

**UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA**



**TRABAJO DE GRADUACIÓN
PARA OPTAR AL GRADO DE
INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**

**DESARROLLO DE UN EDITOR DE LENGUAJE DE MODELAMIENTO
UNIFICADO (UML) PARA LOS DIAGRAMAS DE OBJETOS Y CLASES, CON UN
INTÉRPRETE DE DIAGRAMAS DE CLASES A CÓDIGO C++**

**PRESENTADOR POR:
HENRY WILLY MELARA TORRES
HAZEL PAZ COLATO
JOSUÉ ROLANDO RODRÍGUEZ NAVARRO**

**ASESOR:
LIC. SALVADOR ANTONIO SIBRIAN**

**MAYO 2008
EL SALVADOR, CENTROAMÉRICA**

ÍNDICE

CAPITULO 1 - MARCO REFERENCIAL	1
1.1 ANTECEDENTES	12
1.2 PROBLEMÁTICA	19
1.3 DEFINICIÓN DEL TEMA	20
1.4 JUSTIFICACIÓN	23
1.5 OBJETIVOS	25
1.5.1 GENERAL	25
1.5.2 ESPECÍFICOS	25
1.6 ALCANCES	26
1.7 LIMITACIONES	27
1.8 MARCO TEÓRICO	28
1.8.1 MARCO HISTÓRICO	28
1.8.2 MARCO CONCEPTUAL	28
1.8.2.1 HISTORIA DE HERRAMIENTAS CASE	28
1.8.2.2 COMPONENTES DE UML	29
1.8.2.3 CARACTERÍSTICAS DE MODELADO CON UML	31
1.9 METODOLOGÍAS AUDIOVISUAL	32
1.9.1 RECURSOS AUDIOVISUALES	34
1.9.2 LENGUAJES DE PROGRAMACIÓN	36
1.9.2.1 FRAMEWORK .NET (MARCO DE TRABAJO .NET)	36
1.9.2.2 JAVASCRIPT	40
1.10 METODOLOGÍA Y TÉCNICAS DE INVESTIGACIÓN	41
1.10.1 TÉCNICAS DE RECOLECCIÓN DE DATOS	41
1.10.2 TÉCNICAS DE DISEÑO Y DESARROLLO	41
CAPITULO 2 - DEFINICIÓN DE LA SOLUCIÓN PROPUESTA	45
2.1 METODOLOGÍA DE INVESTIGACIÓN POR ENCUESTAS	46
2.1.1 LA ENCUESTA	46
2.1.2 PASOS A SEGUIR EN LA INVESTIGACIÓN MEDIANTE LA ENCUESTA	48

2.1.3	DISEÑO DE LA MUESTRA	49
2.1.4	TIPOS DE GRÁFICOS	51
2.2	FORMULACIÓN DE LA ENCUESTA.....	52
2.2.1	CALCULO DE LA MUESTRA	53
2.2.2	DATOS GENERALES DE LOS ENCUESTADOS	54
2.3	ANÁLISIS DE RESULTADOS.....	55
CAPITULO 3 - ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PLANTEADA		65
3.1	DIAGRAMAS DE CASOS DE USO	66
3.1.1	DEFINICIÓN DE LOS CASOS DE USO	66
3.1.2	DIAGRAMAS DE ACTIVIDADES.....	68
3.1.3	DIAGRAMA DE CLASES.....	70
3.1.4	DIAGRAMAS CASO DE USO, ACTIVIDADES Y DE CLASES DE UML DE LA SOLUCIÓN PLANTEADA	71
3.1.5	DIAGRAMA DE CLASES DE LA SOLUCIÓN PLANTEADA.....	76
3.2	ANÁLISIS DE HERRAMIENTAS PARA EL DESARROLLO DEL EDITOR E INTÉRPRETE.....	78
3.3	DEFINICIÓN DE LA SOLUCIÓN	79
3.4	DESCRIPCIÓN DE INTERFAZ.....	80
CONCLUSIONES		84
RECOMENDACIONES.....		86
BIBLIOGRAFÍA.....		87
SITIOS WEB.....		87
GLOSARIO DE TÉRMINOS		89
ANEXOS.....		90

ÍNDICE DE ILUSTRACIONES

Ilustración 1	Esquema general del programa	21
Ilustración 2	Esquema general de la ayuda.....	22
Ilustración 3	Diagrama de intérprete.....	22
Ilustración 4	Elementos de UML.....	30
Ilustración 5	Diagrama detallado del Marco de Trabajo .NET	36
Ilustración 6	Vista preliminar de la Infraestructura común del Lenguaje.....	38
Ilustración 7	Ciclo de vida del sistema (Modelo en cascada)	42
Ilustración 8	Caso de uso - actores involucrados	71
Ilustración 9	Caso de uso del editor de Diagramas de clases y objetos.....	72
Ilustración 10	Diagrama de actividades del editor	72
Ilustración 11	Caso de uso de los recursos audiovisuales	73
Ilustración 12	Diagrama de actividades del intérprete.	74
Ilustración 13	Caso de uso del intérprete	74
Ilustración 14	Diagramas de actividades de metodología.....	75
Ilustración 15	Diagrama de clases de la aplicación Parte 1.....	77
Ilustración 16	Diagrama de clases de la aplicación Parte 2.....	77

ÍNDICE DE TABLAS

Tabla 1	Diagrama de la estructura interna del Entorno de Común de Ejecución para Lenguajes (CLR por sus siglas en inglés)	38
Tabla 2	Requerimiento de equipo para el desarrollo	43
Tabla 3	Requerimiento de equipo para la implementación	44
Tabla 4	Presupuesto de desarrollo	101
Tabla 5	Presupuesto de implementación.....	101
Tabla 6	Descripción de gráficos mas comunes para la representación de resultados estadísticos	51
Tabla 7	Elementos de caso de uso.....	67

ÍNDICE DE GRÁFICOS

Gráfico 1	Distribución de estudiantes encuestados por universidades	54
Gráfico 2	Resultados a conciencia acerca si conoce de UML	56
Gráfico 3	Niveles de conocimiento.....	57
Gráfico 4	Porcentaje verdadero de conocimiento	57
Gráfico 5	Frecuencia de aplicaciones mas utilizadas para desarrollar diagramas de UML	58
Gráfico 6	Resultados de las calificaciones de características de las aplicaciones gratuitas y comerciales	60
Gráfico 7	Resultados de preferencia de duración de un video demostrativo	62
Gráfico 8	Resultado de numero de diapositivas idóneas para una clase expositiva.....	62
Gráfico 9	Resultado de tipo de formato de documentos	63
Gráfico 10	Resultados de los proyectos.....	99

INTRODUCCIÓN

En un mundo globalizado en donde las empresas son muy competitivas, se implementan sistemas informáticos que necesitan una planificación temprana para obtener un desarrollo adecuado, evitando así inconvenientes. Debido a estas exigencias, se han creado diferentes metodologías de planificación, las cuales ayudan a evitar gastos innecesarios por el mal diseño de un proyecto.

UML (Unified Modeling Language)² es una metodología aplicada a diferentes ramas de cualquier ámbito profesional, dentro de estas se encuentra la informática, en la cual permite el modelado de sistemas, para un mejor desarrollo de proyectos. La creación de aplicaciones didácticas para la enseñanza de diagramas UML se debe considerar muy importante, ya que el buen uso de estos diagramas facilita al programador desarrollar sistemas robustos de manera eficiente.

Este es el caso del sistema que se describe en el presente documento, cuyo fin es facilitar el proceso de aprendizaje de los diagramas UML por medio de un editor con ayuda multimedia interactiva, integrando un interprete de diagramas de clase a estructuras de clase de código C++. Cabe mencionar que la aplicación se desarrollara con un enfoque didáctico y no con un fin comercial.

Esta aplicación pretende crear conciencia a los estudiantes para que comprendan desde un inicio que la fase de diseño del sistema, es tan importante como las demás fases, y que una de las herramientas más utilizadas son los diagramas UML.

² UML Unified Modeling Language por sus siglas en inglés. Lenguaje de Modelado Unificado.

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA



RECTOR

ING. FEDERICO MIGUEL HUGUET RIVERA

VICERRECTOR

PADRE VICTOR BERMUDEZ

SECRETARIO GENERAL

LIC. MARIO RAFAEL OLMOS

AGRADECIMIENTOS

A Dios, quien me dio la fortaleza para seguir adelante, aun en los peores momentos; porque me enseñó a tener paciencia y comprensión en las malas ocasiones.

A mis padres quienes siempre me apoyaron desde un principio, en las buenas y en las malas, hasta al final de todo; brindándome soporte, ánimos de seguir adelante, en los momentos mas difíciles.

A mi la familia Paz Colato, mama Ina, Marina, Graciela Paz, Gracielita y Gammie, quienes desde el primer momento nos brindaron todo el apoyo, colaboración y consejos. Muchas gracias por sus cuidados y por la hospitalidad excelente que me entregaron cuando llegaba a su casa para las reuniones.

A todos mis familiares y amigos que me brindaron palabras de aliento para poder terminar este ultimo episodio de mi carrera universitaria.

A nuestro asesor Salvador Sibrián y Raúl Pacas (quien nos abandono a medio camino, pero igual le agradezco) y a nuestra lectora Mercy Cáceres quien siempre nos estuvo apoyando en todo, primero como tutora y luego como lectora, y mil disculpas por las veces que no entregamos las cosas a tiempo y muchísimas gracias por todas las palabras de aliento que nos dio al final justo cuando las cosas se pusieron peores.

Henry Willy Melara

AGRADECIMIENTOS

A Dios, por todas las bendiciones que he recibido en mi vida, por darme la fortaleza de seguir y terminar esta etapa de mi vida.

A mi familia, mi papá por su apoyo en mi carrera, a mi mamá por ser mi ejemplo de vida, por darme fuerzas cada vez que necesitaba. Mi hermana Marina por cuidarme y apoyarme siempre, a mi hermana Graciela por su ayuda en mis estudios. A mis sobrinos queridos Graciela y Alejandro, que son parte de mi motivación de superación.

A la familia Melara Torres, por acogerme en su hogar y darme todos esos consejos valiosos y ánimos cuando los necesitaba.

A nuestro asesor Salvador Sibrian, gracias por todo su valioso tiempo, por sus consejos que hicieron que nuestro trabajo fuera un éxito.

A nuestra lectora Ana Mercedes, por todos sus consejos.

A mi amigo Willy, gracias por tu paciencia, comprensión, apoyo en las buenas y en las malas.

A mi amigo Ernesto, por ser mi compañero, amigo y hermano en todo esta etapa de nuestras vidas. Mil gracias.

A mis amigos, Rolando, Geraldine, Carolina por ayudarme y darme ánimos cuando ya no tenía fuerzas.

A Salvador, gracias por existir en mi vida y por todo tu apoyo para hacer realidad mi meta.

Hazel Paz

AGRADECIMIENTOS

A Dios Todopoderoso que me ha permitido finalizar mis estudios profesionales.

A mis queridos padres Mirian Navarro de Rodríguez y Rolando Rodríguez Huevo por el amor y apoyo incondicional que me dieron a lo largo de la carrera y a lo largo de mi vida, por enseñarme que no hay límites, que lo que me proponga lo puedo lograr y que solo depende de mi.

A mi hermana Cesy Rodríguez por su amor, por que siempre me ha brindado su apoyo y siempre ha luchado por mi.

A mis hermanos Karla y Anthony por su cariño y su apoyo.

A Verónica, por su apoyo incondicional, por creer en mí, por compartir mis sueños y hacerlos realidad.

A los protagonistas de este proyecto, Hazel Paz y Willy Melara grandes amigos, por su participación activa en el proyecto, su paciencia, apoyo y su amistad.

A los Docentes, Asesor, tutor y jurado de la Universidad Don Bosco, por formar parte en este proyecto y guiarnos para culminar en éxito

Rolando Rodríguez

CAPITULO 1 - MARCO REFERENCIAL

1.1 ANTECEDENTES

El lenguaje unificado de modelado o UML ("Unified Modeling Language" por sus siglas en ingles) esta consolidado como el lenguaje estándar en el análisis y diseño de sistemas de computo, según la OMG ("Object Management Group" por sus siglas en ingles) entidad regulatoria de desarrollo de estándares para corporaciones. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para representar un sistema de software previo al proceso de su desarrollo. En otros términos, así como en la construcción de un edificio se realizan planos previo a su construcción, en software se deben realizar diseños en UML previa codificación de un sistema.

La enseñanza de los diagramas UML en la carrera del grado de Ingeniería en computación tiene como objetivo que el estudiante tenga conocimientos acerca del uso y las ventajas de estos diagramas. Entre las ventajas se pueden mencionar la unificación de los pasos en el desarrollo de un sistema y la integración de un modelado de negocio, entre otras.

En la actualidad, las universidades con mayor auge de estudiantes de carreras relacionadas a computación en El Salvador como la Universidad Francisco Gavidia (UFG), Universidad Tecnológica (UTEC) y la Universidad Don Bosco (UDB) han integrado en sus planes de estudio la enseñanza de UML en los primeros años en la carrera. Por ejemplo, en la UDB en el plan de estudio del 2003 integraron en el ciclo II la materia Programación II, en el cual se imparte seis horas clase y cuatro horas de laboratorio sobre los diagramas UML. Igualmente, en la UFG han integrado en la materia diseño de software ocho horas clase y ocho horas de laboratorio.

Existen diferentes tipos de métodos de enseñanza los cuales facilitan al docente transmitir los conocimientos acerca de diferentes temas. En el caso de los diagramas UML algunos de los métodos utilizados por los docentes se describen a continuación:

a) Clase demostrativa: La clase demostrativa es aquella técnica en que lo visual es el medio de comunicación utilizado. Normalmente, es un proceso de observación pero puede ser complementada por la participación del estudiante. La clase demostrativa abarca varios elementos: exposición oral, muestra de materiales didácticos, preguntas y pequeños laboratorios evaluativos. Se desarrolla en un salón de clase con un grupo de estudiantes que no exceda la capacidad del docente de controlarlos.

Ventajas:

- Favorece la participación del estudiante sobre alguna interrogante del tema.
- Permite al docente transmitir sus ideas a un buen grupo de estudiantes.

Desventajas.

- Si el grupo de estudiantes es muy grande, no se podrá hacer una evaluación individual más exhaustiva.
- No todos los estudiantes escuchan, ven y entienden con igual claridad.
- La eficiencia del docente durante la charla dependerá de su estado de ánimo, confianza en si mismo y capacidad de hablar en público.

b) Exposiciones: Esta técnica puede utilizar diversas formas que dependen del objetivo inmediato y de la naturaleza del tema; que pueden ser exposiciones grupales o individuales. Estas exposiciones son realizadas por los mismos estudiantes que abordan diferentes conceptos para cumplir de los objetivos de la clase. Se planea cuidadosamente la exposición, determinando objetivos, trazando el tiempo esencial del asunto, previendo tareas específicas y calculando el tiempo que consumirá, que debe de ser entre unos 15 a 25 minutos.

Ventajas:

- Puede emplearse con facilidad.
- Puede usarse con grupos grandes.
- Permite la intervención del docente.

Desventajas:

- Escasa participación en respuestas.
- El expositor debe conocer muy bien el tema y además, ser un buen orador.
- Fácilmente se puede perder el interés de la audiencia.
- No hay buena retroalimentación.

e) Práctica de laboratorio: Es una actividad que tiene como objetivo poner al estudiante ante una situación práctica de ejecución. Este método ayuda a desarrollar aptitudes para la observación y la coordinación de ejercicios prácticos con capacidades de análisis y síntesis. Al estar en contacto con situaciones reales el estudiante presta más atención al material leído, estudiado, observado o discutido en la hora clase.

Ventajas

- Fomenta la investigación.
- Aclara dudas sobre algún tema en específico.

Desventajas

- Se necesita que todos y cada uno de los estudiantes posean las herramientas necesarias.
- El instructor tiene que tener seguridad y pleno dominio del tema que ha de ser tratado.

d) Trabajos grupales: Esta técnica esta basada en diferentes actividades de dinámica de grupo. En el cual tiene como fin activar las motivaciones individuales y de estimular tanto la dinámica interna como la externa, de manera que los estudiantes puedan integrar las metas del grupo. Las técnicas grupales deben aplicarse con un objetivo claro y bien determinado, tales como una actitud cooperante y no competitiva, asi como crear una atmósfera cordial y democrática.

Ventajas:

- Obtención de información sobre diversos aspectos de un mismo tema.
- Tratamiento de temas de actualidad.

Desventajas:

- Necesita tiempo para planificar la reunión.
- Exige control sobre la emotividad de los participantes.
- El estudiante guía debe ser un líder muy hábil y estar muy bien preparado.

Existen diferentes recursos didácticos que se utilizan normalmente en las diferentes universidades citadas anteriormente para la enseñanza de los diagramas UML, las cuales se detallan a continuación:

a) Pizarra con yeso: Recurso tradicional utilizado por una gran cantidad de docentes debido a su práctico uso y efectividad de expresar ideas.

b) Retroproyector de acetatos: Recurso que permite mediante un sistema simple de superposición de hojas de acetato proyectar una imagen o texto.

c) Computadora con proyector: Recurso tecnológico que se conforma por una computadora conectada a un proyector con el cual se muestra una presentación multimedia en formato digital con diversos elementos como texto, animación y/o videos.

d) Folletos: Recurso en formato impreso que complementa la clase expositiva con material relacionado al tema.

e) Bibliográficos: Recursos que apoyan al docente con diferentes libros de texto.

d) Páginas o sitios web: Recursos virtuales en donde el estudiante dispone diferentes materiales digitales o presentaciones acerca de un tema en específico.

f) Programas de aplicación: Recurso informático el cual brinda la facilidad de aplicar los conceptos teóricos a la práctica por medio de programas tales como editores de diagramas UML e interpretes de diagrama UML de clase a diferentes lenguajes.

Existen dos tipos de licenciamiento para programas de aplicación de diagramas UML: una bajo condiciones de libre distribución y otra que se adquiere por medio de licencias comerciales. En ambos casos se han desarrollado editores que simplifican al usuario la tarea de diseñar sistemas, utilizando los diagramas de UML.

Los editores de diagramas UML de libre distribución, están disponibles al público para adquirirlas sin ningún costo. Algunas de estas permiten la generación de código e ingeniería inversa.. Entre los editores mas populares que existen podemos mencionar.

a) Argo UML: Es una aplicación de diagramado de UML desarrollada en lenguaje Java y publicada bajo la licencia libre. Se utiliza para el modelado de sistemas, con la que se realizan diseños en UML, llevados acabo en el análisis y pre-diseño de sistemas de software. Dado que es una aplicación Java, es una aplicación multi-plaforma. Dentro de sus características y debilidades tenemos:

Características:

- Interfaz de usuario en diversidad de idiomas como Inglés, Alemán, Francés, Español, Ruso entre otros.
- Soporte para generación de código en varios lenguajes: Java, PHP, Python, C++ y Csharp (C#).
- Ingeniería inversa.
- Disposición automática del diagrama de clases.
- Generación de imágenes en formato PNG, GIF, JPG, SVG, EPS de los diagramas.
- Soporte para comentarios para múltiples elementos.
- Todos los diagramas de Java v.1.4 están soportados.

Debilidades:

- Carece de botón "deshacer".

- Los modelos usualmente no pueden ser re-abiertos.
- No hay llamadas-reflexivas en los diagramas de secuencia.
- Se debe seleccionar una clase para crear un diagrama de secuencia.

b) DIA: Es una aplicación gráfica de propósito general para la creación de diagramas. Está diseñada de forma modular, con diferentes paquetes de formas para diferentes necesidades. Fue concebida como un sustituto de la aplicación comercial Visio de Microsoft. Se puede utilizar para dibujar diferentes tipos de diagramas y actualmente se incluyen diagramas entidad-relación, diagramas UML, diagramas de flujo, diagramas de redes, diagramas de circuitos eléctricos, etc.

Por el contrario, existen aplicaciones comerciales que además incluyen beneficios, tales como soporte y capacitación para el correcto uso de la aplicación, soporte multiusuario, ingeniería inversa y generación de código Java, Visual Basic, ANSI C++, C#, entre otros. Entre las versiones comerciales se mencionan las siguientes:

a) IBM Rational Rose: Es una poderosa herramienta de modelado visual para el análisis y diseño de sistemas basados en objetos. Cubre todo el ciclo de vida de un proyecto:

- Concepción y formalización del modelo.
- Construcción de los componentes.
- Transición a los usuarios.
- Certificación de las distintas fases.

IBM Rational Rose consta de los siguientes elementos principales:

- Browser ó Navegador, que permite navegar rápidamente a través de las distintas vistas del modelo
- Ventana de documentación, para manejar los documentos del ítem seleccionado en cualquiera de los diagramas.

- Barra de herramientas: para acceder rápidamente a las acciones comunes a ejecutar para cada uno de los diagramas del modelo.

b) Microsoft VISIO: Es un conjunto de software de dibujo vectorial para Microsoft Windows. Las herramientas que lo componen permiten realizar diagramas de oficinas, de bases de datos, de flujo de programas, de UML, etc. que permiten iniciar al usuario en los lenguajes de programación. Visio es un programa para la creación de diagramas que puede ayuda al usuario a crear diagramas empresariales y técnicos en los que se documenten y organicen ideas, procesos y sistemas complejos.

En la investigación realizada dentro de la Universidad Don Bosco, se encontró un trabajo de graduación³, en el cual se realizo un editor de diagrama de casos de uso de UML. Dentro de sus ventajas se encuentran:

- Ambiente amigable y práctico para el usuario.
- La aplicación se ejecuta en plataforma Windows XP.

Dentro de sus desventajas se encuentran:

- No permite imprimir los diagramas utilizando la metodología WYSIWYG (por sus siglas en ingles What You See Is What You Get).
- No posee una documentación para que el usuario complemente su aprendizaje en los diagramas UML de caso de uso.

Este sistema tiene la capacidad de satisfacer las necesidades en la elaboración de los diagramas de caso de uso, ya que implementa las tres fases de los diagramas, a saber: inclusión, extensión y herencia. Además, este almacena en una estructura de datos los elementos del diagrama, componentes y características de un modelo. También permite al usuario trabajar de forma amigable y fácil.

³ Trabajo de graduación "Desarrollo de un editor de Lenguaje de Modelamiento Unificado (UML) para diagramas de Casos de Uso" elaborado por: Gabriela Herrera y Francisco Arévalo Universidad Don Bosco Octubre 2006.

1.2 PROBLEMÁTICA

Debido a la evolución en las técnicas y herramientas de POO (Programación orientada a objetos), las empresas han logrado disminuir costos y tiempos en el desarrollo e implementación de sus sistemas. Por esta razón, se han diseñado programas que ayudan en este proceso, originando así los editores e intérpretes de los diagramas UML, proporcionando una ayuda visual al usuario.

El presente proyecto busca solucionar los siguientes aspectos:

- Dados los diferentes ritmos de aprendizaje de los estudiantes combinado con la metodología de la mayoría de catedráticos en el medio, da como resultado una pobre implementación y utilización de los conceptos de los diagramas de UML de clases y objetos .
- Los estudiantes no cuentan con una herramienta práctica que desarrolle su conocimiento teórico y fortalezcan el aprendizaje de los diagramas de UML de clases y objetos.
- En El Salvador no se cuenta con una herramienta que integre un editor de diagramas UML de clases y objetos, el cual ofrezca orientación al usuario y que a la vez posean un intérprete que convierta diagramas de UML a estructuras de código de lenguaje C++ con el fin de apoyar en forma didáctica a los estudiantes universitarios.
- En el medio existen diferentes programas, tanto comerciales como de uso libre, que tienen la capacidad de ser editores y poseen un intérprete pero poseen la desventaja de ser muy complejos para un estudiante principiante o son liberados al público cuando no han sido probados exhaustivamente.

1.3 DEFINICIÓN DEL TEMA

UML es un lenguaje de modelado de aplicaciones, que se utiliza para el análisis y el diseño del desarrollo de una herramienta de software.

Este paradigma de modelado de diagramas enfatiza en los estudiantes la enseñanza, en forma audiovisual, la funcionabilidad de los diagramas de objetos y de clases de UML. Como ayuda audiovisual se suele utilizar videos demostrativos acompañados de una explicación en audio.

Las partes que integran este programa son:

- Un editor de diagramas de clases y objetos de UML.
- Ayuda interactiva de los diagramas de clases y objetos de UML.
- Intérprete de diagramas de clases a estructuras de clases de código C++.

El programa esta compuesto por 3 importantes componentes, a saber:

- a) Editor de diagramas de clases y objetos de UML: Este brinda al usuario la posibilidad de crear diagramas de clases y objetos, con la ayuda de una barra de herramientas que contiene cada uno de los elementos necesarios para la creación de cada diagrama. Así mismo, posee las opciones de abrir, guardar e imprimir los diagramas realizados en él.

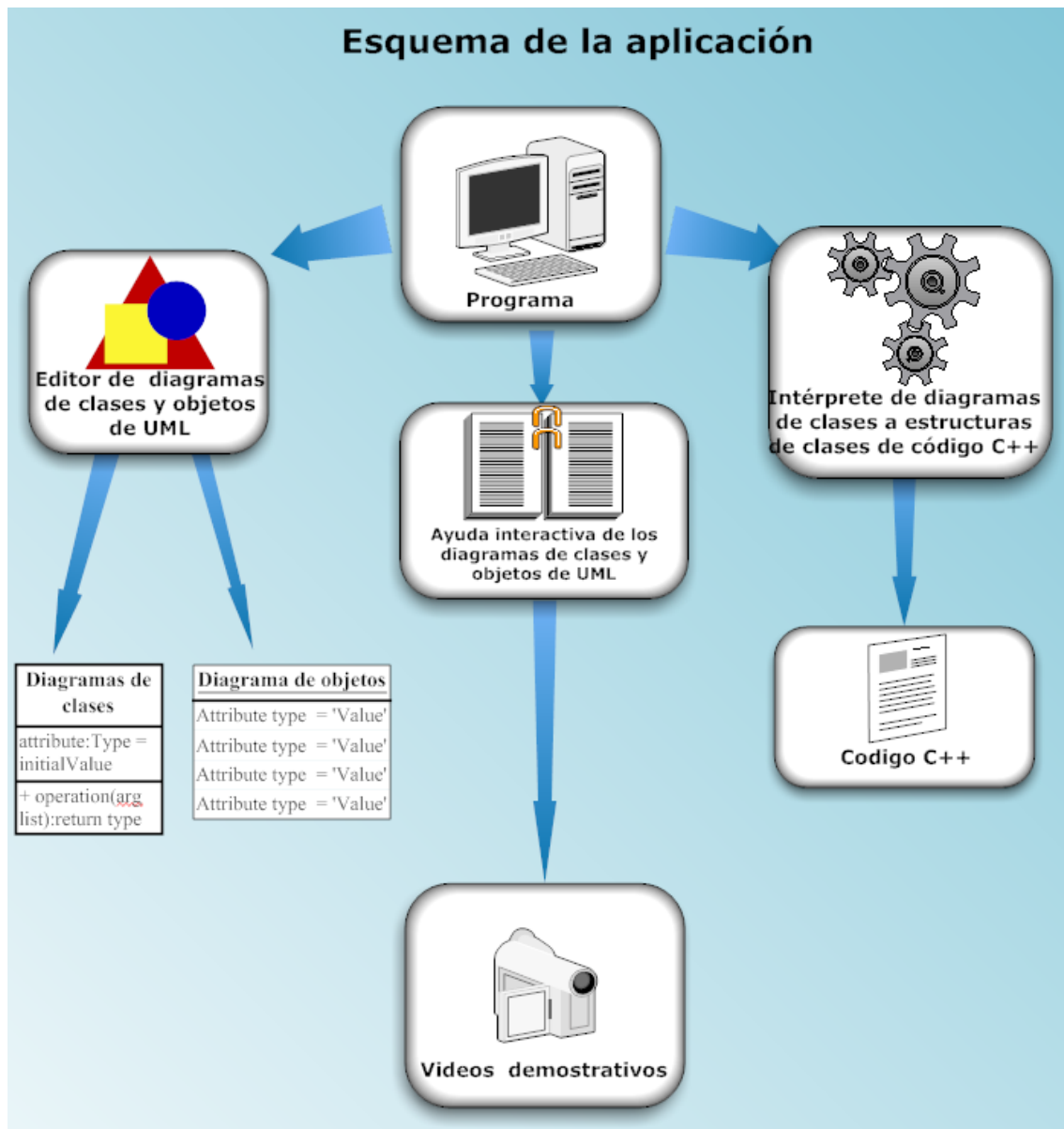


Ilustración 1 Esquema general del programa

- b) Ayuda interactiva de los diagramas de clases y objetos de UML: este componente tiene como finalidad apoyar al estudiante en el aprendizaje de los diagramas de clases y objetos de UML, por medio de ejercicios resueltos, ejercicios para resolver con diferentes grados de complejidad y una lista de contenido de temas. El objetivo de la ayuda consiste en presentar material audiovisual (videos interactivos, audio y texto) que induzcan al estudiante a un mejor aprendizaje y a su vez más práctico de los diagramas por medio del uso del editor.

