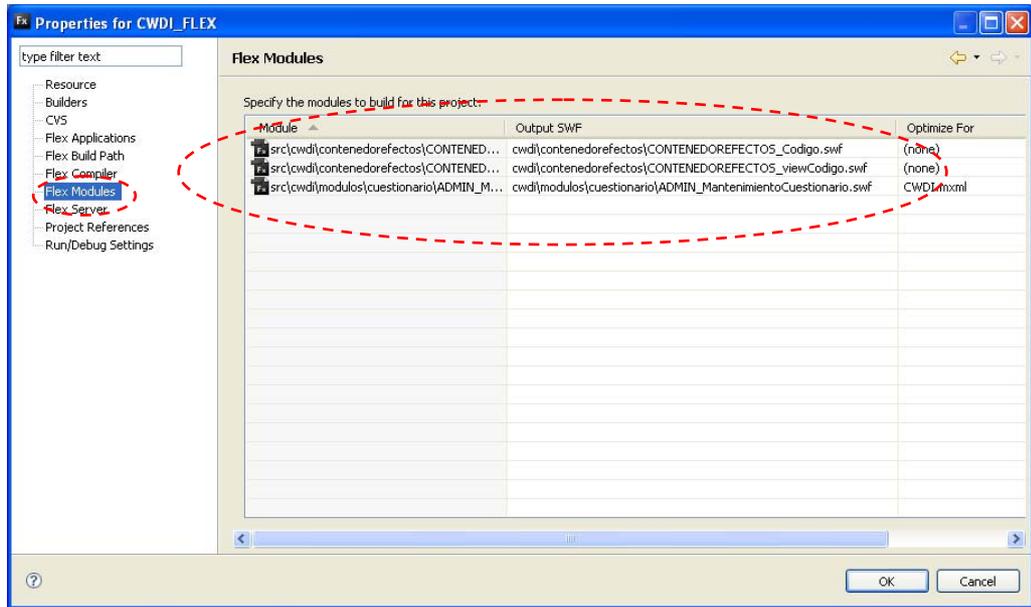


Figura 4.1.- Módulos de la aplicación CWDI

Adicionalmente, para optimizar el tamaño también se procedió a configurar el compilador de FLEX para que redujera el tamaño de la aplicación al mínimo posible. Esto se configuró en el entorno gráfico proporcionado por el IDE de FLEX (Figura 4.2). Esta configuración permitió reducir aún más los tamaños y tiempos anteriormente mencionados.

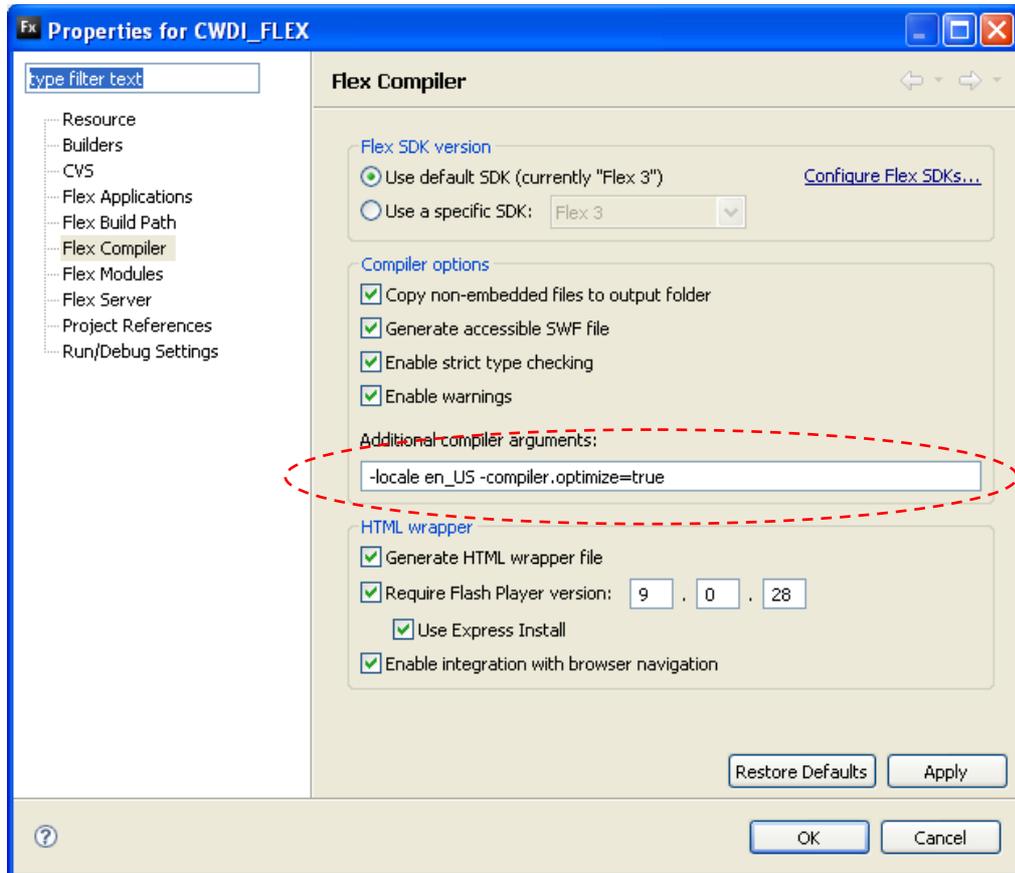


Figura 4.2.- Configuración del Compilador de FLEX

### 4.3 Evaluación de los objetos de aprendizaje

Se evaluó el modulo de Cuestionarios, la encuesta para medir la utilidad de la herramienta se la realizó a 12 alumnos de un paralelo de Fundamentos de Programación dictado en el término II – 2008 durante días previos a rendir el examen correspondiente al primer parcial.

A continuación se detallarán los resultados obtenidos de la encuesta realizada (El contenido de la encuesta se incluye en el Anexo C). Los criterios que se evaluaron fueron:

- Calidad de Contenidos.
- Retroalimentación.
- Motivación.
- Diseño y Presentación.
- Usabilidad.

La Figura 4.3 muestra los resultados, de forma general, obtenidos de la encuesta realizada a los alumnos con respecto al módulo de Cuestionarios.

