



CENTRO UNIVERSITARIO DE SANCTI SPÍRITUS

“JOSÉ MARTÍ PÉREZ”

FACULTAD DE INGENIERÍA

**TESIS EN OPCIÓN AL TÍTULO DE MASTER EN CIENCIAS EN
NUEVAS TÉCNOLOGÍAS PARA LA EDUCACIÓN**

*Título: Simulador para el aprendizaje del
proceso de backtracking.*

Autor: Lic. Aldo Martínez Riverol

Tutor: MSc. Lydia Rosa Ríos Rodríguez

Consultante: MSc. Geysell Guevara Fernández

Sancti Spiritus, 2007

“Año 49 de la Revolución”

*“No desesperes, ni siquiera por el hecho de que no desesperas.
Cuando todo parece terminado, surgen nuevas fuerzas. Eso
significa que vives”.*

Franz Kafka

A mis padres, que se merecen todo y mucho más.

.....Aldo.....

A mis padres....

A mi hermano Alain, a su esposa Yuliet y su hijo, mi sobri querido, Ariel.

A Yaneysi por su ayuda incondicional y su amor inmenso.

A Lidia por su gran ayuda y por ser mi tutora.

A mis compañeros de trabajo y de maestría Sixto y Riqui.

A todos los profesores que impartieron y colaboraron con la maestría.

*A todos los profesores de la Facultad de Ingeniería del
CUSS.*

Y me disculpan si se me queda alguien sin mencionar, les agradezco a todos, gracias por su ayuda.

.....Aldo.....

Resumen

En la presente investigación se realiza la implementación de un simulador para el proceso de Backtracking de la asignatura de programación lógica que se imparte en 4to. Año de la carrera de Ciencias de la Computación en la Universidad Central “Martha Abreu” de las Villas.

Para el diseño del simulador se utiliza el lenguaje UML, y se implementa utilizando tecnología Java específicamente Applet, con el objetivo de satisfacer las necesidades existentes en cuanto a dificultades de aprendizaje de este proceso en los estudiantes.

En el proceso de investigación está implicado un conjunto de métodos teóricos y empíricos. Además el simulador se valida a través del criterio de expertos, los que avalan la pertinencia y la factibilidad del mismo.

Se consideran como palabras claves: PROLOG, Backtracking, JAVA, APPLETS, UML.

Índice

Introducción	7
Capítulo 1. Consideraciones teóricas del uso de las Nuevas Tecnologías de las Informáticas y las Comunicaciones para el desarrollo de simuladores.....	13
1.1 A QUE LLAMAMOS LAS NUEVAS TECNOLOGÍAS DE LAS INFORMÁTICAS Y LAS COMUNICACIONES. (NTIC)	13
1.2 LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES EN LA EDUCACIÓN.	14
1.3 LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES EN LA EDUCACIÓN CUBANA.....	15
1.4 SIMULADORES.....	18
1.5 LOS SIMULADORES EN EL PROCESO DOCENTE.....	19
1.5.1 <i>Simulador del proceso de backtracking de PROLOG visto desde el Enfoque Histórico-Cultural.</i>	21
1.7 INTELIGENCIA ARTIFICIAL.	25
1.8 PROLOG, LENGUAJE DECLARATIVO.....	27
1.9 BACKTRACKING EN PROLOG.	29
1.10 TEACHING.....	30
Capítulo 2. Diseño de la propuesta de solución.....	32
2.1 DIAGNÓSTICO INICIAL DEL PROBLEMA PARA DESARROLLAR EL DISEÑO DE LA PROPUESTA DE SOLUCIÓN.	32
2.1.1 <i>Análisis de la Encuesta aplicada a los estudiantes,</i>	33
2.1.2 <i>Valoración de la Entrevistas realizadas a profesores que han impartido la asignatura.</i>	39
2.2 METODOLOGÍA Y TECNOLOGÍAS PARA EL DESARROLLO DEL SIMULADOR DE BACKTRACKING.....	41
2.2.1 <i>Uso del Software Libre.</i>	41
2.2.2 <i>Tecnología Java</i>	42
2.2.3 JAVA 2 STANDARD EDITION 5.0 (J2SE)	45
2.2.4 NETBEANS COMO IDE.....	46
2.2.5 APPLETS.....	47
2.2.6 W-PROLOG.....	48
2.3 DISEÑO Y MODELACIÓN UTILIZANDO LENGUAJE UML.....	49
2.3.1 <i>Descripción del Negocio y requerimientos del software.</i>	49
2.3.1.1 <i>Diagrama de Casos de Uso del Negocio</i>	49
2.3.1.2 <i>Diagrama de Actividad de Casos de Uso del Negocio</i>	50
2.3.1.3 <i>Diagrama de Clases del Modelo de Objeto del Negocio</i>	51
2.3.1.4 <i>Especificación de los requisitos del software.</i>	51
2.3.2 <i>Diagrama de Casos de Uso del Sistema</i>	53
2.3.3 <i>Diagrama de Interacción</i>	54
2.3.4 <i>Diagrama de Navegación</i>	56
2.3.5 <i>Vistas del Sistema</i>	57
2.4 VALORACIÓN DE LA FACTIBILIDAD DEL DISEÑO DEL SIMULADOR POR EL CRITERIO DE EXPERTOS.	59
Conclusiones.....	62
Recomendaciones	63
Referencias bibliográficas.....	64
Bibliografía	66
Anexos	69

Introducción

En Cuba la educación ha sido pilar de la Revolución desde sus inicios, por ello durante todos estos años el sistema educacional se ha enriquecido de tecnologías y capital humano con el objetivo de fortalecer su estructura y calidad de los graduados.

Los cambios extraordinarios se han ido operando en el mundo en tres campos que condicionan esta nueva era, es decir, en la computación, la información y las comunicaciones que han dado lugar al surgimiento de las Nuevas Tecnologías de Información y las Comunicaciones (NTIC). El país no ha quedado fuera de estos cambios y ha tenido que ir tomando medidas para no quedarse rezagado en este proceso tecnológico que avanza a velocidades vertiginosas, o sea, se encuentra en un constante perfeccionamiento.

El reto actual de la educación en nuestro país consiste no solo en hacer más eficiente el aprendizaje sino también en obtener egresados capaces de organizar su propio aprendizaje e interactuar con la información y el conocimiento de forma exitosa. Esto ha provocado cambios en los instrumentos de transmisión del conocimiento y la interpretación de las representaciones visuales es una necesidad de las actuales y futuras generaciones.

Hoy se ha extendido la educación superior por toda Cuba y ya cada provincia cuenta con un Centro Universitario o Universidad. En todas estas instituciones, de altos estudios, se puede encontrar una carrera de perfil informático ya sea Ingeniería Informática o Ciencias de la Computación. En estas carreras una de las disciplinas más difíciles es la Inteligencia Artificial por su abstracción y complejidad. Esta le permite al estudiante abordar problemas que dependen fuertemente de la experiencia y presentan elementos de incertidumbre y riesgo, y a la vez, que sean adecuados para su solución por métodos deductivos, inductivos u otros procesos cognitivos, y que son de difícil solución descriptivamente o de gran complejidad.

Dentro de esta disciplina se encuentra la asignatura de PROLOG que no es más que programación lógica. Una asignatura que a los estudiantes se les

dificulta su comprensión ya que PROLOG se enmarca en el paradigma de los lenguajes declarativos, lo que lo diferencia enormemente de otros lenguajes más populares tales como Fortran, Pascal, C++, etc. ,es decir, es una forma de programación a la cual ellos no se han enfrentado.

En búsquedas en Internet y en la propia Universidad Central “Marta Abreu” de las Villas se a podido comprobar que sobre este tema existe bibliografía actualizada, investigaciones, tutoriales, grupos dedicados a la investigación como es el caso del Grupo de Inteligencia Artificial de la UCLV. Pero resulta difícil encontrar herramientas informáticas que ayuden a los estudiantes o personas interesadas en el tema de la Programación Lógica a adquirir sus propios conocimientos. Entre los instrumentos para el aprendizaje visual se encuentran los simuladores, estos no solo transmiten información básica sino que permiten ver, estudiar y llegar a entender las características de un fenómeno o aprender que decisiones debe tomar en diferentes circunstancias.

En la UCLV se realizó un trabajo diploma por dos estudiantes y como tutor el profesor Dr. Mateo Lezcano Brito del cual se obtuvo como producto final el software educativo “Teaching”, el cual se estudiará en el desarrollo del presente trabajo, y por sus características no se ajusta completamente a los objetivos que se traza en la presente investigación. También la profesora Msc. Lydia Rosa Ríos Rodríguez, del Centro Universitario “José Martí Pérez” de Sancti Spíritus, ha abordado esta temática para su investigación de doctorado y no ha podido acceder a ninguna herramienta por lo que se ha dado a la tarea de crear una de la cual esta investigación podrá formar parte.

Nos referimos concretamente a que los estudiantes y profesores no han podido contar con un software que simule los procesos que se llevan a cabo en este tipo de programación, específicamente en el proceso de backtracking. Este simulador constituirá sin lugar a dudas un software educativo que contribuirá al desarrollo y mejor comprensión de los conocimientos que necesitan los estudiantes en esta asignatura.

En consecuencia el **problema** de la investigación radica en ¿Cómo elevar los niveles de asimilación del proceso de backtracking en la Programación Lógica en los estudiantes de 4to año de Ciencias de la Computación en la UCLV?

El **objeto** de investigación es el proceso de enseñanza-aprendizaje de la Programación Lógica en la carrera de Licenciatura en Ciencias de la Computación.

Su campo de acción **son los niveles de asimilación del proceso de backtracking en la Programación Lógica.**

El **objetivo general** es implementar un simulador que permita elevar los niveles de asimilación del proceso de backtracking en la Programación Lógica en los estudiantes de 4to año de Ciencias de la Computación en la UCLV.

Como guía para realizar la investigación nos proponemos las siguientes Preguntas Científicas:

1. ¿Cuáles serán los presupuestos teóricos metodológicos que justifican a los simuladores como herramientas que apoyan el proceso de enseñanza-aprendizaje?
2. ¿Cómo se desarrolla actualmente el proceso de enseñanza-aprendizaje del proceso de backtracking en la asignatura PROLOG de la carrera de Lic. en Ciencias de la Computación?
3. ¿Cómo diseñar el simulador para apoyar el aprendizaje del proceso de backtracking en la asignatura de PROLOG?
4. ¿Cómo implementar el simulador para apoyar el aprendizaje del proceso de backtracking en la asignatura PROLOG?
5. ¿Cómo validar el producto final de la investigación?

Al dar respuesta a las interrogantes anteriores surgen las siguientes Tareas Científicas:

1. Determinación de los presupuestos teóricos-metodológicos para la construcción de simuladores para el aprendizaje de PROLOG.
2. Diagnóstico del proceso de enseñanza-aprendizaje del proceso de backtracking en la asignatura PROLOG de la carrera de Lic. en Ciencias de la Computación.
3. Diseñar un simulador para apoyar el aprendizaje del proceso de backtracking en la asignatura PROLOG.
4. Implementar un simulador para apoyar el aprendizaje del proceso de backtracking en la asignatura PROLOG.
5. Valorar los beneficios del simulador una vez terminado por el criterio de experto.

Para la investigación se utilizaron métodos y técnicas de la investigación educacional. Dentro de los métodos teóricos se aplicó el **análisis-síntesis**, la **inducción-deducción** y la **modelación** con el objetivo de la interpretación de la información empírica obtenida y el establecimiento de generalizaciones y regularidades. Dentro de los métodos empíricos se utilizó **observación** con el objetivo de ver como se desarrolla el proceso docente de la asignatura programación lógica; la **revisión de documentos** correspondientes al proceso docente educativo servirá de base para establecer criterios evaluativos con respecto al simulador; la **encuesta**, la cual se aplicó a la muestra seleccionada para diagnosticar el estado real del proceso de enseñanza-aprendizaje de la asignatura Programación Lógica, específicamente el proceso de backtracking; la **entrevista** se utilizó con 2 objetivos:

- Se realizó a profesores que impartieron la asignatura para hacer una caracterización de la situación de la enseñanza del tema de backtracking en la asignatura de Programación Lógica.

- Se realizó al principal cliente del software para conocer los requerimientos y principales características a tener en cuenta en la implementación del simulador de backtracking.

Además se utilizó el **criterio de expertos** con el fin de analizar la pertinencia, confiabilidad y grado de efectividad de la propuesta realizada y dejar además señaladas sus posibles debilidades o factores que dificulten su aplicación.

De los métodos de **nivel estadísticos** se utilizaron la estadística descriptiva y el cálculo porcentual se utilizó en el procesamiento de los resultados de entrevistas y encuestas permitiendo la cuantificación y el procesamiento de los datos para su interpretación.

La selección de la población para la aplicación de las encuestas correspondió a los 36 estudiantes de 4to año de la carrera Licenciatura en Ciencias de la Computación, de la facultad de Matemática, Física y Computación, de la Universidad Central "Martha Abreu" de las Villas que reciben actualmente la asignatura Programación Lógica, haciéndose coincidir con la muestra, lo cual coincide con el 100% de la población. La muestra se escogió de forma intencional ya que la población no es grande y además nos arroja resultados más confiables estadísticamente.

La novedad científica de esta investigación es la creación de un simulador para el proceso de backtracking que apoye el proceso de enseñanza aprendizaje de la Programación Lógica, ya que hasta el momento no se encontró otro antecedente en Cuba para ser usado en las universidades.

Su significación práctica está dada por el simulador obtenido, libre de patentes, realizado bajo la licencia de software libre, que facilite el aprendizaje de la Programación Lógica, además de la posibilidad de accederlo desde la Web o de forma local en las computadoras. El software será fácil de incluir en cualquier sitio, página o cualquier otra herramienta basada en tecnología web. Además facilita el uso de nuevas tecnologías que permiten la enseñanza de esta disciplina a distancia.

La actualidad del tema de esta investigación es la creación de un simulador para el proceso de backtracking de PROLOG para apoyar el proceso de enseñanza aprendizaje de la Programación Lógica, asignatura que se imparte actualmente en las universidades del país y que tiene un amplio campo de investigación y aplicación práctica. Además el software esta implementado utilizando tecnología Java, muy usada por la comunidad de programadores de software libre en el mundo y que en nuestro país se ha comenzado a introducir en los planes de estudio de las carreras de perfil informático.

La investigación está conformada por una introducción, dos capítulos, conclusiones, recomendaciones, bibliografía y anexos. La introducción expresa los aspectos más generales de la investigación. En el primer capítulo se hacen algunas consideraciones teóricas del uso del las TICs para el desarrollo de simuladores.

En el segundo capítulo se presenta el diseño de la propuesta de solución, donde se hace un análisis del estado real del problema, se describe la tecnología a utilizar, las principales herramientas, los requisitos del sistema y el diseño UML. Por ultimo se exponen la validación del producto utilizado el criterio de expertos.

Capítulo 1. Consideraciones teóricas del uso de las Nuevas Tecnologías de las Informáticas y las Comunicaciones para el desarrollo de simuladores.

1.1 A que llamamos las Nuevas Tecnologías de las Informáticas y las comunicaciones. (NTIC)

Resulta innegable el auge cada vez mayor de las NTIC en las diferentes esferas de la sociedad a escala mundial. El desarrollo de la ciencia y la tecnología ha llevado a la sociedad a entrar al nuevo milenio inmerso en lo que se ha dado en llamar "era de la información" e incluso se habla de que formamos partes de la "sociedad de la información". Sin lugar a dudas, estamos en presencia de una revolución tecnológica de alcance insospechado.