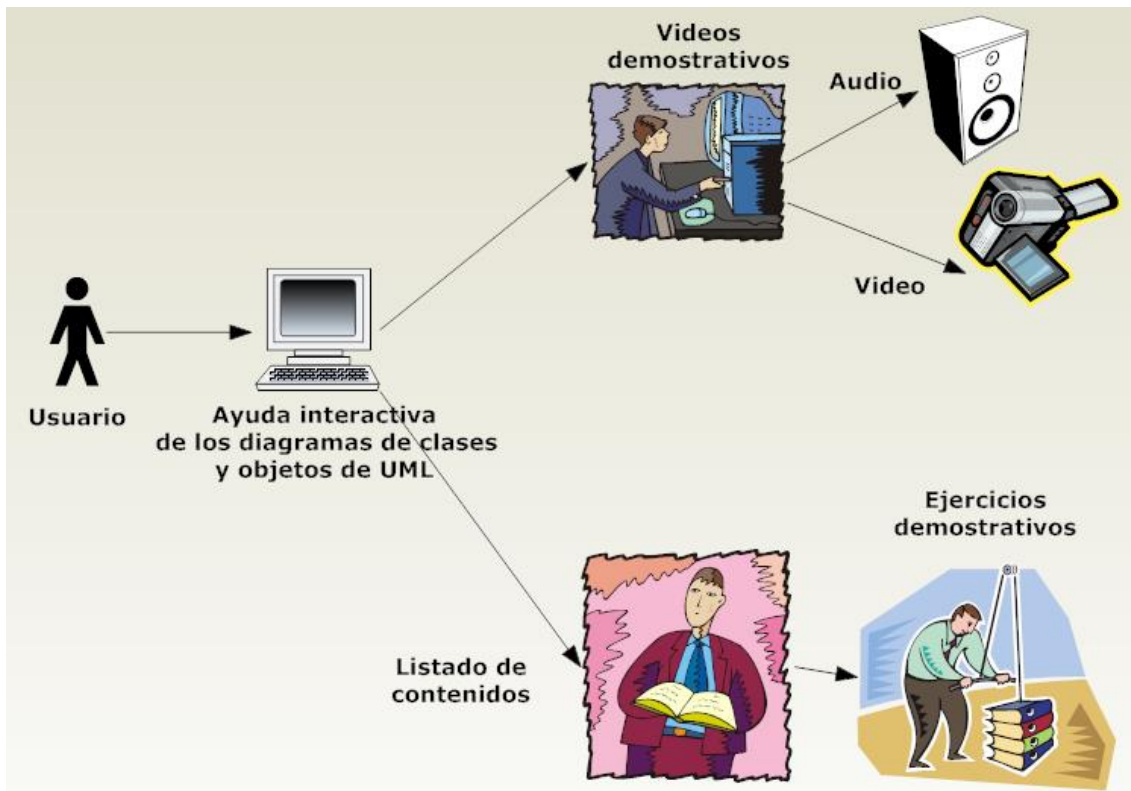


Ilustración 2 . Esquema general de la ayuda

c) Intérprete de diagramas de clases a estructuras de clases de código C++: esta funcionalidad del programa pretende enseñar al estudiante una de las bondades de los diagrama de clase de UML; esta es, convertir el diagrama clases a una estructura de clases de código C++. La estructura de clase resultante se guarda en un archivo de extensión cpp

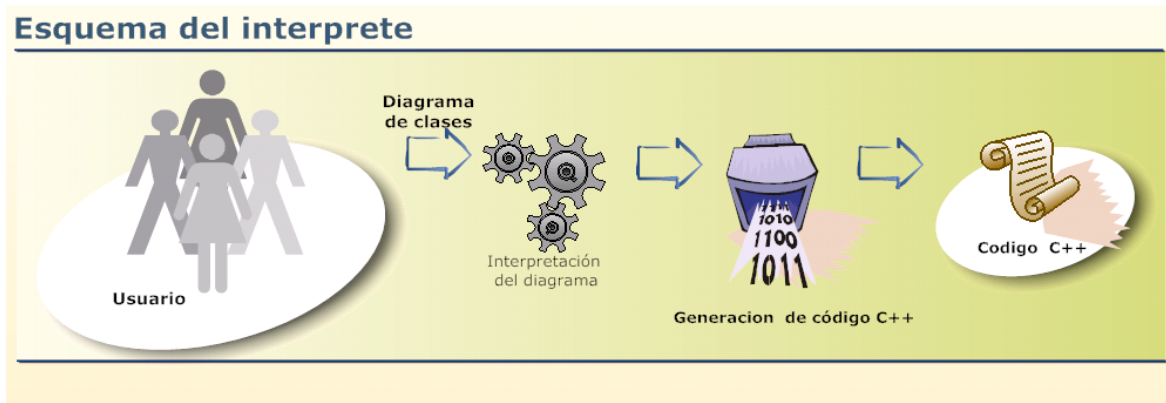


Ilustración 3 Diagrama de intérprete

1.4 JUSTIFICACIÓN

Con la creación de una aplicación para la enseñanza de diagramas UML, se busca facilitar la asimilación de los diagramas de clases y objetos, a través de la creación de un editor de diagramas de clases y objetos. Así, el estudiante es orientado por la ayuda interactiva, que consta de una recopilación de texto, audio y video acerca de la aplicación y de ejercicios prácticos.

La presencia de un producto en el mercado de software, que integre una ayuda audiovisual, modelado de diagramas de forma grafica y un intérprete que convierta diagramas de clase de UML a estructuras de clases de códigos C++, que complemente el estudio que conlleva al uso de los diagramas UML. “Como consecuencia en la práctica, las personas que desarrollan software se ven en la necesidad de rediseñar sus aplicaciones continuamente, debido a una mala planificación durante la fase de definición de requerimientos.”⁴

La aplicación puede ser utilizada por estudiantes y docentes como una herramienta didáctica, en cualquier tipo de clase expositiva, de tal forma sea complemento de la misma. Aquí es donde se aprovecha las ventajas que presenta la aplicación, tales como: listado de contenido, ejercicios resueltos y audio y videos que guían al usuario.

La aplicación de modelado de diagramas integra tres partes principales, las cuales se justifica a continuación:

- Editor: al contar con un modelador de graficas que representan los diagramas de clases y objetos, permite facilitar la planificación de la aplicación que se desea desarrollar.
- Ayuda interactiva: El método audiovisual es un instrumento, que se plantea

⁴ Basados en el artículo realizado por la compañía Microsoft. Ver grafico anexo 4.3

como una opción a múltiples interrogantes que no consigue la metodología normalmente empleada en la enseñanza. Al incorporar video y se logra obtener mayor atención debido a los estímulos visuales que se utilizan.

- Intérprete: Partiendo del hecho de implementar una herramienta que intérprete los diagramas de clases UML, posteriormente convertirlos en estructuras de clases de código C++, crea una nueva visión que permite al usuario abstraer los gráficos en estructura de códigos fuentes de C++.

La fuerte compatibilidad que existe entre las herramientas que provee Microsoft, permite seleccionar un software de desarrollo como Visual Basic .NET, ya que presenta instrumentos prácticos para la creación del editor y el intérprete.

Para el desarrollo de los videos de la ayuda interactiva, se ha seleccionado Camtasia, esta aplicación realiza la captura de video de pantalla y audio permitiendo guardar en formatos estándar de video. Además, para la elaboración del listado de contenido en formato html, se utilizará la herramienta Quanta+,

Se utilizará C++ como código se salida del intérprete, debido a que este lenguaje presenta una facilidad de poder adquirir diferentes compiladores, por ejemplo: Borland C++, Dev C++, Free C++. Además, C++ posee ya varios años de ser utilizado; existe gran variedad de documentación para su uso.

1.5 OBJETIVOS

1.5.1 GENERAL

Desarrollar una herramienta para el estudiante en la creación y edición de diagramas UML, donde se puedan definir objetos y clases. Esta herramienta debe de ser tanto práctica (con una interfaz que sea muy amigable para el usuario) como segura (presentando material audiovisual de apoyo) También tendrá la funcionalidad de poder interpretar los diagramas de clases y crear archivos que contengan código C++.

1.5.2 ESPECÍFICOS

- Desarrollar un editor de diagramas UML de clases y objetos.
- Construir un intérprete que convierta diagramas de clases de UML a estructura de clases de código C++.
- Incorporar una ayuda interactiva de la aplicación que combine texto, audio, video y ejercicios de forma incremental en su complejidad.

1.6 ALCANCES

El programa esta conformado por tres módulos: un editor, una ayuda interactiva de diagramas de clases y objetos de UML y un intérprete de diagramas de clases a estructuras de clases de código C++. Los cuales cuenta con los siguientes alcances:

El editor de diagramas de clases y objetos de UML cuenta con un espacio de trabajo diseñado para realizar diagramas en formato grafico, por medio de figuras que representa las clases y sus relaciones. Además, cuenta con una barra de herramientas para cada uno de los diagramas que puede realizar.

El programa tiene diferentes funciones con los diagramas: guardarlos en formato cpp, editarlos e imprimirlos.

La ayuda interactiva muestra un listado de contenidos en formato HTML, el cual contendrá conceptos básicos de los diagramas de clases y objetos de UML.

Se contara con ejercicios resueltos en la ayuda interactiva, sobre la aplicación de los diagramas de clases y objetos de UML, los cuales tiene diferentes grados de complejidad desde ejercicios sencillos hasta intermedios.

Además, la ayuda interactiva contara con ejercicios resueltos que llevan una continuidad en su desarrollo, para explicar las diferentes clases y sus relaciones, esto incrementando la complejidad del ejercicio en diferentes pasos.

El aspecto interactivo de la ayuda serán elementos multimedia de video y audio. El video explicara los ejercicios tantos resueltos como lo de continuidad, de forma en el cual explica paso a paso el desarrollo del ejercicio por medio del editor de diagramas de clases y objetos de UML. El audio tiene como finalidad guiar al estudiante de forma auditiva de los diferentes ejercicios.

El programa cuenta con una opción en la cual el estudiante puede obtener a partir de los diagramas de UML estructura de las clases en C++. Esta opción se encuentra en la barra de herramientas del menú, con el de nombre de intérprete.

1.7 LIMITACIONES

- No podrá abrir o modificar diagramas hechos en otros editores.
- La aplicación esta diseñada para usuarios con conocimientos básicos de POO, diagramas de clase y de objetos de UML.
- La aplicación se ejecutará en versiones Microsoft Windows XP con Framework versión 2.0.
- El intérprete a C++ a desarrollar será capaz de convertir el diagrama de clases de UML a estructura de clase de código C++, creando un archivo por cada clase diseñada con el editor.
- El intérprete solamente se comportará exportando diagramas UML de clases a estructuras de clases de código C++ y no podrá realizar el proceso inverso para obtener a partir de la estructura de clases de C++, el diagrama de clases correspondiente.
- La aplicación del editor de diagramas de UML y su intérprete no tiene la capacidad de compilar código C++.
- El editor está basado en los diagramas de objetos y de clases de la versión 1.1 de UML

1.8 MARCO TEÓRICO

1.8.1 MARCO HISTÓRICO

A finales de la década de los 90's se estandarizó los diagramas UML en su versión 1.1 en conjunto con empresas importantes tales como: Rational Software (Grady Booch, Jim Rumbaugh y Ivar Jacobson), Digital Equipment, Hewlett-Packard, i-Logix, IBM, Microsoft, Oracle Corporation, Texas Instrument, entre otras. A finales del año 1997 es presentado y adoptado por la OMG (Object Management Group), convirtiéndose en el primer estándar para el modelado de aplicaciones.

Desde la publicación oficial de UML, la cantidad de las herramientas para el modelado con este tipo de diagramas se popularizó, debido a que al realizar los diagramas manualmente se incurre en mucho tiempo en la edición de las figuras. En cambio, con la ayuda de un editor de diagramas UML esto fue solventado permitiendo al profesional orientarse al análisis de estos.

1.8.2 MARCO CONCEPTUAL

1.8.2.1 HISTORIA DE HERRAMIENTAS CASE

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diferentes aplicaciones informáticas que aumentan la productividad en el desarrollo de software haciendo que se reduzcan los costos de las mismas con respecto al tiempo y el dinero.

Estas herramientas son útiles en todos los aspectos del ciclo de vida de desarrollo del software en tareas como:

- El diseño del proyecto.
- Cálculo de costos.
- El diseño permite la implementación de una parte del código.
- Compilación automática.
- Documentación o detección de errores entre otras.

Ya en los años 70 un proyecto llamado ISDOS diseñó un lenguaje y un producto que analizaba la relación existente entre los requisitos de un problema y las necesidades que éstos generaban, el lenguaje en cuestión se denominaba PSL (Problem Statement Language) y la aplicación que ayudaba a buscar las necesidades de los diseñadores PSA (Problem Statement Analyzer).

Aunque esos son los inicios de las herramientas informáticas que ayudan a crear nuevos proyectos informáticos, la primera herramienta CASE fue Excelerator que salió a la luz en el año 1984 y trabajaba bajo una plataforma PC.

Las herramientas CASE fueron muy utilizadas a principios de los años 90. En la época en la que IBM había conseguido una alianza con la empresa de software AD/Cycle para trabajar con sus mainframes, estos dos gigantes trabajaban con herramientas CASE que abarcaban todo el ciclo de vida del software. Pero poco a poco los mainframes han ido siendo menos utilizados y actualmente el mercado de las Big CASE ha muerto completamente abriendo el mercado de diversas herramientas más específicas para cada fase del ciclo de vida del software.

La clasificación más común de las herramientas CASE basada en las fases del ciclo de desarrollo son: Upper CASE, Middle CASE y Lower CASE. Siendo el Upper CASE la herramienta que ayuda en las fases de planificación, análisis de requisitos y estrategia del desarrollo, entre los cuales están los diagramas UML.

Los diagramas UML son considerados una herramienta CASE con mucho potencial debido a la diversa cantidad de diagramas que posee, puede representar etapas del desarrollo del proyecto enfocando cada uno de los diagramas a las distintas fases del ciclo de vida.

1.8.2.2 COMPONENTES DE UML

UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Existen diferencias importantes entre un método y un lenguaje de

modelado. Un método es una manera de estructurar el pensamiento y las acciones de cada individuo. Además, el método le dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; mientras que el lenguaje de modelado no posee estas instrucciones.

Un modelo es expresado en un lenguaje de modelado. Un lenguaje de modelado consiste de vistas, diagramas, elementos de modelo un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos. (Ver figura 9.1)

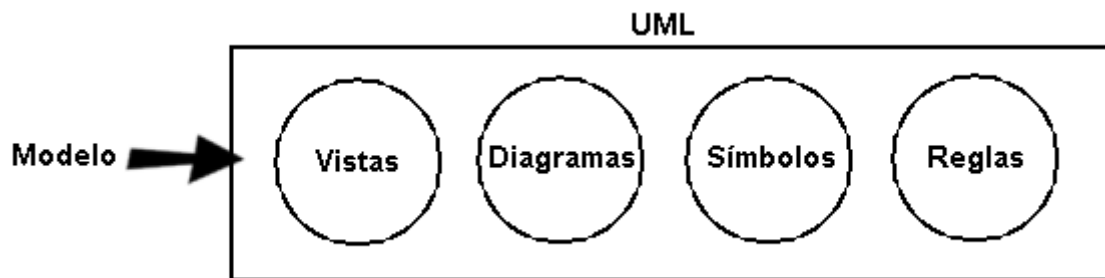


Ilustración 4 Elementos de UML.

Vistas: Las vistas muestran diferentes aspectos del sistema modelado. Una vista no es una gráfica, pero sí una abstracción que consiste en un número de diagramas y todos esos juntos muestran un esquema completa del sistema. Las vistas también unen el lenguaje de modelado a los métodos o procesos elegidos para el desarrollo.

Diagramas: Los diagramas son las gráficas que describen el contenido de una vista. UML tiene nueve tipos de diagramas que son utilizados en combinación para proveer todas las vistas de un sistema: diagramas de caso de uso, de clases, de objetos, de estados, de secuencia, de colaboración, de actividad, de componentes y de distribución.

Símbolos o Elementos de modelo: Las nociones utilizadas en los diagramas son los elementos de modelo que representan conceptos comunes orientados a objetos, tales como clases, objetos y mensajes, y las relaciones entre estos conceptos incluyendo la asociación, dependencia y generalización. Un elemento de modelo es utilizado en varios diagramas diferentes, pero siempre tiene el mismo significado y simbología.

Reglas o Mecanismos generales: Proveen comentarios extras, información o semántica acerca del elemento de modelo; además proveen mecanismos de extensión para adaptar o extender UML a un método o proceso específico, organización o usuario.

1.8.2.3 CARACTERÍSTICAS DE MODELADO CON UML

Aplicación	ArgoUML	FUJABA	mUML	Rhapsody Modeler	Objectteering UML
Compañía (versión evaluada)	Tigris (Oct-2000)	Universidad de Paderborn (Feb-1998)	MountField Computers (Mayo-2001)	I-Logix (Sept-2000)	Modeler softeam (Abril-2001)
Plataforma	JAVA	JAVA	JAVA	WINDOWS, LINUX	WINDOWS, UNIX
Licencia	Libre	Libre	Libre	Versión Demo (con restricciones)	Versión Base (3 meses)
Diagramas que se pueden realizar	De Clases, de Estado, de Actividad, de Casos de Uso, de Colaboración, Deployment	De Clases y de Actividades (Combinación de diagramas de Colaboración con los de Actividad y Estado)	Todos los diagramas UML	Todos los diagramas UML menos los de colaboración y diagramas de despliegue	Todos los diagramas UML
Generación de código	JAVA	JAVA (para Clases y Actividades)	JAVA, CORBA, IDL	C, C++, JAVA	JAVA, C++ (hay que poseer estos módulos)
Ingeniería inversa	NO	JAVA	JAVA (también desde bases de datos)	C, C++, JAVA	SI (módulo)
Exportación de diagramas	GIF, PostScript, EPS, PGML, SVG	PostScript	JPEG, Página Web	No podemos salvar los diagramas (restricción)	Gif, PostScript,, EPS, EMF
Otras características	Apoyo a OCL, control de errores (checklist) automático	Diagramas SDL, edición de constrains y simulación de Objetos	Ingeniería inversa desde Bases de Datos	Importación de modelos Rational Rose	Diversas notaciones diferentes scripts de generación

1.9 METODOLOGÍAS AUDIOVISUAL

El término audiovisual significa la integración e interrelación plena entre lo auditivo y lo visual para producir una nueva realidad o lenguaje. La percepción es simultánea. Se crean así nuevas realidades sensoriales mediante mecanismos como la armonía (a cada sonido le corresponde una imagen), complementariedad (lo que no aporta uno lo aporta el otro), refuerzo (se refuerzan los significados entre sí) y contraste (el significado nace del contraste entre ambos).

En el lenguaje audiovisual, como en los lenguajes verbales, se pueden considerar diversos aspectos o dimensiones: aspectos morfológicos y sintácticos.

a) Aspectos morfológicos: De la misma manera que cuando elaboramos mensajes con los lenguajes verbales utilizamos nombres, verbos, adjetivos y otros elementos morfológicos, los mensajes audiovisuales se construyen utilizando los siguientes elementos morfológicos:

- **Elementos visuales.** Sus elementos básicos son: puntos, líneas, formas, colores, planos y texturas. Con estos elementos las imágenes pueden representar cosas que existen y también cosas que nunca han existido.

- **Elementos sonoros.** Distinguimos: la música, los efectos de sonido, las palabras y el silencio. Funciones: expresiva, ambiental, gramatical.

- Palabra: elemento principal del lenguaje sonoro.
- Música: engendra climas emocionales, enfatiza el realismo de ciertos episodios, hace las veces de las transiciones en el cine.
- Efectos sonoros.
- Silencio.

b) Aspectos sintácticos: Para construir un mensaje verbal, no es suficiente mezclar una serie de nombres, verbos y adjetivos, hay que seguir unas normas sintácticas que permitirán elaborar frases significativas. De la misma manera, cuando creamos un mensaje audiovisual tenemos que seguir unas normas sintácticas que, además, podrán influir en el significado final de nuestro mensaje. Los principales aspectos

sintácticos a considerar son: el plano, el ángulo, la composición, profundidad de campo, distancia focal, ritmo, iluminación, movimiento de cámara y signos de puntuación.

MEDIOS AUDIOVISUALES

Dentro de los medios de comunicación encontramos, los medios audiovisuales que son los que se oyen y se ven. Es decir, son los medios que se basan en imágenes y sonidos para expresar la información. Por lo tanto un punto muy importante hoy en día para el desarrollo de los estudiantes, es una educación audiovisual que permita a los jóvenes una actitud crítica hacia la información que se le presenta, así como que les permita conocerlos para poder aprovecharlos adecuadamente.

Por otro lado, se define a los medios audiovisuales como aquellos elementos que contribuyen a esclarecer la palabra hablada o escrita, considerando a los medios audiovisuales recursos que los docentes utilizan desde hace algún tiempo, como los mapas, objetos, imágenes, etc. Por otra parte, el centro de la actividad de enseñanza tradicional sigue siendo el libro, como fuente de información ágil y más accesible. Los medios audiovisuales ofrecen un soporte completo para realizar acciones que preparan el surgimiento del pensamiento operativo. Pueden ser elementos que motiven al alumno hacia nuevas investigaciones y nuevos aprendizajes, permitiendo acelerar el proceso de enseñanza.

Los medios audiovisuales son un medio que permite muchas acciones positivas en cualquier actividad, ya que permite conocer experiencias, otras que no se pueden dar con los medios tradicionales. Sin embargo para ello es necesaria una formación en los medios para una mayor eficiencia sobre la elaboración, uso y evaluación de medios y recursos audiovisuales e informáticos, no sólo en sus aspectos técnicos, sino fundamentalmente en su integración pedagógica. Con ello los alumnos deben ir formando su propia actitud hacia estos medios, actitud de reflexión y de capacidad crítica.

La utilización de los medios de comunicación audiovisual en el proceso de la enseñanza-aprendizaje depende de:

- La metodología del educador.
- Los contenidos a transmitir.
- El tipo de área educativa con la que se enfrenta el estudiante.
- Las características específicas del grupo.
- La flexibilidad de utilización de cada uno de los medios.
- Las actitudes del educador hacia los medios de comunicación audiovisual.

Los medios de comunicación audiovisual utilizan distintas formas de expresión. Cada medio tiene sus propias características: el video, el audio hacen uso del lenguaje auditivo. Los carteles o la fotografía hacen uso del lenguaje visual. Los periódicos, los libros y las revistas hacen uso del lenguaje escrito. Los textos escolares y todos los medios impresos que utilizan imágenes hacen también un uso combinado de los lenguajes escrito y visual.

1.9.1 RECURSOS AUDIOVISUALES

A medida que se reconoce el valor de los medios audiovisuales, se hace evidente la necesidad de varios de estos recursos para alcanzar las metas de enseñanza. Actualmente, los equipos audiovisuales han extendido su campo de acción y su función dentro de los planes y programas educativos, por ello, es importante el uso de las técnicas y los recursos didácticos, como parte de los cambios y transformaciones educativas.

La necesidad de incorporar recursos audiovisuales que respondan a las interrogantes planteadas por las nuevas tecnologías, permite que se estimulen expectativas dentro del panorama comunicativo, iniciando nuevas posibilidades de encuentro entre la educación y la necesidad de aprender. El valor motivacional que los medios audiovisuales aportan al proceso de enseñanza, promueve a los alumnos hacia un aprendizaje significativo.

Los recursos audiovisuales permiten ofrecer ideas, propuestas, sugerencias y enriquecen la labor del docente y además facilitan experiencias de aprendizaje con mucha creatividad, es por ello, que para mantener la atención de los estudiantes y lograr que se capten los contenidos a facilitar, deben tomarse en consideración estos recursos.

Los recursos audiovisuales que se implementaran en la ayuda interactiva del editor de diagrama UML, están diseñados para facilitar el mejor entendimiento de la aplicación. Para ello se sugieren los siguientes tipos de audiovisuales:

Video didáctico: Elemento audiovisual contemplado para ser utilizado en un proceso directo de enseñanza aprendizaje de forma creativa, cuyos contenidos son dinámicos y se adecuen a las características pedagógicas y educativas de los alumnos.

Audio didáctico: Elemento de audio contemplado para ser utilizado como guía, utilizando las ventajas del medio de audio para brindar al estudiante una mejor comprensión de los ejercicios propuestos.

Lista de contenidos interactivo: Elemento visual contemplado para ser utilizado por los estudiantes para abarcar las diferentes dudas acerca del tema expuesto. Este elemento contempla una lista de contenidos, en el cual esta dividido en preguntas y respuestas de forma clara y concisa.

Todos estos elementos audiovisuales integran diferentes características, que tienen como fin facilitar al estudiante a comprender de manera programada los diferentes diagramas que se pueden implementar en el editor de diagramas de clases y objetos de UML.

1.9.2 LENGUAJES DE PROGRAMACIÓN

1.9.2.1 FRAMEWORK .NET (MARCO DE TRABAJO .NET)

El marco de trabajo .NET es un componente que viene incluido desde el sistema operativo Windows 2003 y provee una gran variedad de librerías predefinidas, utilizadas para realizar tareas tan simples como salida de texto en modo consola, hasta lectura y escritura de un archivo XML. Microsoft⁵ en las versiones anteriores, utiliza una capa de librerías llamada API Win32, la cual provee diferentes servicios como leer memoria, escritura de archivo, manejo de gráficos, entre otros. El marco de trabajo .NET mantiene la misma filosofía de la API Win32, ya que posee sus propias librerías para ejecutar estas y otras tareas.

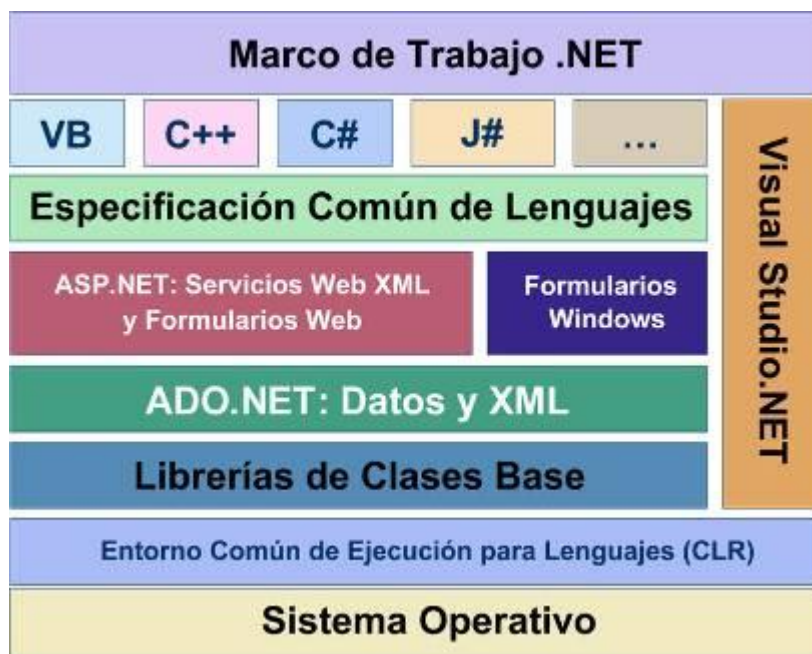


Ilustración 5 Diagrama detallado del Marco de Trabajo .NET. (Imagen tomada del sitio oficial de Microsoft ver bibliografía para mayor referencia)

La diferencia entre ambos es que API Win32 esta diseñada para ejecutar comandos en el Sistema Operativo Windows, en cambio el marco de trabajo .NET, esta

⁵ Microsoft corporación, empresa responsable de la creación de los sistemas operativos basados en tecnología Windows 9x y NT

diseñado para ejecutar comandos sobre cualquier Sistema Operativo compatible con el, en pocas palabras es multiplataforma.

Las principales características del marco de trabajo .NET son:

- **Compatibilidad:** Puede ejecutar aplicaciones desarrolladas con versiones anteriores de .NET.
- **Interoperabilidad:** Permite el acceso a funciones externas al marco de trabajo .Net, como los componentes COM.⁶
- **Motor común en tiempo de ejecución:** Los lenguajes de programación diseñados para el marco de trabajo .NET, compilan el código fuente en lenguaje intermedio llamado Lenguaje Intermedio Común (CIL por sus siglas en ingles), este lenguaje intermedio no es interpretado sino que es compilado de forma que el Compilador en tiempo real (JIT por sus siglas en ingles) lo convierta en código nativo⁷. A este conjunto, CIL y JIT se le llama Infraestructura Común de Lenguaje (CLI por sus siglas en ingles, ver siguiente figura.). Y la implementación realizada por Microsoft del CLI, es conocida como Entorno Común de Ejecución para Lenguajes (CLR por sus siglas en ingles).

⁶ Es una plataforma para la composición de aplicaciones introducida por Microsoft en 1993, y es utilizada para habilitar la comunicación entre procesos y la creación dinámica de objetos en cualquier lenguaje que soporte esta tecnología.

