**Boletín de Informática Educativa** Vol 2, No, 2, 1989
UNIANDES - LIDIE pp.105-117

# PROLOG:
# REFLEXIONES SOBRE SU POTENCIAL EDUCATIVO

**Germán Ricardo Hernández E.**

---

## RESUMEN

La aplicación de PROLOG como herramienta educativa tiene cada vez más una creciente popularidad. Por ello es necesario analizar exhaustivamente las ofertas del lenguaje (a) para construir sistemas de apoyo a procesos educativos, y (b) para convertirse en la base de ambientes de aprendizaje por sí mismo. En el presente artículo se presentan las principales características del lenguaje, como fundamento para analizar su potencialidad de acuerdo con diversos enfoques educativos, así como las peculiaridades que lo convierten en candidato para una exploración más a fondo. Para esto se identifican el tipo de "ideas poderosas" que a través del uso de PROLOG se pueden desarrollar en el alumno, entre las que sobresale la especificación de lo que se desea resolver en una forma declarativa, como contraste a los enfoques hasta ahora promovidos que se concentran en elementos netamente procedimentales. Finalmente se plantean algunos puntos de reflexión que deben ser tenidos en cuenta por quienes, en creciente número, se encuentran interesados en las aplicaciones educativas de PROLOG.

## INTRODUCCION

La popularidad del lenguaje de programación PROLOG ha venido creciendo así como el interés y expectativas de la utilización educativa de la informática debido a la potencialidad que tiene el computador como mecanismo de mediatizar y mejorar el proceso de enseñanza. Esto lleva a quienes se encuentran interesados en esta área a plantearse diversas preguntas acerca del potencial educativo de PROLOG: ¿Cuáles son las diversas ofertas o aplicaciones educativas del lenguaje?, ¿Qué hace de este lenguaje una herramienta interesante para su utilización como ambiente de aprendizaje al estilo del uso que se ha dado a LOGO?, ¿Qué ofertas educativas exclusivas provee PROLOG?.

Para intentar una respuesta entendible a interrogantes como los anteriores, es necesario conocer un poco las características de PROLOG. Por este motivo, para quienes no las conocen, se ha preparado una breve introducción a PROLOG como punto de partida en este trabajo. Quienes ya conozcan del lenguaje pero se interesen por explorar sus usos educativos, pueden obviar la lectura de la sesión introductoria a ir directamente al cuerpo del artículo.

Boletín de Informática Educativa Vol 2, No, 2, 1989
UNIANDES - LIDIE pp.105-117

# UNA INTRODUCCION A PROLOG

PROLOG es un lenguaje de programación desarrollado bajo las ideas matemáticas propuestas por Kawasaki en la Universidad de Edimburgo y aplicadas prácticamente por Colmerauer en la Universidad de Marsella en los años setenta. Este lenguaje ha tenido gran popularidad en los medios académicos europeos para el desarrollo de prototipos de sistemas inteligentes y ha captado gran interés, entre otras cosas, por haber sido promovido por los japoneses dentro del proyecto que busca desarrollar la "Quinta Generación" de computadores, por su sencillez en cuanto hace al reducido número de mecanismos que lo soportan, por el fuerte bagaje matemático relacionado con la lógica y por el renovado enfoque de la programación que promulga.

## Los fundamentos matemáticos de PROLOG

El sólido soporte matemático de PROLOG se hace evidente en el hecho de que gran parte de las cláusulas que conforman un programa corresponden, bien sea a expresiones de la lógica proposicional o a la de predicados de primer orden, para los cuales existe una clara semántica definida por las operaciones lógicas. Un programa se compone de un conjunto de hechos (cláusulas incondicionales) y reglas (cláusulas condicionales), sobre las cuales se realiza un proceso de inferencia que busca la verificación de determinados objetivos, lo que se constituye en su paradigma fundamental de computación. Bajo este enfoque un programa en PROLOG se puede considerar como una teoría que representa el mundo que se está modelando, sobre la que se busca la demostración de teoremas (objetivos) que se desprendan de los axiomas y las reglas de derivación que el programa mismo establece.

A manera de ejemplo, un programa en PROLOG podría ser:

```
humano(X) :- hombre(X).        (1)
humano(X) :- mujer(X).         (2)
mujer(Juana).                  (3)
```

en él se definen cláusulas condicionales o reglas (1 y 2) que establecen las entidades que pertenecen a la clase de los humanos como todas aquellas entidades que sean hombres o mujeres, una cláusula incondicional o hecho (3) que establece que una entidad específica (Juana) como perteneciente a la clase de las mujeres. En este caso X denota una variables que puede tomar distintos valores ("instanciarse") asociados con entidades específicas.