¿Qué son las NTIC? La UNESCO, define a las NTIC como "el conjunto de disciplinas científicas, tecnológicas, de ingeniería y de técnicas de gestión utilizadas en el manejo y procesamiento de la información, sus aplicaciones; las computadoras y su interacción con hombres y máquinas; y los contenidos asociados de carácter social, económico y cultural".[1]

Una forma más sencilla de definir las NTIC sería como un conjunto de equipos, redes, servicios que se integran, en un sistema de información interconectado y complementario. Estas NTIC conforman un sistema integrado por:

Las telecomunicaciones: Representadas por los satélites destinados a la transmisión de señales telefónicas, telegráficas y televisivas; y por la fibra óptica, nuevo conductor de la información que entre sus múltiples ventajas económicas se distingue en transmitir la señal a grandes distancias sin necesidad de usar repetidores y tener ancho de banda muy amplio.

La informática: Se caracteriza por notables avances de hardware y software que permiten producir, transmitir, manipular y almacenar la información con más efectividad, distinguiéndose, las redes locales y globales (INTERNET), los servicios de mensajería electrónica, etc.

La denominación de "Nueva" ha provocado discusiones y criterios opuestos, al punto que muchos especialistas han optado por nombrar simplemente **Tecnologías de la información y las comunicaciones (TIC)**. Esto se debe a que muchas de ellas son realmente antiguas, un ejemplo de ello es el teléfono que data de 1876. El término "Nueva" se le asocia a estas tecnologías fundamentalmente porque en todas ellas se distinguen transformaciones que erradican las deficiencias de sus antecesoras y por su integración como técnicas interconectadas.

1.2 Las Tecnologías de la información y las Comunicaciones en la Educación.

El impacto social de las TICs está reflejado en escuelas y universidades, propiciando modificaciones en las formas tradicionales de enseñar y aprender. Es evidente que el avance y perfeccionamiento de la tecnología no parece tener freno, el reto de los centros educacionales y en particular de las universidades radica en prepararse como institución y preparar a sus profesores para adaptarse a los cambios de manera rápida y efectiva. Entre las claves fundamentales para el éxito está lograr que el aprendizaje se convierta en un proceso natural y permanente para estudiantes y docentes. Es necesario aprender a usar las nuevas tecnologías y usar las nuevas tecnologías para aprender.

En el Congreso Internacional "La Educación y la Informática: Hacia una mayor cooperación internacional" se llegó a la conclusión de que las tecnologías de la información son ya un instrumento útil para mejorar la eficacia interna y externa de los sistemas educativos; su introducción corre el riesgo de aumentar más aún las disparidades ya existentes en la educación en los países y entre los países a menos que se adopten de inmediato medidas correctivas para impedir esta evolución y el ritmo de introducción, las NTIC deberían determinarse en función de un análisis de las necesidades de cada país, teniendo en cuenta el contexto social, económico y cultural.[2]

Es tarea de los educadores utilizar las TICs como medios para proporcionar la formación general y la preparación de sus estudiantes,

contribuyendo a la formación más desarrollada e integral del individuo. Si se tiene en cuenta que la nueva tecnología no garantiza por sí sola el éxito pedagógico, se puede diseñar con mucho cuidado el software educativo donde será utilizada.

Una de las mayores dificultades a vencer para la introducción y la utilización eficiente de las TICs en la educación radica en que esta última es, por lo general, resistente a los cambios, así como poco ágil y efectiva a la hora de adaptarse y enfrentar los nuevos retos. Sin embargo, quizás por primera vez y por su poderoso carácter social; las nuevas tecnologías comienzan a introducirse en la educación, casi al mismo tiempo que lo hacen en otras esferas de la sociedad. Es necesario que se gane conciencia de que el empleo de estos nuevos medios impondrán marcadas transformaciones en el proceso docente, con cambios en los roles que han venido desempeñando estudiantes y profesores. Nuevas tareas y responsabilidades esperan a estos, entre otras, los primeros tendrán que estar mas preparados para la toma de decisiones y la regulación de su aprendizaje y los segundos para diseñar nuevos entornos de aprendizaje y servir de tutor, al pasarse de un modelo unidireccional de formación donde él es el portador fundamental de los conocimientos, a otro más abierto y flexible en donde la información se encuentra en grandes bases de datos compartidos para todos.

1.3 Las Tecnologías de la información y las Comunicaciones en la Educación Cubana.

El desarrollo de la Informática Educativa en Cuba, la utilización de la computación en la enseñanza, en las investigaciones científicas, y en la gestión docente, ha constituido un objetivo priorizado de la Política Nacional Informática desde los primeros años de la Revolución, por lo que el sistema de conocimientos y habilidades que deben tener los integrantes de la sociedad actual y futura, conlleva a una estrategia que tendrá que estar muy ligada a la informática y las nuevas tecnologías. [3]

El papel clave de la ciencia y la tecnología en el desarrollo de cualquier actividad humana se hace cada día más evidente. Frente a los gravísimos retos que enfrenta el mundo en el nuevo milenio, la ciencia y la tecnología

constituyen una herramienta esencial; aún más, para un país, la vitalidad en esta esfera es una cuestión de su propia supervivencia. El desarrollo dependerá de manera creciente del conocimiento, y de la capacidad de producirlo, transferirlo, transformarlo y aplicarlo. [4]

Vamos a ser fortísimos en la Computación (Informática), como ya lo estamos siendo en la Medicina, y no sólo para beneficio de nuestro pueblo, sino de toda la Humanidad. Será también esta una poderosísima fuerza científica, económica y política del país en toda nuestra lucha por desarrollarnos. [5]

En el país existen 600 Joven Club de Computación y Electrónica con la misión de enseñar a nuestros jóvenes a utilizar correctamente esta autopista informática, pues la misma es una realidad innegable que debemos dominar no solo para obtener información, sino porque es una herramienta eficaz para transmitir nuestra verdad, la realidad de la Revolución cubana. [5]

La presencia de computadoras en todas las escuelas de nuestro país, se ha convertido en una realidad objetiva a partir del esfuerzo que realiza nuestro Estado socialista para garantizar laboratorios con computadoras en todas las enseñanzas, la cual comienza desde edades tempranas.

La informatización paulatina del proceso de enseñanza requiere que el personal docente perfeccione los métodos de enseñanza, una base material adecuada y la creación de medios de enseñanza que permitan desarrollar en nuestros estudiantes un mayor interés por el estudio y la superación constante, lo que traería como resultado una preparación íntegra de los mismos, y por tanto, que puedan responder a los requerimientos del progreso social.

Para perfeccionar la educación cubana, hoy más que nunca, se necesita de una concepción científica y metodológica propia, acerca del modo de educar a las nuevas generaciones, la cual esté acorde con las más avanzadas de las ciencias en el mundo, con las mejores tradiciones de nuestra historia y de nuestra cultura, y sobre todo, con las posibilidades exigencias y proyecciones de nuestra sociedad.

Por eso, ante la pedagogía cubana se abren nuevos retos, nuevas perspectivas, nuevas investigaciones, nuevas posibilidades y nuevos métodos de educación que posibilitarán no sólo enriquecer la pedagogía, sino también, lograr el propósito de que nuestro pueblo sea el más culto y preparado del mundo, y además transmitir nuestros conocimientos hacia los hermanos países.

Con la introducción en la escuela cubana de las NTIC a través del uso de la computación, redes, la televisión y el video, se abre ante la educación cubana un caudal de posibilidades para enriquecer la Pedagogía, pues con anterioridad no se contaba de manera masiva con estos medios y por tanto, no hay suficientes experiencias, investigaciones y trabajos relacionados con el uso de estas técnicas en el proceso docente.

Se han creado instalaciones especializadas en la enseñanza de las ciencias de la computación como son la Universidad de las Ciencias Informáticas, que conduce al diseño e implementación de diversos software educativos, y los politécnicos de esa disciplina en los que estudian alrededor de 40 000 alumnos. También la apertura de la carrera de Ingeniería Informática en todos los Centros Universitarios y Universidades del país.

A partir de la ampliación de los laboratorios de computación en todas las escuela y Universidades del país, incluyendo las primarias y círculos infantiles, ha permitido el acceso a esta nueva tecnología de todos los alumnos y profesores del sistema de educación, lo cual permite que cada día se incorporen una gran cantidad de educadores utilicen la computadora como medio de enseñanza y herramienta de trabajo.

En estos momentos, además del uso de los medios tradicionales, los profesores tienen la posibilidad de utilizar las computadoras en el desarrollo del proceso docente educativo a través del empleo de una gran cantidad de Software educativos existentes en el mundo. En nuestro país se han creado muchos de estos Software, principalmente para la enseñanza primaria y secundaria, para ser utilizados por parte de los profesores, lo cual propicia un beneficio en el

aprendizaje de los alumnos, así como en la calidad de la educación.

En nuestras escuelas y universidades, los profesores han comenzado a cambiar los métodos de enseñanza, o sea, a partir de un mayor y eficiente uso de las NTIC hacer más eficiente el proceso de enseñanza-aprendizaje. Lo anterior va permitiendo que los educadores no sólo utilicen en sus clases sus viejos medios, como la tiza y el borrador, sino que también incorporen programas de computación o software educativos a los que se le han incorporados las técnicas de multimedia, la televisión, el video, el correo electrónico, los cursos a distancias, Internet, etc.

1.4 Simuladores

La Real Academia define simulador como: aparato que reproduce el comportamiento de un sistema en determinadas condiciones, aplicado generalmente para el entrenamiento de quienes deben manejar dicho sistema. [6]

Es decir los simuladores reproducen en forma simplificada y a través de un modelo, una situación o fenómeno ya sea físico, químico, matemático y social, donde le permite al individuo explorar y actuar sobre dichos eventos, resolver problemas, aprender procedimientos, llegar a entender las características de un fenómeno o aprender que decisiones debe tomar en diferentes circunstancias.

El uso de simuladores computarizados data de la segunda mitad del siglo pasado. El motor intelectual de su uso se asigna a la contribución de John Dewey en su obra "Educación and Experience" en donde argumentaba en contra del exceso de teoría. [7]

Hoy en día la simulación se utiliza en la industria automotriz, en los programas del espacio, en la aviación, en problemas estadísticos, matemáticos, físicos, etc., en juego por computadoras, en la medicina, en fin cualquier proceso se puede llevar a un simulador.

Este uso de los simuladores esta mayormente dado por la posibilidad de repetir, en condiciones idénticas y a partir de su modelación, procesos y fenómenos, algo

difícil de lograr en condiciones reales, y por tanto, estudiar sistemáticamente sus comportamientos hasta lograr los objetivos deseados. También elimina los riesgos que siempre se presentan en la interacción con la realidad, tanto para dispositivos, instrumentos, etc...

1.5 Los Simuladores en el proceso docente.

Todos los simuladores pueden tomarse como herramientas educativas, simulan situaciones que se pueden presentar en la vida real o procesos que le interese al individuo; este puede analizar los resultados o respuestas provenientes del proceso de simulación y puede ser capaz de sacar sus propias conclusiones y extraer el conocimiento.

Podemos referirnos más específicamente a los **Simuladores Educativos (SE)**, como los que intervienen en el proceso docente. Estos ponen al alumno en la necesidad de opinar, de implicarse, de incorporar un rol en una situación verídica, de escoger sus propias opciones. Se le dan detalles y se le proponen alternativas de actuación, lo que se crea confianza en ellos para implicarse en el estudio de esa realidad.

Los SE están dentro de un conjunto de software de apoyo curricular, generalmente denominados softwares educativos que buscan reforzar, complementar o servir de material pedagógico en diversas asignaturas.

El uso de los SE le brinda al profesor una posibilidad más de investigación en su quehacer pedagógico, pues ahora debe adaptar el proceso, a las nuevas condiciones y evitar el tradicionalismo en las clases, aspecto que le hace daño al desarrollo de la Pedagogía como ciencia. Ahora los educadores tienen la posibilidad de combinar diferentes tipos de métodos y medios con el uso de las nuevas tecnologías, de manera tal que le permita desarrollar y perfeccionar el proceso docente educativo.

Estos permiten la realimentación inmediata, pues los efectos que se logran en el funcionamiento del sistema, fenómeno o proceso que se simula, como resultado de introducir modificaciones en determinados parámetros, resultan inmediatos; lo

que permite corregir la actuación del estudiante en cada momento. Se considera el uso del simulador como una actividad en la que el estudiante no acumula información teórica, sino que la lleva a la práctica, con lo cual esta se identifica con el entrenamiento puramente, y también el alumno gana habilidad en el proceso.

La simulación tiene 2 grandes usos en el proceso docente:

- Durante la enseñanza-aprendizaje.
- En la evaluación.

El Simulador Educativo tiene que estar en estrecha correspondencia con las exigencias y requerimientos del currículo y que el estudiante tiene que sentir la necesidad y la utilidad de su uso de manera independiente.

El uso de los simuladores educativos tiene sus ventajas pero también tiene desventajas.

Dentro de las ventajas tenemos:

- Facilita la enseñanza individualizada.
- Aumenta la motivación y el gusto por aprender. Promueve la enseñanza y aprendizaje de manera activa, y además se estimula a buscar el conocimiento.
- Reduce el tiempo de aprendizaje, cada uno va a su ritmo.
- Mejora el aprendizaje, el alumno avanza hacia nuevos temas únicamente cuando domina los que deben precederle.
- Exigen de un cambio del rol tradicional del profesor. Este no solo es fuente de conocimientos, sino un mentor o animador del aprendizaje.

Algunas desventajas más significativas:

- Cuando se utilizan de forma incorrecta, afectan la comunicación profesor-alumno e intergrupala.
- La disponibilidad de los recursos puede afectar la disponibilidad y uso de

esta herramienta.

- El docente, va perdiendo su papel central, pues el alumno ya no valora tanto el conocimiento que él tiene.

Un simulador para el proceso de backtracking de la programación lógica en la asignatura de PROLOG, facilitaría el autoaprendizaje y la discusión en los grupos de estudiantes, utilizando algo novedoso y desarrollado con las últimas tecnologías de las informáticas y las comunicaciones. Usando esta tecnología educativa, que es el simulador, el estudiante vería diferentes procesos que les resultan invisibles en la teoría y de esta forma visual sería más fácil y rápida su comprensión

La interacción del estudiante con el simulador, sería tan sencillo como que el estudiante introdujera datos y recibiera a cambio una respuesta, la cual él será capaz de evaluar y razonar. Este proceso de intercambio de información del estudiante con el simulador puede ser repetitivo y a medida que el estudiante vaya adquiriendo conociendo se va interesando más por el contenido.

Con el uso del simulador el estudiante va a ser capaz de ir acercándose el mismo a las respuestas utilizando el proceso de ensayo – error, es decir, él va ir probando juegos de datos hasta obtener la respuesta óptima, en este caso visual.

Estos softwares educativos están bien representados desde la tendencia pedagógica Tecnología Educativa pero no podemos dejar de enmarcarlos en el enfoque Histórico-Cultural de L.S. Vigotsky y más en nuestro país donde este constituye el basamento psicológico de la Pedagogía Cubana.