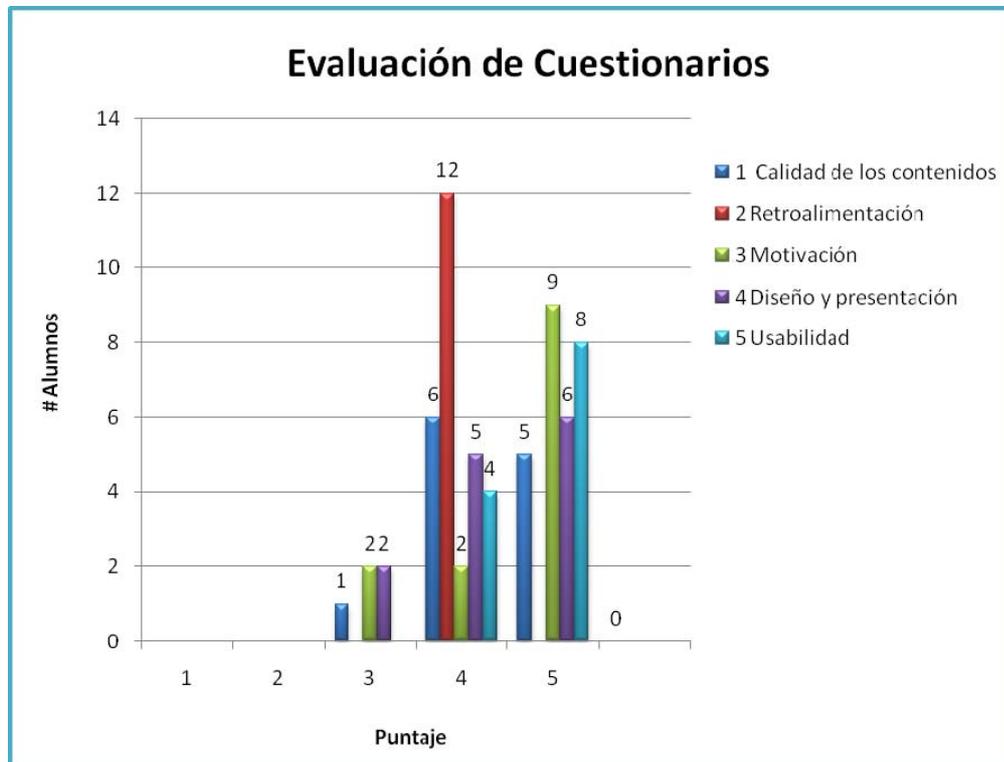


Figura 4.3.- Evaluación del Módulo de Cuestionarios por parte de Alumnos de Fundamentos de Programación

- Calidad de Contenidos:** el 50% de los alumnos concordaron en que la calidad de las preguntas presentadas son buenas y el 42% que era excelente (Figura 4.4), en general se puede decir que los ejemplos que se mostraron a los alumnos fueron buenos, pero que pueden mejorar.



Figura 4.4.- Estadísticas correspondientes a la evaluación de las Calidad de los Contenidos del Cuestionario por parte de los estudiantes.

- **Retroalimentación:** el 100% de los alumnos (Figura 4.5) estuvieron de acuerdo en que el módulo de cuestionarios ofrece una excelente retroalimentación de cada uno de los contenidos. Pero así mismo esta calificación nos indicó que se deben realizar ajustes para que se pueda cumplir su cometido: reforzar los conocimientos del estudiante.



Figura 4.5.- Estadísticas correspondientes a la evaluación de de la Retroalimentación por parte de los estudiantes.

- **Motivación:** el 69% de alumnos (Figura 4.6) concluyeron que los cuestionarios los motivaban a estudiar más los puntos en los que consideraban debían ser reforzados.



Figura 4.6.- Estadísticas correspondientes a la evaluación de Motivación que ofrece el Cuestionario, por parte de los estudiantes.

- Diseño y Presentación:** 39% de los estudiantes concordaron en que el diseño de la aplicación es excelente, mientras que el 39% la consideraron buena (Figura 4.7), en general el diseño es bueno pero puede ser pulido un poco más para que sea más atractivo para el estudiante, lo que conlleva a que los alumnos se vean más motivados a usar la aplicación.



Figura 4.7.- Estadísticas correspondientes a la evaluación del Diseño y presentación los Contenidos del Cuestionario por parte de los estudiantes

- **Usabilidad:** 67% de los estudiantes (Figura 4.8) consideraron que la navegación dentro de la aplicación es excelente. Eso nos indica que la manera para seleccionar un cuestionario y responderlo no es compleja y que una persona se puede guiar bien en la aplicación.

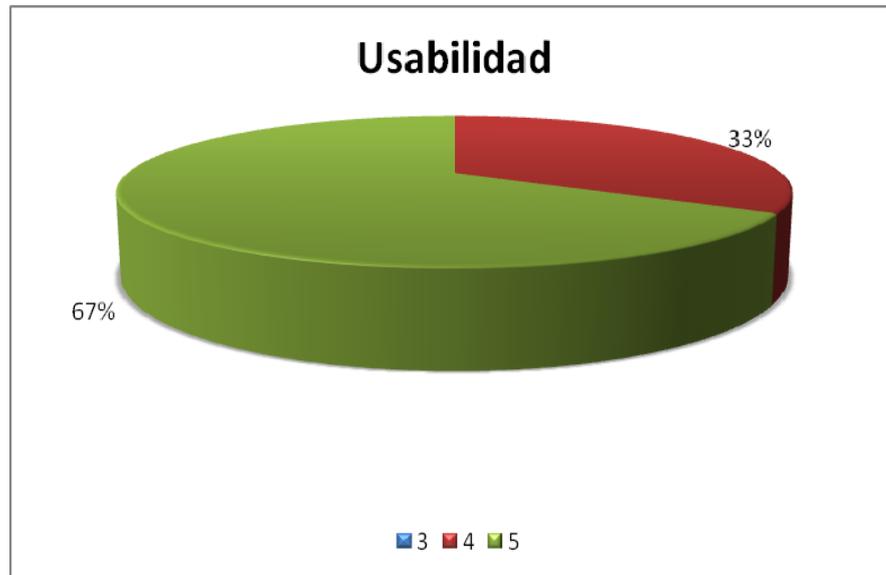


Figura 4.8.- Estadísticas correspondientes a la evaluación de la Usabilidad del Cuestionario por parte de los estudiantes.

La Figura 4.9 muestra los resultados obtenidos de la encuesta realizada a los alumnos con respecto a la aplicación en general:

- Los 12 alumnos (100%) estuvieron de acuerdo en que la aplicación en general, permitirá mejorar sus conocimientos de la materia como una herramienta de autoestudio. Lo que indica que ellos ven la necesidad de tener una aplicación que les permita aumentar sus habilidades respecto al contenido dictado en la materia.