⁷ **código nativo:** es un termino utilizado para especificar la notación de código de maquina. Es decir, aquel que es ejecutado directamente por el microprocesador del CPU.

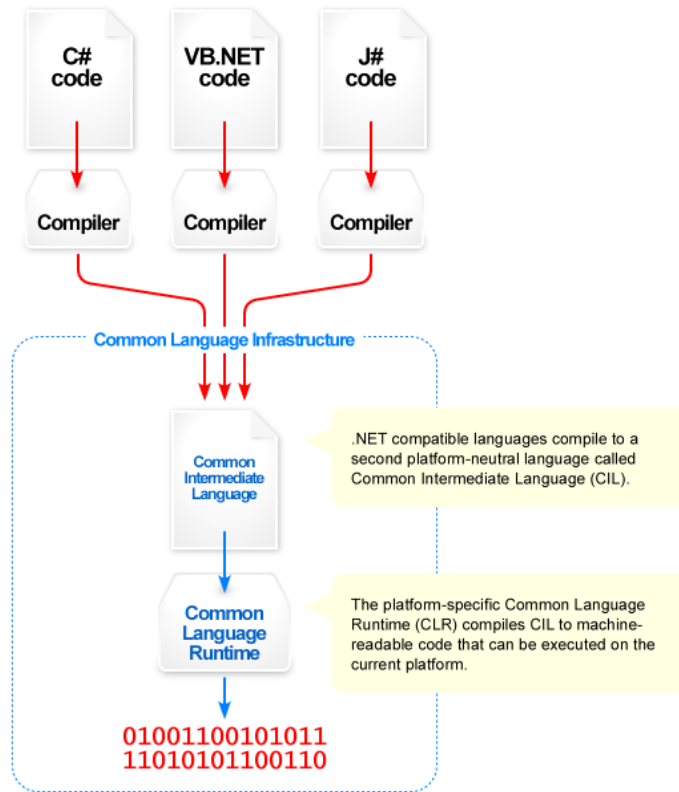


Ilustración 6 Vista preliminar de la Infraestructura común del Lenguaje. (Imagen tomada del sitio oficial de Microsoft ver bibliografía para mayor referencia)



Tabla 1 Diagrama de la estructura interna del Entorno de Común de Ejecución para Lenguajes (CLR por sus siglas en inglés). (Imagen tomada del sitio oficial de Microsoft ver bibliografía para mayor referencia)

- Independencia del lenguaje: En el marco de trabajo .Net se utiliza el Tipo Común del Sistema (CTS por sus siglas en ingles). Es ahí donde se definen todos los posibles tipos de datos y constructores soportados por el modulo de Entorno Común de Ejecución para Lenguajes, y como ellos puedan o no interactuar con otros.
- Librería Base de Clases (BCL por sus siglas en ingles): es una librería de tipos disponibles en todos los lenguajes que utilizan el marco de trabajo .NET, además provee clases que definen ciertas funciones comunes para el desarrollo de aplicaciones (Lectura y escritura de archivos, rendimiento grafico, interacción con base de datos, manipulación de estructuras XML entre otros).
- Despliegue simplificado: El marco de trabajo .Net incluye herramientas e indicaciones para poder realizar instaladores de las aplicaciones desarrolladas de forma tal, que no interfiera con versiones anteriores de la aplicación.
- Seguridad: .NET permite ejecutar aplicaciones con diferentes niveles de seguridad, sin necesidad de crear cajas de arena⁸ separadas.
- Desarrollo simplificado: El marco de trabajo .NET se encarga de la mayor parte de la estructura necesaria para generar software, lo que permite a los programadores centrarse en el código lógico esencial.

Los principales componentes del marco de trabajo .NET son: el conjunto de lenguajes de programación, la BCL y el CLR.

⁸ *Cajas de arena* es un término utilizado en la programación, para hacer referencia a un mecanismo de seguridad utilizado para ejecutar de forma segura, una aplicación. Usualmente se utiliza para ejecutar códigos en desarrollo, aplicaciones de terceros o sin credenciales. Este método proporciona a la aplicación un limitado acceso a recursos del sistema, evitando que tenga acceso a archivos personales del usuario o críticos del sistema y dispositivos de entrada y salida. Un claro ejemplo son los Applets.

1.9.2.2 JAVASCRIPT

JavaScript es un lenguaje interpretado⁹, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del Modelo de Objetos de Documento ¹⁰(DOM por sus siglas en ingles).

Al contrario que Java, JavaScript no es un lenguaje orientado a objetos, ya que no posee Herencia, en lugar de ello, utiliza la re-utilización de procesos conocida como herencia en lenguajes basados en clases, la cual es el resultado de la clonación de objetos ya existentes, que sirven de modelos para poder extender sus funcionalidades. Este tipo de programación se le llama: programación basada en prototipos.

JavaScript fue introducido por Brendan Eich en la empresa Netscape Communications, la misma que fabricó los primeros navegadores Web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0. Luego los demás navegadores lo incluyeron, tales como: Konqueror, las versiones 6 de Internet Explorer, Opera versión 7 y Mozilla.

Tradicionalmente JavaScript se utilizaba en páginas HTML para realizar tareas y operaciones simples en el marco de la aplicación cliente, tales como: validaciones y manejo del flujo de la página en base a eventos del usuario, todo esto sin acceso a funciones del servidor, permitiendo realizar tareas independientes de el y liberándolo de esa carga de trabajo que lo sobrecargaba innecesariamente a medida el numero de usuarios aumentaba el acceso a el, para realizar procesos simples.

⁹ **Lenguaje interpretado** es un lenguaje de programación diseñado para ser ejecutado por medio de un intérprete. También se les conoce como lenguajes de script.

¹⁰ **Modelado de Objetos de Documento**: es una forma de representar los elementos de un documento estructurado (tal como una página web HTML o un documento XML) como objetos que tienen sus propios métodos y propiedades.

A medida fueron evolucionando las páginas Web, se ampliaron las utilidades de JavaScript. Hasta el punto de simular entornos gráficos simples, espacios virtuales y ambientes similares a los creados con Flash, ejemplo de ello es el paquete JQuery, el cual brinda una extensión que permite manejar de forma mas fácil documentos HTML, manejo de eventos, realización de animaciones y agregar interacciones utilizando AJAX¹¹.

1.10 METODOLOGÍA Y TÉCNICAS DE INVESTIGACIÓN

1.10.1 TÉCNICAS DE RECOLECCIÓN DE DATOS

La investigación esta dividida en tres grandes áreas: la investigación para el desarrollo del editor de diagramas de UML, la ayuda interactiva y el interprete, esto por medio de:

- Encuesta: con las encuestas a estudiantes universitarios se buscará recopilar datos estadísticos acerca de las metodologías que mejor resultado ofrezcan al aprendizaje de UML.
- Bibliografía: investigación bibliográficas, análisis de editores existentes en el mercado actual de software
- Sitio Web relacionados con el tema: investigación de artículos y trabajos publicados en Internet.

1.10.2 TÉCNICAS DE DISEÑO Y DESARROLLO

La metodología utilizada en el diseño y desarrollo se basa en un análisis del tipo cascada y se compone de las siguientes etapas: (ver Ilustración 7, pagina 32)

¹¹ **AJAX**, (por sus siglas en ingles) acrónimo de JavaScript asíncrono y XML, es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla.

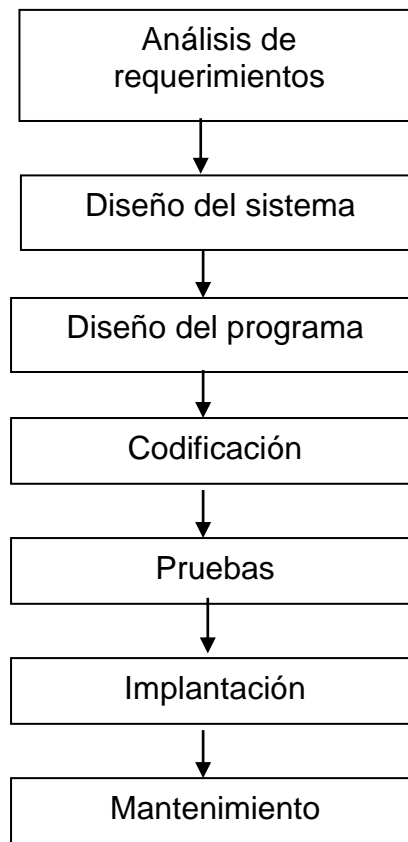


Ilustración **Figure** ¡Error! No hay texto con el estilo especificado en el documento.-1 o en cascada)

a. Análisis de requisitos

Se analizará las necesidades de los usuarios finales de la aplicación, para determinar qué objetivos deberemos cubrir.

Es importante señalar que en esta etapa se deben acordar todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración de la aplicación.

b. Diseño del Sistema

Se organizará el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Se obtendrán así la descripción de la estructura relacional global del sistema y la especificación de lo que deberá hacer cada una de sus partes, así como la manera en que se combinarán unas con otras.

c. Diseño del Programa

Es la fase en donde se realizarán los algoritmos necesarios para el cumplimiento de los requerimientos del usuario, así también los análisis necesarios para saber que herramientas usar en la etapa de codificación.

d. Codificación

Se desarrollará el código fuente, haciendo uso de prototipos así como pruebas y ensayos para corregir errores.

El lenguaje de programación a utilizar para este fin es Microsoft Visual Basic .net y en su versión 2005. Los requerimientos mínimos y óptimos del equipo a utilizar para el desarrollo de la aplicación se listan a continuación:

	Mínimos	Óptimos
Procesador	Celeron 1.7 GHz o superior	Pentium IV 2.8 GHz o superior
Memoria	256 MB	512 MB
Disco Duro	30 GB	40 GB
Unidad óptica	Lector de DVD	Lector de DVD
Sistema operativo	Windows XP PRO	Windows XP PRO
Software de aplicación	Visual Basic .net 2005	Visual Basic .net 2005

Tabla 2 Requerimiento de equipo para el desarrollo

e. Pruebas

Durante esta fase los elementos, ya programados, se unirán para componer la aplicación y se comprobará que funciona correctamente antes de pasar a la fase de implementación.

f. Implantación

En esta fase se realizarán las últimas pruebas, en el lugar donde se implantará la aplicación. Durante la implementación pueden surgir cambios, ya sea para corregir errores o bien para introducir mejoras.

Para la implementación de la aplicación se listan una serie de requerimientos mínimos y óptimos:

	Mínimos	Óptimos
Procesador	Pentium II 450 MHz o superior	Celeron 1.7 GHz o superior
Memoria	128 MB	256 MB
Disco Duro	10 GB (Disponible)	15 GB (Disponible)
Unidad óptica	CDROM	CDROM
Sistema operativo	Windows XP Home	Windows XP Home
Software de aplicación	Service Pack 2 para Windows	Service Pack 2 para Windows

Tabla 3 Requerimiento de equipo para la implementación

CAPITULO 2 - DEFINICIÓN DE LA SOLUCIÓN PROPUESTA

2.1 METODOLOGÍA DE INVESTIGACIÓN POR ENCUESTAS

La estadística es un campo interdisciplinario que provee los fundamentos y técnicas para recolectar, analizar y presentar información de forma efectiva y eficiente, a través de su aplicación en la investigación. Permite estudiar y comprender las diversas dimensiones y factores sociales, educacionales, económicos, tecnológicos, políticos entre otros.

2.1.1 LA ENCUESTA

Es una técnica destinada a obtener datos de varias personas cuyas opiniones aportan sustancialmente a la investigación. Para ello, se utiliza un listado de preguntas escritas a fin de que sean contestadas de la misma manera. Ese listado se denomina cuestionario y es impersonal porque no lleva el nombre ni otra identificación quien lo responde, ya que esos datos son de valor nulo para la investigación.

El método de aplicación de encuestas implica ciertos riesgos, tales como, la falta de sinceridad en las respuestas, manipulación de la información, temor a que la información pueda utilizarse en contra del encuestado, falta de comprensión de las preguntas, entre otros.

Existen varios tipos de preguntas que pueden plantearse en la investigación, se debe de seleccionar las preguntas más certeras, de acuerdo a la naturaleza de la investigación y, sobre todo, considerando el nivel de educación de las personas que van a responder el cuestionario. Según su clasificación, estas pueden ser de acuerdo a su forma y fondo:

Las encuestas se pueden clasificar de diversas formas.

Según su forma:

1. Preguntas abiertas

Las preguntas abiertas requieren que el encuestado escriba con sus propias palabras la respuesta, esta puede ser desde una palabra hasta una composición con varios párrafos de una página o más.

Las preguntas abiertas no son difíciles de escribir para el autor del cuestionario, pero no así el evaluarlas. La calificación se puede basar en palabras clave de la respuesta, tales como la presencia de términos importantes. En este caso, resulta fácil para el encuestador mal interpretar lo escrito, debido a que la respuesta puede estar formada por muchas ideas generales y cosas sin sentido que incluyan los términos que buscan. El proceso adecuado de calificación es leer las respuestas cuidadosamente y buscar la claridad y la lógica. Si el sujeto calificador debe revisar un gran número de encuestas, esto se vuelve cansado, especialmente porque generalmente conocen el material a un nivel superior del que se espera que lo conozcan los encuestados.

2. Preguntas cerradas

Son aquellas donde el encuestador establece todas las posibles respuestas para cada pregunta. Pueden ser de selección simple o selección múltiple, a saber:

- Preguntas de selección simple (dicotómicas): Sólo es posible una respuesta (sí o no, hombre o mujer).
- Preguntas de selección múltiple: Se presentan varias alternativas de posible respuesta. Dentro de estas se encuentran los tipos de respuestas:
 - En abanicos: Son aquellas que presenta una o varias respuestas a la lista que acompaña a la pregunta
 - Preguntas de estimación: son aquellas que incluyen una progresión de intensidad sobre un mismo término.
 - Escala de Likert: Se le conoce como escala sumada debido a que la puntuación de cada unidad de análisis se obtiene mediante la sumatoria de las respuestas obtenidas en cada ítem.

Clasificación de acuerdo con el fondo:

1. Preguntas de hecho: Son acerca de datos concretos sobre el encuestado.
2. Preguntas de acción: Son aquellas que tienen que ver con actividades y decisiones particulares de cada encuestado.
3. Preguntas de intención: Son aquellas que buscan averiguar las intenciones o propósitos del encuestado sobre alguna circunstancia.
4. Preguntas de opinión: Son sobre lo que el encuestado piensa u opina de algún hecho o tema determinado.
5. Preguntas índices o preguntas test: son tipos de preguntas indirectas sobre temas que el encuestado puede considerar indiscretas si se le preguntan en persona.

2.1.2 PASOS A SEGUIR EN LA INVESTIGACIÓN MEDIANTE LA ENCUESTA

Una encuesta es una herramienta de investigación, la cual, requiere seguir los pasos que se exigen como tal. El procedimiento específico de este tipo de investigación supone las siguientes etapas:

1. Formulación del problema y planteamiento de los objetivos que se pretenden con el estudio. Esta fase es común a cualquier investigación, ya que no sería posible recoger datos si no se ha procedido previamente a especificar el problema que se desea estudiar. En esta fase se determinará la población objetivo.
2. Selección y tamaño de la muestra. Constituye una de las fases más importantes del estudio, ya que aquí se especifican los procedimientos que se van a seguir para calcular el tamaño de la muestra en función de nuestros objetivos planteados anteriormente.
3. Construcción y administración del cuestionario piloto. Todo trabajo serio de encuesta supone la realización previa de un estudio piloto, que cubre diversos objetivos: depurar el cuestionario que se va a utilizar en el estudio original, preparar

a los encuestadores, familiarizándolos con el trabajo a realizar y calcular algunos estimadores precisos para la determinación del tamaño muestral.

4. Entrenamiento de los encuestadores, asegurando de esta forma que los datos recabados por ellos sean fiables. Este entrenamiento supone, por una parte, la familiarización con el cuestionario que se va a utilizar, el establecimiento de una forma de entrevistar homogénea, que garantice la igualdad de las respuestas, así como el aprender a manejar situaciones más o menos difíciles.

5. Organización del trabajo de campo. En esta fase se incluye la construcción del cuestionario definitivo y su administración, por el procedimiento que se considere más adecuado (encuesta telefónica, postal o por correo).

6. Análisis y presentación de resultados. Presupone un paso previo, el de codificación e informatización de los datos, lo cual, eliminará errores en dos fases de la investigación: cuando se obtiene la información y cuando esta se introduce a la computadora. El análisis de los datos, persigue dar respuesta a las cuestiones planteadas, que serán de dos tipos: una primera aproximación que nos informa de las opiniones de los encuestados en cada pregunta realizada y, en segundo lugar, los resultados que proceden de cruzar las respuestas de un grupo de preguntas.

2.1.3 DISEÑO DE LA MUESTRA

El muestreo aleatorio es la extracción de una muestra de una población finita, en el que el proceso de extracción es tal que garantiza a cada uno de los elementos de la población la misma oportunidad de ser incluidos en dicha muestra. Esta condición garantiza la representatividad de la muestra porque si en la población un determinado porcentaje de individuos presenta la característica A, la extracción aleatoria garantiza matemáticamente que por término medio se obtendrá el mismo porcentaje de datos muestrales con esa característica¹².

¹² Fuente: Cómo hacer una tesis de graduación con técnicas estadísticas. Biblioteca Central UDB. Clasificación 011.7 B715 2000.

El muestreo aleatorio puede ser de dos tipos:

- Sin reposición de los elementos: los elementos extraídos se descartan para la siguiente extracción.
- Con reposición de los elementos (Muestreo Aleatorio Simple o m.a.s.): las observaciones se realizan sustituyendo los individuos, de forma que la población es idéntica en todas las extracciones y, por tanto, cada observación es independiente de la anterior. En poblaciones muy grandes, la probabilidad de repetir una extracción es tan pequeña que el muestreo puede considerarse con reposición aunque, realmente, no lo sea.

Se selecciono el muestreo aleatorio simple fue ya que es un procedimiento de selección de una muestra por el cual todos y cada uno de los elementos de la población finita N tiene igual probabilidad de ser incluidos en la muestra; entonces, si toda unidad disponible para observación o medición tiene la misma probabilidad de ser escogida, se selecciona una muestra de n observaciones de una población finita de N mediciones.

El tamaño de la muestra n para estimar la proporción poblacional utilizando muestreo aleatorio simple se determinó de la siguiente manera:

$$n = \frac{Z^2 P Q N}{(N-1)E^2 + Z^2 P Q}$$

Donde:

n = Número de encuestas a realizar.

Z = Coeficiente de confianza de que la estimación efectuada se ajuste a la realidad.

Suele exigirse 95% o 95.5% (Z = 2). El 100% obligaría estudiar todos los casos.

P = Proporción poblacional de la ocurrencia de la característica poblacional que se quiere estudiar.

Q = 1- P

N = La población esta definida en la Pág. 57 y el total de la población es de 80 personas según el sondeo realizado, ver tabla 1.

E = Error muestral, probabilidad de aceptar una hipótesis que sea falsa como si fuera verdadera o rechazar una verdadera como si fuera falso. Para 0% de error, muestra debe coincidir con la población. Aceptable entre 4 y 10% de error.

2.1.4 TIPOS DE GRÁFICOS

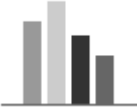



Descripción	Ejemplo
<p>Barras verticales:</p> <p>Muestran las series como conjuntos de barras horizontales agrupadas por categoría. Los valores se representan por la longitud de las barras con relación al eje X. Las etiquetas de las categorías se presentan en el eje Y. Los gráficos de barras suelen utilizarse para comparar valores de categorías.</p>	
<p>Líneas:</p> <p>Representan los valores en dos ejes cartesianos ortogonales entre sí. Las gráficas lineales se recomiendan para representar series en el tiempo y es donde se muestran valores máximos y mínimos; también se utiliza para varias muestras en un diagrama</p>	
<p>Pastel</p> <p>Se emplea un círculo el cual se divide en sectores cuyas medidas angulares son proporcionales a las magnitudes de los valores que representan.</p>	
<p>Burbujas</p> <p>Muestran las series como un conjunto de símbolos. Los valores se representan por la posición del punto en el espacio del gráfico y el tamaño del símbolo.</p>	

Tabla 4 Descripción de gráficos mas comunes para la representación de resultados estadísticos

2.2 FORMULACIÓN DE LA ENCUESTA

Objetivo general de la encuesta: Identificar el nivel de conocimiento de los encuestados acerca de los diagramas de clases y objetos de UML, además descubrir las características de las herramientas de diseño más importantes para los encuestados. Así como también, analizar los métodos de aprendizaje que los encuestados consideren eficientes.

Para evaluar el nivel de conocimiento de los diagramas UML, en los conceptos básicos como prácticos, se decidió aplicar la técnica del muestreo aleatorio en la población objetivo, considerando lo siguiente:

Las personas encuestadas pertenecen a las universidades del área de San Salvador, que cuentan con lo siguiente:

Característica principal: estudiantes universitarios de las carreras de ingeniería y licenciatura en computación.

Características complementarias:

- Población universitaria mayor a 4,000 estudiantes¹³.
- Contar en su pensum¹⁴ de la carrera con materias en las que se impartan los diagramas de clases y objetos de UML.
- Imparten conocimientos del tema de procesamiento de imágenes.

Las universidades que se estudiaron debían poseer la característica principal como requisito, al menos 2 de las características complementarias antes mencionadas y otras particularidades que se detallan a continuación:

- UES: por ser la universidad con mayor experiencia.
- UDB: objeto de estudio.
- UTEC: segunda universidad con mayor población de estudiantes.
- UCA y UFG: facilidad de acceso a los encuestados.

¹³ Fuente: <http://www.mined.gob.sv/>, visitada en marzo 2008: contemplado en el Anexo X Población Estudiantil a Nivel Superior Universitaria, ver anexo

¹⁴ Ver anexos pensum de universidades encuestadas

2.2.1 CALCULO DE LA MUESTRA

El cálculo se realizó de la siguiente manera:

$$n = \frac{Z^2 PQN}{(N-1)E^2 + Z^2 PQ}$$

Sustituyendo,

$$n = \frac{2^2 (0.9)(0.1)80}{(79)0.10^2 + 2^2(0.9)(0.1)}$$

n = **25 Encuestas**

La distribución de dichas encuestas se contempla en la tabla 1.

Universidades	Cantidad de profesionales	% Cantidad de estudiantes	Cantidad de Encuestas
UES	25	31,25	7,81 = 8
UTEC	18	22,50	5,63 = 5
UCA	15	18,75	4,69 = 5
UFG	12	15,00	3,75 = 4
UDB	10	12,50	3,13 = 3
Total	80	100	25

Distribución de población y muestra para el análisis de los resultados de la encuesta.

Así resulto el tamaño de la muestra de 25 encuestados, se realizó una prueba piloto para validar los resultados a un 20% de la muestra en estudiantes de ingeniería y profesionales de las universidades UCA y UDB. El número de encuestas que se realizaron en la prueba piloto fue de 5.

2.2.2 DATOS GENERALES DE LOS ENCUESTADOS

Un aspecto relevante en nuestra encuesta es la distribución de los encuestados en las diferentes universidades del área metropolitana. Esta se detalla a continuación en el grafico #1.

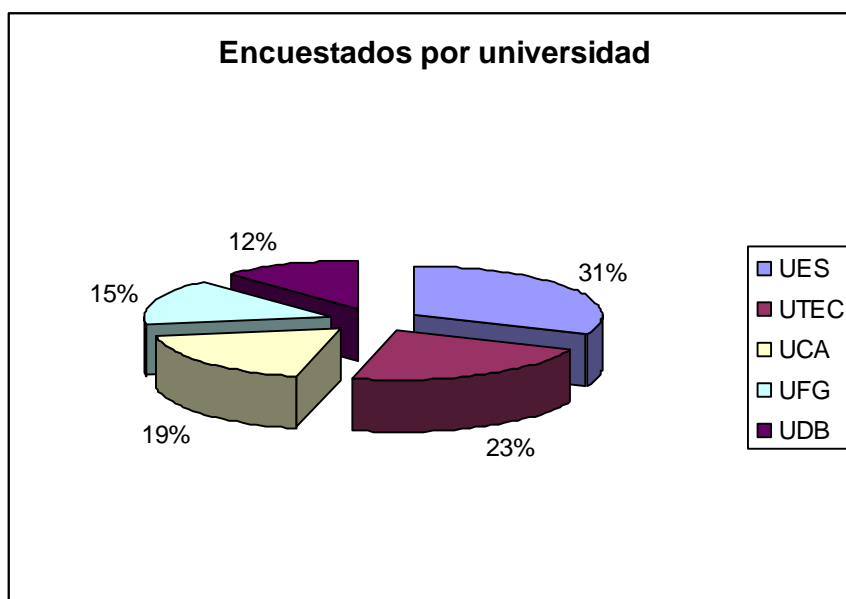


Gráfico 1 Distribución de estudiantes encuestados por universidades

En la ejecución del ejercicio de la encuesta final, las personas responsables de la realización de ella, fueron las mismas que se encargaron de diseñarla, lo cual resulto favorable, ya que no se vio en la necesidad de capacitar a terceros para encargarse del desarrollo de la encuesta.

Los medios que se utilizaron para la realización de la encuesta fueron: medios electrónicos (correo electrónico) y de manera escrita.

2.3 ANÁLISIS DE RESULTADOS

Después de seleccionar las 25 personas que forman parte de nuestra muestra, se realizó con ellos la encuesta¹⁵. Analizando los resultados obtenidos, de la siguiente forma. En la primera parte, se realizó un análisis conjunto de las primeras 5 preguntas, con lo cual se identifica el porcentaje de personas que realmente conoce lo esencial de UML.

En la segunda parte, se identifican los tipos y características de las aplicaciones de diagramas de UML, que para el encuestado resulten más importantes; en la tercera parte, se identificaron los tipos de metodologías de aprendizaje que nuestros encuestados consideraron más eficientes.

A continuación se detallan los objetivos de las preguntas con sus resultados y sus análisis obtenidos en las encuestas realizadas:

Porcentaje de personas que saben UML

Objetivo: Realizar un sondeo directo a conciencia si sabe o no acerca de UML.

Pregunta: ¿Conoce usted como se utilizan los diagramas de clases y objetos de UML?

Resultados:

Respuesta	Porcentaje (%)
Si	92%
No	8%

¹⁵ Ver anexo 4.2

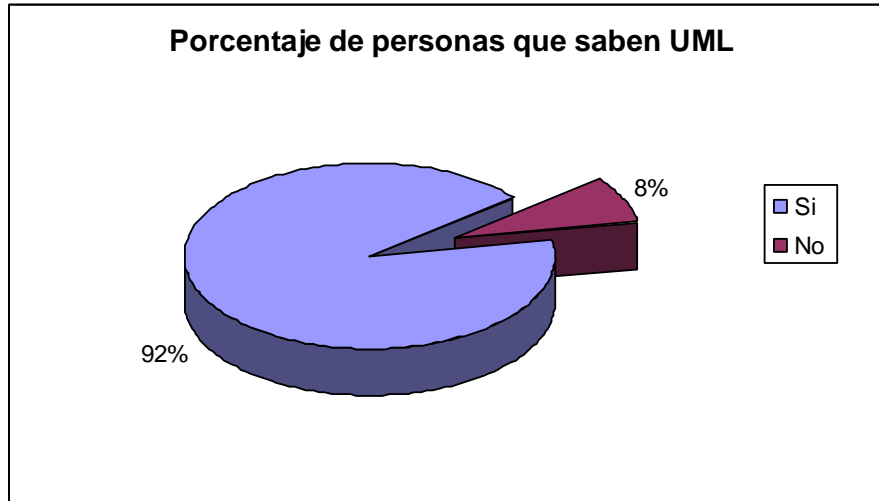


Gráfico 2 Resultados a conciencia acerca si conoce de UML

Análisis:

Los resultados obtenidos con esta pregunta fueron que la mayoría de los encuestados asegura tener un conocimiento, por lo menos básico sobre los diagramas de clases y objetos de UML.

Clasificación de resultados de los niveles de conocimiento

Objetivo: Realizar una medición del conocimiento esenciales acerca de los diagramas de clases y objetos de UML.

Preguntas

¿Qué es un diagrama de clases?

¿Cómo se representa una clase en los diagramas de clases de UML?

¿Qué tipos de relaciones se utilizan en los diagramas de clases y objetos?

¿Cuál es la definición de herencia y generalización?