Si a este programa se establece un objetivo (lo que se desea probar) tal como:

```
humano(Juana).
```

X se "instancia" con Juana (i.e., toma ese valor); PROLOG trata de demostrar si Juana es un hombre por la cláusula primera y, al no poder hacerlo, trata de demostrar que es una mujer, mediante la cláusula segunda; esto se consigue a través de la cláusula incondicional 3, verificando que se cumple el objetivo establecido inicialmente.

## Los mecanismos fundamentales que sirven de base a PROLOG

Una de las características que hacen de PROLOG un lenguaje de gran interés es su reducido número de mecanismos de soporte, entre los que se encuentran el reconocimiento de patrones, la unificación, el reintento ("backtracking"), el manejo de listas y de estructuras de datos flexibles, así como la recursividad.

El mecanismo de reconocimiento de patrones permite identificar las clausulas condicionales o incondicionales, que son de utilidad para demostrar el objetivo que se persigue verificar en determinado momento dentro del proceso de inferencia.

La unificación define un procedimiento para realizar substitución o "instanciación" de parámetros. Este procedimiento se fundamenta en reglas como las siguientes :

- Una constante se unifica con la misma constante.
- Una variable se instancia con cualquier constante.
- Dos variables no instanciadas al unificarse, quedan compartidas; es decir, compartirán el mismo valor una vez que alguna de las dos se instancie con una constante.

El "reintento" (backtracking) permite buscar alternativas para la satisfacción de un objetivo una vez que se llega a la situación en la que no es posible continuar el proceso de inferencia y aún no se ha logrado demostrar lo que se quiere, o no se han obtenido todos los resultados deseados. El mecanismo provee entonces una nueva secuencia que aunque probablemente no es tan promisoria como la anterior, lleva al menos a un resultado adecuado. Eso muestra cómo el orden de las cláusulas dentro de un programa puede tener incidencia en el proceso que se sigue y por ende en los resultados obtenidos.

Adicionalmente, PROLOG ofrece una opción que permite modificar el comportamiento básico del reintento (backtracking): el CUT (o corte), y se representa en un programa por el símbolo "!" . Su uso puede estar destinado a mejorar la eficiencia del programa o puede alterar su significado declarativo. El CUT se usa, por ejemplo, cuando se tienen reglas alternativas que son mutuamente excluyentes: una vez que una de ellas se satisface, no hay mayor razón en perder eficiencia haciendo reintentos sobre las otras, cuando un objetivo no se verifica más adelante. Por ejemplo, si se supone que hay una relación definida por el predicado etapaVida(X,Y), el cual establece una relación discreta entre la edad y la etapa de la vida de los seres humanos, esto se lograría expresar con las siguientes reglas:

```
etapaVida(X,niño):-X<=15.                    (4)
etapaVida(X,joven):-X>15,X<=30.              (5)
etapaVida(X,adulto):-X>30.                   (6)
```

Si a esta base de reglas se plantea un objetivo como

```
etapaVida(7,Y),Y=joven.
```

es claro que la respuesta ha de ser NO ya que este objetivo no se puede demostrar. Sinembargo el mecanismo básico de PROLOG intentará probar el objetivo con cada una de las reglas mediante reintentos, después de satisfacer la regla 4 que instancia Y con niño y fallar al tratar de probar la segunda cláusula del objetivo inicial (Y=joven). Al colocar ! como una cláusula atómica al final de cada regla (por ejemplo etapaVida(X,niño):- X<= 15,!.), se garantiza que, una vez satisfecha la primera parte del objetivo a demostrar, si la segunda parte falla, no habrá más reintentos y el resultado será NO sin mayor pérdida de tiempo.

Adicionalmente, tal como lo ofrecen otros lenguajes como LOGO y LISP, PROLOG incluye estructuras de listas dinámicas que le dan gran poderío y flexibilidad; en ellas se puede hacer referencia explícita a un elemento de una lista o al resto de ella (cola). Por ejemplo

```
miembro(Elemento,[Elemento|_]).              (7)
miembro(Elemento,[_|Lista]):-miembro(Elemento,Lista).  (8)
```

define las reglas que permiten verificar si Elemento es parte de una lista o no. En esta definición de lo que significa ser miembro de una lista se está empleando otra característica de PROLOG: las variables anónimas, que representan a aquellas que no poseen nombre debido al desinterés que el programador tiene en su contenido.