1.5.1 Simulador del proceso de backtracking de PROLOG visto desde el Enfoque Histórico-Cultural.

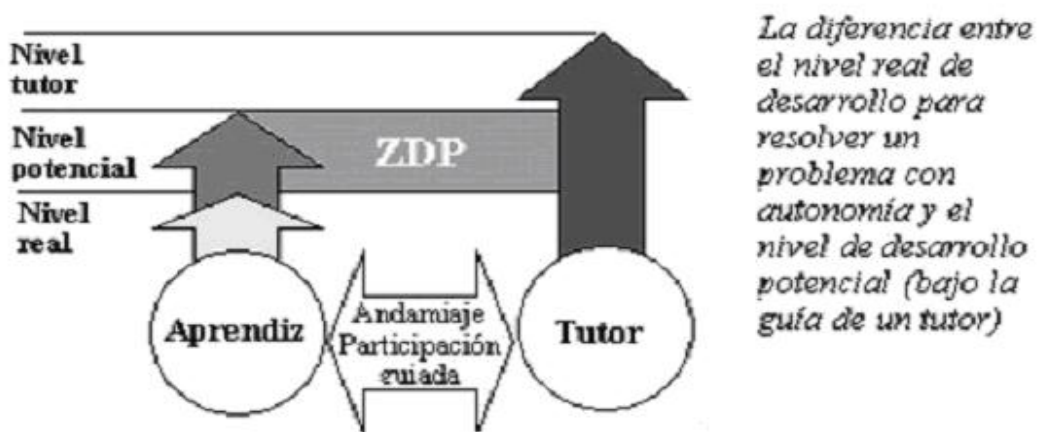
La esencia de la tendencia pedagógica del enfoque histórico-cultural centrada en Vigotsky es una concepción dirigida en lo fundamental a la enseñanza, facilitadora de un aprendizaje desarrollador, en dinámica interacción entre el sujeto cognoscente y su entorno social, de manera tal que se establece y

desarrolla una acción sinérgica entre ambos, promotora del cambio cuantitativo del sujeto que aprende a punto de partida de la situación histórico cultural concreta del ambiente social donde él se desenvuelve.[8]

Vigotsky introduce el concepto de Zona de Desarrollo Próximo (ZDP) que es la distancia entre el nivel real de desarrollo determinado por la capacidad de resolver independientemente el problema y el nivel de desarrollo potencial determinado a través de la resolución de un problema con el apoyo de una persona más preparada.

A continuación podemos ver una gráfica donde se representa lo antes expuesto.

Zona de desarrollo próximo (ZDP)



Para determinar este concepto hay que tener presente dos aspectos: la importancia del contexto social y la capacidad de imitación. Aprendizaje y desarrollo son dos procesos que interactúan. El aprendizaje debe estar relacionado con el nivel de desarrollo del individuo. El aprendizaje se logra más fácil en situaciones colectivas.

La zona de desarrollo próximo permite considerar que lo que puede hacer hoy el individuo con ayuda, le permite apropiarse de conocimientos, para posteriormente utilizar los mismos de forma independiente, y además que sea capaz de transferir su experiencia a nuevas situaciones. Si siempre que el alumno se enfrente a un

nuevo problema, va a necesitar de la ayuda de un compañero mejor preparado o del profesor, entonces la enseñanza no ha sido verdaderamente desarrolladora, como para que pueda asumir nuevas tareas y solucionarlas a partir de la experiencia anterior, por lo que este debe ir desarrollando su propio conocimiento.

De todo lo anterior se infiere, que el proceso de desarrollo está sin dudas mediado por la acción de los otros, lo que demuestra que el proceso pedagógico es un proceso eminentemente interactivo, en el que la comunicación tiene una gran significación.

La enseñanza desarrolladora mediante métodos y procedimientos adecuados debe trabajar teniendo en cuenta el desarrollo alcanzado en cada momento por el alumno, para que se promueva un "desarrollo próximo o futuro", cuyo nivel dependerá de los conocimientos y de las acciones que sea capaz de lograr el estudiante de forma independiente, con la ayuda del profesor. [9]

Esta concepción cambia la tradicional relación entre autoridad y distancia existente entre ambos participantes del proceso, señala como función fundamental del profesor la orientación y guía del estudiante, con el fin de potenciar sus posibilidades.

El simulador del proceso de backtracking de la programación lógica en la asignatura de PROLOG facilitará el proceso de enseñanza-aprendizaje en la asignatura de Programación Lógica. El estudiante utilizará lo disponible en el sistema de relaciones más cercano para propiciar su interés y un mayor grado de participación e implicación personal en las tareas de aprendizaje.

En este caso como lo más cercano al estudiante y lo que le está rodeando constantemente debido al perfil en que se está desarrollando son las NTIC, más específicamente las computadoras, la realización de un simulador educativo, como herramienta informática, facilitaría su formación como futuro profesional de la rama, si tenemos en cuenta que la computadora es un poderoso instrumento que brinda la oportunidad de transformar la pedagogía, hacia una pedagogía

más efectiva y emotiva que cambie, mejore y fortalezca el papel educador del profesor.

Simulando el proceso de backtracking de PROLOG, se brinda más claridad al estudiante a la hora de comprenderlo, pues más fácil le será al mismo llegar a la próxima zona de desarrollo, considerando a este software como una parte importante de su medio histórico-cultural en el que está presente.

En este enfoque se le atribuye gran importancia a la actividad conjunta con otros individuos, en este caso la relación profesor-alumno y entre los alumnos. El profesor no impone sus criterios, este orienta y guía al estudiante con el objetivo de desarrollar sus posibilidades, convertir en realidad las potencialidades de su zona de desarrollo próximo ayudando a este a alcanzar un mayor grado de conocimiento.

Precisamente el desarrollo de este sistema posibilitará además de que el alumno se apoye en una herramienta para su estudio, en la que puede interactuar con ella y además le propiciará el trabajo en grupo y el intercambio de opiniones con el resto de sus compañeros, así como la interacción con el profesor. En caso que la actividad sea presencial, o la educación sea a distancia, se considera al profesor como la persona encargada de guiar el proceso de aprendizaje, en ambos casos se facilita su camino para llegar a su zona de desarrollo próximo al minimizar el esfuerzo para el entendimiento.

Cuando se organiza la enseñanza como actividad conjunta donde interactúan profesor-alumno o alumnos entre sí, se fomentan en los jóvenes el desarrollo de una serie de cualidades de su personalidad y se genera un clima emocional favorable, muy eficaz para el aprendizaje y las relaciones interpersonales.

El simulador debe ser capaz de que el estudiante vaya desarrollando su propio conocimiento, lo que hoy realiza con asistencia o guía de una persona más experta, ya sea el profesor u otro estudiante, en un futuro deberá realizarlo con autonomía sin necesidad de tal asistencia.

Este software se caracterizará por tener organizados los contenidos de forma tal que los estudiantes transiten por estos para aspirar a niveles superiores de desempeño y ejecución, para ello el profesor debe diseñar las tareas a aplicar y ser sensible a los avances progresivos del estudiante y esto está totalmente ideado en la herramienta.

1.7 Inteligencia Artificial.

Se define la **Inteligencia Artificial (IA)** como aquella inteligencia exhibida por artefactos creados por humanos (es decir, artificiales). A menudo se aplica a los computadores. El nombre también se usa para referirse al campo de la investigación científica que intenta acercarse a la creación de tales sistemas. Aunque la IA está rodeada de bastante ciencia ficción, se trata de una rama de la Informática, que trata sobre comportamientos inteligentes, aprendizaje y adaptación en máquinas. [10]

La Inteligencia Artificial nació bajo la expectativa de desarrollar sistemas inteligentes, capaces de ejecutar actividades consideradas propias e inherentes a los seres humanos. Agrupa a todas las ramas que se interesan por la emulación de la inteligencia del hombre y de los procesos que le acompañan y caracterizan: el sentido común, razonamiento, aprendizaje y el habla, entre otras.

La IA se ha abierto espacio y ha logrado, por medio de la representación del conocimiento, la elaboración de sistemas inteligentes, sistemas expertos que para su desarrollo requieren de la ciencia de la computación y la investigación que permiten manipular la información de manera que puedan explicar y simular la conducta inteligente que ocurre en los seres humanos.

La inteligencia artificial es, una de las áreas con más retos de las Ciencias de la Computación. Posee amplias relaciones con disciplinas matemáticas como el Álgebra y la Estadística, tomando de éstas algunas herramientas para desempeñar su labor.

Las principales técnicas aplicadas actualmente por la IA son:

- Sistemas Expertos: Colocan las estrategias de razonamiento y la experiencia de un experto en una caja negra.
- Lógica Difusa: Sistemas que manejan la ambigüedad y vaguedad del lenguaje natural.
- Redes Neuronales: Sistemas que aprenden simulando el trabajo conexionista y en paralelo del cerebro.
- Algoritmos Genéticos: Soluciones evolutivas que se adaptan a cambios.
- Razonamiento Basado en Casos: Sistemas que razonen y aprendan basándose en analogías.
- Máquinas de aprendizaje: Sistemas que derivan conocimiento y reglas de decisión a partir de un conjunto de datos. El ejemplo más conocido es el *data mining*.

Estas son algunas de las áreas de trabajo de la IA:

- Reconocimiento de patrones.
- Visión artificial.
- Robótica.
- Lenguaje natural.
- Multimedia.
- Realidad Virtual.

En nuestro país la IA es una disciplina que forma parte de las carreras de Ingeniería Informática y Ciencias de la Computación, la cual está formada por varias asignaturas como lógica matemática, simulación, programación lógica e inteligencia artificial considerada la asignatura integradora.

Su objetivo fundamental es completar la formación del profesional, capacitándolo en la solución computacional de problemas que requieran la aplicación de los

conceptos y métodos de la IA. Se trata de problemas considerados inabordables por los métodos tradicionales y a su vez no bien estructurados, que no sólo se resisten a una solución algorítmica por su complejidad computacional, sino que además requieren de la introducción de estrategias de búsqueda de soluciones, de reglas heurísticas y finalmente del conocimiento, para el hallazgo no determinista y no siempre óptimo de sus soluciones. [11]

Los elementos generales que abarcan esta disciplina son:

- Fundamentos de la lógica matemática. Interpretación y demostración de teoremas.
- Fundamentos de la programación descriptiva. Programación lógica y funcional.
- Fundamentos de la inteligencia artificial.
- Fundamentos de la teoría de la decisión bajo incertidumbre y de los procesos de inferencia.
- Fundamentos de los modelos de representación y procesamiento del conocimiento.
- Fundamentos teóricos del aprendizaje y los sistemas dinámicos.
- Fundamentos de la ingeniería del conocimiento.

Como puede notarse, esta disciplina juega un papel significativo en la formación y preparación del estudiante.

1.8 PROLOG, lenguaje declarativo.

El desarrollo de PROLOG se inició en 1970 con Alain Coulmeauer y Philippe Roussel, quienes estaban interesados en desarrollar un lenguaje para hacer deducciones a partir de texto. El nombre corresponde a "**PRO**gramming in **LOG**ic" (Programación en lógica). **PROLOG** fue desarrollado en Marsella, Francia, en 1972. El principio de resolución de Kowalski, de la Universidad de Edimburgo pareció un modelo apropiado para desarrollar sobre él mecanismo de

inferencia. Con la limitación de de la resolución de cláusulas de Horn, la unificación condujo a un sistema eficaz donde el no-determinismo inherente de la resolución se manejó por medio de un proceso de exploración a la inversa, el cual se podía implementar con facilidad. [12]

PROLOG es un lenguaje para el procesamiento de información simbólica, se basa en el lenguaje de la Lógica de Primer Orden y se utiliza para resolver problemas en los que entran en juego *objetos* y *relaciones* entre ellos. Por ejemplo, cuando decimos "Jorge tiene una moto", estamos expresando una relación entre un objeto (Jorge) y otro objeto en particular (una moto). Más aún, estas relaciones tienen un orden específico (Jorge posee la moto y no al contrario).

Es un lenguaje declarativo, basado en las matemáticas y en la lógica, donde no se dice cómo hacer una cosa, sino más bien, qué cosa hacer. Formando por una serie de hechos y un conjunto de reglas. Estos hechos y reglas se introducen en una base de datos a través de una operación de consulta por cláusulas de Horn que constituyen reglas del tipo "modus ponens", es decir, si es verdad el antecedente, entonces es verdad el consecuente". Una de las ventajas de la programación lógica es que se especifica qué se tiene que hacer (programación declarativa), y no cómo se debe hacer (programación imperativa).

No obstante, la forma de escribir las cláusulas de Horn es lo contrario de lo habitual. Primero se escribe el consecuente y luego el antecedente. El antecedente puede ser una conjunción de condiciones que se denomina secuencia de objetivos. Cada objetivo se separa con una coma y puede considerarse similar a una instrucción o llamada a procedimiento de los lenguajes imperativos, ejemplo:

Hija (A, B) <- mujer (A), padre (B, A).

Que podría leerse así: "A es hija de B si A es mujer y B es padre de A".

Un programa lógico está formado por un conjunto finito de cláusulas de programa que son hechos o reglas. Por ejemplo:

padre(luis,miguel).-----	hecho
padre(miguel,jose).-----	hecho
padre(jose,juan).-----	hecho
abuelo(X,Y):-padre(X,Z),padre(Z,Y)-----	regla

En PROLOG no existen instrucciones de control. Su ejecución se basa en dos conceptos: la unificación y el backtracking. Gracias a la unificación, cada objetivo determina un subconjunto de cláusulas susceptibles de ser ejecutadas. Cada una de ellas se denomina punto de elección. PROLOG selecciona el primer punto de elección y sigue ejecutando el programa hasta determinar si el objetivo es verdadero o falso. En caso de ser falso entra en juego el backtracking, que consiste en deshacer todo lo ejecutado situando el programa en el mismo estado en el que estaba justo antes de llegar al punto de elección. Entonces se toma el siguiente punto de elección que estaba pendiente y se repite de nuevo el proceso. Todos los objetivos terminan su ejecución "verdadero" o "falso".

PROLOG permite implementar los programas casi directamente a partir de las especificaciones realizadas a partir de un análisis y diseño de la solución desde un alto nivel de abstracción. Además el procedimiento de backtracking está implícito en el propio motor de inferencia.

El lenguaje PROLOG juega un importante papel dentro de la Inteligencia Artificial, y se propuso como el lenguaje nativo de las máquinas de la quinta generación ("Fifth Generation Kernel Language", FGKL) que quería que fueran Sistemas de Procesamiento de Conocimiento. La expansión y el uso de este lenguaje propició la aparición de la normalización del lenguaje PROLOG con la norma ISO (propuesta de junio de 1993).

1.9 Backtracking en PROLOG.

El proceso de backtracking es una de las herramientas más fuerte con la que cuenta PROLOG, este se realiza automáticamente en las ejecuciones de los programa y en la búsqueda de las soluciones.

El backtracking consiste en recordar los momentos de la ejecución donde un

objetivo tenía varias soluciones para posteriormente **dar marcha atrás** y seguir la ejecución utilizando otra solución como alternativa.

El backtracking funciona de la siguiente manera:

Cuando se va ejecutar un objetivo, PROLOG sabe de antemano cuantas soluciones alternativas puede tener. Cada una de las alternativas se denomina **punto de elección**. Dichos puntos de elección se anotan internamente y de forma ordenada. Para ser exactos, se introducen en una pila.