Resultados:

Para evaluar este objetivo, se considero las preguntas del #2 - #5 de la encuesta.(ver anexo #)

Respuesta	Porcentaje (%)
Excelente	39%
Muy bueno	22%
Deficiente	30%
No sabe	9%

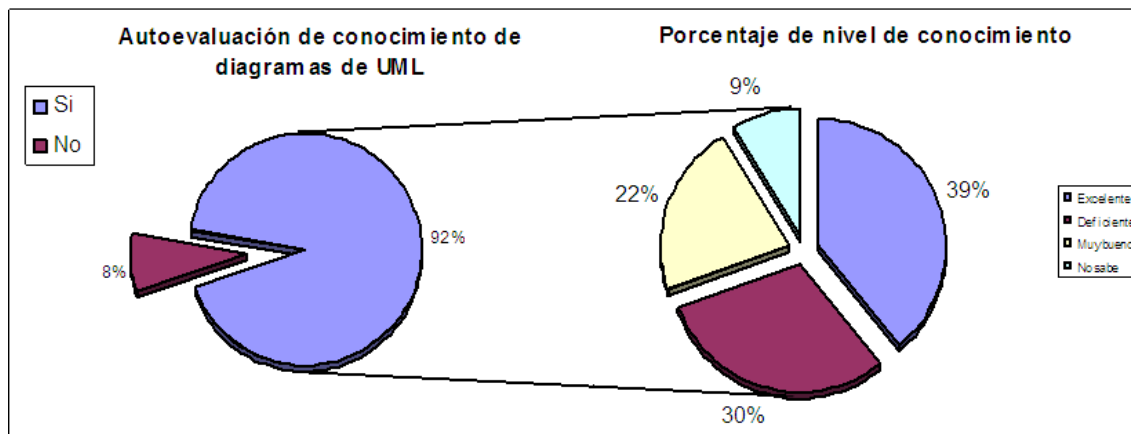


Gráfico 3 Niveles de conocimiento

Análisis:

Para realizar el análisis de estas 4 preguntas, se basó en el 92% de los encuestados que aseguraron saber de UML. Tomando como aceptable las categorías de excelente y muy bueno, se obtiene un total del 61%. Y las categorías restantes conforman un nivel deficiente totalizando un 39%. Podemos observar en el gráfico 5 los resultados reales de la evaluación.

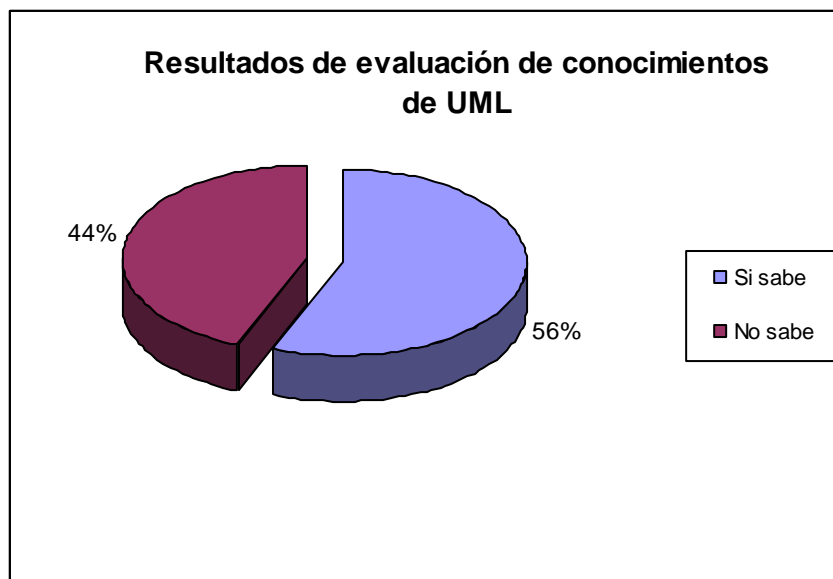


Gráfico 4 Porcentaje verdadero de conocimiento

Después de este análisis, se obtiene un porcentaje real de $(0.92 \times 0.61) \times 100\% = 56\%$ que en efecto si saben de UML, y por el contrario existe un total de $(0.92 \times 0.39) \times 100\% + 8\% = 44\%$ que no saben lo básico de los diagramas de UML.

Porcentaje de herramientas utilizadas para el diseño de diagramas de UML

Objetivo: Identificar las herramientas de diseño de diagramas UML que utilizan los estudiantes, tanto comerciales como gratuitos.

Pregunta: ¿Cuáles de las siguientes herramientas de diseño (comercial o gratuito) ha utilizado para los diagramas de UML?

Resultados:

Respuesta	Cantidad	Porcentaje (%)
Visio	15	26%
Argo UML	14	25%
Power Designer	9	16%
Violet	8	14%
Otros	7	12%
Umbrello	3	5%
Ninguno	1	2%
Rational Rose	0	0%

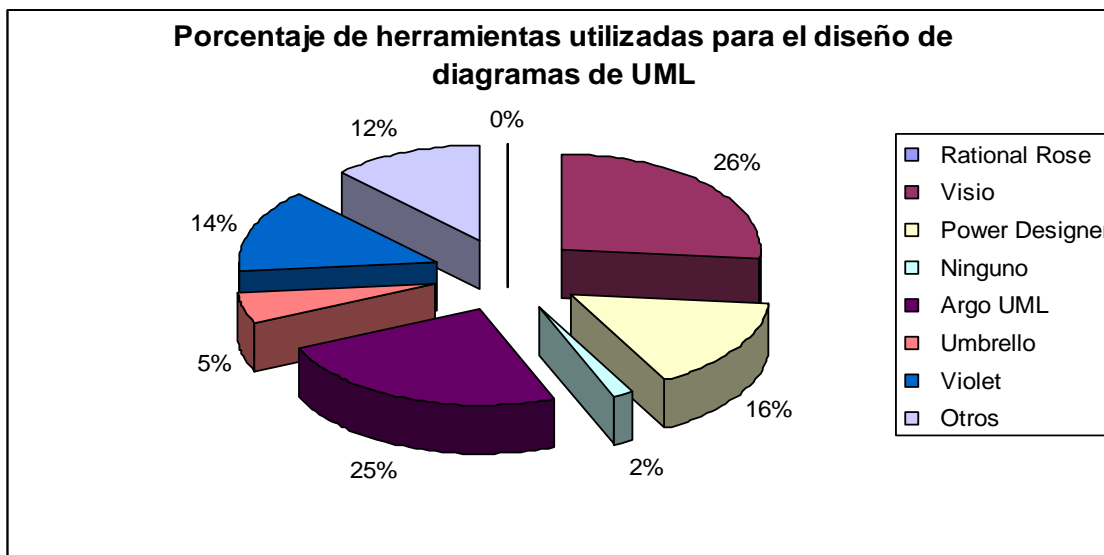


Gráfico 5 Frecuencia de aplicaciones más utilizadas para desarrollar diagramas de UML

Análisis:

El resultado nos indica que los estudiantes han utilizando diferentes aplicaciones tanto comerciales y gratuitas, siendo un 48% de los encuestados que han utilizado aplicaciones comerciales y un 50% que han utilizado aplicaciones gratuitas. Es importante hacer notar que solamente un 2% manifestó no haber utilizado ninguna aplicación, lo cual nos lleva a la conclusión que una gran parte de los encuestados utiliza las aplicaciones que se encuentran en el mercado.

Comparación de características de aplicaciones comerciales y gratuitas

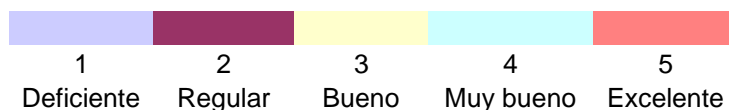
Objetivo: busca evaluar las diferentes características del software comercial y gratuito, para tener una referencia acerca de que características deberían ser consideradas en nuestra aplicación.

Pregunta: ¿Que calificación le colocaría a las siguientes características de las herramientas de aplicación de tipo comercial?

Características \ Calificación	Promedio
Flexibilidad	4
Seguridad	4
Ambiente agradable	4
Fácil de utilizar	4
Fácil instalación	4
Soporte	4
Costo	3

Pregunta: ¿Que calificación le colocaría a las siguientes características de las herramientas de aplicación de tipo gratuito?

Características \ Calificación	Promedio
Flexibilidad	3
Seguridad	3
Ambiente agradable	3
Fácil de utilizar	4
Fácil instalación	4
Soporte	3
Costo	4

Escala de colores de clasificaciones

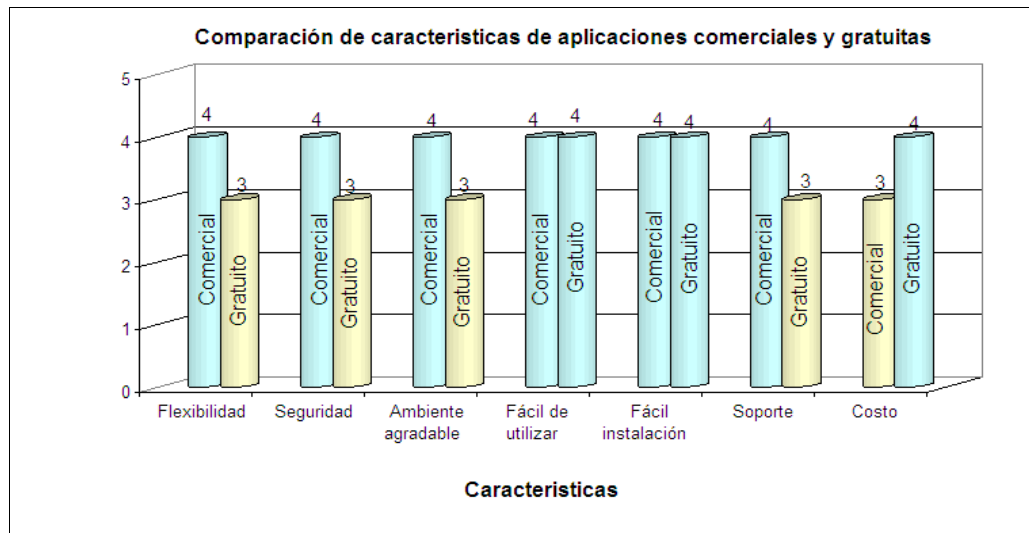


Gráfico 6 Resultados de las calificaciones de características de las aplicaciones gratuitas y comerciales

Análisis:

Los resultados obtenidos en la pregunta 7 nos reflejan que el uso de herramientas de aplicación de uso comerciales, tienen características que a los encuestados consideran relevantes, tales como la seguridad, ambiente agradable, fácil instalación y soporte. En cambio, en la pregunta 8 acerca de aplicaciones gratuitas, los resultados demuestran que todas las características fueron calificadas con menos ponderación que las de uso comercial, sin embargo, los encuestados calificaron con mayor ponderación el costo de las aplicaciones gratuitas. Es de hacer notar que los encuestados calificaron de forma similar ambos tipos de aplicaciones.

Promedio de calificación del uso de metodologías de aprendizaje

Objetivo: Analizar la prioridad de los métodos de aprendizaje que se menciona, para identificar cuales serán implementadas en nuestro programa.

Pregunta: ¿Que nivel de prioridad tiene para usted las siguientes metodologías de aprendizaje?

Resultados:

Metodología \ Prioridad	Prioridad
Clase expositiva	5
Guías	3

Videos demostrativos	4
Preguntas frecuentes	2
Ejercicios resueltos	6
Otros	1

Análisis:

Los resultados según la experiencia de los encuestados, acerca de los elementos didácticos más importantes para su aprendizaje, se agrupan en ejercicios resueltos y videos demostrativos. Mientras que las metodologías con menos prioridad fueron: las preguntas frecuentes y otras, en las que se mencionan los foros, discusiones, tutoriales y lecturas individuales.

Evaluación de tiempo estimado para un video demostrativo, número de diapositivas para una clase expositiva y preferencias del tipo de formato para la lectura de un documento

Objetivos:

- Analizar el tiempo en el cual un video demostrativo sea aceptable por el estudiante para un mayor entendimiento de un tema.
- Analizar la cantidad de páginas que el estudiante considera necesarias para una clase expositiva utilizando diapositivas.
- Identificar el formato en el cual los estudiantes prefieren realizar la lectura de un documento, el cual será implementado en nuestra aplicación.

Pregunta:

¿Cuánto tiempo considera que debe durar un video demostrativo?

Resultados:

Respuesta	Porcentaje (%)
3 a 5 minutos	60%
Mas de 5 minutos	20%
1 a 3 minutos	20%

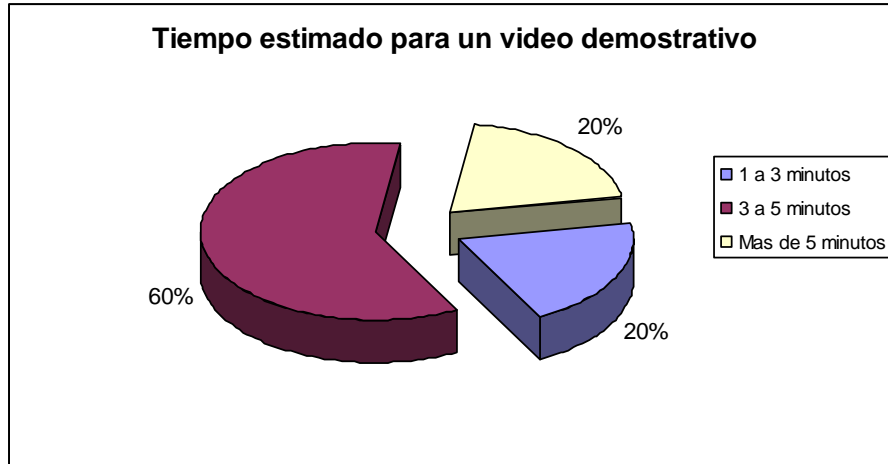


Gráfico 7 Resultados de preferencia de duración de un video demostrativo

Pregunta: ¿Cuántas páginas deberían tener una clase expositiva con diapositivas?

Resultados:

Respuesta	Cantidad
10 a 20 páginas	76%
20 a 25 páginas	24%
Mas de 25 páginas	0%



Gráfico 8 Resultado de numero de diapositivas idóneas para una clase expositiva

Pregunta: Cuando usted lee un documento, ¿Qué tipo de formato prefiere que tenga este para facilitar su lectura?

Resultados:

Respuesta	Cantidad
Formato impreso	60%
Formato digital	40%



Gráfico 9 Resultado de tipo de formato de documentos

Análisis:

En base a los resultados obtenidos de las siguientes preguntas a) evaluación de tiempo estimado para un video demostrativo, b) evaluación de número de diapositivas para una clase expositiva y c) preferencias del tipo de formato para la lectura de un documento, se analizó que la preferencia de los encuestados para aprender cierto conocimiento sobre un tema, es invertir el menor tiempo leyendo. Además de existir una gran preferencia por documentos impresos que digitales.

Resumen de análisis

Los resultados de la encuesta proyectaron que un 92% de los encuestados aseguraron tener un cierto grado de conocimiento acerca de diagramas de clase y objetos de UML, sin embargo, se observó que un 51% de los encuestados posee conocimientos esenciales, por el contrario un 49% carece de estos.

En base a las 3 características más importantes para los encuestados, sobre las aplicaciones comerciales y de uso gratuito para realizar diagramas de UML, se encontró que estas fueron las siguientes: ambiente agradable, seguridad y costos. Considerando los resultados, se desarrollara una aplicación que cumpla con las expectativas obtenidas en el análisis de resultados de la encuesta, para el desarrollo del editor, intérprete y ayuda interactiva.

Como último punto, los resultados según la experiencia de los encuestados, acerca de las diferentes metodologías con mayor importancia para su aprendizaje, son ejercicios resueltos y videos demostrativos. Mientras que las metodologías con menor ponderación son: las preguntas frecuentes y otras, en las que se mencionan los foros, discusiones, tutoriales y lecturas individuales.

CAPITULO 3 - ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PLANTEADA

A continuación se presentan los diagramas UML de caso de uso, de actividades y de clases de nuestra aplicación, junto con concepto teórico de cada uno de los tipos de diagramas.

3.1 DIAGRAMAS DE CASOS DE USO

3.1.1 DEFINICIÓN DE LOS CASOS DE USO

Un caso de uso es un modelo de la interacción entre los usuarios externos de un sistema de información y el sistema de información mismo. Dicho de una manera precisa, un actor es un usuario que tiene una función específica. Un diagrama de casos de uso es un conjunto de casos de uso.

Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos. Los casos de uso tienen las siguientes características:

- Los Casos de Uso (Ivar Jacobson¹⁶) describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario.
- Permiten definir los límites y las relaciones entre el sistema y el entorno.
- Son descripciones de la funcionalidad del sistema independientes de la implementación.
- Cubren la carencia existente en métodos previos (OMT, Booch¹⁷) en cuanto a la determinación de requisitos.
- Separan el conjunto de necesidades atendiendo a la categoría de usuarios que participan en el mismo.
- Están basados en el lenguaje natural, es decir, es accesible por los usuarios.

¹⁶ En 1995 se incorporo con su compañía Objectory a Rational Software Corporation aportando el metodo OOSE (Object Oriented Software Engineering) mediante la metodología de casos de uso.

¹⁷ OMT (Object Modeling Technique) Tecnica de modelado orientada a objetos incorporada a UML por Grady Booch y James Rumbaugh de Rational Rose Software Corporation.

La implementación de este tipo de diagrama es para establecer las transacciones principales que contiene el sistema, además, de definir las funcionalidades que debe cumplir para satisfacer todos los requerimientos. El objetivo principal de incorporar los casos de uso en el sistema es porque se puede visualizar detalladamente las funciones más importantes utilizadas en la aplicación.

Un diagrama de casos de uso consta de los siguientes elementos:



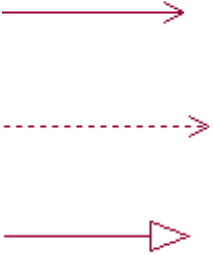


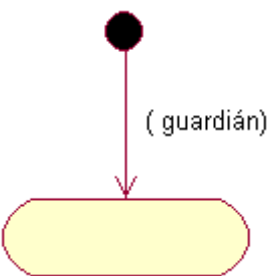

Elemento	Descripción	Tipo
Actor	Representa participantes en las cajas de uso. Los actores pueden ser personas o cosas.	
Caja de uso	Se utiliza para representar capacidades. A la caja de uso se le da un nombre y una descripción mediante texto. Este último debe describir como inicia y finaliza la caja de usos, e incluye una descripción de la capacidad descrita por el nombre de la misma.	
Conector	Dado que los diagramas de casos de uso tienen múltiples actores y en virtud de que las cajas de uso pueden estar asociadas con los actores y con otras cajas de uso, se utilizan los conectores para indicar la manera en que ambos están asociados. Tipos de conectores: Asociación: es un conector de línea simple, se usa para mostrar cuales actores están relacionados con las cajas de uso. Dependencia: es una línea punteada con una flecha direccional. La flecha apunta hacia la caja de usos de la que depende. Generalización: es una línea dirigida con un triangulo hueco. La palabra generalización en UML significa herencia, la flecha apunta hacia la cosa sobre la cual nos estamos expandiendo.	
Estereotipo Incluir	Una dependencia rotulada con el estereotipo incluir significa que, finalmente, la caja de usos dependiente es para volver a usar la caja de la que depende.	
Estereotipo Extender	Se usa para agregar más detalle a una dependencia, lo cual significa que estamos agregando más capacidades.	

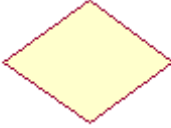
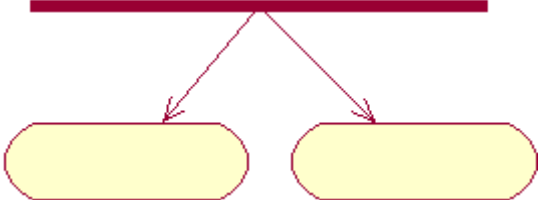
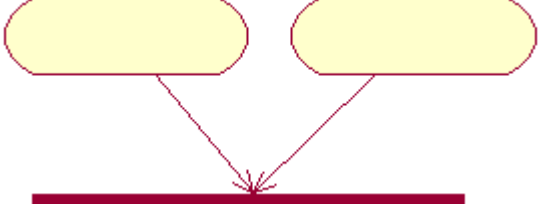
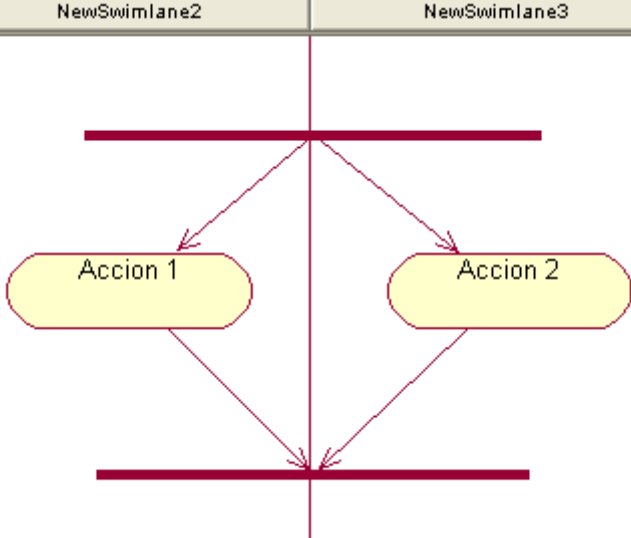

Tabla 5 Elementos de caso de uso

3.1.2 DIAGRAMAS DE ACTIVIDADES

Los diagramas de actividades ayudan a los usuarios a describir en forma visual la secuencia de acciones que le conduzcan a través de la estructura de la tarea. Un diagrama de actividades de UML tiene cierta similitud al diagrama de flujo dado que comparten algunos pero los diagramas de actividades, a diferencia de los de flujo, pueden modelar comportamiento paralelo. Además, son útiles para captar, visualizar y describir un conjunto ordenado de acciones, desde un principio hasta un final.

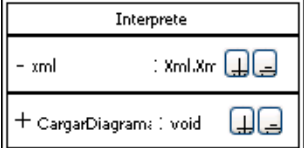
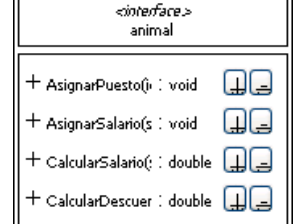
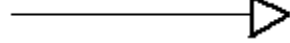

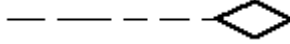
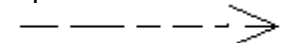
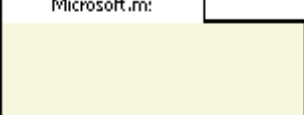
Un diagrama de actividades consta de los siguientes elementos:

Elemento	Descripción	Tipo
Nodo inicial	Circulo relleno que indica el inicio del flujo.	
Flujo de control	Es una flecha dirigida hacia fuera del nodo inicial. El flujo de control empieza en el símbolo que pierde foco y apunta hacia el objeto que lo aumenta y se conecta con esta.	
Condición guardián	Esta actúa como un centinela que requiere que se pase una prueba antes que el flujo continúe.	
Nodo de acción	Son las cosas que hacen o que suceden en un diagrama de actividades, y representa el camino que se sigue para saltar de acción en acción.	

<p>Nodos de decisión y de fusión</p>	<p>Los nodos de decisión y de fusión usan el mismo símbolo y transmiten la ramificación y la fusión condicionales.</p>	
<p>Bifurcación</p>	<p>Una bifurcación existe para describir comportamiento paralelo. Tiene una transición de entrada y muchas transiciones de salida.</p>	
<p>Unión</p>	<p>Tiene muchas transiciones de entrada, cada una de las cuales vienen de una actividad ejecutada en paralelo con las otras actividades, y tiene una transición de salida que se inicia cuando todas las actividades en paralelo han finalizado.</p>	
<p>Carriles</p>	<p>Se utilizan para representar que o quien es responsable de una parte de una actividad.</p>	
<p>Nodo final de actividad</p>	<p>Se utiliza para representar la finalización de una actividad.</p>	

3.1.3 DIAGRAMA DE CLASES

Los diagramas de clases se utilizan para representar las clases de un sistema y las relaciones entre ellas. Una misma clase puede aparecer en uno o más diagrama como facilidad visual para no mostrar todas las clases en un solo diagrama monolítico de clases. El mayor valor es mostrar las clases y sus relaciones desde varias perspectivas, de una manera que ayudara a transmitir la comprensión más útil.

Elemento	Descripción	Tipo
Clase	Es el elemento más común en un diagrama de clases. Se representa por medio del símbolo de clasificador, que se divide en regiones rectangulares con el nombre de la clase, sus atributos y métodos.	
Interfaces	Son equivalentes a clases abstractas puras, lo que quiere decir que no tendrá código ejecutable.	
Relaciones	Es una conexión entre elementos. En UML se encuentran los siguientes tipos de relaciones: -Asociación: el conector de asociación es una línea continua, si es dirigida. -Agregación: se representa con un diamante hueco, una recta y de manera opcional, una flecha de figura de palillos. -Generalización: se refiere a una relación del tipo “es uno de posibilidad de sustitución” y se representa con un conector de línea continua con triangulo hueco en uno de los extremos. -Realización: se refiere a heredar de interfaces de realización o las propias interfaces. El conector es casi idéntico a uno generalización, excepto que la línea de conexión es punteada con un triangulo hueco -Dependencia: en este tipo de relación una clase depende de la otra. El símbolo es una línea punteada con una flecha de figura de palillos.	Asociación  Agregación  Realización  Dependencia 
Paquete	Tiene la apariencia de una carpeta de archivos. Un paquete se puede implementar como un espacio de nombre o un subsistema.	

3.1.4 DIAGRAMAS CASO DE USO, ACTIVIDADES Y DE CLASES DE UML DE LA SOLUCIÓN PLANTEADA

Editor de diagramas de clase y objetos

El caso de uso “editor” permite al usuario realizar diagramas de clases y objetos de UML.

Descripción:

1. El usuario abre la aplicación: Realizar un diagrama de clases y/u objetos
2. Abrir un diagrama: el usuario podrá abrir un diagrama hecho en el editor anteriormente y modificarlo.
3. Guardar diagrama: el usuario luego de realizar o modificar un diagrama, podrá guardarlo en una ubicación
4. Imprimir el diagrama: el usuario podrá imprimir el diagrama realizado, antes o después de guardarlo

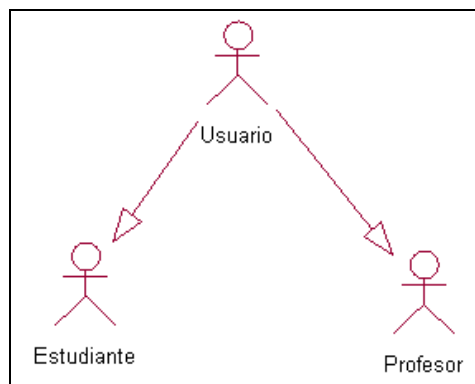


Ilustración 8 Caso de uso - actores involucrados

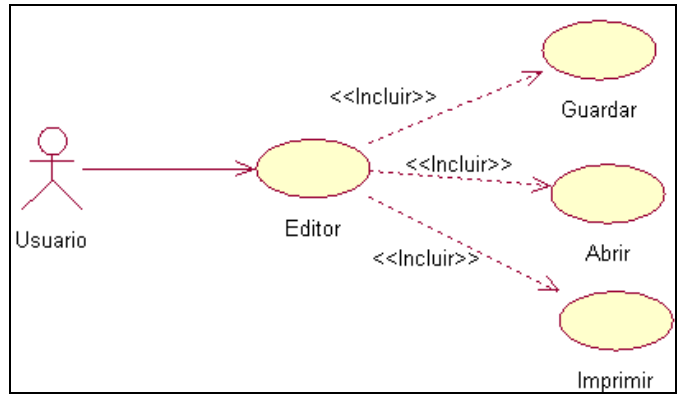


Ilustración 9 Caso de uso del editor de Diagramas de clases y objetos

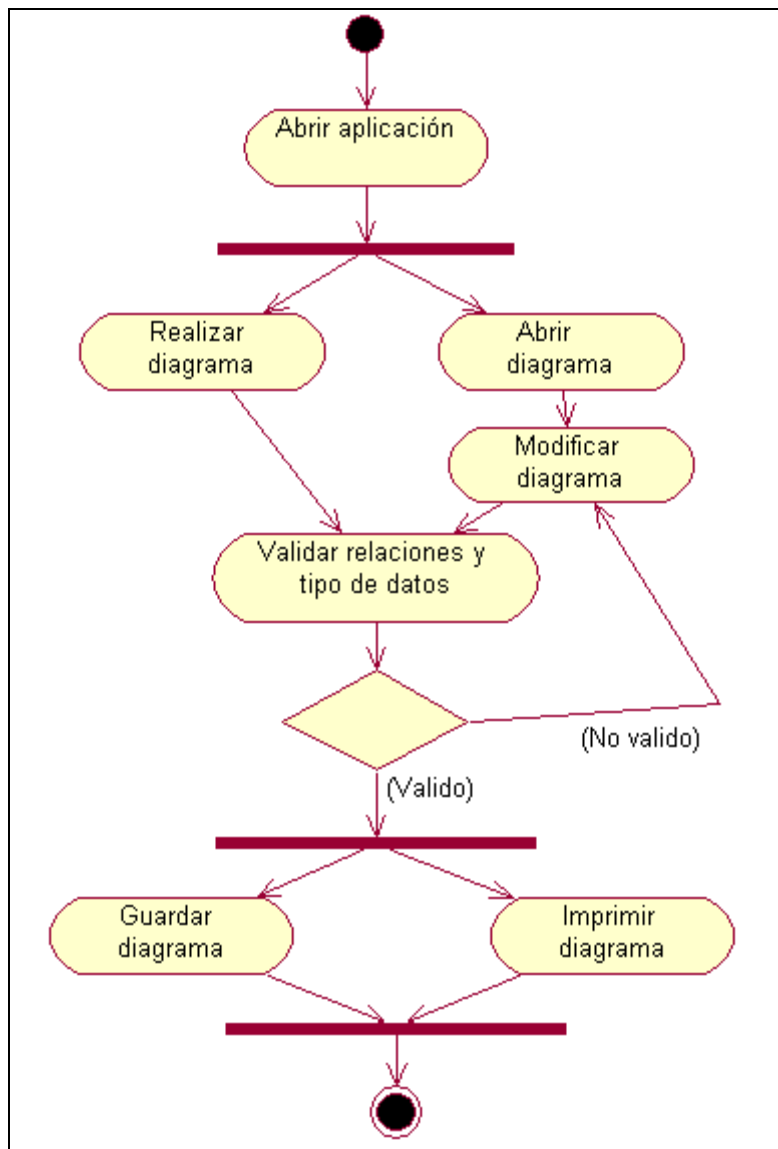


Ilustración 10 Diagrama de actividades del editor

Usar metodologías de aprendizaje

Este caso de uso le permite al usuario seleccionar el tipo de metodología que le ayude a la comprensión de los diagramas de clases y objetos de UML.