Finalmente, PROLOG permite definiciones recursivas (p.ej., como la de miembro) y estructuras de datos que, al igual que en el caso de LISP y LOGO, corresponden a una notación uniforme con los programas; esto que permite que un programa se modifique a sí mismo durante el proceso de ejecución y que genere como resultado un comportamiento no determinístico, es decir que no siempre genere los mismos resultados ante las mismas entradas.

La interpretación procedimental y declarativa de los programas en PROLOG.

Hasta el momento, en esta presentación se ha hecho referencia indiscriminada a dos aspectos diferentes que están involucrados en un programa PROLOG: lo que significan las relaciones que conforman el programa y cómo es ejecutado el programa. Sinembargo, una diferenciación de estas dos facetas facilita que el programador se concentre en QUE es lo que representa el programa, antes que enCOMO es que el interpretador PROLOG

produce resultados. Esta dualidad presenta una peculiaridad de PROLOG que, a diferencia de enfoques netamente procedimentales como los promovidos por lenguajes imperativos o aplicativos (PASCAL, BASIC, LISP, LOGO), ofrece un enfoque declarativo en el que son abstraidas las consideraciones respecto del control.

Desafortunadamente este enfoque declarativo no es siempre suficiente para el desarrollo de aplicaciones prácticas de alguna envergadura, ya que se requiere hacer uso de mecanismos que modifiquen el control provisto por PROLOG para la obtención de los resultados de un programa. La existencia de predicados de control, implementados en el lenguaje, destruye la homogeneidad provista por la lógica y le introduce elementos que desvirtúan la caraterística de ser medio de Programación en Lógica. Dentro de estos predicados se encuentran el CUT y un conjunto de predicados de corte netamente procedimentales como aquellos de entrada y salida (READ, WRITE, NL, etc.).

## PROLOG vs. Programación Lógica

PROLOG, a pesar del claro soporte matemático, presenta discrepancias con algunos aspectos establecidos en la lógica proposicional y de predicados. Clocksin y Mellish [1] identifican entre otras: diversos predicados conllevan información de control; se presentan situaciones en la que se contradice la naturaleza autocontenida del cálculo de predicados al permitir a PROLOG adicionar nuevos axiomas durante la ejecución de un programa; es posible que una variable lógica pueda tomar el valor de una proposición que hace parte de un axioma; y finalmente, la implementación de la negación en PROLOG, no corresponde a la semántica de dicha operación en la lógica formal.

## PROLOG como lenguaje de desarrollo de sistemas inteligentes

Vistos los fundamentos de PROLOG, cabe preguntarse las razones por las cuales es aplicable para el desarrollo de sistemas inteligentes. Sinembargo, antes de responder esta pregunta se debe intentar solucionar ¿qué es un sistema inteligente?. Esta es sin lugar a dudas una pregunta difícil de responder. Campbell [2] hace un esfuerzo por caracterizar un amplio espectro de sistemas que proveen diversos grados de "inteligencia". Para ello identifica tres atributos que, combinados en distintas formas, producen diferentes niveles en este tipo de sistemas.

Un primer atributo es el comportamiento presentado por el sistema en lo que se refiere a la demostración de habilidades que comúnmente están asociadas con las que realiza el ser humano. Esta situación puede llevar a lo que el autor denomina la "falacia M-C": los humanos se comportan en forma C; usando un método M se puede construír un mecanismo que se comporta de modo C; de esto se desprende que los humanos hacen uso de M al comportarse de ese modo C.

Un segundo atributo de un sistema inteligente es la aplicación de técnicas de programación propias de la Inteligencia Artificial en su construcción, entre las que se

encuentra el "reintento" como uno de los principales mecanismos hasta ahora aplicados, y que se ajusta a múltiples ejemplos de comportamiento inteligente que encajan dentro de esta estrategia de solución de problemas. Esto ha llevado a proponer que el principio fundamental de la Inteligencia humana es un proceso parecido al del "reintento", fundamentado en lo que se ha dado en llamar el principio de la búsqueda heurística [3]. Otra técnica propia de la Inteligencia Artificial y de los lenguajes que permiten más adecuadamente su aplicación, es el procesamiento simbólico, entendido como la organización conceptual de modelos que se construyen en torno a entidades acerca de las que existen "trozos" (chunks) de información y sobre los que realizan diferentes tipos de procesamiento. Este que no es un concepto plenamemente definido, pero ha servido de fundamento para la aplicación tradicional de enfoques de la Inteligencia Artificial fundamentados en la búsqueda heurística.

Finalmente, un tercer atributo propio de un sistema inteligente es su capacidad de aprender o de automodificarse a lo largo de su ejecución, lo cual se obtiene a través de la eliminación de la barrera entre datos y programa, existente en lenguajes tradicionales (Fortran, Pascal, Cobol, etc).