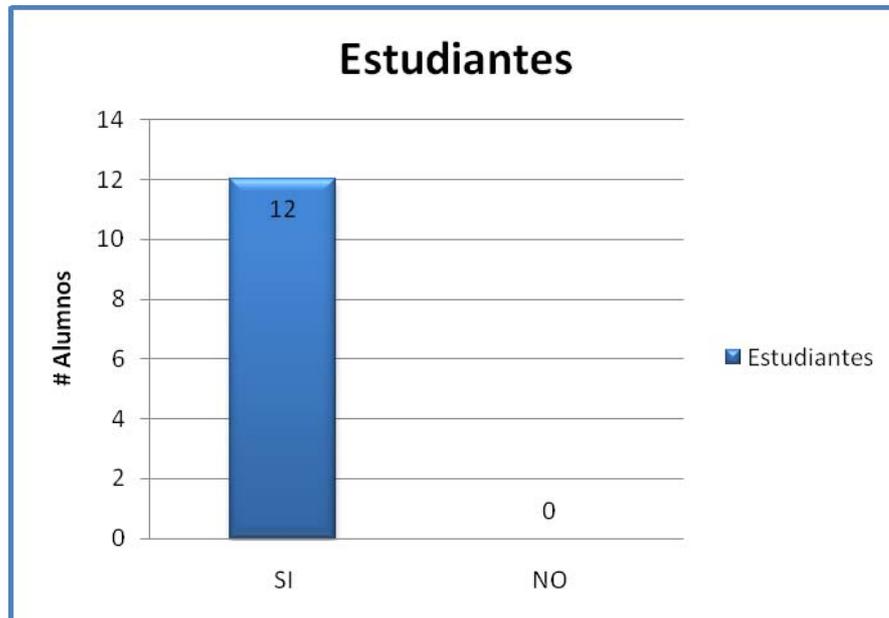


Figura 4.9.- Evaluación General de la herramienta por parte de Alumnos de Fundamentos de Programación

#### 4.4 Estadísticas producidas por el sistema

En las pruebas mencionadas en la sección 4.3, los alumnos resolvieron cuestionarios generados dinámicamente. Una vez que se concluyó la prueba se recogieron los datos proporcionados por el sistema.

La aplicación permite generar estadísticas valiosas para el instructor como por ejemplo, le permite observar número de respuestas correctas e incorrectas de cada estudiante con sus respectivos porcentajes.

La Figura 4.10 nos muestra el detalle de cuántas preguntas fueron contestadas correctamente para el cuestionario llamado SUMARIO, la

calificación que obtuvieron(PC) calificado sobre 100, la cantidad de respuestas incorrectas(PI) y su correspondiente porcentaje(PI%).

## Reporte de Calificaciones

Apellido	Nombre	Título	PC	Calificación	PI	PI(%)
2	Estudiante	SUMARIO	10	55.56	8	44.44
3	Estudiante	SUMARIO	2	22.22	7	77.78
CARRASCO	VERONICA	SUMARIO	6	33.33	3	16.67
DELGADO	MARIA BELEN	SUMARIO	6	66.67	3	33.33
DOYLET PARRA	SOLANGE MELISSA	SUMARIO	6	66.67	3	33.33
GUEVARA	HENRY	SUMARIO	5	55.56	4	44.44
LOMA MONTOYA	ANDRES	SUMARIO	4	44.44	5	55.56
LUNA CASTRO	JOSE GABRIEL	SUMARIO	8	88.89	1	11.11
MUÑOZJADAN	ALEXIS YANIRA	SUMARIO	4	44.44	5	55.56
RODRIGUEZ	JONNATHAN	SUMARIO	5	55.56	4	44.44
VALVERDE	CARLOS ANDRES	SUMARIO	10	43.48	8	34.78
VELEZ	CARLOS ANDRES	SUMARIO	7	77.78	2	22.22

Figura 4.10.- Reporte de la evaluación realizada a los alumnos de Fundamentos de Programación Semestre II-2008

De acuerdo a la Figura 4.11, la cual está basada en los datos del reporte mostrado en la Figura 4.10, podemos concluir que alrededor de un 67% de estudiantes no obtuvieron una calificación mayor a 60, lo que indica que la mayoría de alumnos tiene todavía puntos débiles con respecto al

contenido de la materia, previo al examen del primer parcial. Este tipo de reportes y estadísticas le permitirán al profesor hacer un seguimiento de los posibles puntos en los que tienen problemas los alumnos pues como el profesor es el encargado de crear los cuestionarios, le es más fácil determinar en qué temas tienen problemas los estudiantes.

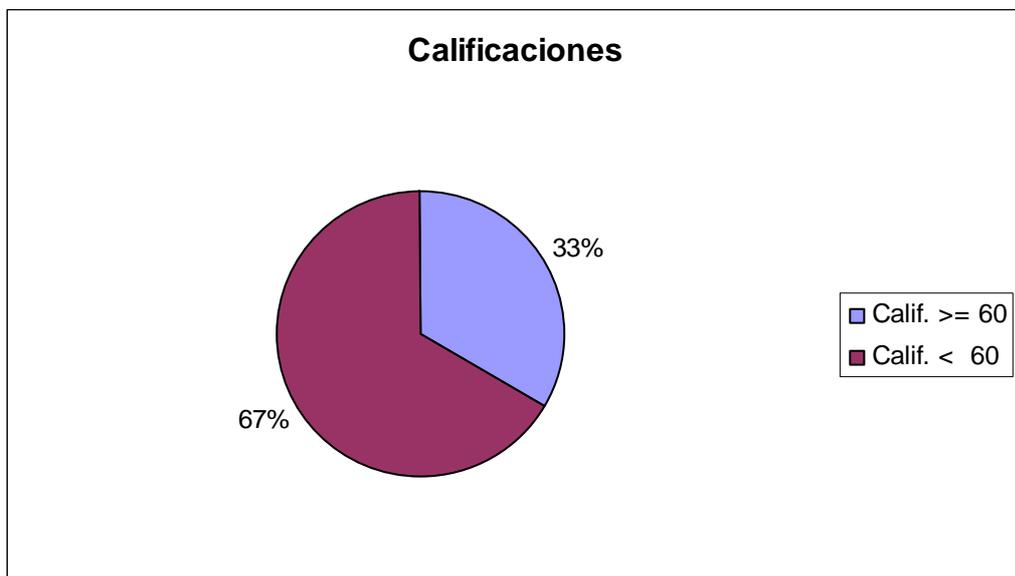


Figura 4.11.- Estadística de la evaluación realizada a los alumnos de Fundamentos de Programación Semestre II-2008

## **CONCLUSIONES Y RECOMENDACIONES**

1. Una vez terminada la implementación de la herramienta, y realizadas las evaluaciones a estudiantes de la materia se puede concluir que se ha desarrollado una herramienta que permite generar OAs de calidad para ser utilizado por estudiantes de cursos introductorios de programación con el fin de reforzar conceptos aprendidos durante la clase.
2. La elección de los tipos de objetos de aprendizaje que se desarrollaron: Teoría, resolución Cuestionarios, y Código, se basó en las principales actividades que se desarrollan durante una curso regular de programación, lo que permite tener una aplicación sencilla que

servirá como herramienta de soporte en el proceso de aprender a programar.