1. Seleccionar tipo de metodología: el usuario selecciona el tipo de metodología que considera mas adecuada a su consulta

- 1.1 Usar ejercicios resueltos
- 1.2 Usar ejercicio de continuidad
- 1.3 Usar listado de contenido
- 1.4 Usar preguntas frecuentes

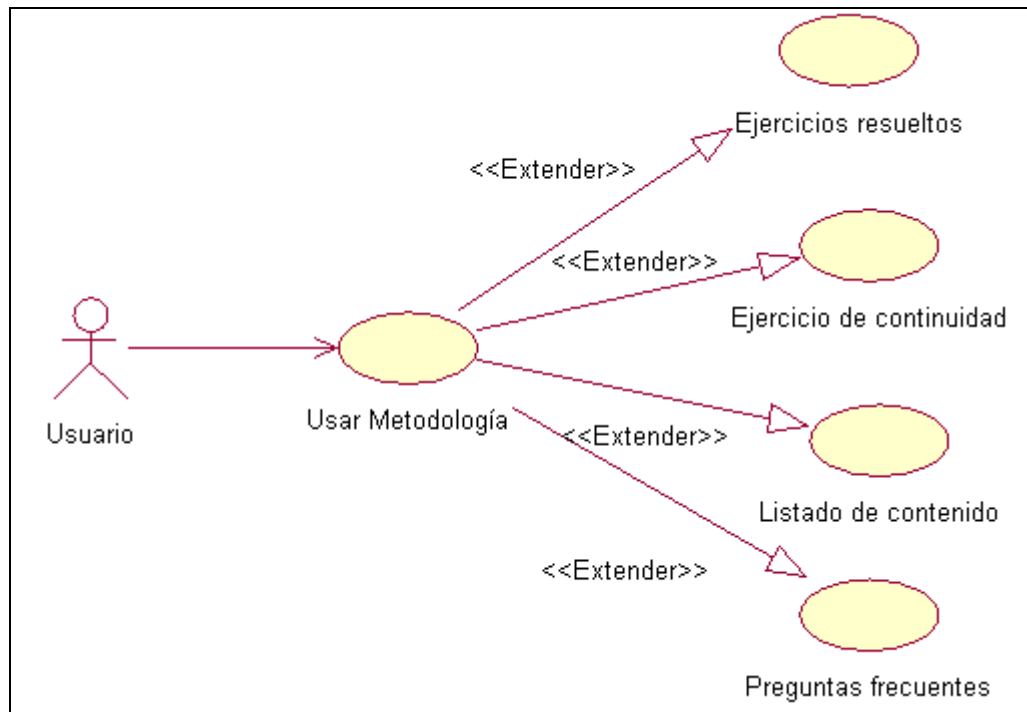


Ilustración 11 Caso de uso de los recursos audiovisuales

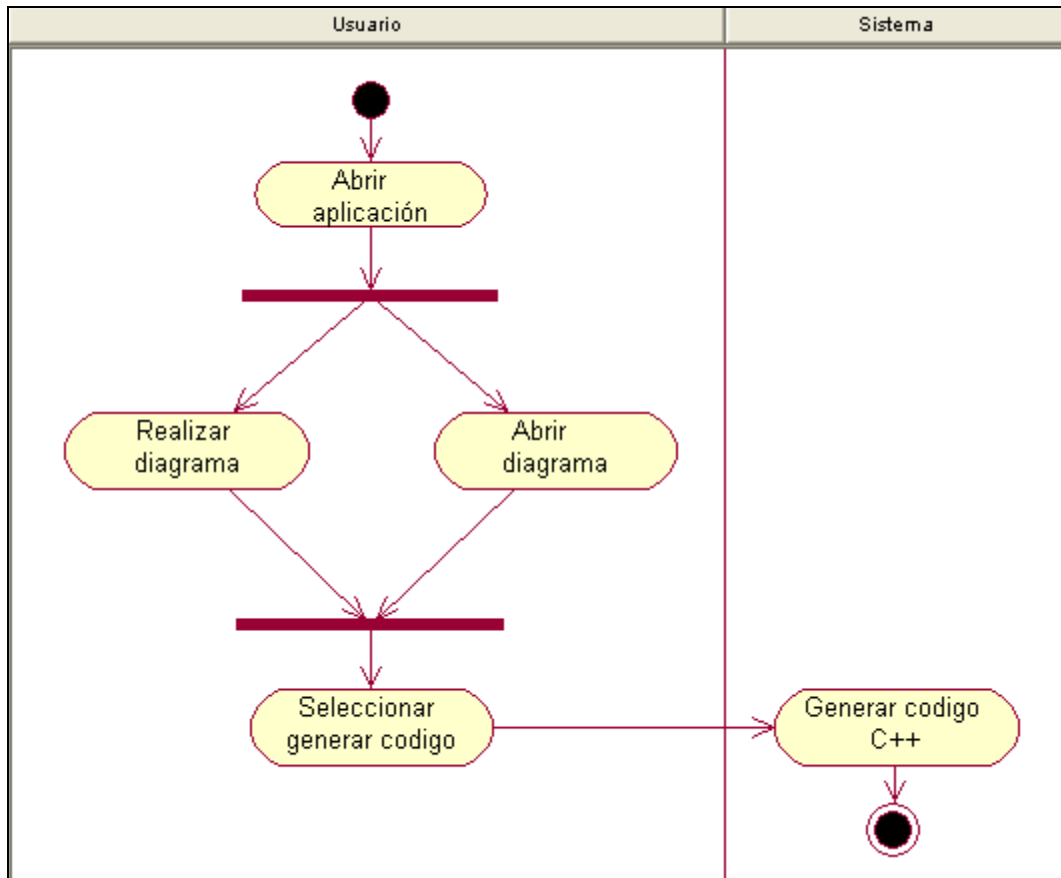


Ilustración 12 Diagrama de actividades del intérprete.

Interprete de código intermedio

El caso de uso "Usar intérprete" permite al usuario activar la generación de código C++ a partir del código intermedio generado con el diagrama de clases.

Descripción:

El usuario accionará la generación del intérprete, el cual tomara el código intermedio generado por el editor y lo convertirá en código C++.

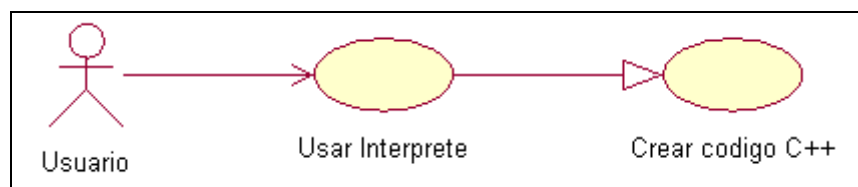


Ilustración 13 Caso de uso del intérprete

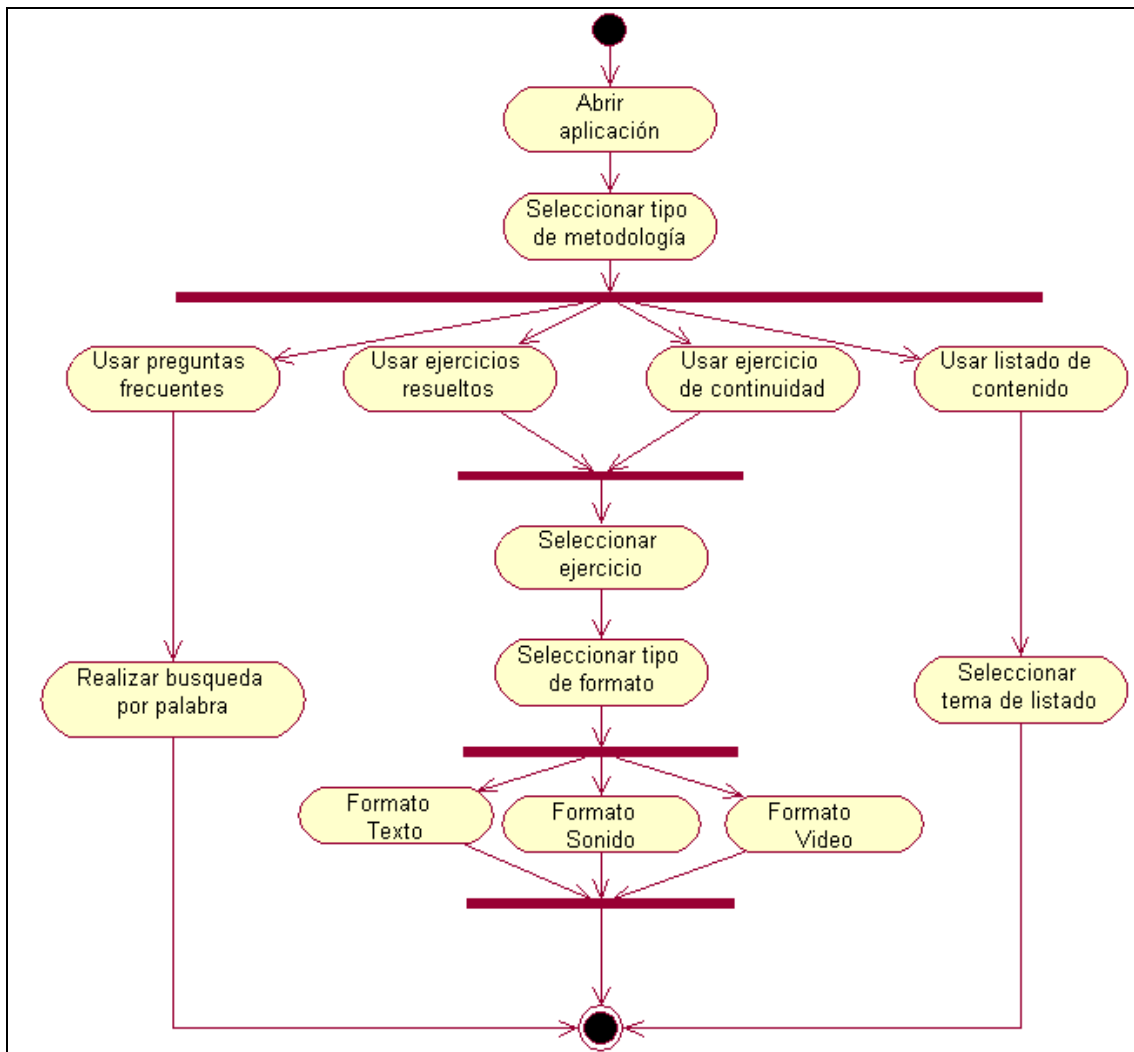


Ilustración 14 Diagramas de actividades de metodología

3.1.5 DIAGRAMA DE CLASES DE LA SOLUCIÓN PLANTEADA

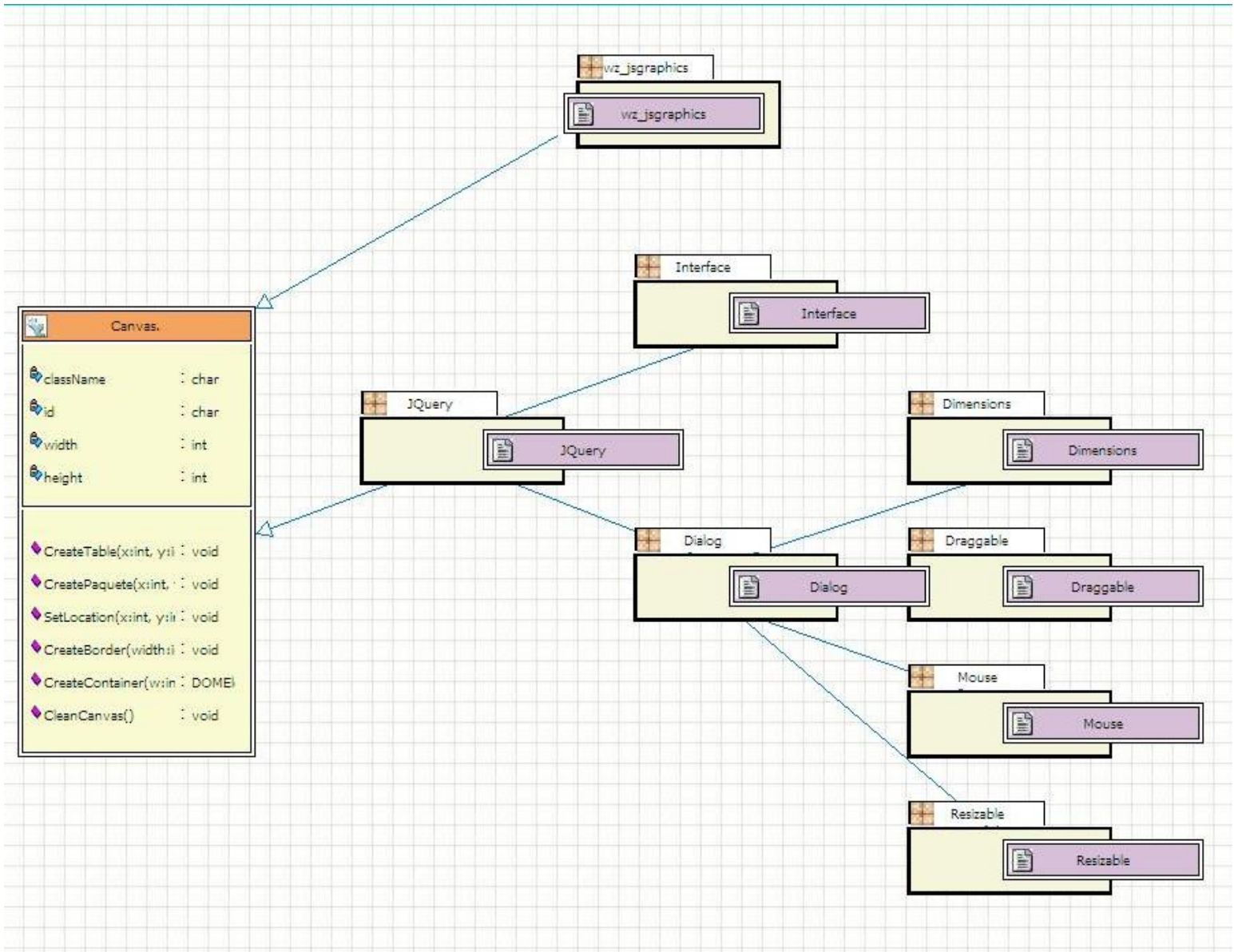


Ilustración 15 Diagrama de clases de la aplicación Parte 1

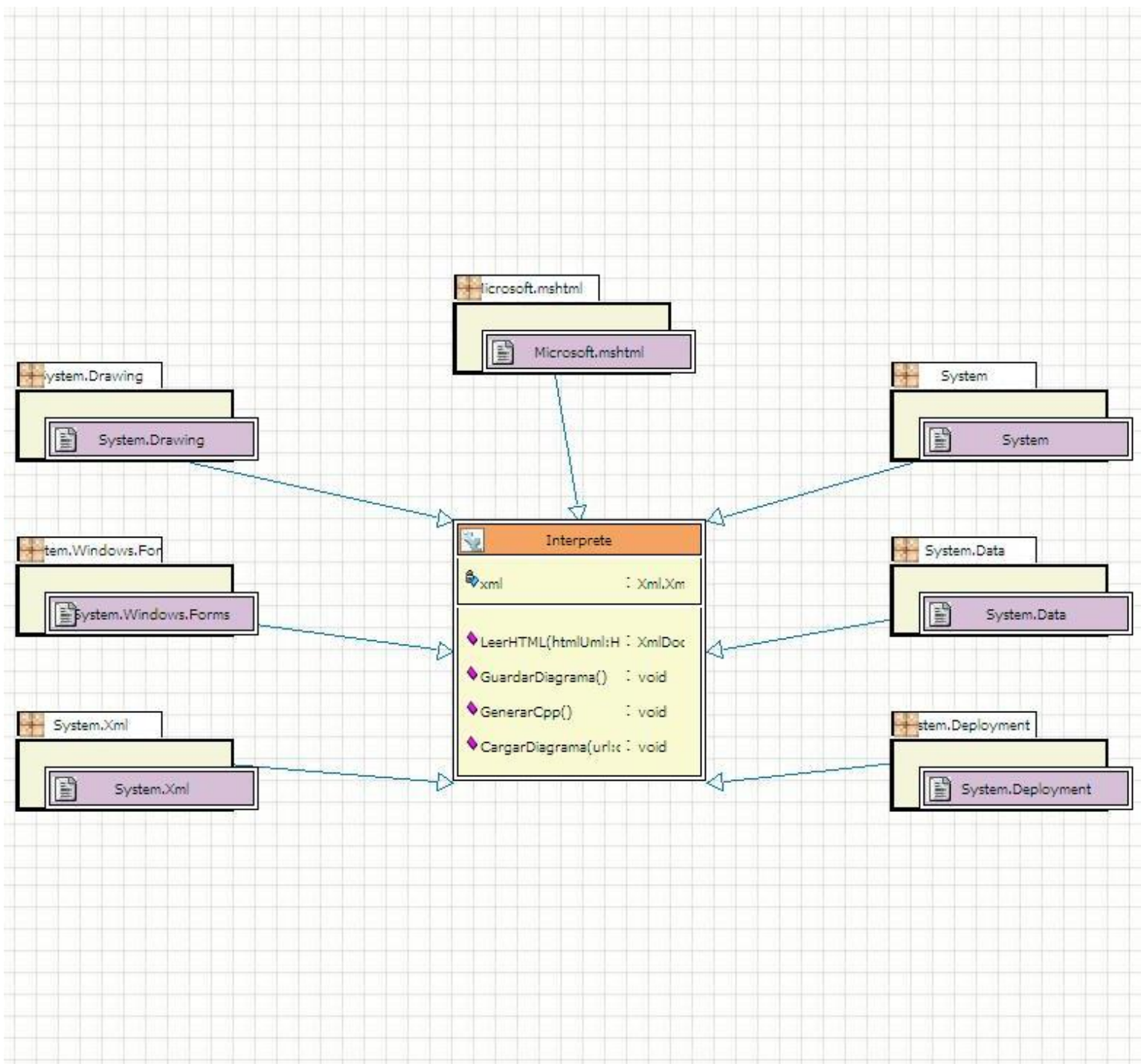


Ilustración 16 Diagrama de clases de la aplicación Parte 2

3.2 ANÁLISIS DE HERRAMIENTAS PARA EL DESARROLLO DEL EDITOR E INTÉRPRETE.

Para el desarrollo del editor de diagramas de clase y objetos de UML, existieron muchas opciones en cuanto al lenguaje de desarrollo, como por ejemplo: Java, Javascript, HTML, Visual Basic .NET, Visual Basic 6.0, C# y C++. Pero existieron factores determinantes para seleccionar un lenguaje,

Basado en la especialización que los integrantes adquirieron en el transcurso de sus estudios universitarios, se optó por utilizar los lenguajes de programación HTML, Visual Basic .NET, Visual Basic 6.0, Javascript y C++, siendo estas las herramientas con mayor experiencia adquirida para el desarrollo de este tipo de aplicación.

Una vez determinado los posibles lenguajes para desarrollar el editor e intérprete, se investigó las facilidades que estos tenían para poder desarrollar en ellos, una aplicación con gráficos simples en dos dimensiones, resultando como factibles: HTML y Javascript. Sin embargo, el uso de estos lenguajes solo podría limitarse para el desarrollo de espacio de trabajo del editor (donde se realizarán los diagramas), y no para realizar el intérprete, porque estos lenguajes son limitados en cuanto a la escritura en disco duro, lo cual no permitiría que existiese la función de poder guardar y abrir los diagramas realizados en el editor. Por lo que se realizó otra selección con estos últimos lenguajes, pero ahora buscando el que mejor se acoplara para desarrollar el intérprete, resultando como factibles: Visual Basic .NET, Visual Basic 6.0 y C++. Sin embargo, entre ellas la mejor opción es Visual Basic .NET, ya que presenta mejoras significativas en comparación a su versión anterior (Visual Studio 6.0, a la cual pertenecen C++ y Visual Basic 6.0) permitiendo una mejor portabilidad de las aplicaciones desarrolladas en ellas y menos dependencias de librerías externas (Ver tema 3.2.1 y 3.2.2) .

Luego se investigó sobre la programación gráfica en dos dimensiones en estos lenguajes, y se encontró que existían herramientas, que son adicionales, enfocadas para la programación gráfica avanzada en tres y dos dimensiones, las cuales son:

DirectX¹⁸ y OpenGL¹⁹; por lo que no resulta factible en tiempo de desarrollo, utilizar alguna de estas herramientas, ya que el estimado para llegar a un nivel intermedio es demasiado grande. En vista de todo esto, se decidió utilizar una combinación de lenguajes; utilizando Javascript y HTML para desarrollar el entorno de trabajo del editor y Visual Basic .NET para la parte del interprete y las funciones de abrir y guardar diagramas hechos con el editor. A continuación se detalla la investigación de las herramientas a utilizar en el desarrollo del editor e interprete de diagramas de UML.

3.3 DEFINICIÓN DE LA SOLUCIÓN

El sistema de apoyo para el aprendizaje de los Diagramas de clase de UML consistirá en tres módulos:

- 1) Editor de Diagramas de clases y objetos.
- 2) Ayuda audiovisual
- 3) Interprete de Diagramas de clases a código C++

Editor de diagramas de clase y objetos: Se encargara de proporcionar un área de trabajo al usuario, donde podrá realizar relaciones entre clases, crear o eliminar clases, objetos e interfaces, editar las propiedades y funciones de las clases e interfaces.

La ayuda audiovisual: es un apartado donde se encontraran temas relacionados a los diagramas de clases y objetos, así como ejemplos interactivos, videos demostrativos, preguntas frecuentes, listado de contenidos y grabaciones de audio.

El interprete de diagramas de clases a código C++: Es un modulo que crea archivos con extensión cpp y que contienen la estructura de las clases creadas en el Editor de diagramas de clases de UML.

¹⁸ *DirectX es una colección de librerías creadas para facilitar las complejas tareas relacionadas con la programación orientada a juegos en la plataforma Microsoft Windows.*

¹⁹ *OpenGL es una especificación estándar que define un conjunto de librerías multilenguaje y multiplataforma para desarrollar aplicaciones que utilicen gráficos 2D y 3D. Fue desarrollada por Silicon Graphics Inc. (SGI) en 1992. Su nombre viene del inglés Open Graphics Library, cuya traducción es biblioteca de gráficos abierta (es de libre uso). se utiliza en campos como diseño grafico, realidad virtual, representación científica y de información, simulación de vuelo o desarrollo de videojuegos.*

3.4 DESCRIPCIÓN DE INTERFAZ

La aplicación consta de una área de trabajo en la cual, el usuario puede crear diferentes paquetes, clases, interfaces y objetos. Además, crear relaciones entre ellos. Esto lo puede realizar por medio de la barra de componentes y el menú contextual.

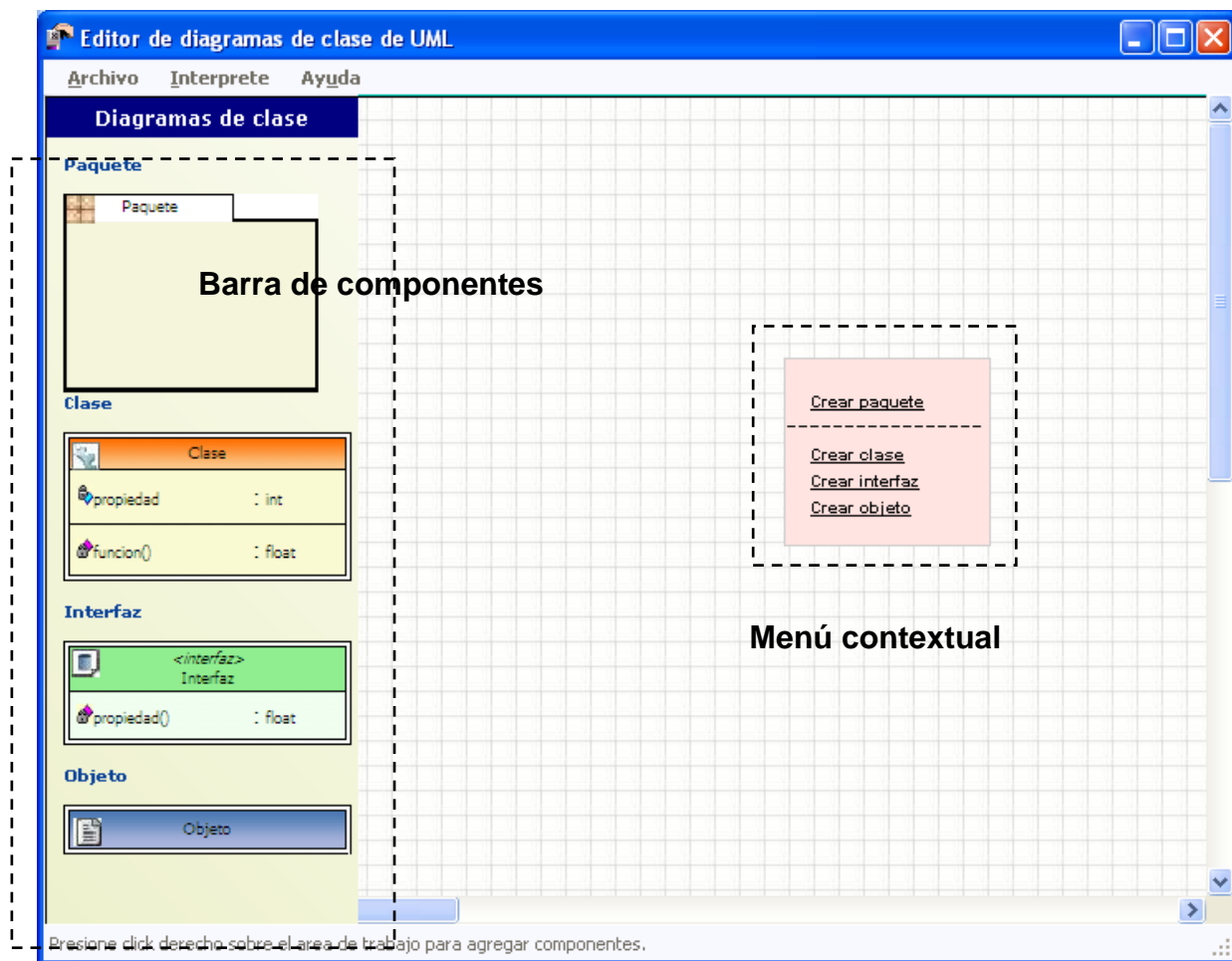


Gráfico 10 Barra de componentes y menú contextual de la aplicación.

Creación de paquete, clase, interfaz y objeto.