Si se aceptan los atributos mencionados como los que hacen a un sistema inteligente, son claras las razones que convierten a PROLOG en un lenguaje apropiado para su desarrollo: su soporte directo al "reintento", la uniformidad del programa y los datos, la capacidad de construir sistemas que muestren comportamiento similar al de los humanos mediante el uso de los mecanismos de reconocimiento de patrones, la unificación y el mismo "reintento", el soporte a esquemas recursivos y al procesamento simbólico a través de las estructuras jeráquicas y recursivas.

## LAS POSIBILIDADES EDUCATIVAS DE PROLOG

Briggs[4] presenta diferentes alternativas para emplear PROLOG con fines educativos:

— Como herramienta de desarrollo de sistemas expertos.
— Como lenguaje de desarrollo de sistemas para la construcción de sistemas expertos.
— Como lenguaje de desarrollo de aplicaciones educativas.
— Como lenguaje de desarrollo de sistemas para la construcción de aplicaciones educativas.
— Como soporte para la enseñanza de programación lógica.
— Como lenguaje de programación.

La discusión siguiente toma en cuenta esta categorización, pero se desarrolla alrededor de tres tópicos : los Sistemas Expertos (SE) y las herramientas de construcción de SE, las aplicaciones y herramientas de desarrollo en PROLOG, y la enseñanza de PROLOG y la programación lógica.

Boletín de Informática Educativa Vol 2, No, 2, 1989
UNIANDES - LIDIE pp.105-117

Los sistemas expertos y las herramientas de construcción de sistemas expertos

Un sistema experto es un sistema computacional que hace uso del conocimiento de experto para obtener niveles adecuados de rendimiento en un área o dominio de conocimientos reducido [5]. Por su parte una herramienta de construcción de sistemas expertos, también conocida como "concha" (shell), provee diversos mecanismos (lenguajes, paradigmas de consulta, esquemas de representación y de inferencia) para el acopio, validación y tratamiento del conocimiento que conforma el sistema experto.

Con respecto a las ofertas de los sistemas expertos para la educación, distintos autores [6][7][8] han reseñado sus posibilidades y requerimientos. Dentro de las posibilidades se destaca la transparencia de los mecanismos de razonamiento seguidos para la solución de un problema por parte del sistema experto y de la estructura de los conocimientos que lo conforma. En cuanto a los requerimientos para la aplicación educativa de los sistemas expertos, se requiere proveerlo de mecanismos que le otorgen una verdadera utilidad educativa.

Una alternativa para promover utilidad educativa consiste en adicionar al cuerpo de conocimientos que componen el sistema, elementos de manejo del diálogo educativo con el alumno, mediante la inserción de conocimiento acerca de diversas estrategias instruccionales y dialogales que varían a lo largo del proceso de interacción según un modelo dinámico del alumno; éste representa lo que el aprendiz conoce en determinado momento. Sinembargo esta estrategia empleada en GUIDON [9], un sistema desarrollado para darle uso educativo a MYCIN [10], el cual a su vez es un sistema experto para diagnóstico de enfermedades, presenta desventajas por cuanto el razonamiento que sigue el sistema no se realiza necesariamente de acuerdo con la misma línea o estrategia empleada por el experto humano.

Situaciones como la anterior han llevado a la aplicación de enfoques alternativos, como por ejemplo hacer explícito el conocimiento acerca del proceso de diagnóstico y apartarlo del conocimiento existente acerca de una enfermedad específica. Esta diferenciación permite la aplicación de estrategias generales de diagnóstico más ajustadas a las empleadas por un facultativo, lo que se refleja en una más fácil transferencia del conocimiento hacia el alumno, y ha sido aplicada para el desarrollo de NEOMYCIN[11], una versión ajustada de MYCIN que, como en el caso anterior, es usada por GUIDON, el cual maneja educativamente el diálogo con el usuario.

El grado de control sobre el proceso de interacción con el sistema experto determina diversos enfoques educativos con que éste se puede emplear.

Bajo un esquema cerrado (pero dinámico), el sistema se convertiría en un componente de un Sistema Inteligente de Tutoría en el que se llevan a cabo diversas estrategias instruccionales que promueven sea aprendizajes memorísticos, o también, a partir de

ejemplos o reglas, analogías o descubrimiento.

Bajo esquemas que provean menor control sobre la interacción por parte del alumno, su uso sigue un enfoque más cercano al de un simulador, en el que se ofrece al alumno un modelo externo que induce a un aprendizaje de tipo experiencial y conjetural que apoya un proceso de descubrimiento, el cual se ve complementado con las opciones de explicación del razonamiento seguido por el sistema.