Se escoge el primer punto de elección y se ejecuta el objetivo **eliminando** el punto de elección en el proceso.

Si el objetivo tiene éxito se continúa con el siguiente objetivo aplicándole estas mismas normas.

Si el objetivo falla, PROLOG da marcha atrás recorriendo los objetivos que anteriormente sí tuvieron éxito (en orden inverso) y deshaciendo las ligaduras de sus variables. Es decir, comienza el backtracking.

Cuando uno de esos objetivos tiene un punto de elección anotado, se detiene el backtracking y se ejecuta de nuevo dicho objetivo usando la solución alternativa. Las variables se ligan a la nueva solución y la ejecución continúa de nuevo hacia adelante. El punto de elección se elimina en el proceso.

El proceso se repite mientras haya objetivos y puntos de elección anotados. De hecho, se puede decir que un programa PROLOG ha terminado su ejecución cuando no le quedan puntos de elección anotados ni objetivos por ejecutar en la secuencia.

Además, los puntos de elección se mantienen aunque al final la conjunción tenga éxito. Esto permite posteriormente conocer todas las soluciones posibles.

1.10 Teaching.

Se realizó una minuciosa búsqueda en Internet y solamente se encontró una herramienta realizada en la UCLV llamada **Teaching**. La materialización de este trabajo se aprecia a través de una aplicación en Java, la cual permite entre otras

cosas, explicar de forma visual cómo se alcanzó un objetivo, cómo se instancian las variables, la animación de lo que ocurre en cada paso, así como el árbol de cláusulas obtenido. Del mismo modo se puede explicar por qué una conclusión particular no puede ser alcanzada [13]. Esta herramienta simula 3 ejemplos relativamente complejos para estudiantes que comienzan a ver este tipo de programación, no permite ir a pasos anteriores para si el estudiante tuvo alguna duda volver a repetir el o los pasos donde surgió la duda además no se especializa en un solo proceso por lo que no profundiza en ningún contenido.

Por lo expuesto anteriormente se decidió desarrollar una herramienta que constará con ejemplos clásicos, sencillos y ordenados por el grado de complejidad sobre un solo proceso, **Backtracking**. Dará la posibilidad de navegar por los pasos de la ejecución, es decir el estudiante podrá ir atrás o hacia delante según vaya entendiendo el ejemplo el cual él selecciona para simular. De esta forma el estudiante constará con una aplicación que lo ayudará en el autoaprendizaje y un mejor entendimiento de este proceso tan usado en el lenguaje de programación PROLOG, como lo requiere este caso.

Este simulador formará parte de una aplicación basada en un mapa conceptual sobre PROLOG, y se ubicará exactamente en el concepto Backtracking. Dicha aplicación, estará libremente disponible en Internet por lo que exige ser soportada por cualquier navegador, libre de patentes comerciales y de libre adquisición.

Para desarrollar una aplicación web con estas características lo más recomendado serían los Applets de Java. Estos se pueden desarrollar con varios entornos de desarrollo integrado (IDE) pero se escogió el NetBeans IDE 5.0 que se puede obtener de forma gratuita conjunto con la plataforma J2SE 5.0 Update 7 en <http://java.sun.com/j2se/1.5.0/download-netbeans.html>. Además usaremos el W-PROLOG para interpretar el código PROLOG de los ejemplos a simular. En el próximo capítulo se estudian estas herramientas demostrando su eficacia en la implementación del producto que se requiere.

Capítulo 2. Diseño de la propuesta de solución.

2.1 Diagnóstico inicial del problema para desarrollar el diseño de la propuesta de solución.

Antes de comenzar el diseño del simulador se realizó un diagnóstico con el objetivo de conocer el estado y las principales deficiencias del proceso de enseñanza-aprendizaje de la asignatura Programación Lógica, específicamente en el proceso de backtracking. El diagnóstico se dividió en 2 etapas, la primera en encuestas a los estudiantes y la segunda en entrevistas a profesores de la asignatura.

La encuesta (**Anexo 1**) se aplicó a los 36 alumnos de 4to año de Licenciatura en Ciencia de la Computación, de la facultad de Matemática-Física-Computación de la Universidad Central “Martha Abreu” de las Villas, siendo esta la muestra seleccionada de forma intencional, la cual representa el 100% de la población. Se realizó de forma anónima, explicando a los estudiantes sobre el objetivo general de nuestra investigación, solicitándoles su cooperación. Esta se confeccionó con el objetivo de caracterizar el aprendizaje del proceso de backtracking de la asignatura Programación Lógica.

Para el procesamiento de la encuesta se utilizó la estadística descriptiva y el cálculo porcentual, permitiendo llegar a conclusiones y diagnosticar el proceso de enseñanza-aprendizaje de la Programación Lógica.

La entrevista (**Anexo 2**) permitió corroborar con los educandos las insuficiencias en el proceso de enseñanza-aprendizaje de la asignatura Programación Lógica y en el tema del backtracking, para ello se aplicó a 5 profesores universitarios de los cuales 3 son Máster en Ciencias y 2 son Doctores en el tema de Inteligencia Artificial. Todos poseen más de 6 años de experiencia impartiendo docencia y han impartido la asignatura.

2.1.1 Análisis de la Encuesta aplicada a los estudiantes.

Pregunta 1:

Evalúa el interés de los estudiantes por la asignatura. De los 36 encuestados, 7 (19.44%) tienen un interés muy bueno por la asignatura, 25 (69.44%) de bueno y solamente 4 (11.11%) estudiante tiene un interés regular (**Tabla 2.1**). Esto demuestra que 32 (88.89%) estudiantes tienen interés entre muy bueno y bueno, y al menos todos muestran algún interés pues no hay ningún que no tenga interés por PROLOG (**Figura 2.1**).

Tabla 2.1 Interés de los estudiantes por la asignatura

P1	FRECUENCIA	%
Muy Bueno	7	19.44%
Buena	25	69.44%
Regular	4	11.11%
Ninguno	0	0.00%
TOTAL	36	100.00%



Figura 2.1 Interés de los estudiantes por la asignatura

Pregunta 2:

Esta evalúa la dificultad con el estudio de la asignatura. De los 36 estudiantes, 32 (88.89%) expresaron que la asignatura era difícil, 4 (11.11%) manejable y ninguno fácil (**Tabla 2.2**). De aquí se deriva que para los estudiantes esta es una asignatura con un alto grado de dificultad (**Figura 2.2**).

Tabla 2.2 Dificultad con el estudio de la asignatura.

P2	FRECUENCIA	%
Difícil	32	88.89%
Manejable	4	11.11%
Fácil	0	0.00%
TOTAL	36	100.00%



Figura 2.2 Dificultad con el estudio de la asignatura.

Pregunta 3:

Evalúa el cubrimiento bibliográfico que tienen los estudiantes de la asignatura. 18 (50 %) estudiantes considera que la bibliografía sobre la programación lógica es escasa, 16 (44.4 %) que no es muy abundante y 2 (5.5 %) cree que es abundante para un total de 36 encuestados (**Tabla 2.3**). De aquí podemos inferir que existe un déficit de bibliografía sobre la asignatura (**Figura 2.3**).

Tabla 2.3 Cubrimiento bibliográfico que tienen los estudiantes de la asignatura.

P3	FRECUENCIA	%
Escasa	18	50
No muy abundante	16	44.4
Abundante	2	5.5
TOTAL	36	100



Figura 2.3 Cubrimiento bibliográfico que tienen los estudiantes de la asignatura.

Pregunta 4:

Evalúa la comprensión que tiene los estudiantes del proceso de Backtracking (**Tabla 2.4**). De los 36 alumnos encuestados, 3 (8.33%) consideran que es fácil, 9

(25%) normal y 24 (66.67%) difícil. De aquí se saca la conclusión que este proceso es difícil para la mayoría de los estudiantes (**Figura 2.4**). Estos plantean que la mayor dificultad es la abstracción del proceso y su complejidad.

Tabla 2.4 Compresión que tiene los estudiantes del proceso de Backtracking

P4	FRECUENCIA	%
Fácil	3	8.33%
Normal	9	25.00%
Difícil	24	66.67%
TOTAL	36	100.00%

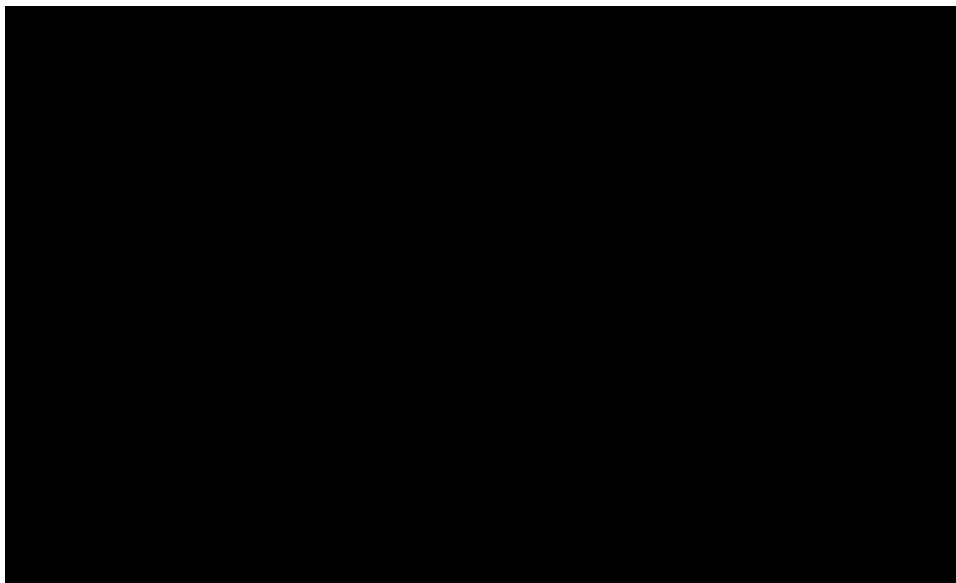


Figura 2.4 Compresión que tiene los estudiantes del proceso de Backtracking

Pregunta 5:

Evalúa la orientación del estudio al proceso de backtracking en la asignatura. De un total de 36 encuestados, 13 (36.11%) plantean que es buena, 17 (47.22%) regular y 6 (16.67%) insuficiente (**Tabla 2.5**). En este resultado predominan las respuestas de regular e insuficiente por lo que se demuestra que la orientación no cubre las expectativas de los estudiantes en su totalidad y que son

necesarias otros métodos (**Figura 2.5**).

Tabla 2.5 Orientación del estudio al proceso de backtracking en la asignatura.

P5	FRECUENCIA	%
Buena	13	36.11%
Regular	17	47.22%
Insuficiente	6	16.67%
TOTAL	36	100.00%

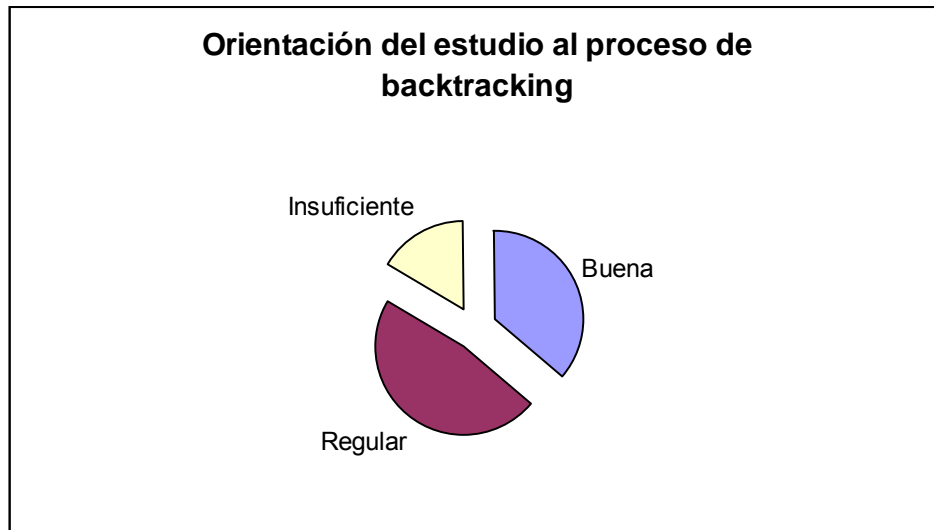


Figura 2.5 Orientación del estudio al proceso de backtracking en la asignatura.

Pregunta 6:

Informa sobre el conocimiento de algún recurso informático accesible para comprender mejor el proceso de backtracking, de los 36 alumnos que se le aplicó la encuesta, 25 (69.44%) plantean no conocer ningún recurso informático que les brinde esta posibilidad, 11 (30.56%) expresaron lo contrario (**Tabla 2.6**). De este resultado, se infiere que la mayoría de los estudiantes no tienen conocimiento sobre la existencia de algún recurso informático que les ofrezca la posibilidad de

comprender mejor el proceso de backtracking (**Figura 2.6**).

Tabla 2.6 Conocimiento de algún recurso informático accesible para comprender mejor el proceso de backtracking.

P6	FRECUENCIA	%
Sí	11	30.56%
No	25	69.44%
TOTAL	36	100.00%

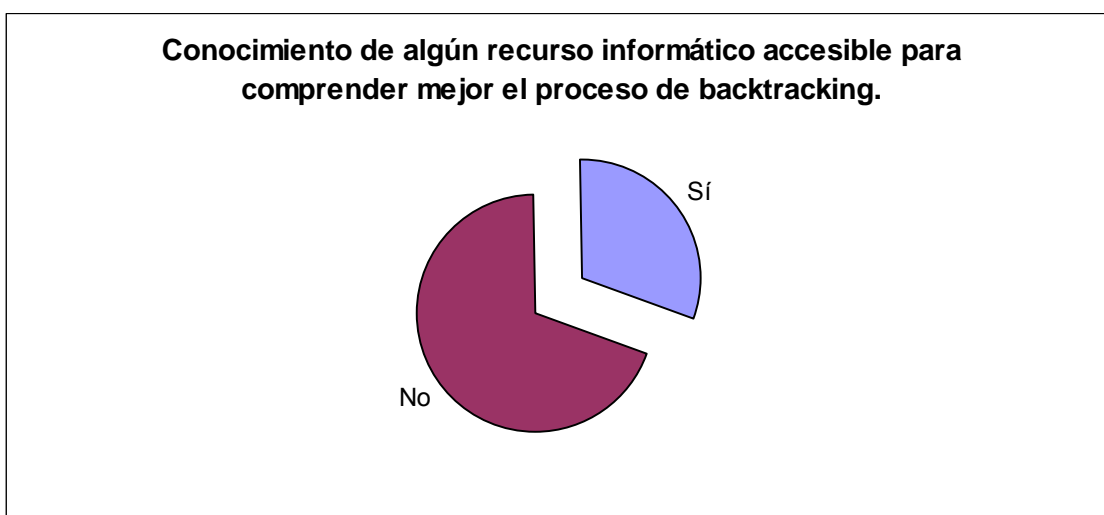


Figura 2.6 Conocimiento de algún recurso informático accesible para comprender mejor el proceso de backtracking.