3. Durante la evaluación realizada a los estudiantes se pudo observar la motivación de cada uno al contestar las preguntas presentadas por la aplicación y su interés al revisar la retroalimentación brindada una vez terminada la prueba. Esto nos permite concluir que herramientas como la desarrollada ayudan a captar la atención del estudiante sin la presión de una evaluación escrita u oral.
4. Aplicar la estructura de OAs representa, por un lado, un ahorro de tiempo para el profesor en la preparación de recursos de aprendizaje de calidad, y por otro lado, una disponibilidad constante de dichos recursos para el alumno.
5. De acuerdo con las pruebas que realizamos con los alumnos de Fundamentos de Computación sobre el OA Cuestionario, podemos concluir que el 100% de los estudiantes, que formaron parte de la evaluación, concordaron en que esta herramienta les permitía poner a prueba sus conocimientos y que además les facilitaría comprender, de mejor manera, el contenido dictado en la materia.

6. Utilizar el entorno de desarrollo(IDE) que ofrece Flex nos permitió disminuir el tiempo de implementación de la herramienta, pues el IDE ofrece un panel con componentes predefinidos que se utilizan generalmente para crear formularios Web tales como botones, campos de texto, calendarios,etc. Adicionalmente, nos facilitó el manejo de efectos y de modificación de las propiedades de cada componente de la manera más sencilla posible debido al entorno grafico que ofrece.
  
7. Aplicar el esquema de modularización ofrecido por el Framework de Flex vs. el esquema de Aplicación que el marco de trabajo ofrece por defecto, permitió crear un RIA de menor tamaño. Lo que permitió mejorar significativamente el tiempo de espera para la descarga de la aplicación en alrededor de un 50%.

Las recomendaciones que se pueden presentar con la elaboración de la presente tesis son:

1. Se recomienda para futuras adecuaciones, permitir el ingreso de OAs con contenido de tipo multimedia, es decir videos, por ejemplo.
  
2. Implementar funcionalidad que permita el ingreso de datos desde el teclado, pues actualmente sólo se puede hacer a través de la definición de una variable con un valor estático.

3. Un valor agregado que puede tener la aplicación en un futuro sería el de permitir que el compilador no solamente funcione con las librerías estándar de C, es decir que se puede utilizar la librería <conio.h> por ejemplo, ya que es utilizada con cierta frecuencia en los ejemplos dados en clases.