En cuanto a las herramientas de construcción de sistemas expertos, éstas proveen un ambiente para la formalización del conocimiento, haciendo uso de esquemas de representación (reglas, marcos, scripts, lógica, etc.) y mecanismos de inferencia predeterminados; ésto exige un profundo dominio del área de desarrollo por parte de quien construye el sistema, para poder llegar al nivel de formalización que se requiere. Ejemplos de herramientas de desarrollo programadas en PROLOG son el APES, ADVISOR, SLOTS y muchos otros que cada día se ofrecen en mayor número [12].

Las herramientas de construcción de sistemas expertos en manos del profesor permiten crear sistemas expertos en un área determinada, así como micromundos externos a la concepción del alumno con las posibilidades educativas ya mencionadas. En manos del estudiante estas herramientas se convierten en un mecanismo de creación y utilización de sus propios micromundos, con las restricciones y posibilidades establecidas por las características inherentes al paquete general respecto al esquema o esquemas de representación y mecanismos de inferencia soportadas, facilidades de desarrollo, interfaces provistas, etc.

La aplicabilidad de PROLOG para el desarrollo de este tipo de sistemas se desprende de lo mencionado anteriormente al presentar las características que lo hacen una herramienta que soporta adecuadamente el desarrollo de sistemas inteligentes.

## Aplicaciones y herramientas de desarrollo en PROLOG

Debido a que PROLOG es un lenguaje de aplicación general, es posible utilizarlo para el desarrollo de sistemas de corte más "tradicional" (i.e. herramientas de productividad y paquetes específicos), los cuales igualmente poseen utilidad educativa [13]. PROLOG ofrece propiedades tales como la abstracción de los mecanismos de control (condicional, repetitivo) a través del "reintento" que lo convierten en una herramienta para la construcción rápida de estos sistemas [14]. Sinembargo, el costo a pagar es la eficiencia del producto, generada por una aplicación indiscriminada de un mecanismo general que no explota las características mismas de la aplicación que se ha desarrollado.

PROLOG por otra parte, al manejar homogéneamente las instrucciones de un programa y sus datos, se convierte en un vehículo apropiado para la construcción de herramientas de desarrollo de aplicaciones, análogas a las herramientas de construcción de sistemas expertos mencionadas anteriormente, pero que se diferencian en que, en los sistemas

expertos, su contenido se refiere a la experiencia de experto, en tanto en el caso de las aplicaciones se refiere más al conocimiento en un dominio de conocimiento determinado no necesariamente asociado a un tipo de "experiencia".

Nichos et. al.[12] enumeran varios proyectos que se vienen adelantando en la línea de herramientas de desarrollo de aplicaciones educativas y su incorporación en el currículo para la enseñanza de diversas asignaturas. Estos sistemas construidos por el PEG (PROLOG Educación Groups de la Universidad de Exeter- Gran Bretaña) incluyen entre otros:

DETECT, un sistema que permite actividades de descubrimiento, obtención y verificación de evidencias, ha sido empleado en la enseñanza de Historia y de estrategias generales de solución de problemas; PLACES, un sistema para la organización coherente de información acerca de lugares geográficos; PLAN, un sistema para la escritura de aventuras que permite al usuario crear su propio mundo imaginario sobre el cual puede explorar y desarrollar habilidades cognitivas de tipo general como la concepción y ejecución de planes, y diversas estrategias de solución de problemas; LINX, un sistema que provee los mecanismos para construír simulaciones u programas que requieran la aplicación de criterios de decisión y una estructura de consulta ramificada como la que se aplica en proceso de identificación de diversos tipos de evidencia; y SLOTS, un sistema que provee soporte para la construcción verificación y procesamiento de conocimiento.

Al igual que en el caso de las herramientas de construcción de sistemas expertos, estos sistemas de desarrollo ofrecen un ambiente apto para la construcción y utilización, por parte del alumno, de diversos tipos de micromundos para jugar juegos de aventuras, realizar simulaciones o manipular información estructurada, y por parte del profesor para construír modelos que el estudiante debe descubrir.

## La enseñanza de PROLOG y de la Programación Lógica

Las diversas ofertas educativas de PROLOG hasta ahora enumeradas no le son propias exclusivamente a PROLOG; aún más, sistemas como MYCIN, NEOMYCIN y GUIDON han sido desarrollados en INTERLISP, otro lenguaje utilizado en el campo de la Inteligencia Artificial. En este sentido PROLOG es otro lenguaje de programación que permite una construcción rápida y adecuada de diversos tipos de sistemas que dadas las condiciones necesarias, pueden ofrecer gran potencial educativo.