Pregunta 7:

Esta pregunta dice cuales han sido los recursos informáticos usado por los 11 estudiantes que plantearon conocer algún recurso de estos, que les brinda la posibilidad de comprender el proceso de backtracking (**Tabla 2.7**), 2 (18.18%) refirieron que han utilizado tutoriales inteligentes, 2 (18.18%) afirmaron que han interactuado con algún simulador, ninguno con un entrenador y 7 (63.64%) otro tipo de recurso. Este resultado demuestra que la mayoría de los estudiantes no ha interactuado con ningún simulador que les permita comprender mejor el proceso de backtracking en lenguaje PROLOG (**Figura 2.7**), esta es la razón por la que esta investigación queda plenamente justificada.

Tabla 2.7 Recursos informáticos usado por los 11 estudiantes.

P7	FRECUENCIA	%
Tutorial Inteligente	2	18.18%
Simulador	2	18.18%
Entrenador	0	0.00%
Otro	7	63.64%
TOTAL	11	100.00%

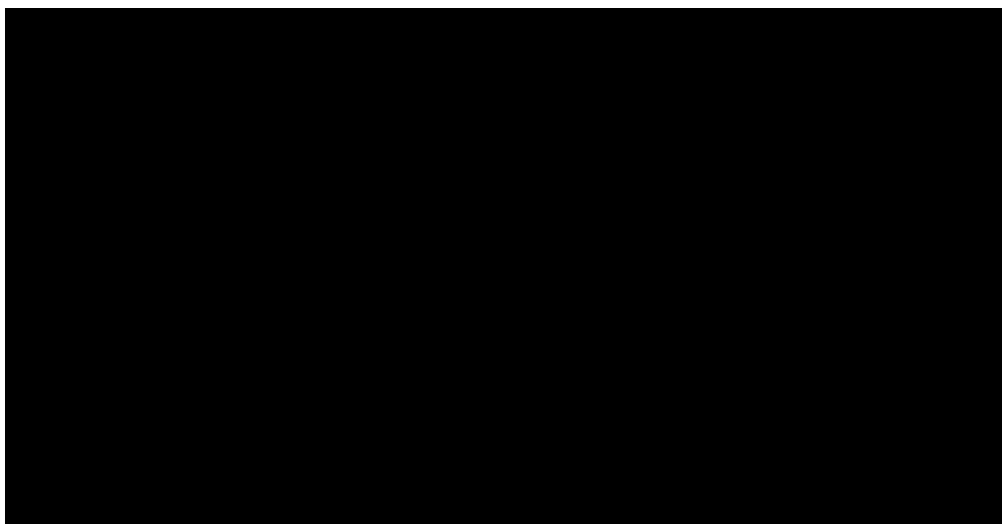


Figura 2.7 Recursos informáticos usado por los 11 estudiantes.

2.1.2 Valoración de la Entrevistas realizadas a profesores que han impartido la asignatura.

Como resultado de las entrevistas realizadas se pudo apreciar que los principales problemas que se presentan son:

- Una parte significativa de los profesores utiliza las mismas conferencias de cursos anteriores. Estas conferencias están en formato digital publicadas en la red. Además de escasos libros y artículos que ellos ponen en la biblioteca para ser consultados sin posibilidad de

prestamos externos. También orientan búsquedas en internet donde la información no está concentrada y resulta difícil encontrarla.

- La mayoría de los profesores expresaron que existen problemas con la motivación y el interés de los estudiantes, esto puede estar dado porque enfrentan a este paradigma de programación después de haber trabajado por mucho tiempo con un estilo imperativo, lo que trae como resultado que rechacen el nuevo paradigma y que lleguen, incluso, a cuestionarse su utilidad.
- Los profesores plantean que la asimilización de los contenidos no es muy bueno y que los resultados de la asignatura de manera general no son satisfactorios. Plantearon que las posibles causas están en que la Programación Lógica constituye un nuevo estilo de programación al que el estudiante hasta este momento no se ha enfrentado y por lo abstracto que resulta su explicación y entendimiento. Además no cuentan con suficientes herramientas que los apoyen en el proceso de enseñanza-aprendizaje de los contenidos de la asignatura.
- Todos los profesores coincidieron en que la comprensión del proceso de backtracking presenta dificultades. Señalan que este contenido es muy abstracto y difícil, necesitando mucho tiempo para impartirlo y ejercitarlo. Muchas veces el tiempo planificado para la clase no alcanza y por tanto tienen que dejar algunos ejercicios orientados para el autoestudio. Plantean que es necesario un adecuado aprendizaje del proceso de backtracking debido a que este constituyen la base para el resto de los contenidos que se impartirán en la asignatura.
- El total de los profesores utilizó como materiales complementarios los gráficos en el pizarrón. Manifiestan no tener a su alcance ninguna herramienta que sustituya esta técnica tradicional y concuerdan en que este método es deficiente, ya que necesitan dedicar demasiado tiempo a un mismo tema porque se dificulta el entendimiento.

2.2 Metodología y Tecnologías para el desarrollo del simulador de backtracking.

Antes de hacer un análisis de las herramientas y lenguajes que se utilizarán para el diseño e implementación del simulador se debe tener en cuenta que el software finalmente estará accesible desde un mapa conceptual, realizado por la Msc. Lydia Rosa Ríos. Partiendo del hecho de que el mapa conceptual se encontrará accesible desde la Web, por lo que se hace necesario que el simulador sea un software que se pueda ejecutar desde una página Web, sin necesidad de descargarlo. Además de acceder a él gratuitamente y sin tener en cuenta patentes.

2.2.1 Uso del Software Libre.

El software que se va realizar se quiere que no requiera patente y que este de forma gratuita y de libre adquisición en Internet por lo que debe ser realizado bajo el concepto de software libre. **Software libre** (en inglés *free software*) es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. [14]

El término de software libre –lo contrario del software patentado– se aplica a las aplicaciones informáticas que están libremente disponibles bajo un acuerdo de licencia pública de manera que cualquiera puede adaptarlos y mejorarlos. Las quejas más comunes sobre los programas de software patentados de Microsoft son sus costos relativamente elevados y que, cuando fallan, sólo Microsoft puede repararlos. Cada vez son más las organizaciones de toda clase que optan por soluciones libres para sus necesidades informáticas, porque los productos libres son generalmente más baratos, más fiables y más fáciles de reparar cuando fallan.

Los softwares libres nos dan:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus

necesidades. El acceso al código fuente es una condición previa para esto.

- La libertad de distribuir copias.
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

2.2.2 Tecnología Java

Desde ya hace un tiempo venimos escuchando en todas partes sobre la tecnología Java, esta está compuesta por 2 partes:

- El lenguaje de programación, y
- La plataforma

El lenguaje de programación Java

El lenguaje Java es de alto nivel es decir que las instrucciones o sentencias son escritas con palabras similares a las de los lenguajes humanos (mayormente en inglés). Esto facilita la escritura y comprensión del código al programador. Sus características más importantes son:

- Lenguaje orientado a objetos.
- Java es un lenguaje sencillo.
- Independiente de plataforma
- Brinda un gran nivel de seguridad
- Capacidad multihilo
- Gran rendimiento
- Creación de aplicaciones distribuidas
- Su robustez o lo integrado que tiene el protocolo TCP/IP lo que lo hace un lenguaje ideal para Internet.

Tradicionalmente se han dividido los lenguajes en compilados e interpretados. Los primeros necesitan ser traducidos por un programa llamado compilador al lenguaje máquina, que es el que entiende el ordenador. Como ejemplo de estos

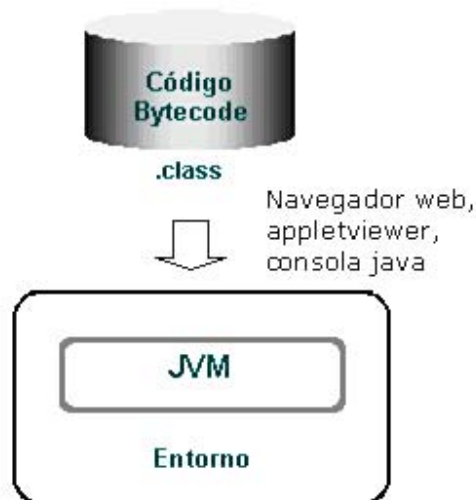
lenguajes podríamos citar a C, C++, Visual Basic, Clipper, etc. Los interpretados, en cambio, son traducidos mientras se ejecutan, por ejemplo HTML, WML o XML, por lo cual no necesitan ser compilados.

Así pues la diferencia entre estos lenguajes radica en la manera de ejecutarlos. Mientras que los compilados sólo se compilan una vez y lo hacen pasando todo el programa a código máquina (si da un error aunque sea en la última línea no podríamos ejecutar nada de nada), en el momento que lo hemos compilado correctamente se genera un archivo .exe que se puede ejecutar tantas veces como queramos sin tener que volver a compilar. Los interpretados en cambio, cada vez que los queramos ejecutar tendremos que interpretarlos línea a línea, es más lento, pero puede ocurrir un error en la última línea y a diferencia de los compilados, el programa se ejecuta justo hasta la línea que produce el error.

Java está diseñado para que un programa escrito en este lenguaje sea ejecutado independientemente de la plataforma (hardware, software y sistema operativo) en la que se esté actuando. Esta portabilidad se consigue haciendo de Java un lenguaje medio interpretado medio compilado, pues se coge el código fuente, se compila a un lenguaje intermedio cercano al lenguaje máquina pero independiente del ordenador y el sistema operativo en que se ejecuta (llamado en el mundo Java bytecodes).



Finalmente, se interpreta ese lenguaje intermedio por medio de un programa denominado máquina virtual de Java (JVM), que sí depende de la plataforma.



Los java bytecodes permiten el ya conocido “write once, run anywhere” (compila una sola vez y ejecútalo donde quieras). Podemos compilar nuestros programas a bytecodes en cualquier plataforma que tenga el compilador Java. Los bytecodes luego pueden ejecutarse en cualquier implementación de la máquina virtual de Java (JVM). Esto significa que mientras el ordenador tenga un JVM, el mismo programa escrito en Java puede ejecutarse en Windows, Solaris, iMac, Linux, etc.

Lo anterior expuesto demuestra por lo que se escogió este lenguaje de programación para implementar el simulador del proceso de backtracking en PROLOG.

La plataforma Java

Con plataforma nos referimos al ambiente de hardware y software en donde el programa se ejecuta, por ejemplo, plataformas como Linux, Solaris, Windows 2003 y MacOS. En casi todos los casos las plataformas son descritas como la combinación del sistema operativo y el hardware. La plataforma Java se diferencia de estas plataformas, es que es una plataforma sólo de software y se ejecuta sobre las otras plataformas de hardware.

La plataforma Java tiene 2 componentes:

- La máquina virtual de Java (JVM)

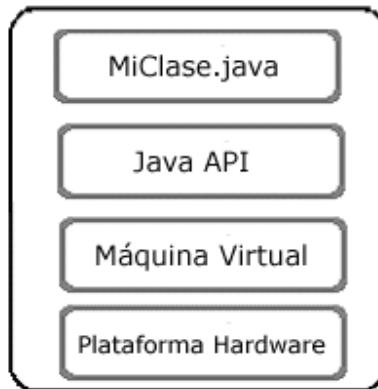
- El Java API (Application Programming Interface)

La (JVM); es la base de la plataforma Java y es llevada a diferentes plataformas de hardware.

El Java API es una gran colección de componentes de software que proporcionan muchas utilidades para el programador, por ejemplo, los API's para las interfases gráficas. Los API's de Java están agrupados en librerías de ciertas Clases e interfaces, estas librerías son conocidas como paquetes.

El siguiente gráfico describe un programa que se está ejecutando sobre la plataforma Java.

Como vemos, el Java API y la máquina virtual aíslan al programa del hardware.



2.2.3 Java 2 Standard Edition 5.0 (J2SE)

Esta plataforma fue la última que lanzó Sun Microsystems y permite hacer todo tipo de aplicaciones (Escritorio con Ventanas, escritorio consola, aplicaciones Web, Applets, etc).

La plataforma Java 2 incluye importantes novedades y mejoras, entre las que destaca el incremento de la velocidad de ejecución, la mejora de la "recogida de basura" (garbage collection), nuevas librerías de clases, modelo de seguridad mucho más completo, y un mayor soporte a la creación de software para cualquier país del mundo.

Algunas de las principales características de esta plataforma son:

- Plantillas (genéricos) — provee conversión de tipos (type safety) en tiempo de compilación para colecciones y elimina la necesidad de la mayoría de conversión de tipos (type casting).
- Metadatos — también llamados anotaciones, permite a estructuras del lenguaje como las clases o los métodos, ser etiquetados con datos adicionales, que puedan ser procesados posteriormente por utilidades de proceso de metadatos.
- Enumeraciones — la palabra reservada `enum` crea una lista ordenada de valores (como `Dia.LUNES`, `Dia.MARTES`, etc.). Anteriormente, esto solo podía ser llevado a cabo por constantes enteras o clases construidas manualmente (enum pattern).
- Varargs (número de argumentos variable) — El último parámetro de un método puede ser declarado con el nombre del tipo seguido por tres puntos (e.g. `void drawtext(String... lines)`). En la llamada al método, puede usarse cualquier número de parámetros de ese tipo, que serán almacenados en un array para pasarlos al método.

2.2.4 Netbeans como IDE

NetBeans es el programa de desarrollo para aplicaciones Java de licencia libre más utilizado por los usuarios. Este cuenta con las funcionalidades necesarias para el desarrollo de todos los tipos soportados bajo este estándar. El NetBeans IDE es un producto libre y gratuito sin restricciones de utilización.

Este IDE de desarrollo creado por Sun, el cual mantiene una licencia Open Source sobre Netbeans, ofrece ventajas extras sobre otros IDE's y llega a ser considerado el mejor editor de programas JAVA. Este software consta por el momento de dos partes, el IDE (entorno de desarrollo) NetBeans, y NetBeans Tools Platform (Plataforma de herramientas). Además de lo gratuito de la herramienta, esta posee un poderoso trasfondo de ejecución que permite ejecutar sin problemas aplicaciones de escritorio con la máquina virtual de Java y en

aplicaciones Web, el servidor JSP Tomcat.

Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

Dada todas estas características y funcionalidades que brinda el NetBeans, él será el IDE escogido para desarrollar el software.

2.2.5 Applets

Un applet es un código JAVA que carece de un método main, por eso se utiliza principalmente para el trabajo de páginas web, ya que es un pequeño programa que es utilizado en una página HTML y representado por una pequeña pantalla gráfica dentro de ésta. El applet se almacena en el servidor web y se llama desde una de sus páginas, ejecutándose en la máquina del cliente.

Ventajas de usar applets

- El applet es independiente de la máquina y el sistema operativo. El programador no tiene que hacer ningún cambio en función de la plataforma en la que se ejecutan.
- Evita problemas de actualización y distribución: si el programador quiere hacer una nueva versión del applet, la deposita en el servidor Web y automáticamente todos los clientes que accedan a partir de ese momento se descargan la nueva versión. No hay que recurrir al engorroso y caro sistema de mandarles un CD con el nuevo software.
- Debido a las restricciones de seguridad del propio lenguaje, los errores de programación no alteran el sistema operativo del cliente.