# **ANEXOS**

# ANEXO A

## Diagrama de modelo estándar LOM

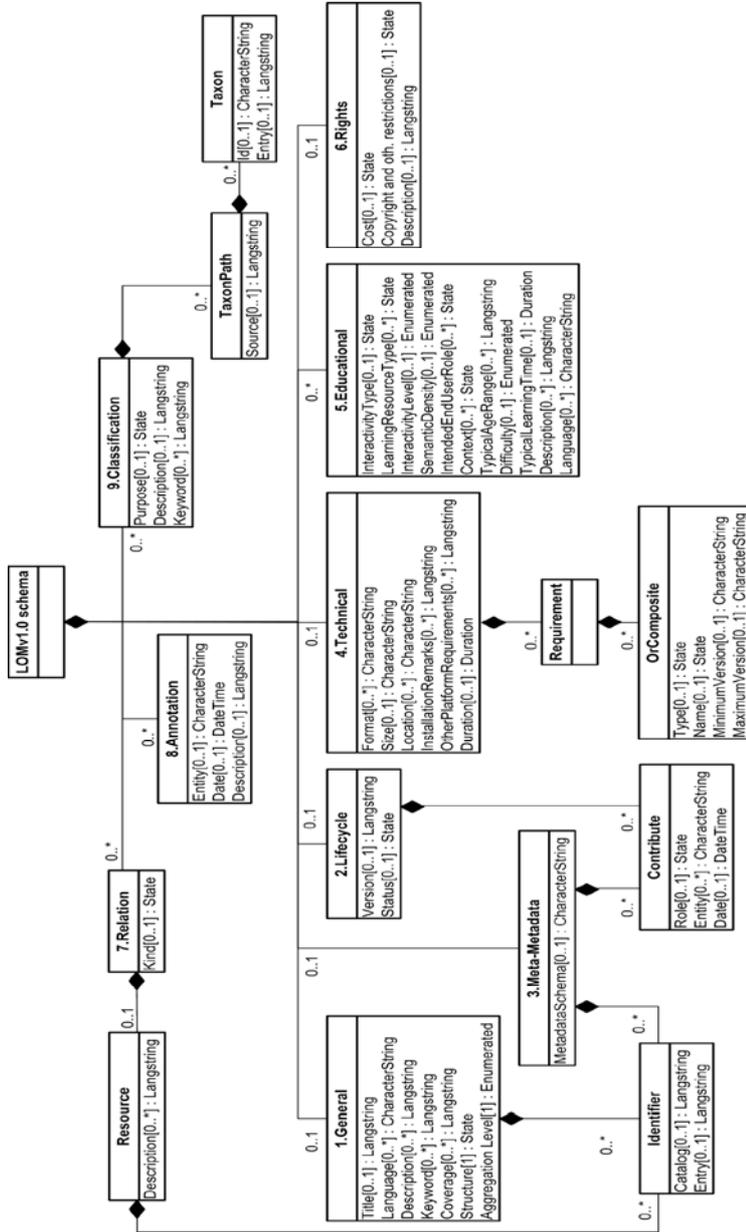


Figura A.1.- Esquema del estándar LOM

## ANEXO B

### Errores Básicos de Compilación

Error	Descripción
'variable' undeclared	<p>C es un lenguaje tipado. Esto significa que hay que declarar las variables antes de usarlas. C es sensible a mayúsculas, por lo que las variables pepe y Pepe son distintas.</p> <p>Las causas del error pueden ser:</p> <ul style="list-style-type: none"><li>• No se ha declarado la variable.</li><li>• Se ha escrito mal (cuidado con las mayúsculas).</li></ul> <p>Este error normalmente se acompaña con el mensaje: prog.c:7: (Each undeclared identifier is reported only once prog.c:7: for each function it appears in.) que indica que, aunque la variable que no se ha declarado se utilice varias veces en la función, sólo se informará de la primera aparición.</p>
Parse error before 'string'	<p>El compilador ha encontrado algo que no entiende. Las sentencias de un programa C empiezan normalmente con una palabra reservada, un nombre de variable o un nombre de función.</p> <pre>main() { into printf("Hola, mundo\n"); }</pre> <ul style="list-style-type: none"><li>• Falta un paréntesis ')' o una llave de cierre '}' en los alrededores de la línea indicada.</li><li>• Hay más llaves de la cuenta.</li><li>• Falta un punto y coma (;) en la línea anterior.</li><li>• Hay una expresión mal construida.</li><li>• La condición de un if o de un while no va entre paréntesis.</li></ul>
Missing terminating " character	<p>No se encuentra el carácter de terminación de la definición de una cadena. Este mensaje suele venir acompañado por un mensaje que indica la posición donde el compilador cree que puede haberse producido el fallo.</p>

	<code>printf("Coord. vertices (x1&lt;=x2, y1&lt;=y2)\n");</code>
undefined reference to 'funcion'	<p>Se ha encontrado lo que parece una llamada a una función, pero no encuentra el código de dicha función. El nombre de la función desaparecida es la última palabra de la segunda línea (en el ejemplo anterior, <code>juan</code>).</p> <pre>/tmp/ccabbz60.o: In function 'pepe': /tmp/ccabbz60.o(.text+0x10): undefined reference to 'juan'</pre>
Incompatible type for argument 1 of 'funcion'	<p>Este error aparece cuando se le pasa a una función un argumento con un tipo que no concuerda con el esperado. La principal causa de este error es que no se ha respetado el orden de los argumentos definido por la cabecera de la función.</p>

Tabla C.1.- Tabla de Errores Generales de Compilación con GCC.

## ANEXO C

### Formato Evaluación del Módulo Cuestionario

**EVALUACION: CUESTIONARIO**

	Bajo				Alto	
	1	2	3	4	5	NA
<b>1. Calidad de los contenidos:</b> crees que el contenido mostrado en el cuestionario te ayudará a mejorar tus habilidades en la materia de Fundamentos de Programación?.						
<b>2. Feedback (retroalimentación):</b> las preguntas tienen una retroalimentación adecuada a tus necesidades?						
<b>3. Motivación:</b> Los cuestionarios te motivaron más para estudiar determinados puntos de la materia con el fin de reforzar tus conocimientos?						
<b>4. Diseño y presentación:</b> el diseño visual favorece para fortalecer tus conocimientos de la materia?.						
<b>5. Usabilidad:</b> Hay una facilidad de navegación?, es una interfaz predictiva ?						

6. **En general, crees que esta herramienta te permitirá mejorar tus conocimientos de la materia. Es decir te serviría como herramienta de autoestudio?**