Donde sí resultan singulares las características inherentes a PROLOG es en su utilización como ambiente de enseñanza, es decir como un sistema heurístico abierto, que hace uso de las características propias del lenguaje para, a través de la enseñanza de su programación, desarrollar habilidades cognitivas y metacognitivas en el alumno, aplicando un enfoque similar al popular uso de LOGO con fines educativos.

Papert [15] en su propuesta de la utilización del LOGO como medio para desencadenar en el alumno un proceso en el que se haga consciente acerca de sus propios procesos mentales, resalta lo que denomina "el principio de la popularización epistemológica" que establece que el conocimiento es descomponible, característica análoga a la modularidad en el campo de la informática. LOGO con su soporte a los procedimientos puros, se convierte entonces en un agente a través del cual el estudiante se puede constituir en constructor de nuevas teorías. Elemento central de este enfoque educativo es la noción de "Ideas Poderosas" , es decir ideas que una vez aprehendidas por el estudiante en un dominio de conocimiento, pueden ser transferidas o generalizadas para su aplicación en otros.

De otra parte, tomando "el principio de la búsqueda heurística" como fundamento de un comportamiento inteligente, las razones esgrimidas para la aplicación de PROLOG en la construcción de sistemas inteligentes dan piso para que se considere que, haciendo uso apropiado de estos mecanismos, el alumno pueda comprender la naturaleza de un dominio específico y del proceso de pensamiento, además de aprehender "ideas poderosas" en forma similar a lo que se reclama que se consigue a través de LOGO.

El potencial de los lenguajes de programación para la generación de "ideas poderosas" por parte del alumno no se ve circunscrito a aquellos comúnmente asociados con desarrollos en el campo de la Inteligencia Artificial. Las técnicas generales de programación y las peculiaridades de lenguajes específicos pueden colaborar en el desarrollo cognitivo del alumno [16]. Así por ejemplo, el aprendizaje de técnicas generales de programación puede colaborar en la generación de "ideas poderosas" tales como:

- Planeación de acciones independientemente de su ejecución.
- Concientización acerca del proceso mismo de pensamiento.
- Independencia entre los procedimientos y los datos.
- Interpretación de las fallas en el programa como situaciones a corregir de las cuales aprender.
- Concepción de variable como mecanismo de representar múltiples instancias a ser consideradas como una misma idea.
- Iteración (para ... haga…).
- Selección de múltiples escogencias (si ... entonces ... si no...).

Por su parte, el aprendizaje de técnicas de programación estructurada promueve la aprehensión de nuevas ideas como:

- Aplicación de la estrategia de "divide y vencerás" para la solución de problemas complejos.
- La denotación sensible de conceptos.
- La aplicabilidad de la recursión.

De otra parte LOGO a través de los micromundos sintónicos que ofrece, permite al

alumno:

- Aprehender las reglas que gobiernan el micro mundo.
- Aprehender el comportamiento de los elementos involucrados y su relación con el dominio de los números.
- Aprender a dar instrucciones.
- Manipular listas
- Intercambiar datos y procedimientos, lo que permite la programación automodificable.
- Comprender la recursión en estructuras de datos.

Finalmente PROLOG con su doble enfoque (declarativo y procedimental), además de sus mecanismos de reconocimiento de patrones, reintento, unificación y estructuras de datos jerárquicas y recursivas, permite promover adicionalmente:

- Clara especificación de lo que se desea antes de cómo obtenerlo.
- Especificación de la relación existente entre diversos conceptos.
- Especificación clara de hechos, objetos, relaciones y acciones.
- Clasificación de acuerdo con diversos criterios.
- Manipulación de hechos contenidos en una base de datos.
- Aplicación de estrategias sistemáticas de exploración de alternativas en un espacio de posibilidades.
- Planteamiento de hipótesis y el proceso sistemático a seguir para su verificación.

Este potencial respecto a la promoción de nuevas "ideas poderosas", hace pensar en forma optimista frente a las posibilidades educativas de PROLOG.

Sin embargo, existen dificultades propias del ambiente PROLOG en lo que hace a su sintonía [7], esto es, a la naturalidad con que el alumno puede emplearlo en su interacción, lo que establece una nueva diferencia con LOGO. En efecto, se han reportado problemas considerables en procesos de edición y depuración de los programas por parte de los alumnos[17]. Esta situación genera una aparente contradicción con afirmaciones en el sentido de que programar en PROLOG es fácil, ya que consiste en la expresión de lo que se desea realizar en forma más natural [18] (p.ej., esta afirmación se ve comprometida por el uso de los CUT que afectan considerablemente la transparencia perceptual del sistema).