2.2.6 W-PROLOG

W-PROLOG es un interpretador implementado en JAVA para un lenguaje como PROLOG. La implementación es extremadamente portable y puede ser ejecutada como un applet en un Browser. W-PROLOG tiene una agradable interfaz para usuarios mejor que los sistemas PROLOG (los cuales proporcionan una interfaz típica con líneas de comandos). [15]

El sistema W-PROLOG es pequeño y es comparativamente simple (**Figura 2.8**). Sin embargo esto no lo hace particularmente rápido. W-PROLOG esta disponible tanto para Unix como para Windows, además puede ser ejecutado como un applet o como una aplicación individual, también puede ser llamada desde otro programa.

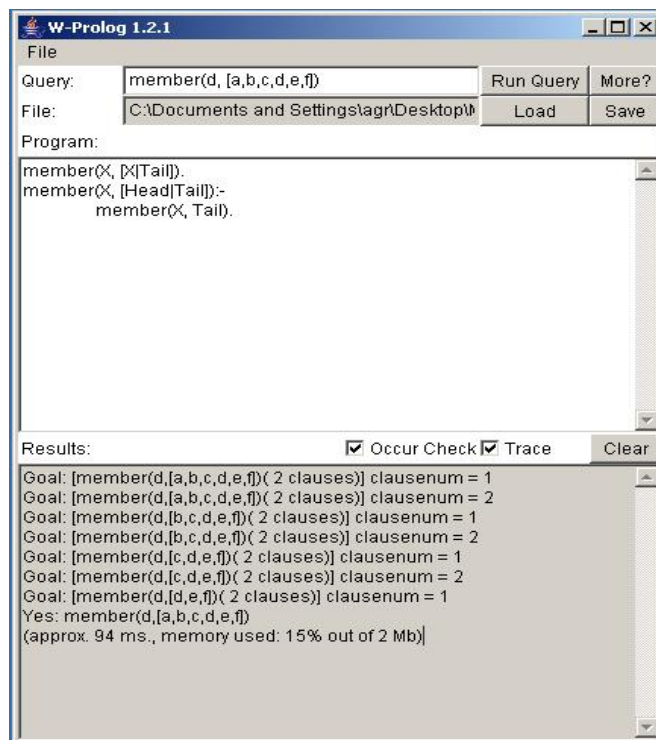


Figura 2.8 Wprolog

2.3 Diseño y Modelación utilizando lenguaje UML

UML es el lenguaje de modelado de sistemas de software más conocido en la actualidad; es el estándar internacional aprobado por la OMG (Object Management Group), consorcio creado en 1989 responsable de la creación, desarrollo y revisión de especificaciones para la industria del software.

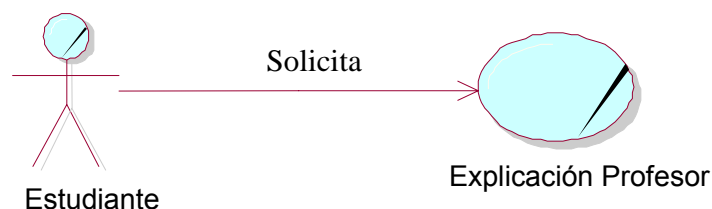
UML son un grupo de especificaciones de notación orientadas a Objeto, las cuales están compuesta por distintos diagramas, que representan las diferentes etapas del desarrollo de un proyecto de software. A continuación describimos y desarrollamos varios de estos diagramas.

2.3.1 Descripción del Negocio y requerimientos del software.

Se utilizó la entrevista (**Anexo 3**) como técnica para identificar el negocio y obtener los requerimientos del software, tanto funcionales con los no funcionales. Esta se le aplicó a la MsC Lidia Rosa Ríos, principal cliente del software. Los resultados se reflejan a continuación mediante diagramas, utilizando el lenguaje UML y los requerimientos del software.

2.3.1.1 Diagrama de Casos de Uso del Negocio

El modelo de Casos de Uso del Negocio es un modelo que describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como socios y clientes, es decir, describe las funciones que el negocio pretende realizar y su objetivo básico es describir cómo el negocio es utilizado por sus clientes y socios.

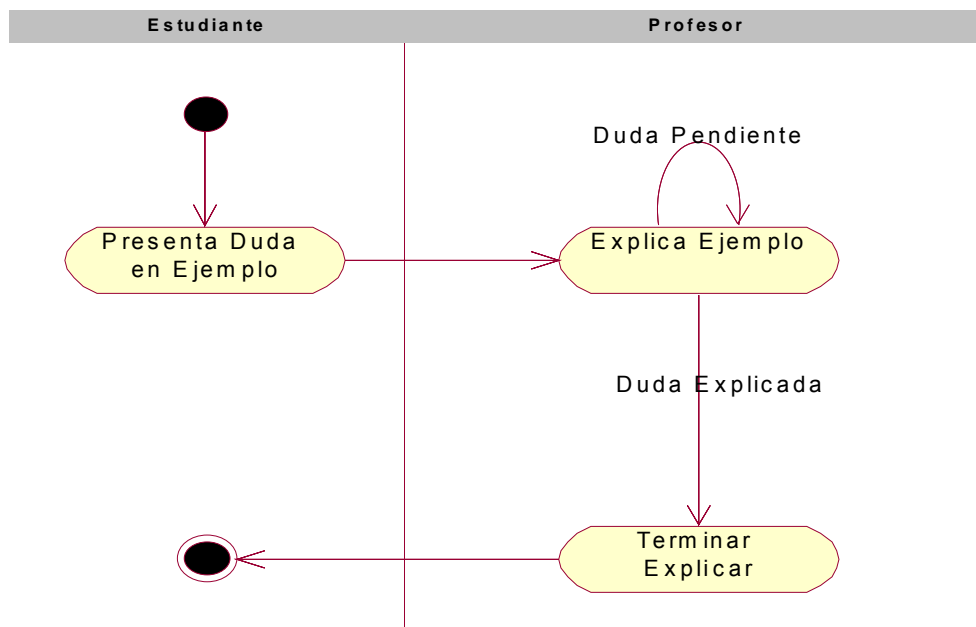


2.3.1.2 Diagrama de Actividad de Casos de Uso del Negocio

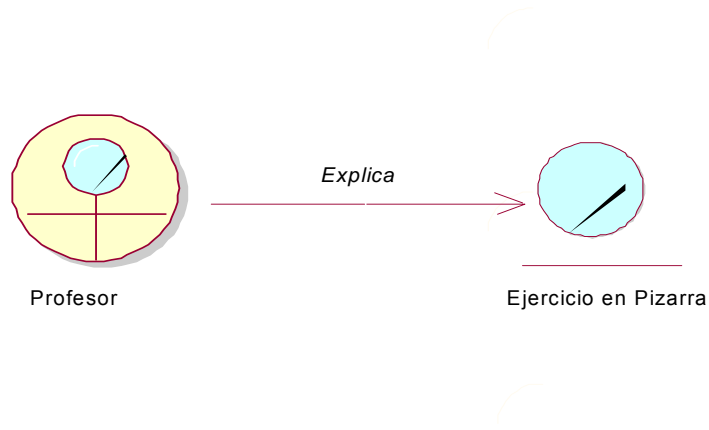
El modelo de Casos de Uso del Negocio es un modelo que describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como socios y clientes, es decir, describe las funciones que el negocio pretende realizar y su objetivo básico es describir cómo el negocio es utilizado por sus clientes y socios.

Los diagramas de actividad se usan para mostrar cómo se construyen los diferentes flujos de trabajo o los procesos dentro de un sistema, cómo se inician, los variados caminos alternativos que se pueden tomar desde el inicio hasta el fin y dónde puede ocurrir el procesamiento paralelo durante la ejecución.

Un diagrama de actividades generalmente no modela el comportamiento exacto de un sistema de software (como lo hace un diagrama de secuencia), sino los procesos y los flujos a un muy alto nivel. Las actividades generalmente serán realizadas por uno o más casos de uso; la actividad describe el proceso que se desarrolla y tanto el caso de uso como un actor usará el sistema para realizar toda o parte de una actividad.



2.3.1.3 Diagrama de Clases del Modelo de Objeto del Negocio



2.3.1.4 Especificación de los requisitos del software.

La obtención de requerimientos es un paso muy importante para el posterior desarrollo del software, pues un error en estas fases iniciales puede dar al traste con un sistema que no cumpla las expectativas de los usuarios y difícilmente aporte valor agregado al negocio para el que debe ser concebido.

La especificación de requisitos es un factor crítico para el éxito o fracaso de proyectos de desarrollo de software. Una gran proporción de los errores detectados en aplicaciones pueden ser debido a especificaciones de requisitos incorrectas o a un pobre entendimiento del dominio del problema.

Los requisitos no son más que:

- Condiciones que debe cumplir un sistema para satisfacer un contrato, una norma o una especificación.
- Condición o capacidad que necesita el usuario para poder resolver un problema o conseguir un beneficio determinado.

Requerimientos funcionales.

Los requerimientos funcionales no son más que la determinación clara y concisa de qué debe ser capaz de hacer el sistema, éstas se corresponden con opciones que ejecutará el software, operaciones realizadas de forma oculta o condiciones extremas a determinar por el sistema.

- Seleccionar el ejemplo. El software debe permitirle al usuario escoger el ejemplo que él desee.
- Mostrar el código del ejercicio. El software debe mostrar el código del ejercicio para usuario vea como esta implementado.
- Mostrar las variables del problema a simular. El software debe permitirle al usuario ver las variables y el valor que estas están tomando.
- Dejar un historial. El software debe ir dejando un historial del código al que se fue accediendo en cada paso.
- Mostrar la explicación del ejemplo. El software debe mostrar una explicación detallada del ejemplo seleccionado para simular.
- Ejecutar paso a paso. El software debe permitirle al usuario ejecutar paso a paso el ejemplo seleccionado.
- Regresar a pasos anteriores. El software debe permitirle al usuario regresar a pasos anteriores donde halla podido quedar dudas.

Requerimientos no funcionales.

Se refieren a las propiedades emergentes del sistema como la fiabilidad, el tiempo de respuesta, la capacidad de los dispositivos de entrada/salida, y la representación de datos que se utiliza en las interfaces del sistema.

Requisitos de performance.

Tiempo de respuesta rápido. El sistema debe tener un tiempo de respuesta rápido ante cualquier solicitud del usuario.

Alto rendimiento. El sistema debe implementar varias transacciones por segundo para dar una respuesta rápida al usuario y evitar demoras innecesarias.

Portabilidad.

El sistema deberá funcionar sobre cualquier plataforma.

Seguridad.

El sistema debe permitir el acceso todos los estudiantes o profesores pero no debe permitir que se le modifiquen datos.

Confiabilidad.

Se indican los requerimientos para la confiabilidad del sistema, de la habilidad de la aplicación de comportarse consistentemente de manera aceptable para el usuario.

Disponible todo el tiempo. El sistema debe estar disponible todo el tiempo para trabajar de forma tal que se pueda acceder las 24 h.

Preciso en la información. El sistema debe ser preciso en la información que le suministra al usuario para evitar cualquier tipo de error.

Mantenimiento. El sistema debe estar bien documentado, de forma tal que el tiempo de mantenimiento sea mínimo.

Interfaz.

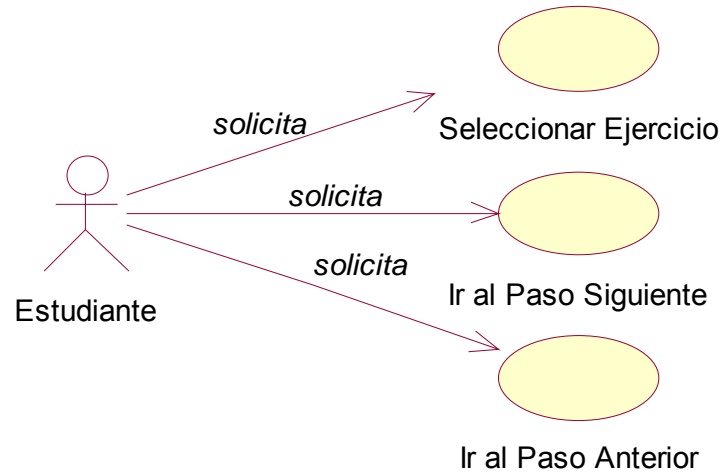
Interfaz de usuario. El sistema debe tener una interfaz sencilla, intuitiva. En general, fácil de usar y agradable a la vista del usuario.

Interfaz de comunicación. El sistema debe permitirle al usuario conectarse a él desde cualquier PC de la red.

2.3.2 Diagrama de Casos de Uso del Sistema

La siguiente figura muestra en el diagrama de casos del sistema, que se realizó basándose, en que representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos

interactúan.



2.3.3 Diagrama de Interacción

Los diagramas de interacción no son más que una descripción del modo en el que cada operación detectada en los diagramas de secuencia lleva a cabo sus responsabilidades y modifica el estado del sistema. En UML los diagramas de interacción pueden representarse a través de los Diagramas de Colaboración y/o de los Diagramas de Secuencia.

El tipo de diagrama seleccionado para construir los diagramas de interacción fue el de Secuencia, debido a que muestra cómo los objetos se comunican unos con otros en una secuencia de tiempo, qué sucede en cada momento, y para ello contienen objetos con sus ciclos de vida y los mensajes que se envían entre ellos ordenados secuencialmente.[16]

El diagrama de secuencias es el núcleo de un modelo dinámico, y muestra todos los cursos alternos que pueden tomar los casos de uso. Los diagramas de secuencias se componen de 4 elementos que son: el curso de acción, los objetos, los mensajes y los métodos.