Como se ha mencionado, la aplicación puede agregar, modificar y eliminar paquetes, clases, interfaces y objetos; así también, crear relaciones entre ellas. Para realizar estas tareas tenemos la ayuda de la barra de componentes a continuación se detalla el flujo para cada uno de los componentes:

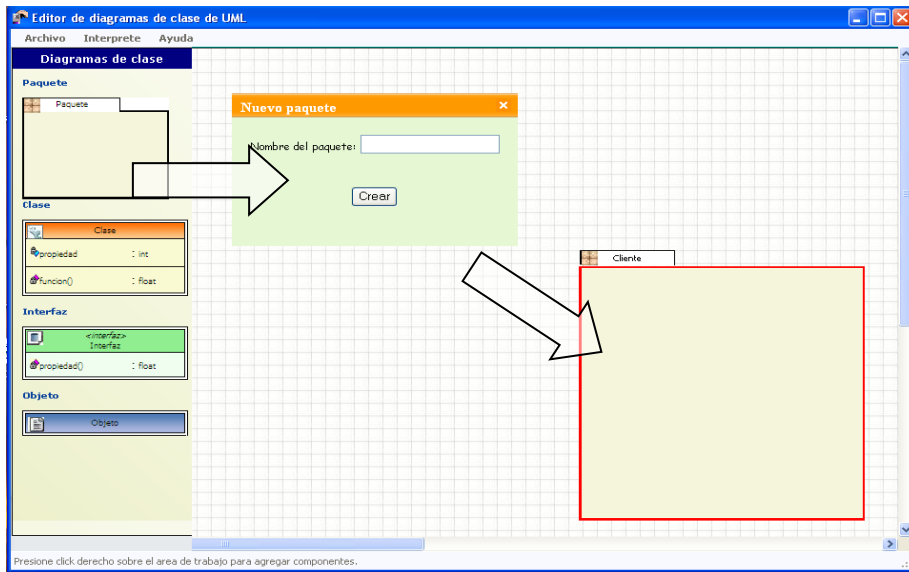


Gráfico 11 Agregar un paquete con de la barra de componentes.

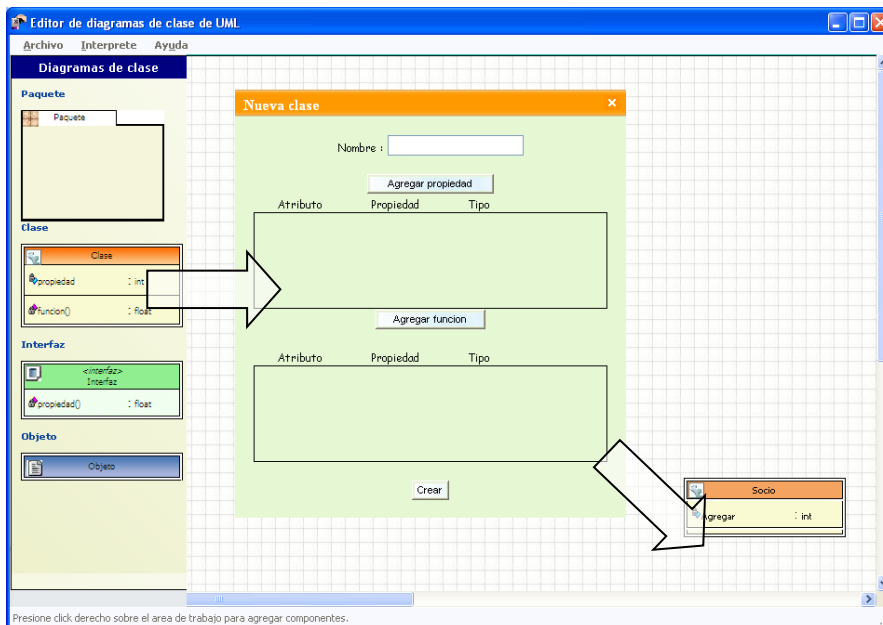


Gráfico 12 Agregar una clase con la barra de componentes.

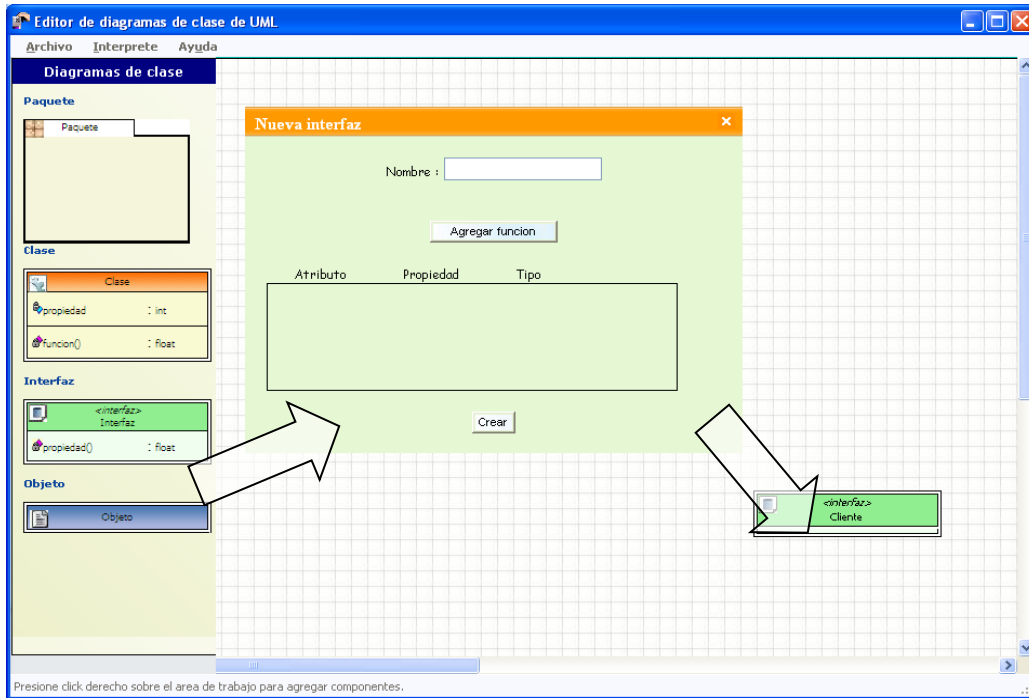


Gráfico 13 Agregar una interfaz con la barra de componentes.

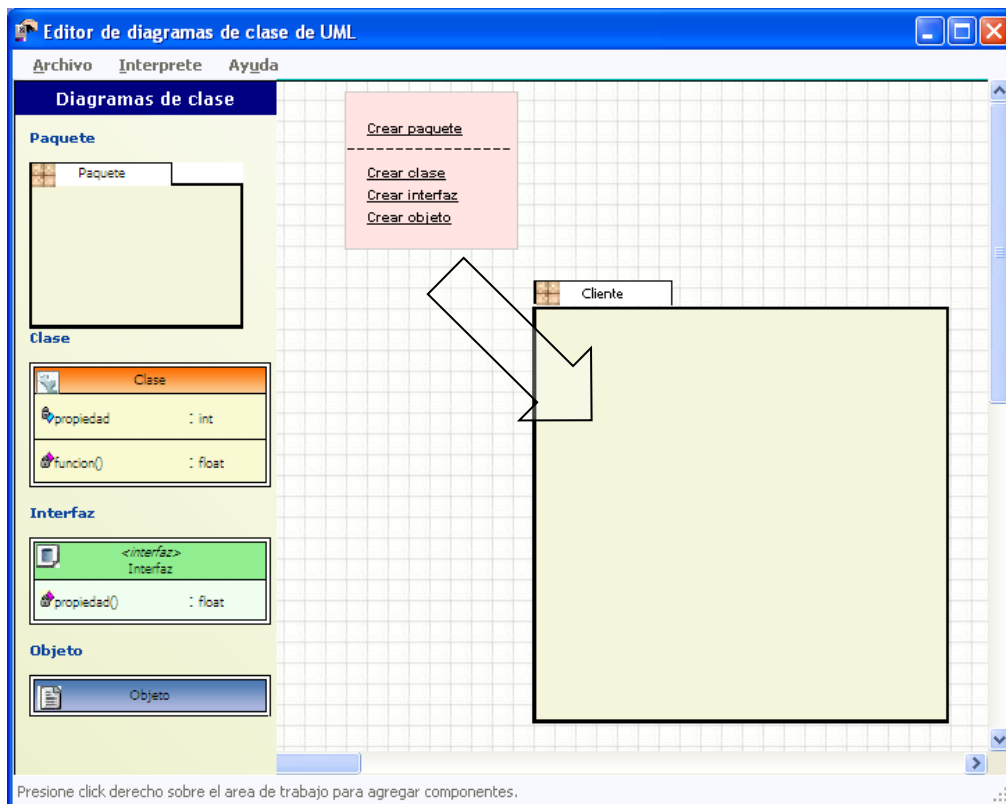


Gráfico 14 Creación de un paquete con menu contextual

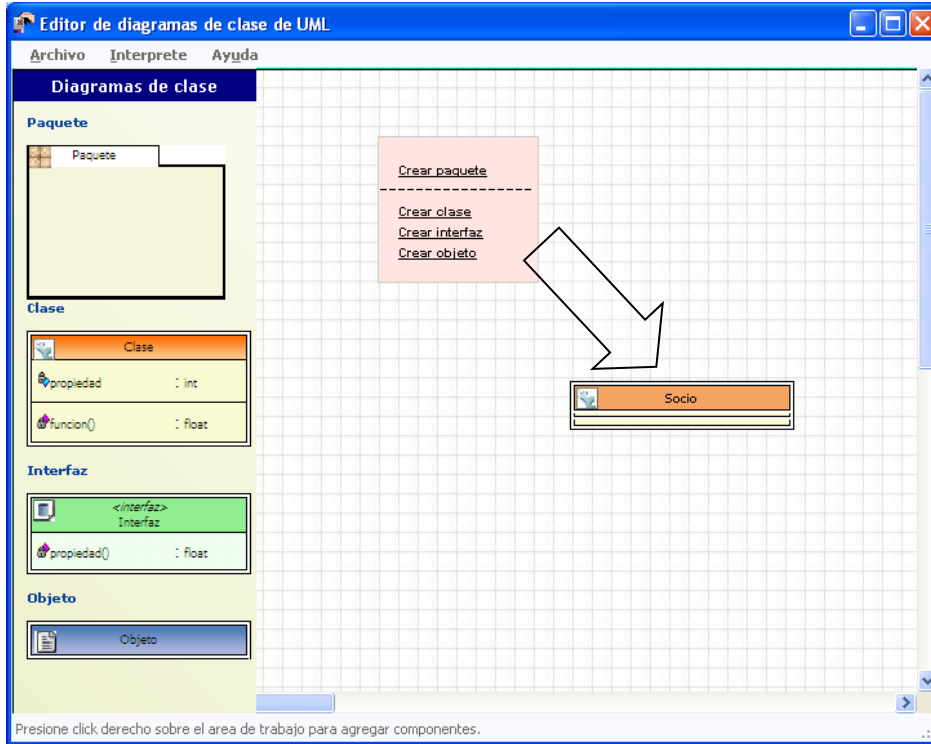


Gráfico 15 Creación de una clase por medio del menu contextual

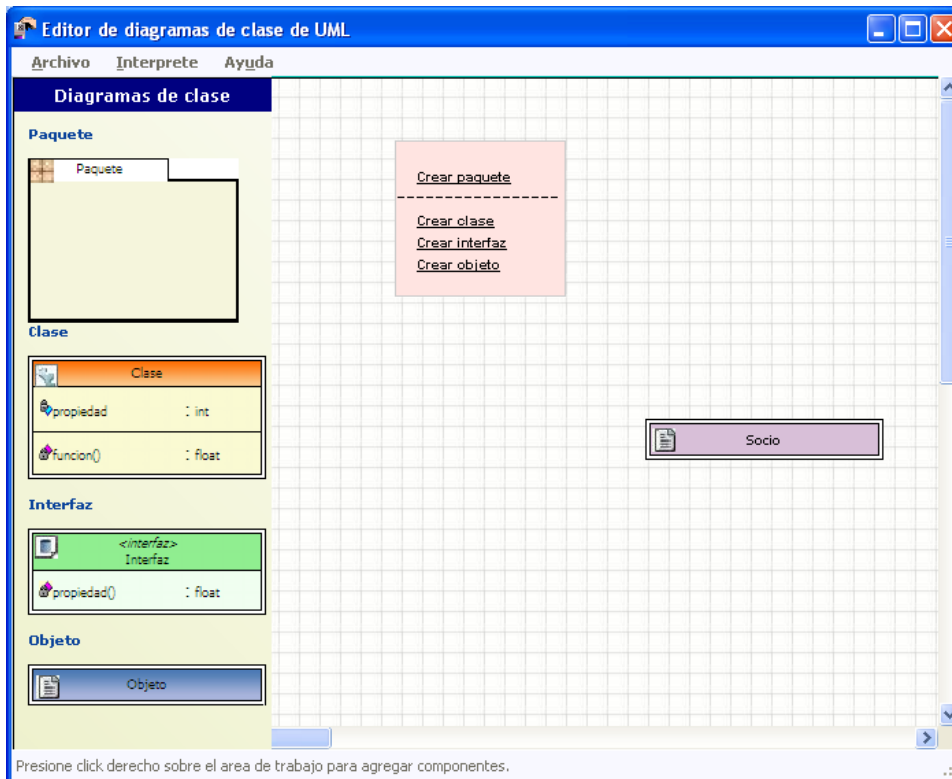


Gráfico 16 Creación de objeto por medio del menu contextual

CONCLUSIONES

En el presente trabajo de graduación hemos desarrollado el diseño de una aplicación educativa de editor de diagramas UML de clases y objetos. El propósito de ello ha sido, por un lado, apoyar a las personas que deseen aprender a utilizar dichos diagramas de modelado; y por otro, revelar el conocimiento que tiene las personas acerca del uso de los diagramas UML. Esto se llevo a cabo por medio de una investigación que mostró que muchas personas manifiestan tener un conocimiento amplio de los diagramas UML para utilizarlos, pero en la realidad no lo tienen.

Ambos diagramas UML tanto de objetos y de clases fueron seleccionados debido a su importancia ya que estos representan las clases y objetos que son utilizados dentro de cualquier sistema. Apoyándonos en esta afirmación, dentro del propio diseño del editor se encuentra un análisis que se desarrollo utilizando los diagramas de clases de UML.

El editor de diagramas UML cuenta con una ayuda en el cual consta de un conjunto de ejercicios para ser resueltos por la aplicación. Este beneficio ayudara a interpretar mejor los diagramas de clase y objeto de UML.

De acuerdo a las bases de la investigación en lo referido a las herramientas didácticas, el comportamiento de las personas se oriento mas por lo audiovisual, lista de contenido y ejercicios resueltos. Este dato nos ayudo a darle forma a la aplicación e integrar los elementos necesarios para la ayuda de la aplicación con el fin de lograr el máximo entendimiento de los diagramas UML de clases y objetos.

En cuanto a la compatibilidad de la aplicación se concluyo que al desarrollarla utilizando framework .net y otras tecnologías como CSS, DHTML y javascript se logro una compatibilidad que satisface los requerimientos necesarios para correr la aplicación de cualquier de la familia de Windows XP. Siendo este el sistema operativo más utilizado a nivel mundial. Igualmente, la aplicación esta empaquetada

con todas las librerías necesarias para que el usuario no tenga ningún problema a la hora de instalarlo, y además poderlo transportar a todos los equipos que desea ya que el tamaño es inferior a los 40MB.

Los usuarios de la aplicación que utilicen el intérprete para visualizar los diagramas en código C++, ayudara a la comprensión de los diagramas de UML con la generación de código C++. Este código generado partiendo de un diagrama de clase UML podrá servir como una guía para que el usuario pueda visualizar mejor la programación orientado a objeto.

RECOMENDACIONES

A continuación se proporciona aspectos que a partir de toda nuestra investigación realizada, se consideran necesarios mejorar y tomar en cuenta para el futuro.

A la Universidad Don Bosco

Fomentar más el uso de herramientas de diseño de sistemas informáticos, debido a que presenta menos problemas durante la fase de desarrollo, implementación y mejora.

A los Estudiantes.

Combinar diferentes lenguajes de programación para tener mejores resultados.

Realizar una evaluación a conciencia para saber si cada quien posee los conocimientos básicos de UML, de no ser así, indagar en el tema, para conocer las ventajas que este tipo de diagramas ofrecen.

Apoyarse en los diagramas UML para el diseño sistemas informáticos en la vida laboral.

BIBLIOGRAFÍA

- R. SCHACH STEPHEN, ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS CON UML Y EL PROCESO UNIFICADO, México, Editorial McGraw-Hill Interamericana, 2005.
- KIMMEL, PAUL, MANUAL DE UML, México, Editorial McGraw-Hill Interamericana, 2007.
- WEITZENFELD ,ALFREDO, INGENIERÍA DE SOFTWARE ORIENTADA A OBJETOS CON UML, JAVA E INTERNET, México, Internacional Thomson Editores, 2005.
- LÓPEZ, KARLA, MERINO GARCÍA, ANGELA DEL ROSARIO TRABAJO DE GRADUACIÓN DESARROLLO DE SOFTWARE DIDÁCTICO PARA EL PROGRAMA DE EDUCACIÓN PARVULARIA SECCIÓN TRES: SEIS AÑOS, El Salvador, 2005.
- AREVALO LOPEZ, FRANCISCO JAVIER, HERRERA CARRILLO, GABRIELA EUGENIA. DESARROLLO DE EDITOR DE LENGUAJE DE MODELAMIENTO UNIFICADO (UML) PARA DIAGRAMA DE CASO DE USO, Trabajo de graduación para optar a grado de ingeniero en ciencias de la computación, Universidad Don Bosco, El Salvador 2006.

SITIOS WEB

- <http://www.di.uniovi.es/~claudio/isoft/recursos/INTIS.pdf>
Universidad de Oviedo
Departamento de Informática
Última visita: 27/Ene/07
- <http://es.wikipedia.org/wiki/UML>
Enciclopedia en línea
Última visita: 27/Ene/07
- <http://www.omg.com.mx/>
OMG INTERNATIONAL S.A. de C.V.
Última visita: 26/Ene/07

- <http://www.awprofessional.com/articles/article.asp?p=347699&rl=1>
Artículo sobre UML
Pearson Education, Addison-Wesley Professional
Última visita: 26/Ene/07

- http://iteso.mx/~snaval/materiales_didacticos/curso%20de%20UML.ppt
Diapositiva sobre UML
Por Benjamín Rodríguez
Última visita: 26/Ene/07

- <http://www-gris.det.uvigo.es/~avilas/UML/node22.html>
Grupo de Redes e Ingeniería de Software
Ana Fernández Vilas
Última visita: 03/Feb/07

- <http://www.creangel.com/uml/actividad.php>
Addison Wesley Ed. James Rumbaugh, Ivar Jacobson y Grady Booch.
Rational Software Corporation.
Año 2000. ISBN: 84-7829-037-0.

- <http://download.microsoft.com/download/2/4/d/24df163e-6b5d-43ae-9538-4146988d2edf/050524-MSF-UML-VS-AlexSanchez1.0.ppt>
Diapositiva sobre UML en la actualidad
Última visita: 15/Feb/07

- http://exa.unne.edu.ar/depar/areas/informatica/anasistem2/public_html/apuntes/maf/cap3.htm
Información del ciclo de vida en cascada
Última visita: 12/Jul/07

GLOSARIO DE TÉRMINOS

D

DFD: Abreviación utilizada para Diagrama de Flujo de Datos. Es una de las notaciones empleadas en la fase de diseño de un programa.

O

OOPSLA'95: Por su abreviación en inglés significa (Object-Oriented Programming, Systems, Languages & Applications) Abreviación en inglés de Programación Orientada a Objetos, Sistema, Lenguajes y Aplicaciones. Es una conferencia anual de la Asociación para el material de cálculo (Association for Computing Machinery por sus siglas en inglés) en la que se cubren temas de programación orientada a objetos, lenguajes y aplicaciones. Entre los logros está la ayuda al desarrollo de la programación orientada a objetos,

OMG: Consorcio de la industria computacional que define estándares como CORBA, UML y IIOP.

P

POO: Abreviación utilizada para Programación Orientada a Objetos, es una metodología de diseño de software y un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos) y comportamiento (esto es, procedimientos o métodos).

U

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por la OMG.

ANEXOS

1. ENCUESTA PILOTO

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN



Encuesta Piloto sobre Lenguaje Unificado de Modelado

Objetivo: Esta encuesta tiene como finalidad identificar el nivel de conocimiento de los estudiantes universitarios acerca de los diagramas de clases y objetos del Lenguaje Unificado de Modelado (UML por sus siglas en inglés). Además, descubrir las herramientas de diseño que utilizan para desarrollar dichos diagramas. Así como, analizar los métodos de aprendizaje que los estudiantes consideren eficientes.

Las conclusiones que se obtengan, serán de gran utilidad para el desarrollo de un editor e interprete de diagramas de clases y objetos de UML, que incorpore una ayuda audiovisual.

a) Conocimiento de UML

Indicaciones

Subraye la respuesta mas adecuada a cada una de las preguntas que se listan a continuación:

1. ¿Conoce usted como se utilizan los de diagramas de UML?

- Si
- No
- He escuchado acerca de ello.

Si su respuesta fue No pase a la pregunta #10, caso contrario continúe con la siguiente pregunta.

2. ¿Qué es un diagrama de clases?

- Se refiere a heredar de interfaces de realización.
- Representa la herencia de las clases.
- Representa las clases y sus interrelaciones.

3. ¿Cómo se representa una clase en los diagramas de clases de UML?

- Flecha, rectángulo y rombo.
- Con un clasificador, que contiene el nombre de la clase, propiedades y métodos.
- Diagramas de estado.

4. ¿Qué tipos de relaciones se utilizan en los diagramas de clases y objetos?

- Generalización, asociación, realización y composición
- Relación de uno a uno, de uno a muchos
- Asociación, relación de muchos a muchos

5. ¿Cuál es la definición de herencia y generalización?

- La herencia no se utiliza en la Programación Orientada a Objetos (POO por sus siglas en ingles). La generalización es un tipo de relación en los diagramas de clases y objetos de UML.
- La herencia es una característica requerida de la POO. La generalización es sinónimo de herencia.
- La herencia y la generalización son relaciones que se utilizan en los diagramas de entidad relación.

b) Software Comercial - Software Gratuito

Indicaciones

Seleccione las respuestas en base a su conocimiento. Puede seleccionar más de una respuesta.

6. ¿Cuáles de las siguientes herramientas de diseño del tipo software comercial conoce?

- Rational Rose
- Visio
- Power Designer
- Otro: _____

7. ¿Cuáles de las siguientes herramientas de diseño del tipo software gratuito conoce?

- Argo UML
- Umbrello
- Violet
- Otros: _____

Indicaciones

En una escala de 1 a 5 (donde 1 es deficiente y 5 es excelente), seleccione la casilla que se adecue mas a su opinión.

8. ¿Que calificación le colocaría a las siguientes características de las herramientas de software de tipo comercial?

-Flexibilidad	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Seguridad	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Ambiente agradable	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Fácil de utilizar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Facilidad de instalación	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Soporte	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Costo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

9. ¿Que calificación le colocaría a las siguientes características de las herramientas de software de tipo gratuito?

-Flexibilidad	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Seguridad	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Ambiente agradable	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Fácil de utilizar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Facilidad de instalación	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Soporte	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-Costo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

c) Metodología de aprendizaje

Indicaciones

Coloque del 1 al 6 el nivel de prioridad según su criterio, siendo 1 la más importante y 6 la menos relevante.

10. ¿Que nivel de prioridad tiene para usted las siguientes metodologías de aprendizaje?

-Clase expositiva

- Guías
- Videos demostrativos
- Preguntas frecuentes
- Ejercicios resueltos
- Otros: _____

Indicaciones

Coloque una X a la casilla con la respuesta que mejor le parezca. Solo seleccione una respuesta.

11. ¿Cuánto tiempo considera que debe durar un video demostrativo?

- 1 a 3 minutos
- 3 a 5 minutos
- Mas de 5 minutos

12. ¿Cuántas páginas deberían tener una clase expositiva con diapositivas?

- 10 a 20 páginas
- 20 a 25 páginas
- Mas de 25 páginas

13. Cuando usted lee un documento, ¿Qué tipo de formato prefiere que tenga este para facilitar su lectura?

- Formato digital
- Formato impreso

Gracias por su tiempo, apreciamos su ayuda.

2. ENCUESTA FINAL

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN



Encuesta sobre Lenguaje Unificado de Modelado

Objetivo: Esta encuesta tiene como finalidad identificar el nivel de conocimiento de los estudiantes universitarios acerca de los diagramas de clases y objetos del Lenguaje Unificado de Modelado (UML por sus siglas en inglés). Además, descubrir las herramientas de diseño que utilizan para desarrollar dichos diagramas. Así como, analizar los métodos de aprendizaje que los estudiantes consideren eficientes.

Las conclusiones que se obtengan, serán de gran utilidad para el desarrollo de un editor e interprete de diagramas de clases y objetos de UML, que incorpore una ayuda audiovisual.

d) Conocimiento de UML

Indicaciones

Subraye la respuesta mas adecuada a cada una de las preguntas que se listan a continuación:

1. ¿Conoce usted como se utilizan los diagramas de UML?

- Si
- No

Si su respuesta fue No pase a la pregunta #10, caso contrario continúe con la siguiente pregunta.

2. ¿Qué es un diagrama de clases?

- Se refiere a heredar de interfaces de realización.
- Representa la herencia de las clases.
- Representa las clases y sus interrelaciones.

3. ¿Cómo se representa una clase en los diagramas de clases de UML?

- Flecha, rectángulo y rombo.
- Con un clasificador, que contiene el nombre de la clase, propiedades y métodos.
- Diagramas de estado.

4. ¿Qué tipos de relaciones se utilizan en los diagramas de clases y objetos?

- Generalización, asociación, realización y composición
- Relación de uno a uno, de uno a muchos
- Asociación, relación de muchos a muchos

5. ¿Cuál es la definición de herencia y generalización?

- La herencia no se utiliza en la Programación Orientada a Objetos (POO por sus siglas en ingles). La generalización es un tipo de relación en los diagramas de clases y objetos de UML.
- La herencia es una característica requerida de la POO. La generalización es sinónimo de herencia.
- La herencia y la generalización son relaciones que se utilizan en los diagramas de entidad relación.

e) Aplicación Comercial – Aplicación Gratuita

Indicaciones

Seleccione las respuestas en base a su conocimiento. Puede seleccionar más de una respuesta.

6. ¿Cuáles de las siguientes herramientas de diseño (comercial o gratuita) ha utilizado para los diagramas de UML?

- Rational Rose
- Visio
- Power Designer
- Ninguno
- Argo UML
- Umbrello
- Violet
- Otros: _____

Indicaciones

En una escala de 1 a 5 (donde 1 es deficiente o económico y 5 es excelente o elevado), seleccione la casilla que se adecue mas a su opinión.

7. ¿Que calificación le colocaría a las siguientes características de las herramientas de aplicación de tipo comercial?

- Flexibilidad

1	2	3	4	5
---	---	---	---	---
- Seguridad

1	2	3	4	5
---	---	---	---	---
- Ambiente agradable

1	2	3	4	5
---	---	---	---	---
- Fácil de utilizar

1	2	3	4	5
---	---	---	---	---
- Fácil instalación

1	2	3	4	5
---	---	---	---	---
- Calidad del soporte

1	2	3	4	5
---	---	---	---	---
- Costo

1	2	3	4	5
---	---	---	---	---

8. ¿Que calificación le colocaría a las siguientes características de las herramientas de aplicación de tipo gratuito?

- Flexibilidad

1	2	3	4	5
---	---	---	---	---
- Seguridad

1	2	3	4	5
---	---	---	---	---
- Ambiente agradable

1	2	3	4	5
---	---	---	---	---
- Fácil de utilizar

1	2	3	4	5
---	---	---	---	---
- Fácil instalación

1	2	3	4	5
---	---	---	---	---
- Calidad del soporte

1	2	3	4	5
---	---	---	---	---
- Costo

1	2	3	4	5
---	---	---	---	---

f) Metodología de aprendizaje

Indicaciones

Ordene del 1 al 6 en nivel de prioridad según su criterio, siendo 1 la menos relevante y 6 la más importante, sin repetir el mismo número en otra casilla en más de una casilla.

9. ¿Que nivel de prioridad tiene para usted las siguientes metodologías de aprendizaje?

- Clase expositiva
- Guías

- Videos demostrativos
- Preguntas frecuentes
- Ejercicios resueltos
- Otros: _____

Indicaciones

Coloque una X a la casilla con la respuesta que mejor le parezca. Solo seleccione una respuesta.

10. ¿Cuánto tiempo considera que debe durar un video demostrativo?

- 1 a 3 minutos
- 3 a 5 minutos
- Mas de 5 minutos

11. ¿Cuántas páginas deberían tener una clase expositiva con diapositivas?

- 10 a 20 páginas
- 20 a 25 páginas
- Mas de 25 páginas

12. Cuando usted lee un documento, ¿Qué tipo de formato prefiere que tenga este para facilitar su lectura?

- Formato digital
- Formato impreso

Gracias por su tiempo, apreciamos su ayuda.

3. RESULTADOS DE ENCUESTA REALIZADA POR MICROSOFT

¿Cual ha sido el resultado de sus proyectos al no utilizar una herramienta diagramas de sistemas?



Gráfico 17 Resultados de los proyectos

El 53% de encuestados opina que sus proyectos realizados sin ayuda de una herramienta de diagramas de sistema es cuestionado al final, el 31% fue cancelado y solo un 16% tuvo éxito.

4. POBLACION ESTUDIANTIL A NIVEL SUPERIOR UNIVERSITARIA

Cuadro resumen de las universidades autorizadas por el Ministerio de Educación. La población considerada para el instrumento de medición fue la del 2006.