SI

NO

7. **Que le añadirías a la aplicación?**

# ANEXO D

## Casos de Uso

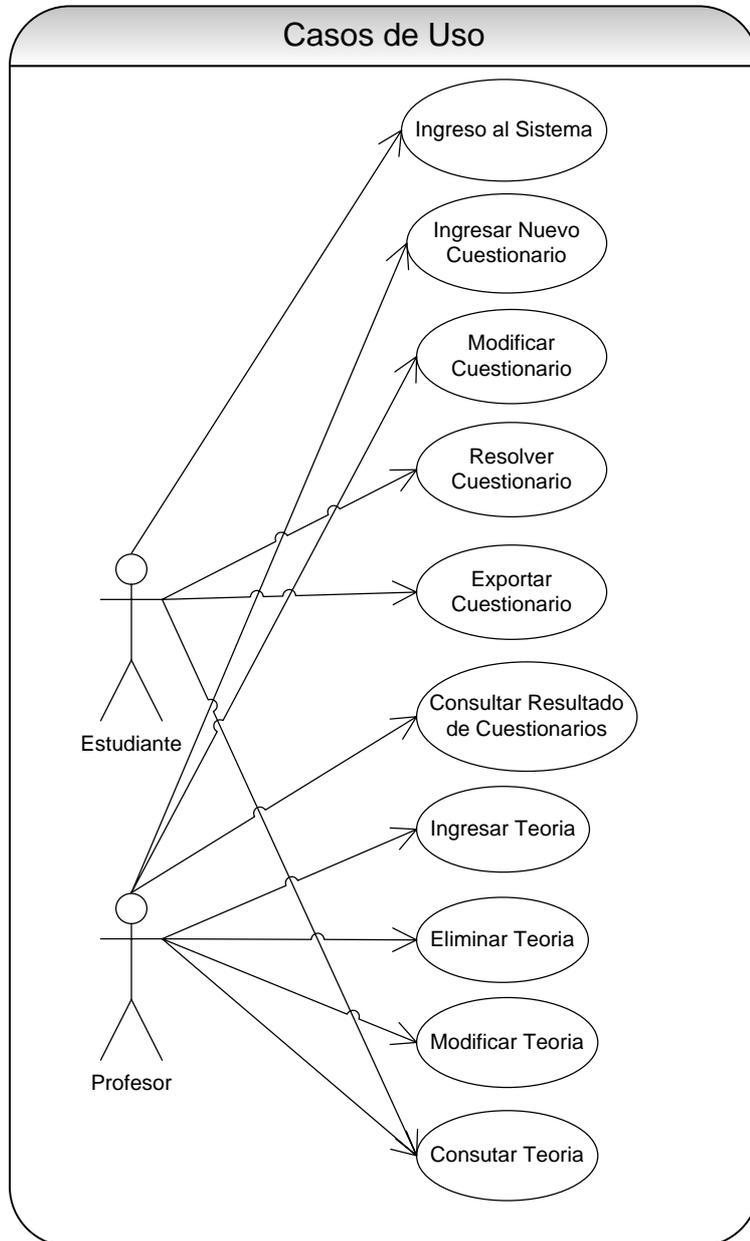


Figura D.1.a.- Diagrama de Casos de Uso

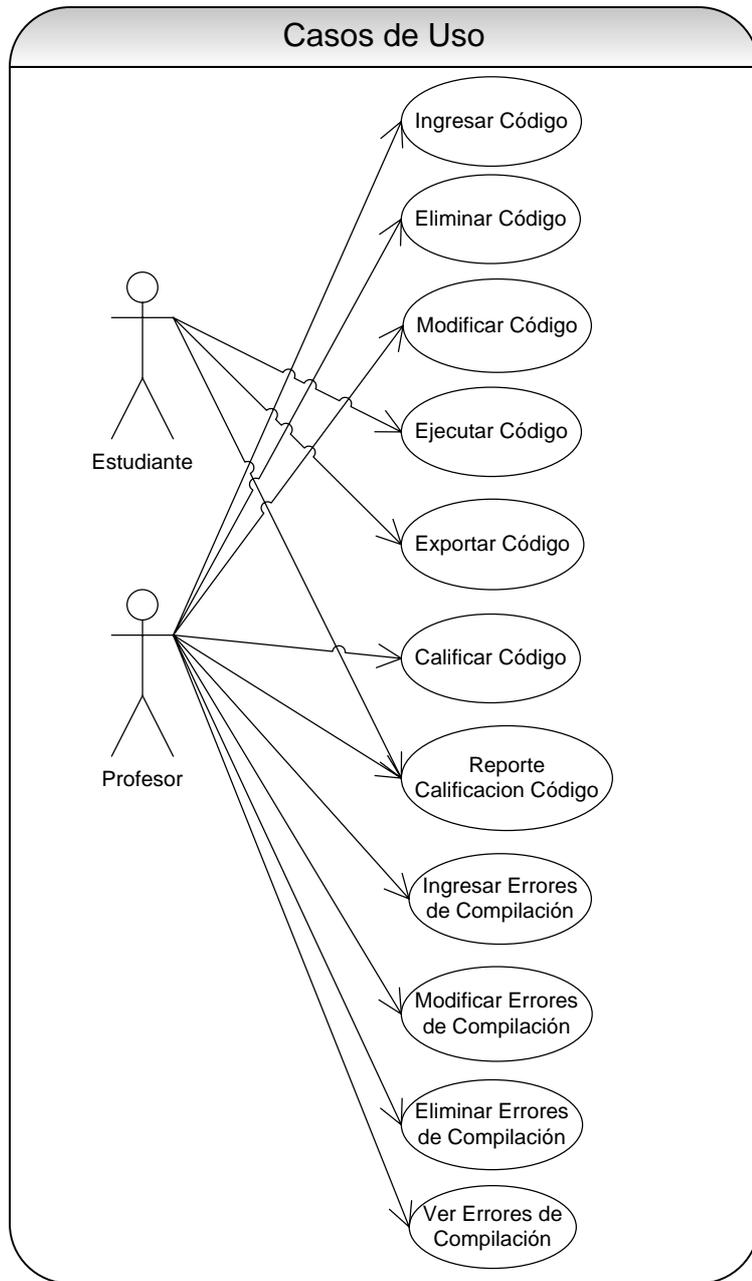


Figura D.1.b.- Continuación del Diagrama de Casos de Uso

## **Actores**

- Estudiante o invitado
- Profesor

## **Listado de Casos de uso**

1. Ingresar al sistema.
2. Ingresar nuevo cuestionario.
3. Modificar cuestionario.
4. Eliminar Cuestionario.
5. Resolver cuestionario.
6. Exportar Cuestionario resuelto.
7. Consultar resultados de cuestionarios.
8. Ingresar Teoría.
9. Eliminar Teoría.
10. Modificar Teoría.
11. Consultar Teoría.
12. Ingresar Código.
13. Modificar Código.
14. Eliminar Código.
15. Ejecutar Código.
16. Exportar Código.
17. Calificar Código.
18. Reporte Calificación Código.
19. Ingresar Errores de Compilación
20. Modificar Errores de Compilación

21. Eliminar Errores de Compilación.

22. Ver Errores de Compilación

## **Descripción de los casos de uso**

### **Caso de Uso 1: Ingresar al Sistema**

**Actor:** Estudiante, Profesor

**Descripción:**

Permite que un usuario ingrese en el sistema.

**Notas:**

En este caso el usuario y clave se encuentran previamente registrados, o también puede ser una persona común que ingresa como invitado

**Escenarios:**

- 1.1 Ingreso exitoso al sistema.
- 1.2 Ingreso no exitoso debido por usuario o contraseña incorrectos.
- 1.3 Ingreso no exitoso por fallas externas

### **Caso de Uso 2: Ingresar Cuestionario**

**Actor:** Profesor

**Descripción:**

El profesor ingresa el cuestionario al sistema, es decir, registra las preguntas y las posibles respuestas, ingresa la información correspondiente al estándar LOM para definir así el nuevo objeto de aprendizaje.

**Notas:**

El profesor adicionalmente puede configurar el tiempo que aproximadamente tomaría la resolución del cuestionario.

**Escenarios:**

- 2.1. Ingreso exitoso de los datos del cuestionario.
- 2.2. Ingreso no exitoso debido a datos erróneos.
- 2.3. Ingreso no exitoso por fallas externas

**Caso de Uso 3: Modificar Cuestionario**

**Actor:** Profesor

**Descripción:**

El profesor modifica un cuestionario ya existente en el sistema, es decir, puede modificar las preguntas, las posibles respuestas, y la información correspondiente al estándar LOM para definir así el nuevo objeto de aprendizaje.

**Notas:**

El profesor adicionalmente puede modificar el tiempo que aproximadamente tomaría la resolución del cuestionario.

**Escenarios:**

- 3.1. Modificación exitosa de los datos del cuestionario.
- 3.2. Modificación no exitosa debido a datos erróneos.
- 3.3. Modificación no exitosa por fallas externas

#### **Caso de Uso 4: Eliminar Cuestionario**

**Actor:** Profesor

**Descripción:**

El profesor puede eliminar un cuestionario ya existente en el sistema

**Notas:**

El Cuestionario no se elimina físicamente de la base de datos, sino que solo se cambia su estado a uno Inactivo, es decir se realiza una eliminación lógica.

**Escenarios:**

- 4.1. Eliminación exitosa de los datos del cuestionario.
- 4.2. Eliminación no exitosa por fallas externas

#### **Caso de Uso 5: Resolver Cuestionario**

**Actor:** Estudiante, Profesor

**Descripción:**

El usuario elige un cuestionario y lo resuelve.

**Notas:**

El usuario selecciona entre las diferentes respuestas que se muestran en pantalla la respuesta que considera es correcta. Adicionalmente tiene un tiempo límite para resolverlo.

**Escenarios:**

- 5.1. Resolución exitosa del cuestionario, podrá ver la retroalimentación y puntuación obtenida. Adicionalmente se enviara un correo al profesor indicando toda la información de la evaluación.
- 5.2. Resolución no exitosa del cuestionario, podrá ver retroalimentación y puntuación obtenida a su vez que se enviara un correo al profesor que creó el cuestionario, con toda la información obtenida de la evaluación.
- 5.3. Error al intentar resolver el cuestionario por factores externos.
- 5.4. Error al enviar el correo.