Diagrama de Secuencia: Seleccionar Ejercicio

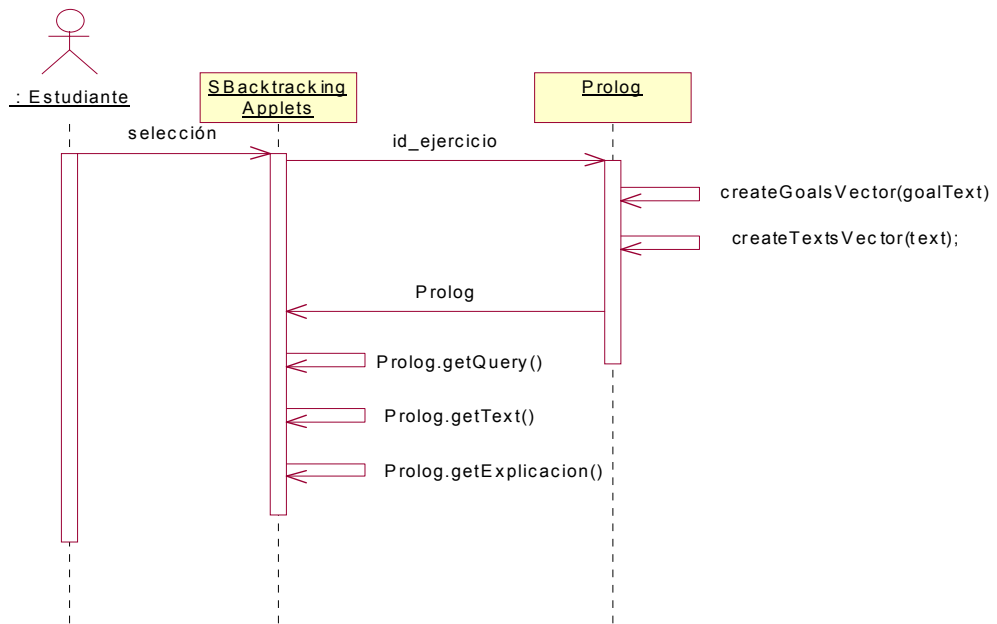
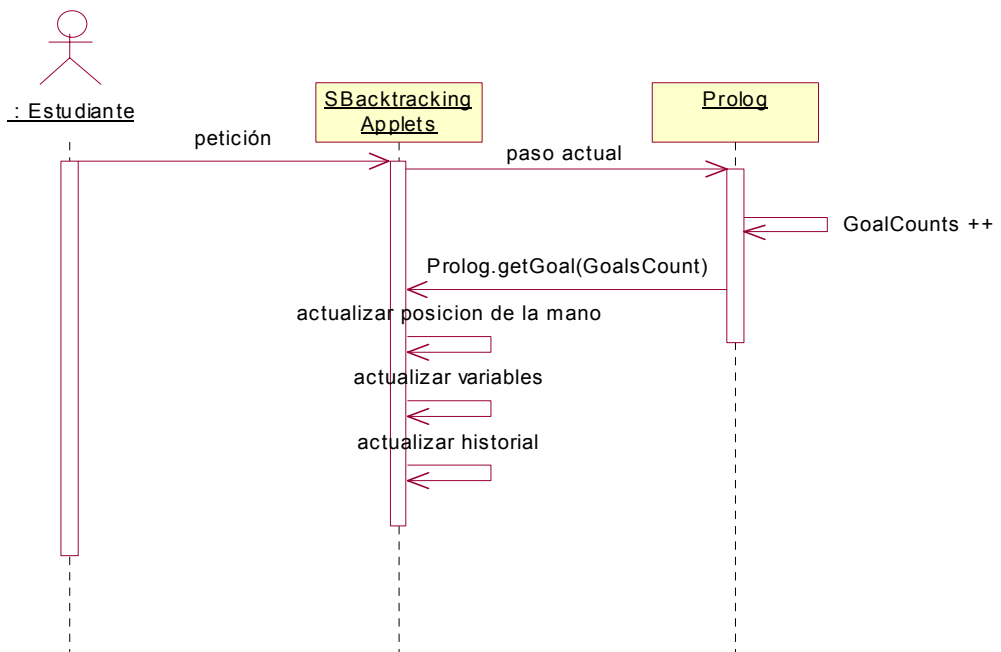
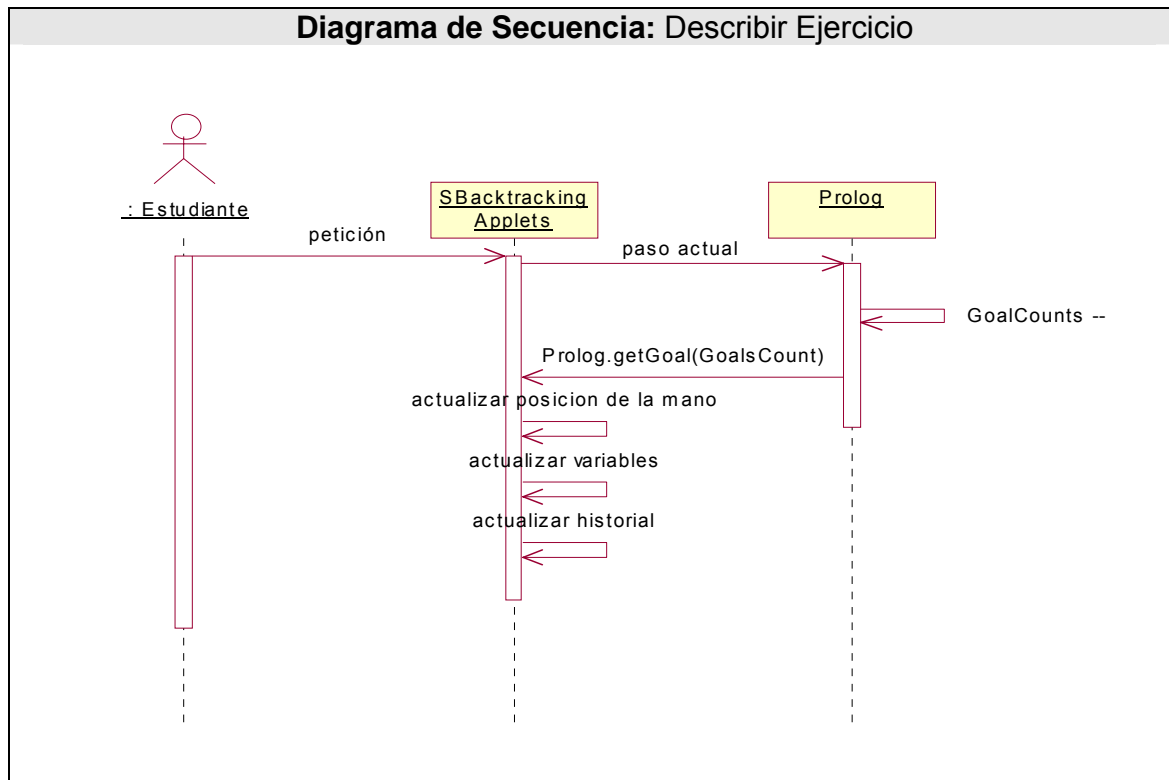
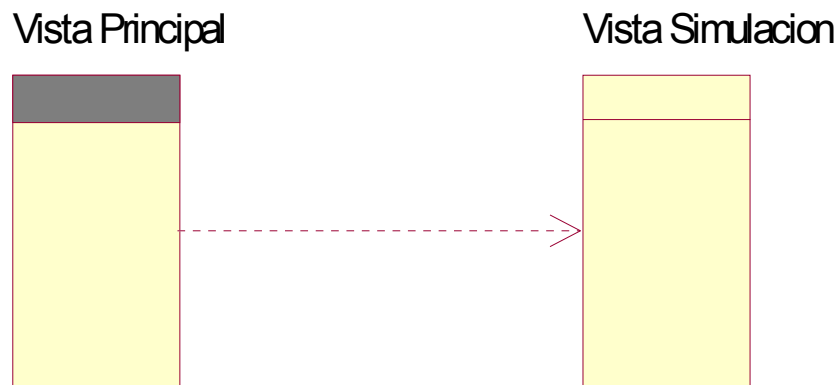


Diagrama de Secuencia: Ir al Paso Siguiente



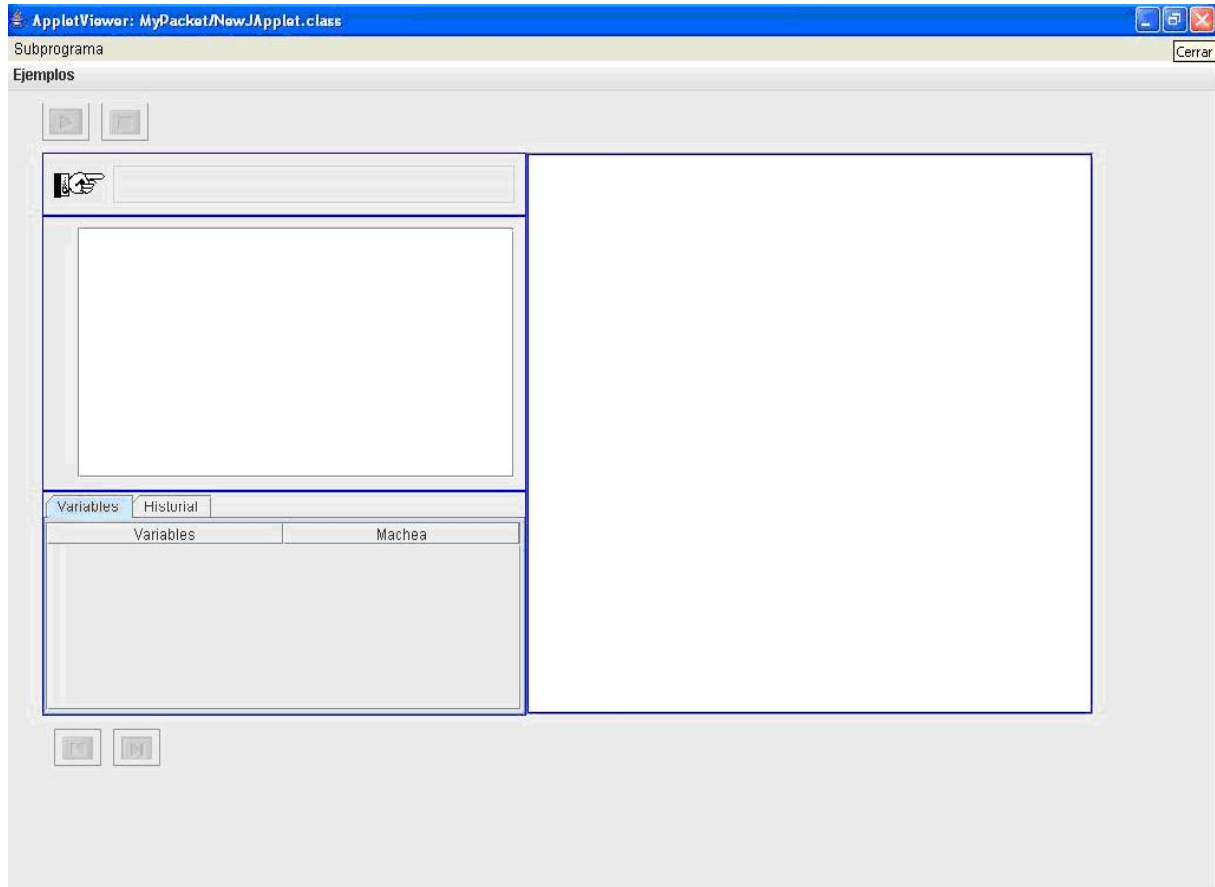


2.3.4 Diagrama de Navegación



2.3.5 Vistas del Sistema

Vista principal



Vista de la Simulación

AppletViewer: MyPacket/NewJApplet.class

Subprograma

Ejemplos

bisabuelo(clara,jaime).

```

progenitor(clara,jose).
progenitor(tomas,jose).
progenitor(tomas,isabel).
progenitor(jose,ana).
progenitor(jose,patricia).
progenitor(patricia,jaime).
bisabuelo(X,Y):- progenitor(X,T),progenitor(T,Z),progenitor(Z,Y).
        
```

Variables	Machea
X	clara
T	
Z	
Y	jaime

Se intenta satisfacer 'T' del primer objetivo de la conjunción de objetivos, 'progenitor(clara,T)'. El objetivo coincide con el primer hecho de la base de datos, donde se quedará su marca de posición asociada, y la variable 'T' se instancia a 'jose'. Dentro de una misma pregunta, las variables con el mismo nombre se refieren al mismo objeto, por tanto se sigue comprobando si se puede demostrar que 'progenitor(jose,Z), progenitor(Z,jaime)'.
 Para el objetivo 'progenitor(jose,Z)' se empieza a buscar en la base de datos desde el principio, hasta encontrar un hecho que haga coincidencia sintáctica. El primer hecho que coincide es 'progenitor(jose,ana)', por tanto Y se instancia a 'ana' y en este hecho se queda la marca de posición para este objetivo.
 Ahora se comprueba si se puede demostrar que es verdad 'progenitor(ana,jaime)'. Como este hecho no está en la base de datos, esta no es una solución, y se señala en el árbol de derivación natural, con el símbolo 'h'. La última variable que se había instanciado se desinstancia, y se sigue buscando en la base de datos, desde la marca de posición asociada, otro hecho que coincida con el objetivo actual. En este caso, progenitor(jose,patricia).
 Y se instancia a patricia. Ahora se demostrará si se cumple 'progenitor(patricia,jaime)'. Como este sí que es un hecho de la base de datos, la instanciación 'T=jose' y 'Z=patricia' es una solución a la conjunción de objetivos del ejemplo, y se señala en el árbol con el símbolo '.'.
 Y no hay más soluciones, porque no hay otro hecho de la base de datos que coincida con 'progenitor(clara,T)'.

2.4 Valoración de la factibilidad del diseño del simulador por el criterio de expertos.

Con el objetivo de analizar la pertinencia, confiabilidad y grado de aplicación del simulador, se procede a consultar un grupo de personas que poseen sobre el tema poder de expertos, de forma tal que la opinión de estos posibilite la validación del mismo y dejen además señaladas sus posibles debilidades o factores que dificulten su aplicación.

Para determinar el número de expertos a consultar se utilizó la siguiente expresión:

$$N = \frac{p(1-p) * k}{i^2}$$

Donde:

N : Número de expertos.

p : Proporción de error estimado. 0.01.....0.05

i : Nivel de precisión 0.005.....0.10

k : Constante computarizada.

Fiabilidad (%) K

1- α	K
99	6.6564
95	3.8416
90	2.6896

Tomando:

$$p = 0.01$$

$$i = 0.1$$

Fiabilidad del 99% para una $k = 6.6564$

Luego de aplicar la fórmula descrita anteriormente se obtuvo que $N \approx 6.5$ por lo que se utilizarán en el estudio un total de 7 expertos.

Para la selección de los 7 expertos se tuvo en cuenta sus años de experiencias en la enseñanza superior, los resultados alcanzados en su labor profesional y el conocimiento en temas de aplicaciones java y programación lógica. Estos aportaron opiniones para la evaluación de la propuesta sobre su nivel de aplicabilidad, necesidad de introducción, actualidad y rigor científico. **(Anexo 4)**

La labor de los 7 expertos representando el 100%, esta comprometida con el nivel de educación superior. La actividad profesional de los expertos seleccionados comprendió:

- 2 profesores que imparten la asignatura de la Programación Lógica en la UCLV para un 28,5%.
- 1 del grupo de inteligencia artificial en la UCLV, que representa un 14,3%.
- 1 pedagogo relacionado con la educación superior y la informática, representando el 14,3%.
- 2 profesores del departamento de Ingeniería Informática en el Centro Universitario José Martí, para un 28,5%.
- 1 desarrollador de aplicaciones Java, el cual representa un 14,3%.

La guía elaborada **(Anexo 5)** consta de cuatro aspectos o ítems sobre los criterios, puntos de vista y argumentos a considerar por los expertos.

Como resultado del procesamiento de las respuestas a la guía, se identificaron las regularidades en relación con los juicios de mayor consenso emitidos por ellos, acerca de cada temática indagada, los que se relacionan a continuación:

El nivel de aplicación en la educación superior.

Su nivel de aplicación es alto porque simula el proceso de backtracking de la programación lógica, proceso sumamente importante en este tipo de

programación, la cual presenta dificultades en el aprendizaje y dispone de muy pocos recursos para su aprendizaje. Brinda a los estudiantes la posibilidad comprender como funciona este proceso y leer una explicación de cada uno de los ejercicios simulados.

Necesidad de introducción.

Es necesario introducir el uso del simulador de backtracking, pues en la actualidad son escasas las posibilidades de simular el proceso de backtracking que propicie una mejor comprensión este proceso de la Programación Lógica. Da la posibilidad de ver las variables y sus valores en cada paso, una descripción de como se resuelve el ejemplo y un historial del código que se va ejecutando. Además se corresponde con los enfoques actuales de educación superior del país el diseño e implementación del simulador propuesto en esta investigación.

Fiabilidad Funcional

No se necesita la descarga o la instalación del simulador pues este se ejecuta en condiciones normales, en cualquier tipo de computadora, navegador y sistema operativo que se utiliza, solamente se tiene que tener la maquina virtual de Java (JVM) que se descarga de forma rápida y gratis de Internet. Permite acceder sin dificultad a sus diferentes opciones.