POBLACIÓN ESTUDIANTIL A NIVEL SUPERIOR Años 2005-2006

CUADRO No. 1

No.	UNIVERSIDADES	2,005	2,006	Diferencia	(%)
1	ALBERT EINSTEIN	1,227	898	(329)	(26.81)
2	AUTÓNOMA DE SANTA ANA	778	803	25	3.21
3	CAPITÁN GENERAL GERARDO BARRIOS (*)	3,820	3,494	(326)	(8.53)
4	CATÓLICA DE OCCIDENTE (A)	2,994	3,067	73	2.44
5	CENTROAMERICANA JOSÉ SIMEÓN CAÑAS (A)	9,083	9,279	196	2.16
6	CRISTIANA DE LAS ASAMBLEAS DE DIOS	789	778	(11)	(1.39)
7	DE EL SALVADOR (*)	37,248	36,416	(832)	(2.23)
8	DE ORIENTE (*)	4,210	4,183	(27)	(0.64)
9	DE SONSONATE	1,926	1,939	13	0.67
10	DON BOSCO (A)	3,596	3,795	199	5.53
11	DR. ANDRÉS BELLO (*)	3,858	4,547	689	17.86
12	DR. JOSÉ MATÍAS DELGADO (A)	5,337	5,632	295	5.53
13	EVANGÉLICA DE EL SALVADOR (A)	3,090	3,132	42	1.36
14	FRANCISCO GAVIDIA (*) (A)	8,330	8,768	438	5.26
15	LEONARDO DA VINCI	270	250	(20)	(7.41)
16	LUTERANA SALVADOREÑA	306	273	(33)	(10.78)
17	MODULAR ABIERTA (*)	3,852	3,923	71	1.84
18	MONSEÑOR OSCAR ARNULFO ROMERO	745	880	135	18.12
19	NUEVA SAN SALVADOR	751	686	(65)	(8.66)
20	PANAMERICANA (*)	1,734	1,921	187	10.78
21	PEDAGÓGICA DE EL SALVADOR	3,043	3,397	354	11.63
22	POLITÉCNICA DE EL SALVADOR	1,121	1,059	(62)	(5.53)
23	SALVADOREÑA ALBERTO MASFERRER (A)	1,558	1,670	112	7.19
24	SALVADOREÑA ISAAC NEWTON	279	292	13	4.66
25	TÉCNICA LATINOAMERICANA	374	343	(31)	(8.29)
26	TECNOLÓGICA DE EL SALVADOR (A)	13,480	14,290	810	6.01
A)	TOTAL UNIVERSIDADES	113,799	115,715	1,916	1.68

(*) Incluye Centros Regionales

(A) Institución Acreditada a diciembre 2006.

5. PRESUPUESTO

En la tabla 6 se detallan los elementos necesarios para llevar a cabo la aplicación en la parte de desarrollo. Además, en la tabla 7 se muestran los correspondientes costos para la implementación.

CANTIDAD	DESCRIPCIÓN	MONTO (US \$)	TOTAL (US \$)
3	Arrendamiento de computadoras para desarrollo de aplicación.	\$300.00	\$900.00
3	Honorarios de programadores	\$800.00	\$2,400.00
3	Licencia de Visual Basic .net 2005	\$850.00	\$2,250.00
		Total	\$5,500.00

Tabla 6 Presupuesto de desarrollo

CANTIDAD	DESCRIPCIÓN	MONTO (US \$)	TOTAL (US \$)
1	Computadora de escritorio	\$550.00	\$550.00
1	Licencia de Windows XP Home Edition	\$125.00	\$125.00
		Total	\$675.00

Tabla 7 Presupuesto de implementación

En el caso que la institución, cuente con computadoras y licencias para el desarrollo e implementación, que cumplan con la descripción que se encuentra en las tablas 12.1 y 12.2, se considerará dentro del presupuesto solamente los honorarios de los programadores.

Depreciación Método de línea recta

En el método de depreciación en línea recta se supone que el activo se desgasta por igual durante cada periodo contable. El método de la línea recta se basa en el número de años de vida útil del activo, de acuerdo con la fórmula:

(Costo – valor de desecho) /Años de vida útil = monto de la depreciación para cada año de vida del activo.

Ecuación 1 Método de la línea recta

La depreciación anual para la computadora será el costo de \$550.00 con una vida útil estimada de cuatro años y un valor de recuperación de \$50.00, usando el método de la línea recta será:

$$(\$550.00 - \$50.00) / 4 \text{ años} = \text{Gasto de depreciación anual de } \$125.00$$

Ecuación 2 Calculo de la depreciación

CRONOGRAMA

MANUAL DE USUARIO

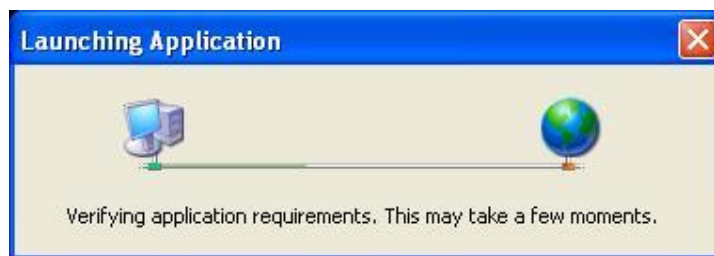
Instalación de la aplicación

Pasos:

- 1- Seleccionamos el icono de EditorUMLXML.exe y damos doble clic.



- 2- El instalador se ejecuta descomprimiendo los archivos necesarios, como vemos en la imagen



- 3- A continuación se confirma la ubicación en la cual se instalará la aplicación, damos un clic en el botón **Install**.



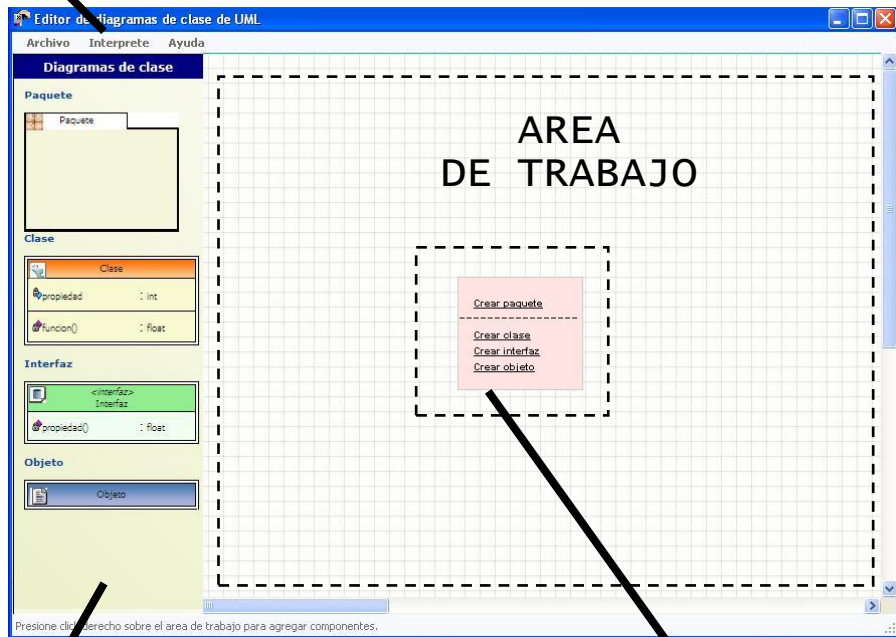
- 4- Para finalizar se instalarán todos los componentes necesarios, confirmando por medio de la ventana siguiente:



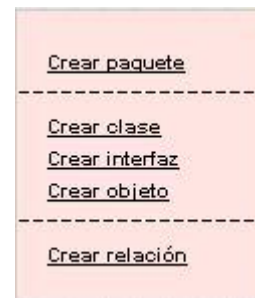
Descripción de ventana principal

Archivo Interprete Ayuda

Barra de menú



Menú de componentes



Menú contextual

Descripción de Menús

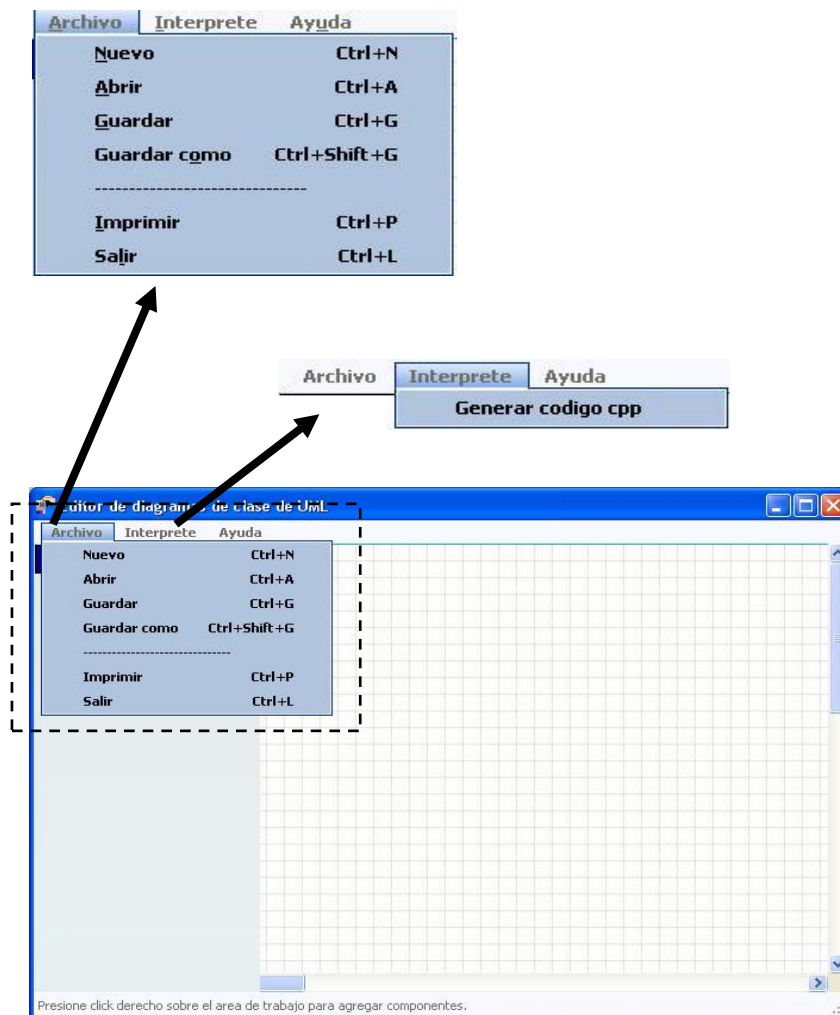
Menú contextual

Archivo

- Nuevo: Crea un nuevo documento en blanco
- Abrir. Abre un documento existente
- Guardar. Guarda el documento con formato xml
- Imprimir: imprime el documento
- Salir: Cierra la aplicación

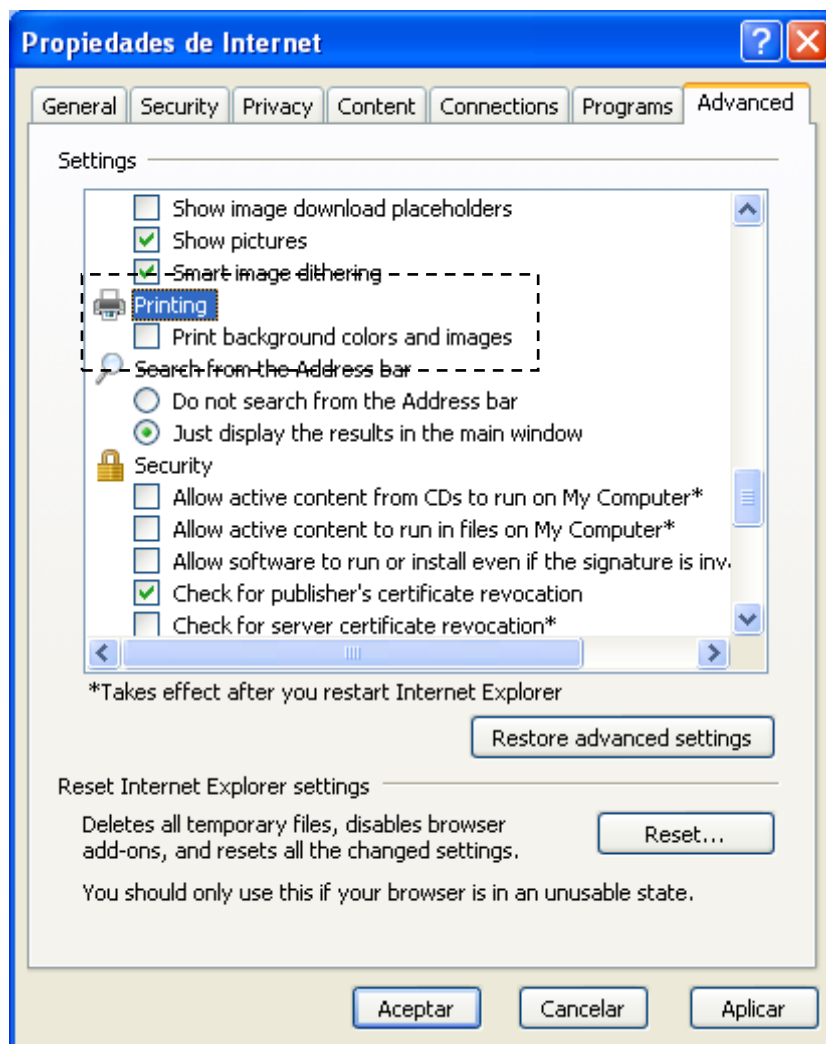
Interprete

Generar código cpp: Transforma a código C++ el diagrama de clases y/u objetos que se encuentre en el área de trabajo en ese momento.



Antes de realizar la impresión, se debe configurar la opción "Imprimir fondo e imágenes" de Internet Explorer. Los pasos son los siguientes:

- 1- Clic en menú Inicio
- 2- Clic en Panel de Control
- 3- Abrir la opción "Opciones de Internet" y aparecer una ventana con las opciones de Internet Explorer
- 4- Seleccionar la pestaña "Avanzado"
- 5- Buscar el grupo "Impresión" que se encuentra al final del listado que se presenta.
- 6- Verificar que la opción "Imprimir colores de fondo e imágenes" se encuentre con un cheque. De no ser así, chequearla.
- 7- Clic en OK



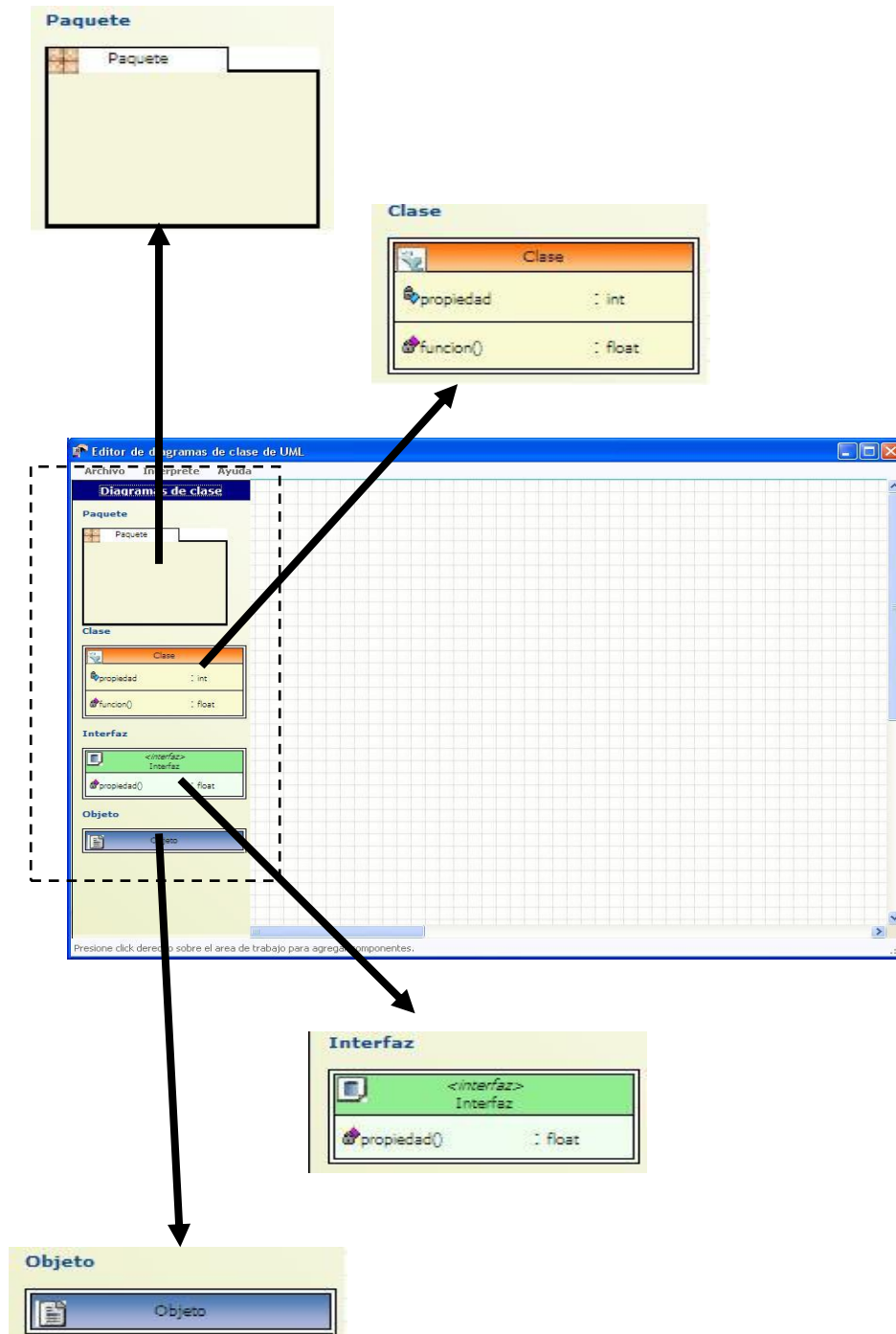
Menú de componentes

Paquete: permite la creación de un paquete

Clase: Permite la creación de una clase, con propiedades y funciones

Interfaz: permite la creación de una interfaz

Objeto: permite la creación de un objeto



Menú contextual

Para acceder a este menú debemos dar clic derecho sobre el área de trabajo. Este menú permite crear:

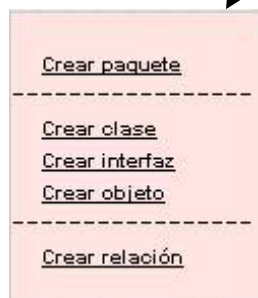
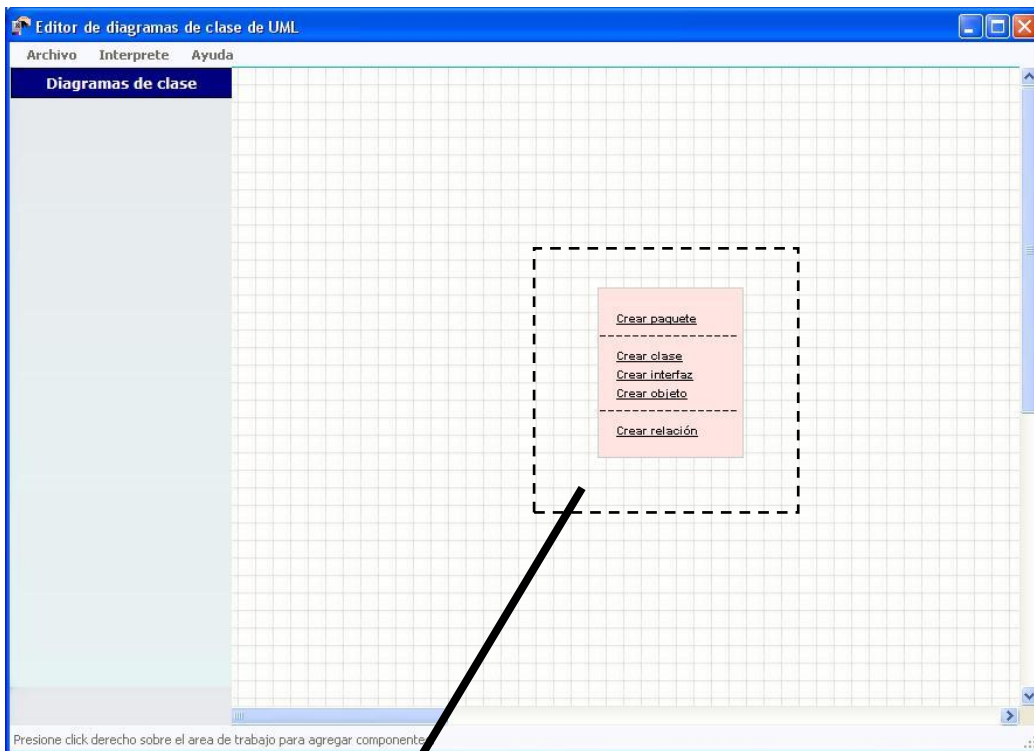
Crear paquete: permite la creación de un paquete

Crear clase: Permite la creación de una clases, con propiedades y funciones

Crear interfaz: permite la creación de una interfaz

Crear objeto: permite la creación de un objeto

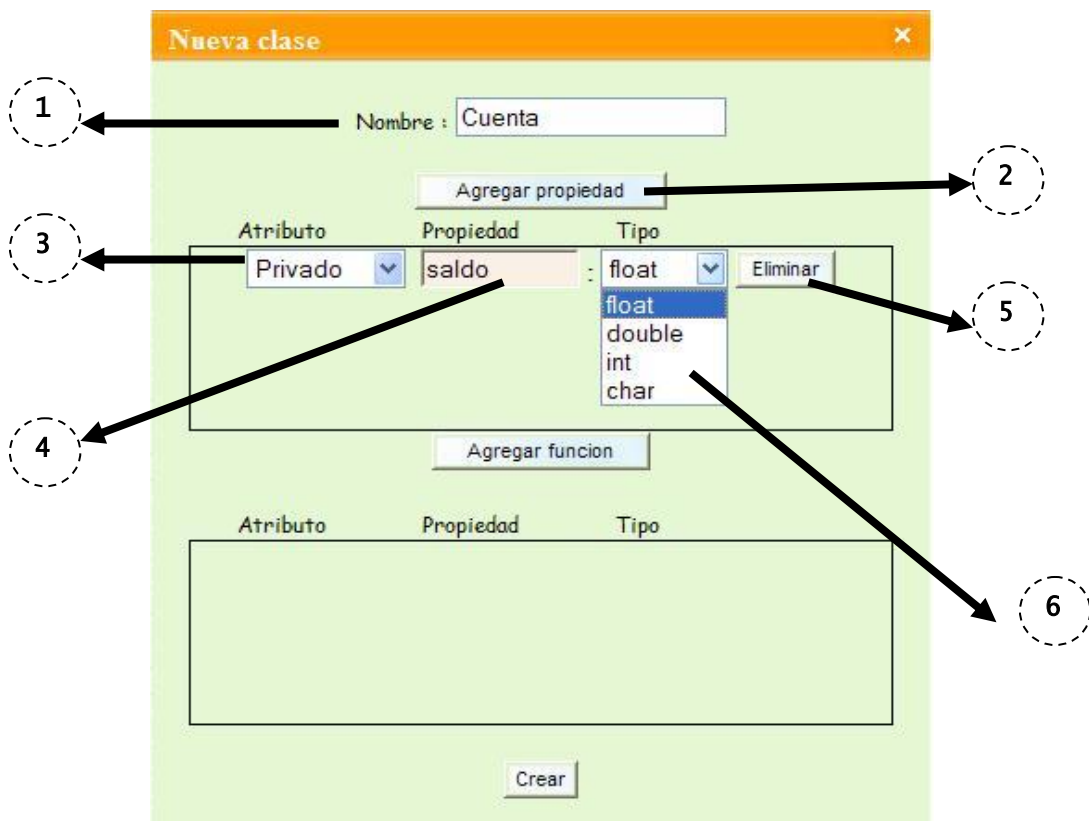
Crear relacion: permite la creacion de una relacion entre dos clases, objetos o interfaces



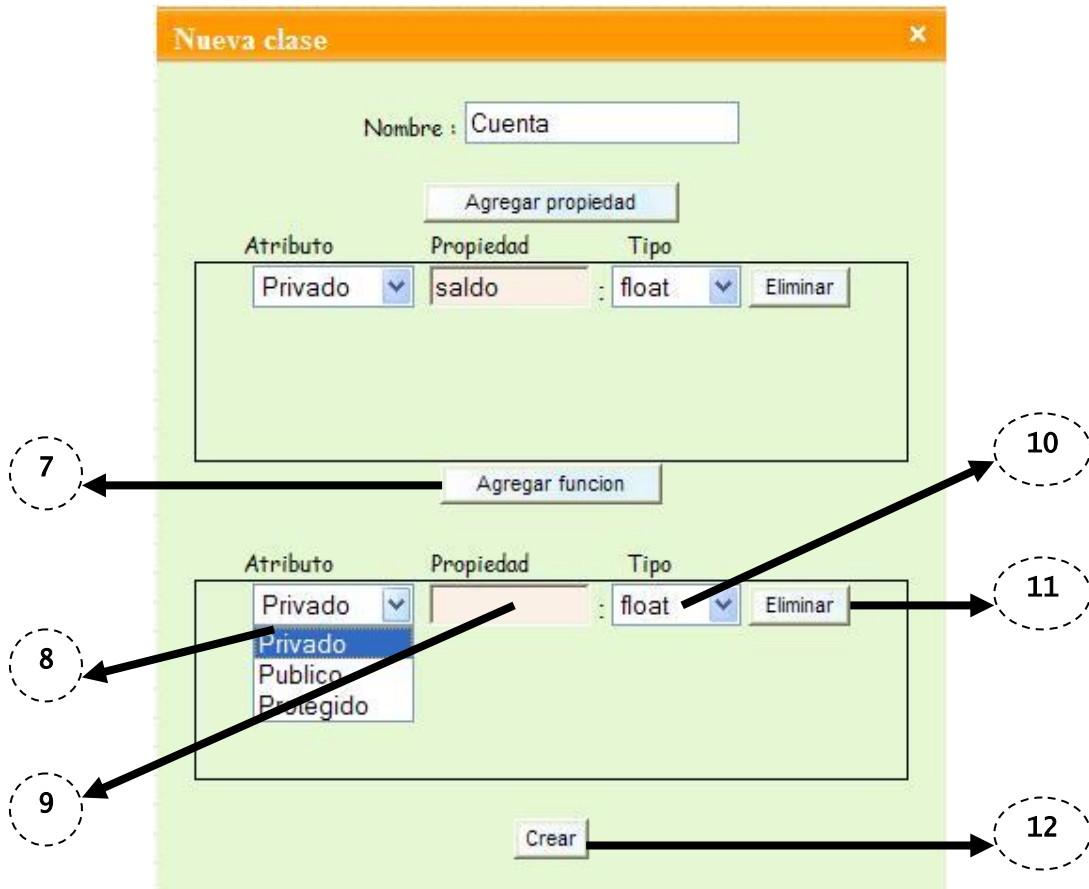
Como hacer una clase

Para crear una clase podemos seleccionar Clase del menú de componentes o Crear clase del menú contextual dando clic derecho sobre el area de trabajo, a continuación se presenta la siguiente ventana

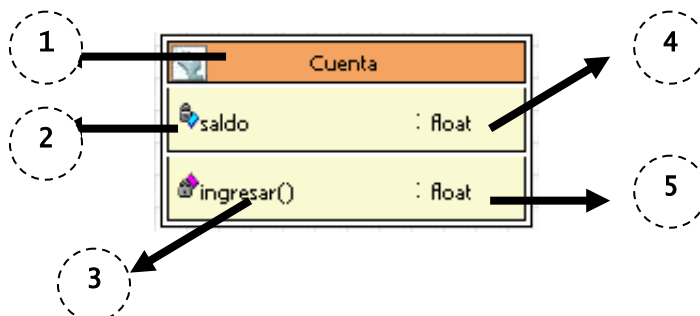
1. Nombre de la clase
2. Botón Agregar propiedad: permite agregar una propiedad a la clase.
3. Atributo: coloca el atributo a la propiedad de la clase, este puede ser privado, público o protegido.
4. Propiedad: nombre de la propiedad de la clase
5. Tipo: se selecciona el tipo de la propiedad de la clase esta puede ser flota, double, int y char.
6. Botón eliminar: permite eliminar una propiedad de la clase.