**Caso de Uso 6: Exportar Cuestionario resuelto**

**Actores:** Profesor, Estudiante

**Descripción:**

El usuario elige la opción de exportar en la resolución del cuestionario, el archivo a descargarse esta en formato PDF.

**Notas:**

Esta opción la podrá realizar cuando ya haiga realizado la resolución del cuestionario.

**Escenarios:**

- 6.1. Exportación del PDF realizada exitosamente.
- 6.2. La exportación no se realizo por que el cuestionario aun no ha sido evaluado.
- 6.3. La exportación del PDF no se dio por fallas externas.

**Caso de Uso 7: Consultar resultados de cuestionarios**

**Actores:** Profesor, Estudiante

**Descripción:**

El usuario va al menú lateral del sistema y en la sección Reportes verifica sus calificaciones.

**Notas:**

La información se muestra como gráfico estadístico y como un reporte.

**Escenarios:**

- 7.1. La consulta fue realizada exitosamente.
- 7.2. La consulta no fue realizada exitosamente. No se encontraron datos para mostrar

### **Caso de Uso 8: Ingresar Teoría**

**Actores:** Profesor

**Descripción:**

El usuario profesor realiza el ingreso de la teoría relacionada a la materia de Fundamentos de Programación.

**Notas:**

Puede publicar imágenes que faciliten la comprensión de la teoría.

**Escenarios:**

- 8.1. Se ha ingresado exitosamente la teoría.
- 8.2. No se ha ingresado exitosamente la teoría debido a problemas de conexión a la Base

### **Caso de Uso 9: Eliminar Teoría**

**Actores:** Profesor

**Descripción:**

El usuario profesor realiza la eliminación de alguna publicación, siempre y cuando este contenido sea existente en la Base de datos, que considere que no es relevante para la materia de estudio.

**Notas:**

La eliminación de una publicación de teoría es de manera lógica, es decir se mantendrá físicamente el contenido desechado.

**Escenarios:**

9.1. Se ha eliminado exitosamente la teoría.

9.2. No se ha eliminado exitosamente la teoría debido a problemas de conexión a la Base

### **Caso de Uso 10: Modificar Teoría**

**Actores:** Profesor

**Descripción:**

El usuario profesor realiza la modificación de alguna publicación, siempre y cuando este contenido sea existente en la Base de datos.

**Notas:**

Puede publicar imágenes que faciliten la comprensión de la teoría.

**Escenarios:**

10.1. Se ha modificado exitosamente la teoría.

10.2. No se ha modificado exitosamente la teoría debido a problemas de conexión a la Base.

### **Caso de Uso 11: Consultar Teoría**

**Actores:** Profesor, Estudiante

**Descripción:**

El usuario realiza una vista del contenido publicado por el profesor seleccionando uno de los ítems que se encuentran en el menú teoría.

**Notas:**

El visor utiliza la metáfora de un libro.

**Escenarios:**

11.1. Se ha visualizado exitosamente la teoría.

11.2. No se ha visualizado exitosamente la teoría debido a problemas de conexión a la Base.

**Caso de Uso 12: Ingresar Código**

**Actores:** Profesor

**Descripción:**

El usuario realiza el ingreso del enunciado de un problema, y propone el código de solución.

**Notas:**

Se realiza una verificación previa de que el código propuesto como solución no tenga errores de compilación.

**Escenarios:**

12.1. Se ha ingresado el enunciado y el código propuesto como solución exitosamente.

12.2. No se ha ingresado exitosamente el enunciado y el código propuesto como solución debido a problemas de conexión con la Base de Datos.

- 12.3. No se ha ingresado exitosamente el enunciado y el código propuesto como solución debido a que ocurrieron errores de compilación.

### **Caso de Uso 13: Modificar Código**

**Actores:** Profesor

**Descripción:**

El usuario realiza la modificación del enunciado de un problema, y propone un nuevo código de solución.

**Notas:**

Se realiza una verificación previa de que la modificación del código propuesto como solución no tenga errores de compilación.

**Escenarios:**

- 13.1. Se ha modificado el enunciado y el código propuesto como solución exitosamente.
- 13.2. No se ha modificado exitosamente el enunciado y el código propuesto como solución debido a problemas de conexión con la Base de Datos.
- 13.3. No se ha modificado exitosamente el enunciado y el código propuesto como solución debido a que ocurrieron errores de compilación.

### **Caso de Uso 14: Eliminar Código**

**Actores:** Profesor

**Descripción:**

El usuario elimina el enunciado de un problema, y el código propuesto como solución.

**Notas:**

La eliminación del enunciado y código se realiza de manera lógica.

**Escenarios:**

- 14.1. Se ha eliminado el enunciado y el código propuesto como solución exitosamente.
- 14.2. No se ha eliminado exitosamente el enunciado y el código propuesto como solución debido a problemas de conexión con la Base de Datos.

### **Caso de Uso 15: Ejecutar Código**

**Actores:** Profesor, Estudiante

**Descripción:**

El usuario selecciona el enunciado que desea resolver y propone el código correspondiente para solucionarlo.

**Notas:**

Se realiza una verificación previa para que el código propuesto no tenga errores de compilación.

**Escenarios:**

- 15.1. Se ha ejecutado exitosamente el código propuesto como solución.
- 15.2. No se ha ejecutado exitosamente el código propuesto como solución.

**Caso de Uso 16: Exportar Código****Actores:** Profesor**Descripción:**

El usuario solicita la descarga del archivo en formato PDF del código que propuso como solución.

**Notas:****Escenarios:**

- 16.1. Se ha descargado el archivo exitosamente.
- 16.2. No se ha descargado el archivo exitosamente debido a problemas de conexión con el servidor.

**Caso de Uso 17: Calificar Código****Actores:** Profesor**Descripción:**

Al usuario profesor le llega una notificación vía correo electrónico en el que se le informa que un estudiante a resuelto un ejemplo, el profesor

procede a ingresar al sistema para revisar la solución propuesta y emite una calificación acorde con la solución.

**Notas:**

La calificación no debe ser menor a 0 ni mayor a 100 para que sea considerada válida.

**Escenarios:**

- 17.1. Se ha ingresado exitosamente la calificación de la solución propuesta.
- 17.2. No se ha ingresado exitosamente la calificación de la solución propuesta debido a errores de conexión con el servidor.
- 17.3. No se ha ingresado exitosamente la calificación de la solución propuesta debido a que se ha ingresado una calificación inválida.