La anterior inconsistencia se resuelve al identificar la diferencia entre la naturalidad de la sintaxis y la semántica del lenguaje. PROLOG presenta inconvenientes en cuanto a su sintaxis cuando se le utiliza como medio educativo; ésto ha llevado a la creación de diferentes interfaces más amistosas: Micro-PROLOG, SIMPLE, MITSI, y EMITSI[19], cada una de las cuales persigue solucionar inconsistencias y dificultades presentadas por las otras, permiten una comunicación en un lenguaje más cercano al humano. A manera de

ejemplo un programa expresado en PROLOG como:

gusta(jose,comer). hace_ponques(X):-tiene_receta(X),gusta(X,comer) se programaría en EMITSI (traducido al español) como:

Jose gusta comer.
alguien hace ponques si alguien tiene receta y alguien gusta comer.

Lo anterior representa, sin lugar a dudas, un avance hacia la sintonía del lenguaje.

Finalmente, en cuanto a la enseñanza de programación lógica, PROLOG tiene un gran potencial que se circunscribe esencialmente a la lógica proposicional y de predicados de primer orden, mediante la utilización de un subconjunto del lenguaje en el que los componentes procedimentales estén excluídos y los mecanismos de inferencia empleados por PROLOG no se hagan explícitos [4]. Esta aplicación del lenguaje podría promover la generación de "ideas poderosas" tales como el razonamiento lógico y la identificación de la estructura lógica del micromundo con el que interactúa el estudiante.

## ALGUNOS PUNTOS DE REFLEXION

Aun cuando la presentación de los usos educativos de PROLOG lleva a dar una respuesta afirmativa a la pregunta de si éste tiene algo que ofrecer a la educación, es necesario reflexionar acerca de diversos puntos que son claves para el logro de los objetivos educativos que se persigue lograr mediante su uso.

Respecto a la aplicación de PROLOG para la construcción de sistemas que apoyen actividades educativas, en números anteriores del Boletín de Informática Educativa se han dado criterios específicos para mejor aprovechamiento de esta tecnología [7] [8].

En cuanto al uso como ambiente de aprendizaje, en este mismo número se resaltan variables que deben tomarse en cuenta para la construcción y utilización de software educativo que promueva el desarrollo cognitivo [20] y que, en últimas tienden a lo que Pea [21] denomina una nueva ciencia instruccional para la programación que responda a objetivos educativos que igualmente deberían ser replanteados como resultado de los problemas detectados al usar LOGO bajo este enfoque: problemas de transferencia, dificultades cognitivas con el lenguaje a niveles avanzados, necesidad de apoyo por parte del profesor, entre otros. Estas recomendaciones han de ser aplicadas en el caso de PROLOG dentro del marco de la potencialidad mostrada mediante la identificación de "ideas poderosas" que promueve. Algunas estrategias instruccionales concretas (aproximaciones "de arriba a abajo" y de "abajo a arriba") han comenzado a ser exploradas arrojando resultados que deben ser analizados cuidadosamente para un mejor logro de los fines educativos [19][22].

Un paso adelante en la línea de ambientes de aprendizaje es la construcción de sistemas inteligentes que apoyen una mayor automatización en la provisión de diversas estrategias, donde el lenguaje pueda ser empleado en diferentes modos complementarios: en modo ejecución, en el que el sistema presente al alumno sistemas que pueda manipular; en modo demostración, en el que se le presenten adicionalmente explicaciones de las acciones realizadas y, finalmente, en modo "entrenamiento" (coaching) en el que se analiza el proceso de resolución de problemas seguido por el estudiante y cuando se requiera lo compara con las técnicas empleadas por un experto solucionador de problemas en esta área. Este enfoque, que resulta similar al mencionado para los tutores inteligentes, se diferencia en que tanto el control como la posibilidad de selección de los diversos modos, continúan en las manos del alumno [23].

Otro punto de reflexión acerca del uso educativo de PROLOG (o cualquier otra herramienta) es la dimensión social de la interacción con el computador que comienza a plantear nuevos retos y posibilidades sobre el enfoque de individualización promovido en el pasado. Este interés de la socialización en el uso de las herramientas computacionales, forzado por factores de tipo práctico (p.ej., escasez de recursos) comienza a mostrar los beneficios de la colaboración y la discusión entre compañeros [17].