7. Botón Agregar función: permite agregar una función de la clase.
8. Atributo: coloca el atributo de la función de la clase, este puede ser privado, público o protegido.
9. Propiedad: nombre de la función de la clase
10. Tipo: coloca el tipo de la función de la clase



Partes de una clase



- 1- Nombre de la clase
- 2- Nombre de la propiedad
- 3- Nombre de la función
- 4- Tipo de dato de la propiedad
- 5- Tipo de dato de la función

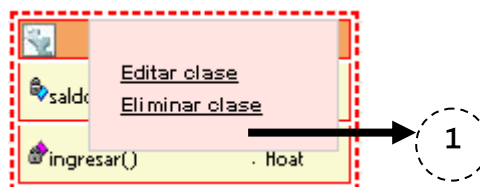
Los tipos de datos de las propiedades y funciones de una clase son los siguientes:

	Propiedad Privado		Función Privado
	Propiedad Protegido		Función Protegido
	Propiedad Publico		Función Publico

Como editar una clase

Una clase puede ser editada en su nombre, los atributos y tipos de sus propiedades y funciones.

- 1- Para editar una clase, solamente damos un clic derecho sobre ella y damos clic en Editar clase como se muestra en la figura



2- A continuación, se presenta la ventana en la cual podemos modificar el nombre, modificar, agregar o eliminar las propiedades y funciones de la clase.

The screenshot shows a window titled "Modificar clase" with a close button (X) in the top right corner. The window has a light green background. At the top, there is a text input field labeled "Nombre:" containing the text "Cuenta". Below this is a button labeled "Agregar propiedad".

Atributo	Propiedad	Tipo	
Privado	saldo	float	Eliminar

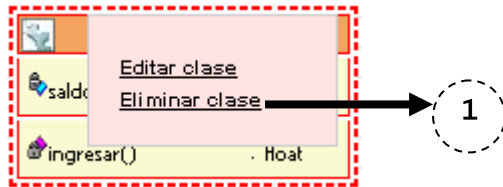
Below the table is a button labeled "Agregar funcion".

Atributo	Propiedad	Tipo	
Privado	ingresar()	float	Eliminar

At the bottom of the window is a button labeled "Actualizar". A black arrow points from this button to a dashed circle containing the number "2".

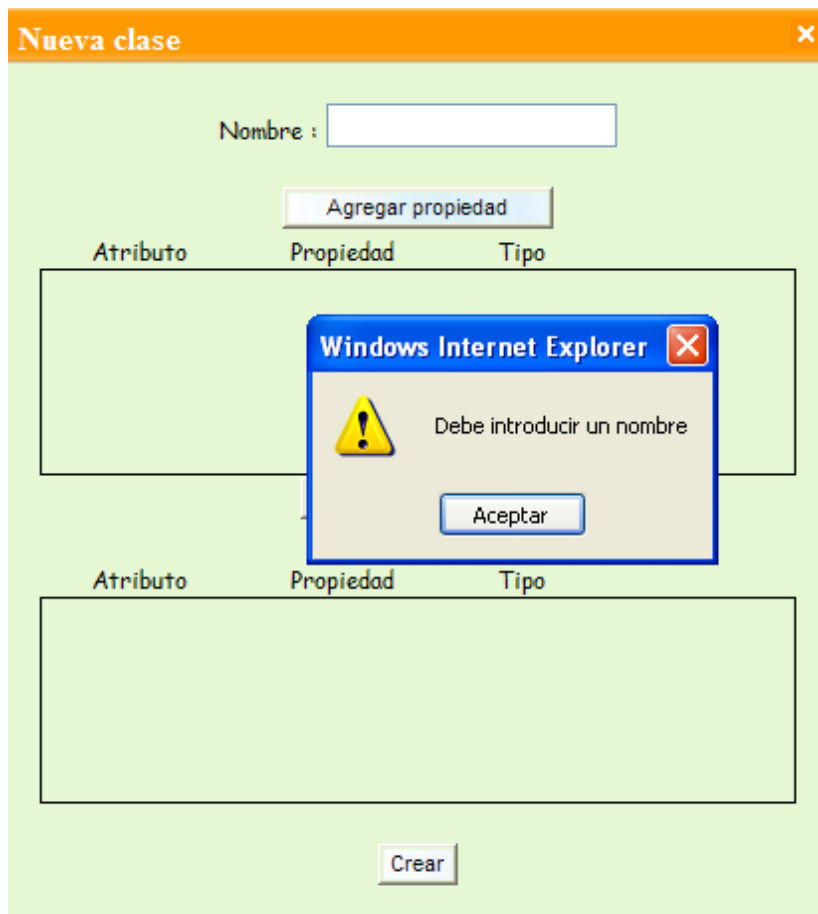
Como eliminar una clase

1- Para eliminar una clase, solamente damos un clic derecho sobre ella y seleccionamos Eliminar clase del menú contextual.

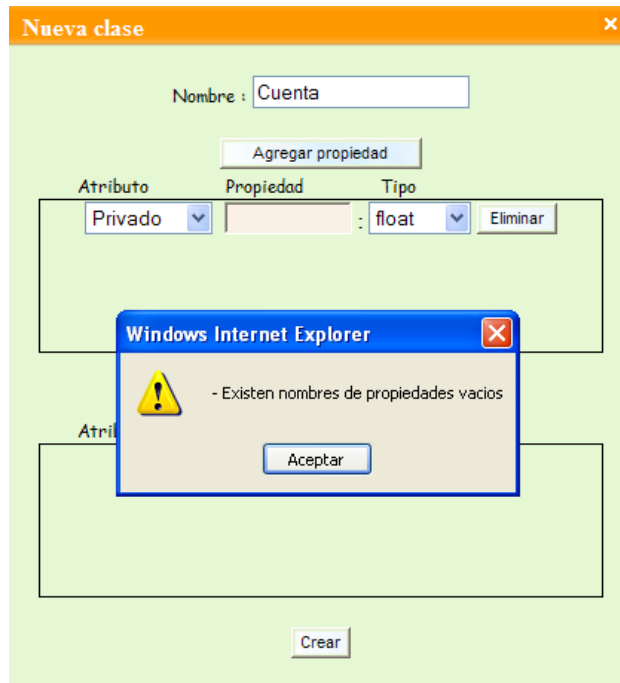


Mensajes de validación

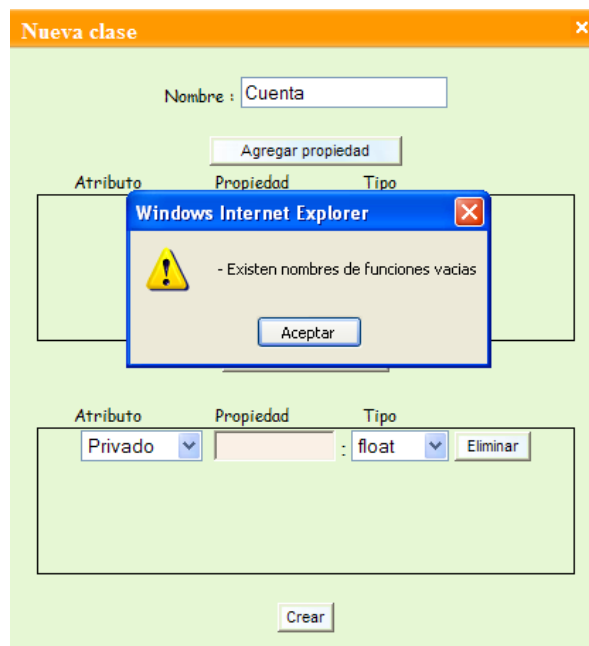
Cuando tratamos de crear una nueva clase sin especificarle un nombre, la aplicación genera el siguiente mensaje de validación que nos advierte del error generado, para solucionar se tiene que colocar por lo menos el nombre de la clase.



Cuando tratamos de crear una nueva clase sin especificarle el nombre de la propiedad, la aplicación genera el siguiente mensaje de validación que nos advierte del error generado, para solucionar se tiene que colocar el nombre de la propiedad.



Cuando tratamos de crear una nueva clase sin especificarle el nombre de la propiedad, la aplicación genera el siguiente mensaje de validación que nos advierte del error generado, para solucionar se tiene que colocar el nombre de la propiedad.



Como hacer un objeto

Para crear un objetos podemos:

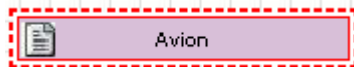
Seleccionar **Objeto** del barra de menú o dar clic derecho en el area de trabajo y seleccionar **Crear objeto** del menú contextual.

- 1- Colocamos un nombre al objeto
- 2- Damos clic en el boton de **Crear** y automáticamente se colocara la figura del objeto en el área de trabajo.



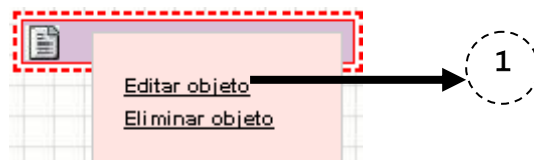
Partes de un objeto

Un objeto en los diagramas de objetos se representa por un rectangulo con el nombre del objeto dentro de el.



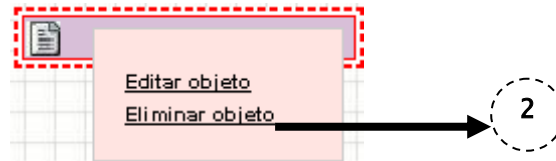
Como editar un objeto

1- En el caso de un objeto, este puede ser modificado solamente en su nombre. Para lo cual damos un clic derecho sobre el y seleccionamos **Editar objeto** del menú contextual.



Como eliminar un objeto

2-Al igual que las clases, un objeto puede ser eliminado del área de trabajo. Para esto damos un clic derecho sobre el objeto y seleccionamos **Eliminar objeto** del menú contextual.



Mensajes de validación

Al tratar de crear un nuevo objeto sin especificarle su nombre, la aplicación genera el siguiente mensaje de validación que nos advierte del error generado, para solucionarlo se tiene que colocar el nombre del objeto.



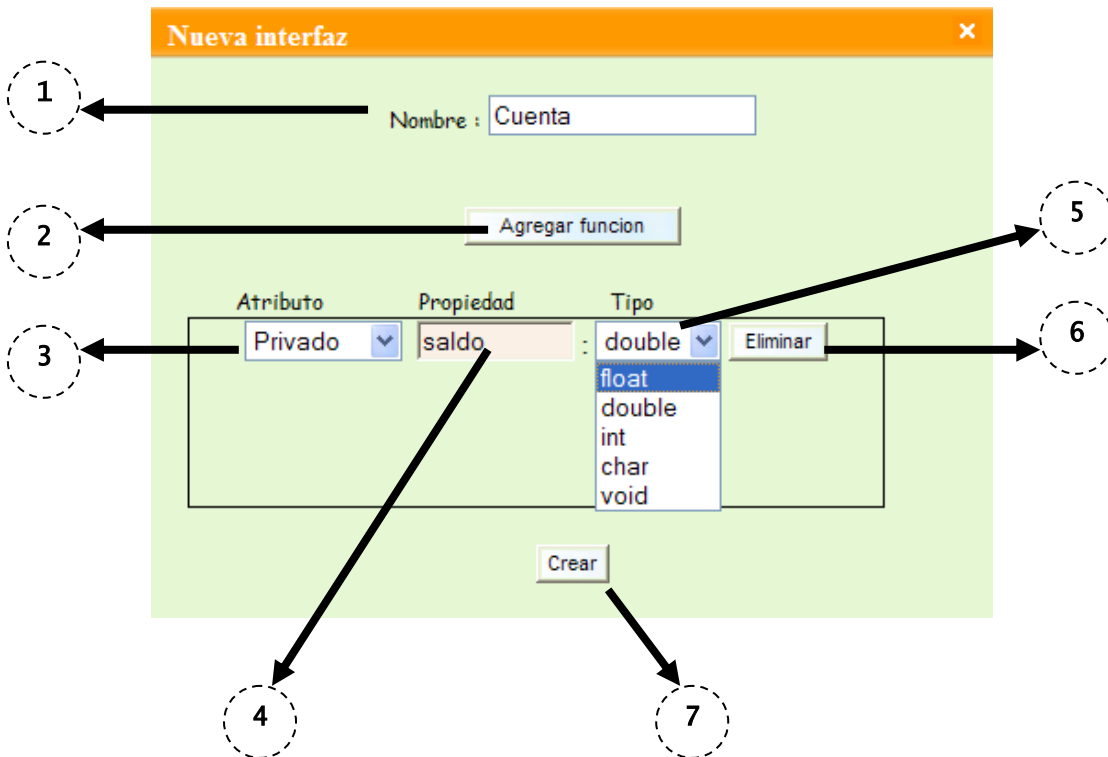
Como hacer una interfaz

Para crear una interfaz podemos seleccionar la figura de **Interfaz** del menú de componentes o dar clic derecho en el area de trabajo, seleccionamos **Crear interfaz** del menú contextual.

Partes de una interfaz

Las partes que conforman una interfaz son

- 1- Nombre de la interfaz
- 2- Botón Agregar función: permite agregar una o mas funciones a una interfaz
- 3- Atributo de la función: puede ser publico, privado o
- 4- Nombre de la función
- 5- Tipo de la función, esta puede ser double, flota, int, char o void.
- 6- Botón Eliminar, permite eliminar una función creada en la interfaz



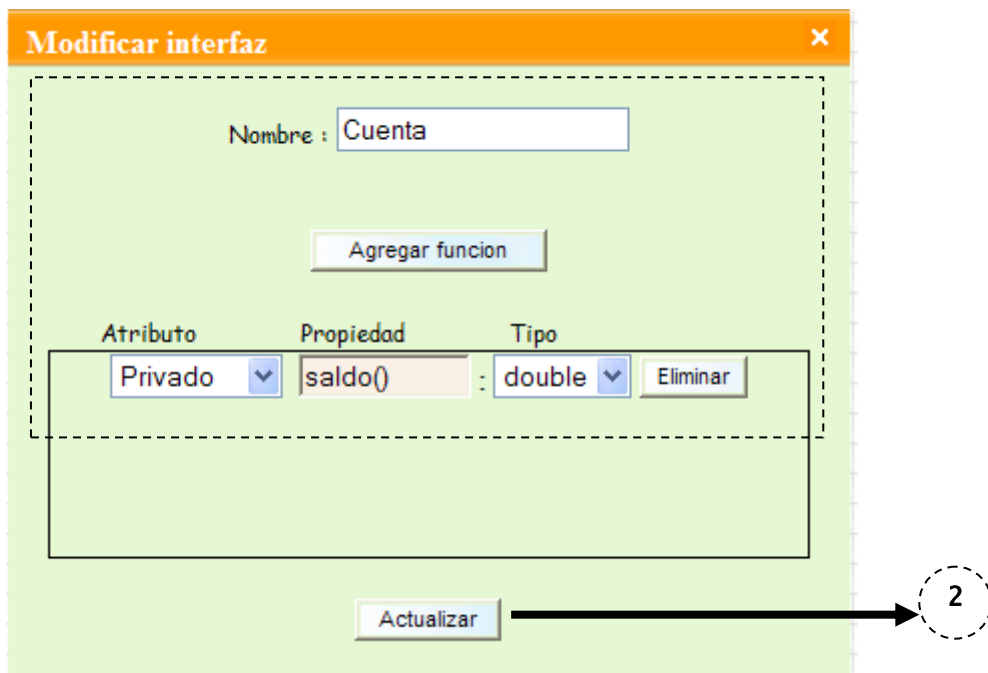
Como editar una interfaz

En el caso de una interfaz esta puede ser modificada en su nombre y en las funciones que posea.

1- Para eso damos clic derecho sobre la interfaz y seleccionamos Editar interfaz del menú contextual.



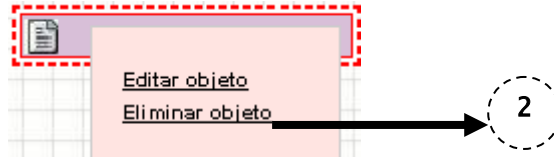
2- Se carga la ventana con los valores en ese momento en la configuración de la interfaz y verificamos los datos y damos clic en el botón de **Actualizar**



Como eliminar una interfaz

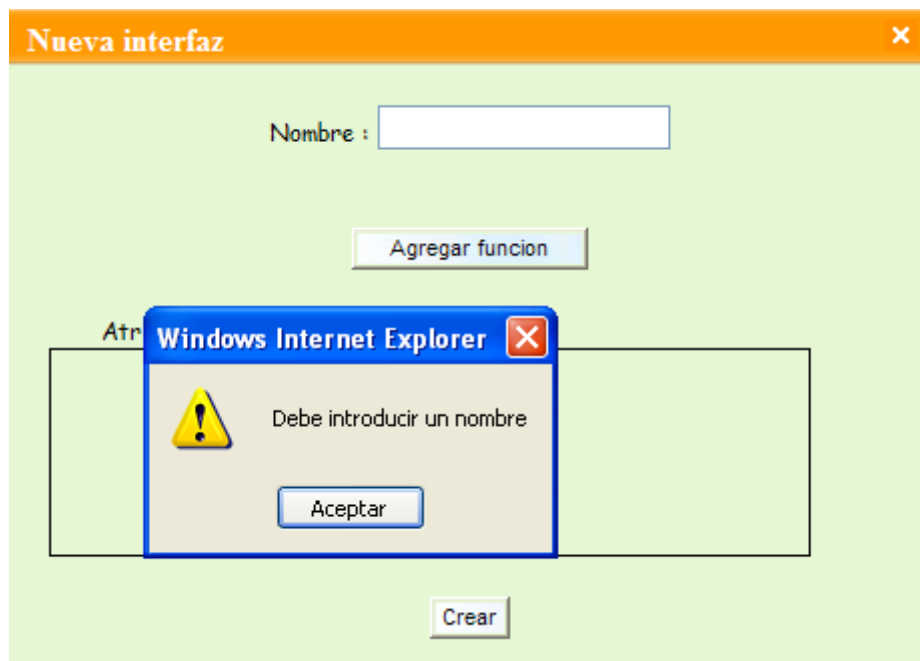
Como eliminar un objeto

2-Al igual que las clases, un objeto puede ser eliminado del área de trabajo. Para esto damos un clic derecho sobre el objeto y seleccionamos **Eliminar objeto** del menú contextual



Mensajes de validación

Al tratar de crear una nueva interfaz sin especificarle su nombre, la aplicación genera el siguiente mensaje de validación que nos advierte del error generado, para solucionarlo se tiene que colocar el nombre de la interfaz.



Como hacer un paquete

Un paquete puede contener una clase, un objeto, una Interfaz o un conjunto de clases, objetos y relaciones.

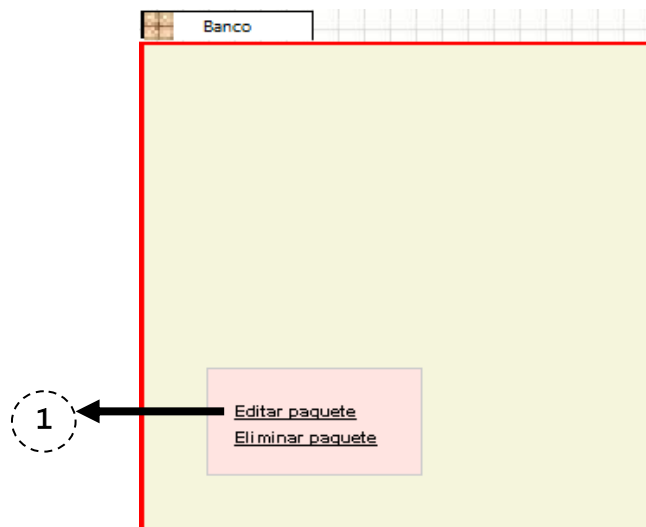
Para crear un paquete podemos seleccionar la figura Paquete del menú de componentes o clic derecho en el area de trabajo y seleccionamos Crear paquete del menú contextual.

- 1- Colocamos el nombre del paquete
- 2- Botón **Crear**, permite de la creación del paquete con el nombre que se coloco.



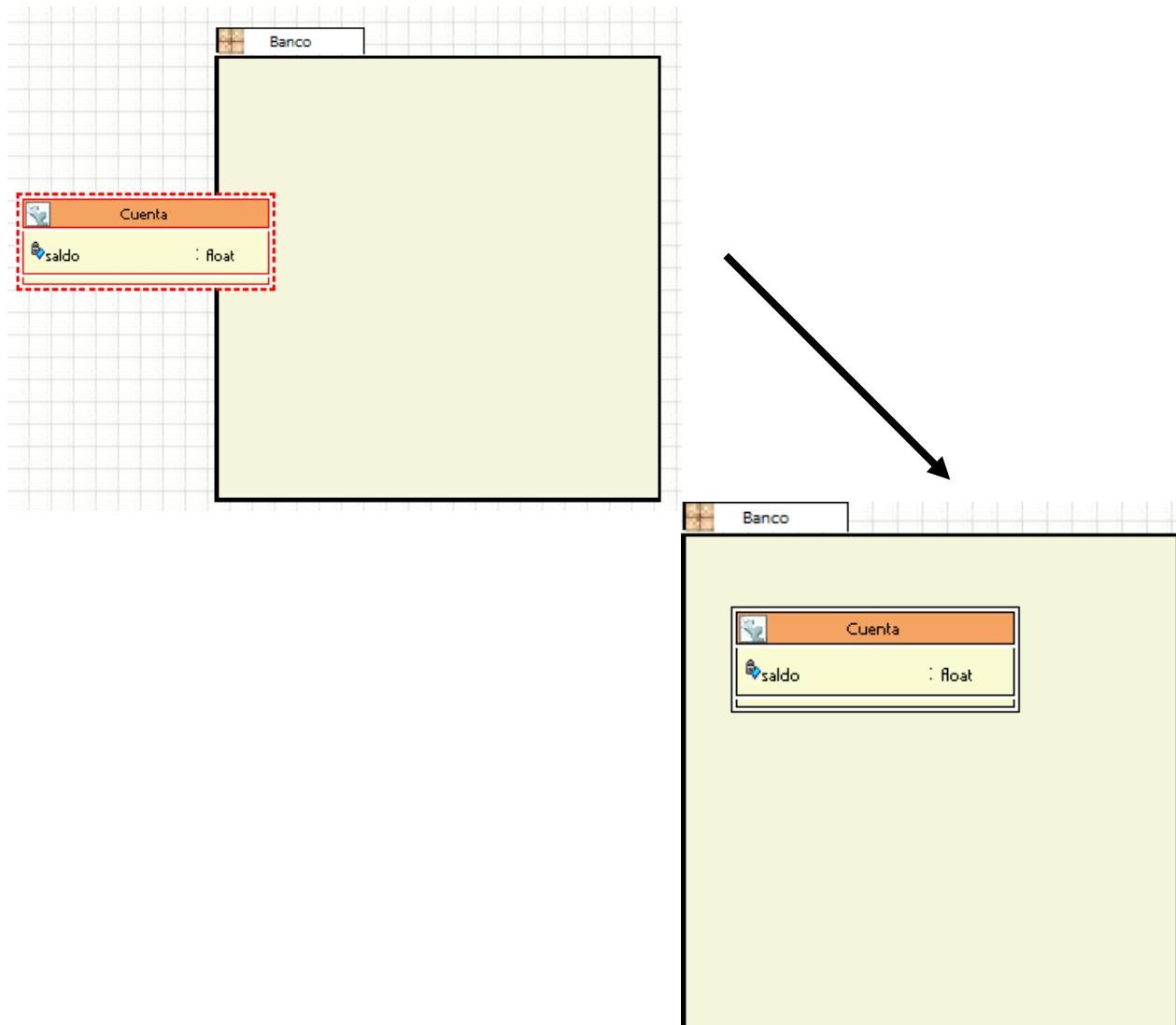
Como modificar un paquete

Para modificar una paquete damos clic derecho sobre el y seleccionamos **Editar paquete**.



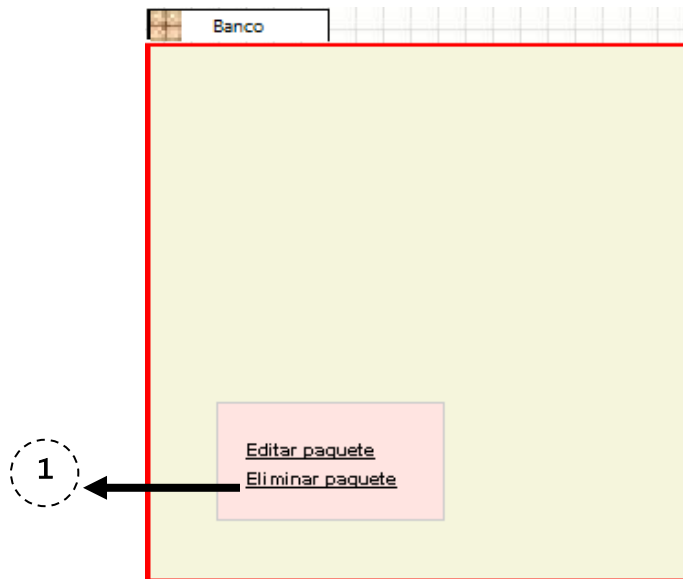
Como introducir una clase en un paquete

Para introducción una clase u objeto dentro de un paquete solamente arrastramos el componente dentro del paquete.



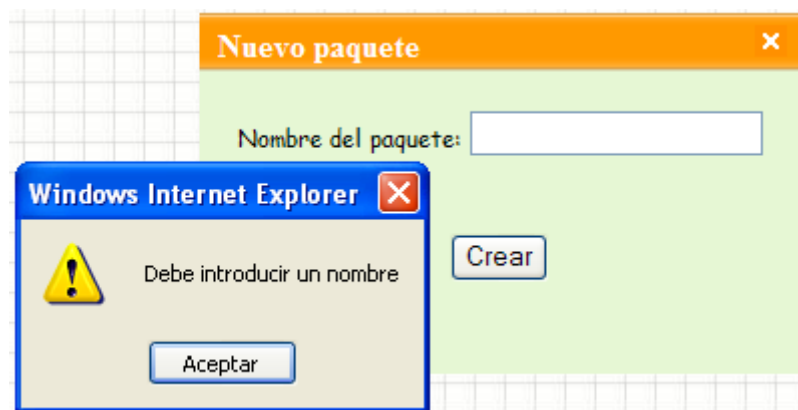
Como eliminar un paquete

Para eliminar una paquete, damos un clic derecho sobre el y seleccionamos **Eliminar paquete** del menú contextual.



Mensajes de validación

Al tratar de crear un nuevo paquete sin especificarle su nombre, la aplicación genera el siguiente mensaje de validación que nos advierte del error generado, para solucionarlo se tiene que colocar el nombre del paquete.

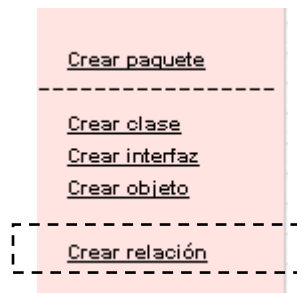


Como hacer una relación entre dos clases

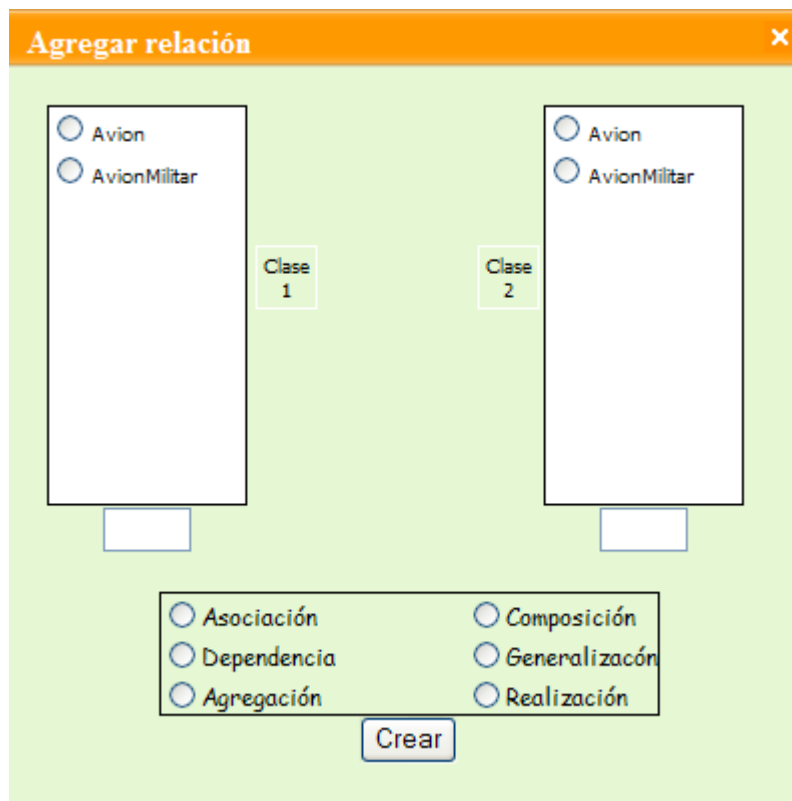
Cuando tenemos dos clases u objetos o mas, podemos hacer una relación entre ellos.



Para crear una relación, seleccionamos crear relación del menú contextual



Luego, seleccionamos las clase que se van a relacionar y el tipo de relación entre ellas.



Como asignar multiplicidad a una relación

Para colocar la multiplicidad de una relación, lo hacemos por medio de las cajas de texto, cuando estamos creando la relación.