**Caso de Uso 18: Reporte Calificación Código**

**Actores:** Profesor, Estudiante

**Descripción:**

El usuario selecciona la opción Calificación Código para ver las calificaciones que obtuvieron sus alumnos.

**Notas:**

**Escenarios:**

- 18.1. Se muestra el reporte de calificaciones pues hay calificaciones registradas.
- 18.2. No se muestra el reporte de calificaciones pues ningún alumno ha desarrollado algún ejercicio.
- 18.3. No se muestra el reporte de calificaciones debido a un error de conexión con el servidor.

### **Caso de Uso 19: Ingresar Errores de Compilación**

**Actores:** Profesor

**Descripción:**

El usuario ingresa ejemplos de errores que pueden ocurrir generalmente al momento de compilar con el GCC para ayudar a los alumnos.

**Notas:**

**Escenarios:**

- 19.1. Se ha ingresado exitosamente el error a la base de datos.
- 19.2. No se ha ingresado exitosamente el error debido a problemas de conexión con el servidor.

### **Caso de Uso 20: Modificar Errores de Compilación**

**Actores:** Profesor

**Descripción:**

El usuario modifica ejemplos de errores, que se encuentren registrados en la base de datos, que pueden ocurrir generalmente al momento de compilar con el GCC para ayudar a los alumnos.

**Notas:**

**Escenarios:**

- 20.1. Se ha modificado exitosamente el error a la base de datos.
- 20.2. No se ha modificado exitosamente el error debido a que no se encuentra esa información en la base de datos.
- 20.1. No se ha modificado exitosamente el error debido a problemas de conexión con el servidor.

**Caso de Uso 21: Eliminar Errores de Compilación**

**Actores:** Profesor

**Descripción:**

El usuario elimina ejemplos de errores que pueden ocurrir generalmente al momento de compilar con el GCC para ayudar a los alumnos.

**Notas:**

La eliminación es de tipo lógico, es decir se mantiene físicamente el registro del error de compilación que se desea eliminar.

**Escenarios:**

- 21.1. Se ha eliminado exitosamente el error a la base de datos.

21.2. No se ha eliminado exitosamente el error debido a que no se encuentra esa información en la base de datos.

21.3. No se ha eliminado exitosamente el error debido a problemas de conexión con el servidor.

### **Caso de Uso 22: Ver Errores de Compilación**

**Actores:** Profesor, Estudiante

**Descripción:**

El usuario puede consultar en el sistema el error que se le ha generado en el compilador, a través de una búsqueda en base al nombre del error.

**Notas:**

**Escenarios:**

22.1. Se ha encontrado exitosamente el error de compilación solicitado.

22.2. No se ha encontrado exitosamente el error de compilación solicitado debido a un error de conexión con el servidor.

22.3. No se ha encontrado exitosamente el error de compilación solicitado debido a que no se encuentra registrado en la base de datos

# BIBLIOGRAFÍA

(1) PROBLEMS IN LEARNING AND TEACHING PROGRAMMING- a literature study for developing visualizations in the Codewitz-Minerva Project [En Línea]  
[http://www.cs.tut.fi/~edge/literature\\_study.pdf](http://www.cs.tut.fi/~edge/literature_study.pdf)

(2) Why Students Drop Out CS1 Course? [En Línea]  
<http://www.cs.kent.ac.uk/events/conf/2006/icer/icer-talks/2006-icer-kinnunen-malmi.pdf>

(3) Aplicaciones de internet ricas [En Línea]  
[http://es.wikipedia.org/wiki/Aplicaciones\\_de\\_Internet\\_Ricas](http://es.wikipedia.org/wiki/Aplicaciones_de_Internet_Ricas)

(4) Learning Technology Standards Committee. IEEE Standard for Learning Object Metadata. IEEE Standard 1484.12.1, Institute of Electrical and Electronics Engineers, New York 2002. [En Línea]  
[http://ltsc.ieee.org/wg12/files/LOM\\_1484\\_12\\_1\\_v1\\_Final\\_Draft.pdf](http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf)

(5) Unidades de aprendizaje, una propuesta de complemento a los objetos de aprendizaje. [En Línea]

[http://www.usal.es/~teoriaeducacion/rev\\_numero\\_06\\_2/n6\\_02\\_art\\_hernandez.htm](http://www.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_hernandez.htm)

(6) Los Repositorios de Objetos de Aprendizaje como soporte para los entornos e-learning [En Línea]

[http://www.biblioweb.dgsca.unam.mx/libros/repositorios/objetos\\_aprendizaje.htm](http://www.biblioweb.dgsca.unam.mx/libros/repositorios/objetos_aprendizaje.htm)

(7) Standard for CS Introductory courses by ACM [En Línea]

<http://www.sigcse.org/cc2001/cs-introductory-courses.html>

(8) Que es Flex? [En Línea] <http://www.holaflex.com/?p=34>

(9) FLEX [En Línea] [http://es.wikipedia.org/wiki/Adobe\\_Flex](http://es.wikipedia.org/wiki/Adobe_Flex)

(10) Arquitectura cliente / servidor [En Línea]

<http://barrapunto.com/articles/05/12/30/1255225.shtml>

(11) Protocolos de comunicación soportados por Flex [En Línea]

<http://www.madeinflex.com/2008/06/12/protocolos-de-comunicacion/>

(12) Apache-Tomcat [En Línea]<http://tomcat.apache.org/tomcat-6.0-doc/index.html>

(13) Checklist for Evaluating SREB-SCORE Learning Objects [En Línea]  
[http://www.sreb.org/programs/edTech/pubs/2007pubs/07T05\\_Checklist\\_Eval\\_SREB-SCORE.pdf](http://www.sreb.org/programs/edTech/pubs/2007pubs/07T05_Checklist_Eval_SREB-SCORE.pdf)

(14) Learning Objects Review Instrument (LORI) [En Línea]  
[http://api.ning.com/files/YGI6OCOWuUjNiGULKX1uofYsdcvKb2j\\*8u2tiw4oXVQQzPd03qupBqNojROpv4xJAfi8V4Y4glEj\\*NOQmvjxgAMXxzOi7Yhq/lori\\_learning\\_object\\_review.pdf](http://api.ning.com/files/YGI6OCOWuUjNiGULKX1uofYsdcvKb2j*8u2tiw4oXVQQzPd03qupBqNojROpv4xJAfi8V4Y4glEj*NOQmvjxgAMXxzOi7Yhq/lori_learning_object_review.pdf)