Aspectos técnicos-estéticos

El simulador se implemento utilizado como lenguaje el Java, aspecto que lo hace novedoso, esta basado en Applets los cuales son muy seguros y estos se almacena en el servidor web y se llama desde una de sus páginas, ejecutándose en la máquina del cliente. La interfaz del software es amigable y sencilla. Cuenta con una pantalla principal desde la cual se accede a todas las opciones.

Conclusiones

- La sistematización teórica realizada sobre las TIC y la Programación Lógica permitió corroborar que existen carencias en cuanto a medios para apoyar el proceso de enseñanza-aprendizaje de backtracking en la asignatura de Programación Lógica.
- Los estudios realizados permitieron constatar que existen insuficiencias en el aprendizaje de los estudiantes para asimilar el proceso de backtracking expresado en dificultades de comprensión de este proceso.
- El simulador implementado se caracteriza por una interfaz sencilla, estabilidad en el sistema y confiabilidad en los datos. Además de facilitar el aprendizaje del proceso de backtracking, ayudando al estudiante a comprender como el lenguaje Prolog realiza este proceso en la búsqueda de soluciones de los problemas.
- Los expertos consultados avalan la pertinencia del simulador por considerar que cumple con los requisitos necesarios para su introducción.

Recomendaciones

- Proponer a la dirección de la Facultad de Matemática Física Computación de la Universidad Central “Martha Abreu” de las Villas (UCLV) la publicación del software para que permita a los estudiantes acceder a este desde cualquier computadora conectada a la red universitaria.
- El Grupo de Inteligencia Artificial de la UCLV continuar profundizando las concepciones teóricas metodológicas de esta investigación con el objetivo de diseñar y desarrollar nuevas herramientas informáticas que apoyen al estudiante en el aprendizaje de la asignatura de Programación Lógica.
- Se recomienda a la dirección de la Facultad de Ingeniería del Centro Universitario “José Martí Pérez” el estudio del simulador del proceso de backtracking para una futura inserción en la carrera de Ingeniería Informática.

Referencias bibliográficas

- [1]. Mura, Raúl. Las NTIC y su relación con la educación.[citado 4 de febrero de 2006]. Disponible en:
http://www.elearningamericalatina.com/edicion/abril1/na_2.php
- [2]. García, Luis. Influencia de las NTIC en la enseñanza. Su repercusión en la sociedad. [citado 4 de febrero de 2006]. Disponible en:
<http://www.edutec.es/edutec01/edutec/comunic/TSE10.html>
- [3]. León, Marco. Los softwares educativos. Una Alternativa en la actualidad. [citado 10 de febrero de 2006]. Disponible en:
<http://www.monografias.com/trabajos29/alternativa-ducacion/alternativa-educacion.shtml>
- [4]. Vecino, Fernando. Intervención en el XXI Seminario de Perfeccionamiento para Dirigentes Nacionales de la Educación Superior. La Habana, Cuba. 1999.
- [5]. Castro, Fidel. Al pronunciar las palabras finales del acto oficial de inauguración de los nuevos 300 Joven Club de Computación y Electrónica, 7 de marzo del 2006 [citado 10 de marzo de 2006]. Disponible en:
<http://www.jrebelde.cubaweb.cu/2006/enero-marzo/mar-8/cuba.html>
- [6]. Diccionario de La Real Academia Española. "Simulador". [citado 8 de marzo de 2006]. Disponible en: <http://buscon.rae.es/diccionario/drae.htm>
- [7]. Brend D, Ruben. Historia y Eficacia de la Simulación. [citado 1 de marzo de 2006]. Disponible en: <http://gerentevirtual.com/historia.htm>
- [8]. Alonso, Ileana y González, Lino. Tendencias pedagógicas contemporáneas. [citado 1 de marzo de 2006]. Disponible en:
<http://www.monografias.com/trabajos6/tenpe/tenpe.shtml>
- [9]. Zilberstein Toruncha, José. Procedimientos didácticos que Propician un

aprendizaje desarrollador en la asignatura Ciencias Naturales, Tesis presentada en opción al grado de Doctor en Ciencias Pedagógicas, Ciudad de la Habana, 1996.

[10]. Enciclopedia libre, Wikipedia. Inteligencia Artificial. [citado 15 de febrero de 2006]. Disponible en: http://es.wikipedia.org/wiki/Inteligencia_artificial

[11]. Lezcano Brito, Mateo. Ambientes de aprendizaje por descubrimiento para la disciplina Inteligencia Artificial. Santa Clara, 1998.

[12]. Enciclopedia libre, Wikipedia. PROLOG. [citado 1 de marzo de 2006]. Disponible en: <http://es.wikipedia.org/wiki/PROLOG>

[13]. Pérez, Yasel y Gutiérrez, Eduardo. Trabajo Diploma. Un método para elaborar simulaciones del lenguaje Prolog. Santa Clara, Cuba. 2006

[14]. Enciclopedia libre, Wikipedia. Software Libre. [citado 25 de agosto de 2006]. Disponible en:

http://es.wikipedia.org/wiki/Software_libre

[15]. Barrera, Ivan. Aplicación de Programación Lógica en Java. [citado 5 de septiembre del 2006]. Disponible en:

<http://html.rincondelvago.com/aplicaciones-de-prolog-en-java.html>

[16]. Álvarez, Sofía, Hernández Anaisa. Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML. La Habana, 2000

Bibliografía

1. Alonso, Ileana y González, Lino. Tendencias pedagógicas contemporáneas. [accedido el 3 de marzo de 2006]. Disponible en:
<http://www.monografias.com/trabajos6/tenpe/tenpe.shtml>
2. Álvarez, Sofía, Hernández Anaisa. Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML. La Habana, 2000
3. Barrera, Ivan. Aplicación de Programación Lógica en Java. [accedido el 12 de septiembre del 2006]. Disponible en:
<http://html.rincondelvago.com/aplicaciones-de-prolog-en-java.html>
4. Brend D, Ruben. Historia y Eficacia de la Simulación. [accedido el 9 de marzo de 2006]. Disponible en: <http://gerentevirtual.com/historia.htm>
5. Castro, Fidel. Al pronunciar las palabras finales del acto oficial de inauguración de los nuevos 300 Joven Club de Computación y Electrónica, 7 de marzo del 2006 [accedido el 10 de marzo de 2006]. Disponible en:
<http://www.jrebelde.cubaweb.cu/2006/enero-marzo/mar-8/cuba.html>
6. Diccionario de La Real Academia Española. "Simulador". [accedido el 8 de marzo de 2006]. Disponible en: <http://buscon.rae.es/diccionario/drae.htm>
7. Enciclopedia libre, Wikipedia. Inteligencia Artificial. [accedido el 10 de febrero de 2006]. Disponible en: http://es.wikipedia.org/wiki/Inteligencia_artificial
8. Enciclopedia libre, Wikipedia. PROLOG. [accedido el 10 de diciembre de 2006]. Disponible en: <http://es.wikipedia.org/wiki/PROLOG>
9. Enciclopedia libre, Wikipedia. Software Libre. [accedido el 30 de agosto de 2006]. Disponible en: http://es.wikipedia.org/wiki/Software_libre
10. Fisher, J. Prolog: Tutorial. [accedido el 9 de enero de 2007]. Disponible en:
http://www.csupomona.edu/~jrfisher/www/prolog_tutorial/contents.html

11. García, Luis. Influencia de las NTIC en la enseñanza. Su repercusión en la sociedad. [accedido el 22 de febrero de 2006]. Disponible en:
<http://www.edutec.es/edutec01/edutec/comunic/TSE10.html>
12. Hernández Sampier, Roberto. (2003). Metodología de la investigación 1. Editorial Félix Varela. La Habana.
13. Hernández Sampier, Roberto. (2003). Metodología de la investigación 2. Editorial Félix Varela. La Habana.
14. Introducción a Java J2SE. [accedido el 9 de septiembre de 2006]. Disponible en: <http://diario.de/medianero.script/introduccion-a-java-j2se.html>
15. Jacobson, Ivar, Booch, Grady. (2000). El proceso Unificado de desarrollo de Software. PEARSON EDUCACIÓN, S, A. Madrid.
16. Lenguaje de programación Java. [accedido el 17 de noviembre de 2006]. Disponible en:
http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java.
17. León, Marco. Los softwares educativos. Una Alternativa en la actualidad. [citado 4 de febrero de 2006]. Disponible en:
<http://www.monografias.com/trabajos29/alternativa-ducacion/alternativa-educacion.shtml>
18. Lezcano Brito, Mateo. Ambientes de aprendizaje por descubrimiento para la disciplina Inteligencia Artificial. Santa Clara, 1998.
19. López, Ignacio. NetBeans. [accedido el 3 de septiembre de 2006]. Disponible en: <http://ame.endesa.es/confluence/display/AMEDev/NetBeans>
20. Mura, Raúl. Las NTIC y su relación con la educación. [citado 19 de febrero de 2006]. Disponible en:
http://www.elearningamericalatina.com/edicion/abril1/na_2.php
21. Pérez, Yasel y Gutiérrez, Eduardo. Trabajo Diploma. Un método para elaborar simulaciones del lenguaje Prolog. Santa Clara, Cuba. 2006

22. Salinas Caro, Patricio. Tutorial de UML. [accedido el 7 de septiembre del 2006]. Disponible en: <http://www.dcc.uchile.cl/~psalinas/uml/introduccion>.
23. Tuckler, Eddy. Temas generales de Prolog [accedido el 20 de marzo de 2006]. Disponible en:
<http://www.monografias.com/trabajos5/prolog/prolog.shtml>
24. Ugaz, Max. Lo nuevo de rational rose 2000. [accedido el 16 de septiembre del 2006]. Disponible en:
<http://www.usmp.edu.pe/publicaciones/boletin/fia/info36/proyectos.html>
25. Vecino, Fernando. Intervención en el XXI Seminario de Perfeccionamiento para Dirigentes Nacionales de la Educación Superior. La Habana, Cuba. 1999.
26. Zilberstein Toruncha, José. Procedimientos didácticos que Propician un aprendizaje desarrollador en la asignatura Ciencias Naturales, Tesis presentada en opción al grado de Doctor en Ciencias Pedagógicas, Ciudad de la Habana, 1996.

Anexos

Anexo 1. Encuesta realizada a los estudiantes

Estudiante:

Por estar usted recibiendo la asignatura Programación Lógica ha sido seleccionado para responder algunas preguntas que permitirán conocer los problemas que existen con el aprendizaje del proceso de backtracking en la Programación Lógica. La sinceridad de sus respuestas garantizará la realidad de los resultados. Está encuesta es anónima.

Objetivo: Caracterizar el aprendizaje del proceso de backtracking de la asignatura Programación Lógica.

MARQUE CON UNA "X"
1- ¿Cómo es tu interés por el tema de la programación lógica? a) Muy Bueno ____ b) Bueno ____ c) Regular ____ d) Ninguno ____
2- Cómo consideras que es el estudio de la asignatura a) Fácil ____ b) Manejable ____ c) Difícil ____
3- El cubrimiento bibliográfico de la asignatura es: a) Escaso ____ b) No muy abundante ____ c) Abundante ____
4- La comprensión del proceso de backtracking te resulta a) Fácil ____ b) Normal ____ c) Difícil ____ ¿Porqué? _____
5- Cómo consideras que es la orientación del estudio al proceso de backtracking a) Buena ____ b) Regular ____ c) Insuficiente ____ ¿Porqué? _____
6- Conoce usted algún recurso informático accesible para comprender mejor el proceso de backtracking Si ____ No ____
7- Si su respuesta es afirmativa seleccione que tipo de recurso informático ha utilizado a) Tutorial Inteligente ____ b) Simulador ____ c) Entrenador ____ d) Otros ____

Anexo 2. Guía de la Entrevista a profesores que han impartido la asignatura de Programación Lógica.

Objetivo: Caracterizar la situación actual de la enseñanza de la asignatura Prolog y del tema Listas.

Preguntas:

1. ¿Qué tipos de materiales orienta para el estudio independiente?
2. ¿Cómo usted considera el interés de los estudiantes por la asignatura?
3. ¿Cómo asimilan los estudiantes el contenido de la asignatura?
4. ¿Cómo es la comprensión del proceso de backtracking por parte de los estudiantes?
5. ¿Qué métodos complementarios utiliza en la clase para apoyarse?

Anexo 3. Guía de la Entrevista para identificar el negocio y obtener los requerimientos del software.

Objetivo: Identificar el negocio y obtener los requerimientos del software, tanto funcionales con los no funcionales.

- 1- Quiénes intervienen en el proceso de enseñanza del proceso de backtracking.
- 2- Cómo se realiza la explicación de este proceso por parte de los profesores.
- 3- De qué forma se presentan los ejemplos
- 4- Cómo quisieras ver la traza que genera el código.
- 5- ¿Te gustaría que estuviera presente la explicación del ejemplo?
- 6- Quisieras la oportunidad de ir a pasos anteriores.
- 7- Cómo te gustaría el tiempo de respuesta de la simulación de los ejemplos.
- 8- Sobre qué plataforma te gustaría que funcionara.
- 9- Que tiempo quisieras que estuviera disponible la herramienta.
- 10- Cómo te gustaría que fuera la interfaz.

Anexo 4. Composición del Grupo de Especialistas consultados.

Nombre y Apellidos	Entidad	Labor
Dra. Zenaida García	Universidad Central “Martha Abreu” de Las Villas Facultad Matemática Física y Computación.	Profesora de la asignatura Programación lógica.
Dr. Mateo Lezcano	Universidad Central “Martha Abreu” de Las Villas Facultad Matemática Física y Computación.	Grupo de Ingeniería Artificial
Lic. Maikel León	Universidad Central “Martha Abreu” de Las Villas Facultad Matemática Física y Computación.	Profesor de la asignatura Programación lógica
MSc. Aurelio Hernández Reyes	Centro Universitario “José Martí” de Sancti Spíritus	Profesor Departamento de Ingeniería Informática
Jorge Fardales	Centro Universitario “José Martí” de Sancti Spíritus	Profesor Departamento de Ingeniería Informática
Vladimir Caballero	Centro Universitario “José Martí” de Sancti Spíritus	Profesor Área de Informatización CUSS
Guido Cecilio	Centro Universitario “José Martí” de Sancti Spíritus	Profesor Área de Informatización CUSS

Anexo 5. Guía para la validación del Simulador de Backtracking.

Compañero (a): Usted ha sido seleccionado por su experiencia en la enseñanza superior, como dirigente y por el nivel docente metodológico que posee, para que dé sus valoraciones sobre la propuesta: “Simulador para el aprendizaje del proceso de backtracking”. Sobre la propuesta usted podrá expresar su criterio teniendo en cuenta los siguientes aspectos:

1. Necesidad de su introducción: (actualidad del tema, finalidad)
2. Fiabilidad funcional: (eficiencia, facilidades de uso)
3. Aspectos técnicos – estéticos (calidad de los contenidos, navegación, interacción, originalidad y tecnología de avanzada)
4. Aspectos psicológicos: (Atractivo, adecuación a los destinatarios)

Datos del evaluador

Nombre y apellidos: _____

Institución: _____

Teléfono: _____

Correo-e: _____