Desafortunadamente todas estas nuevas aproximaciones se ven seriamente limitadas por la inexistencia de nuevos paradigmas educativos que respondan a objetivos que también requieren de una reformulación. De otra parte surgen dudas acerca del proceso cognitivo mismo del alumno y de la posible distorsión que implica la "falacia C-M" mencionada anteriormente, que lleva a concluir que el proceso de pensamiento del alumno corresponde al modelo de procesamiento de información montado a través de un lenguaje de programación. Si el enfoque consistente en el uso generalizado de lenguajes de programación en el currículo prevalece, se corre el peligro de que "las escuelas ayuden mucho a los estudiantes a pensar como ordenadores. Por contraste, ¿ Quién les enseñar a pensar de otras maneras?, ¿Dónde encajará, por ejemplo el estilo cognitivo que llamamos arte?, ¿Tendrán ahora las escuelas más o menos tiempo y dinero para equilibrar el modelo del "ordenador" del pensamiento?. El arte se transformará en arte LOGO, que, después de todo, existe en el repertorio de la programación de Papert. Si así ocurre, será peor que no enseñar arte en absoluto" [24].

Ante todas las dudas que permanecen, lo único que aparece claro es la necesidad de continuar ahondando cuidadosamente en las posibilidades educativas de PROLOG, lo que requiere de una aplicación apropiada para la generación de resultados igualmente apropiados.

## REFERENCIAS

[1] Clocksin, W.F., Mellish, C.S. (1981) Programming in PROLOG. Berlín: Springer Verlag.

[2] Campbell, J.A. (1984) Three uncertainities of AI. en M.Yazdani y A. Narayanan (Eds.) Artificial Intelligence human effects. Chichester: Ellis Horwood Series.

[3] Newell, A., Simon, H.A.(1976) Computer science as empirical inquiry: symbols and search. ACM Communications, 19, (3).

[4] Briggs, J.H. (1988) Why teach PROLOG ? The uses of PROLOG in education . en PROLOG Children and Students. New York: Nichols publishing.

[5] Waterman, D. (1986). A guide to Expert Systems . Addison Wesley.

[6] Harmon, P., King D. (1985). Expert Systems . John Wiley and sons.

[7] Galvis, A.H. (1988) Ambientes de enseñanza-aprendizaje enriquecidos con computador. en Boletin de Informática Educativa. 1 (2), p.117-139.

[8] Mariño, O. (1988). Informática educativa: tendencias y visión perspectiva. en Boletín de Informática Educativa. 1 (1), p. 11-35..

[9] Clancey, W. (1979) Tutoring rules for guding a case method dialogue . IJMMS No. 11, pg 25-49.

[10] Shortliffe, E.H. (1976) Computer-based medical consultations: MYCIN. New York. American Elsevier.

[11] Clancey, W., Shortliffe, E.H. (1984). Readings in Medical Artificial Intelligence: the first decade. Addison Wesley.

[12] Nichols, J., Briggs, J., Dean, J., O'Connell, K., Raffan, J., Wild, M. (1988). PROLOG tools in education. en PROLOG Children and Students. Nichols publishing.

[13] Hernández, G.R. (1988) Usos educativos de herramientas de productividad. En A.H. Galvis y S.C. Prieto (Editores). Usos educativos de los computadores. Bogotá: SENA.

[14] Hernández, G.R. (1988) An attempt to unify software development methodologies around the idea of design for change . Exeter, U.K! : University of Exeter (Tesis de Grado).

[15] Papert, S. (1981). Desafío a la mente. Editorial Galpago.

[16] Yazdani, M. (1988) Artificial Intelligence Powerful Ideas and Education. Exeter, U.K. : DSC, University of Exeter. Reporte W138.

[17] O'Colburn C., Light, P. (1988) Peers, problem-solving and programming: proyects and prospects. en PROLOG, Children and Students. Nichols publishing.

[18] Bratko, I. (1986). PROLOG Programming for Artificial Intelligence. Addison Wesley.

[19] Cumming, G. Abbott, E. (1988). PROLOG and Expert Systems for children learning. en Ercoli, Lewis (eds), Artificial Intelligence tools in education. Elsevier Science Publishing co.

[20] Escobar, H. (1989) Ambientes computacionales y desarrollo Cognitivo: una perspectiva psicológica. En este número.

[21] Pea, R.D. (1987) LOGO Programming and Problem solving . en Scanlon y O'Shea (Eds.) Educational Computing. John Wiley sons.

[22] Scherz, Z. Maler, O., Shapiro, E. (1988). Learning with PROLOG - A new approach. en PROLOG, Children and Students. Nichols publishing.

[23] Feurzeig. (1985). Algebra slaves and agents in a LOGO-based mathematics curriculum.. Conference AI and Education. Exeter, U.K. : University of Exeter.

[24] Roszak, T. (1988). El culto a la información. Editorial Crítica.