

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

INSTITUTO DE INGENIERÍA
MAESTRÍA EN CIENCIAS EN INGENIERÍA DE SISTEMAS



“IMPLEMENTACIÓN DE UN ALGORITMO GENÉTICO PARA
REGULAR EL INTERCAMBIO DE INFORMACIÓN
HUMANO-COMPUTADORA EN SESIONES DE APRENDIZAJE DEL
LENGUAJE SIGNADO MEXICANO ”

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS EN INGENIERÍA DE SISTEMAS

PRESENTA

JOSÉ CORNELIO BARRAZA BELTRÁN

DIRECTOR

M. en C. RAFAEL VILLA ANGULO

RESUMEN de la tesis que presenta **José Cornelio Barraza Beltrán**, como requisito parcial para la obtención del grado de MAESTRO EN CIENCIAS en INGENIERÍA DE SISTEMAS. Mexicali, Baja California, México. Junio del 2008.

**IMPLEMENTACIÓN DE UN ALGORITMO GENÉTICO PARA
REGULAR EL INTERCAMBIO DE INFORMACIÓN
HUMANO-COMPUTADORA EN SESIONES DE APRENDIZAJE DEL
LENGUAJE SIGNADO MEXICANO**

Resumen aprobado por:



M. en C. Rafael Villa Angulo
Director de tesis

En este trabajo de tesis se presenta el diseño y la implementación de un algoritmo genético para regular el intercambio de información Humano-Computadora en sesiones de aprendizaje automatizado del Lenguaje Signado Mexicano. Implementado bajo un paradigma de enseñanza-aprendizaje conductivista, el algoritmo ofrece al usuario sesiones de aprendizaje tomando la posición de maestro virtual y emulando funciones de un maestro real, tales como: 1) explora en forma dinámica todo el dominio de la información que se desea enseñar al usuario, 2) varía de forma automatizada el contexto de un subconjunto de información con el fin de que el usuario lo aprenda o lo reafirme, 3) modula de forma gradual la incorporación de información nueva al acervo de información ya aprendida por el usuario, 4) es capaz de identificar información olvidada por el usuario e integrarla a la sesión de aprendizaje, y 5) ofrece al usuario la capacidad de presentar un concentrado de toda la información con propósitos de reforzamiento y práctica.

Este documento de tesis está estructurado de la siguiente forma: En el capítulo uno se presentan los antecedentes del proyecto, el planteamiento del problema, y se establecen los objetivos general y específicos para la elaboración de este trabajo. En el capítulo dos se presenta una descripción general de las teorías de aprendizaje, y se analiza el ambiente que rodea la integración social de las personas con discapacidad auditiva y del habla. Igualmente se presenta una introducción a los temas de sistemas tutores inteligentes, algoritmos genéticos y al lenguaje signado mexicano. En el capítulo tres se analizan el paradigma de aprendizaje y la estrategia implementada para la enseñanza del Lenguaje Signado Mexicano. Igualmente se presenta el diseño del algoritmo genético. En el capítulo cuatro se presenta la implementación del algoritmo, las pruebas realizadas para su funcionamiento, y se analizan los resultados. Finalmente, en el capítulo cinco se presentan las conclusiones y aportaciones del trabajo realizado.

Palabras claves: Algoritmos genéticos, Sistemas tutores inteligentes, Lenguaje signado mexicano, Interacción humano-computadora.

ABSTRACT of the thesis presented by **José Cornelio Barraza Beltrán**, as a partial requirement to obtain the MASTER OF SCIENCE degree in SYSTEMS ENGINEERING. Mexicali, Baja California, México. June 2008.

**IMPLEMENTATION OF A GENETIC ALGORITHM TO REGULATE
THE HUMAN-CUMPUTER INFORMATION INTERCHANGE IN
MEXICAN SIGNED LANGUAGE LEARNING SESSIONS**

Abstract approved by:



MSc. Rafael Villa Angulo
Thesis director

This masters thesis work presents the design and implementation of a genetic algorithm for regulating the human-computer information interchange in Mexican Signed Language learning sessions. Implemented under a conductivist learning paradigm, the algorithm takes the position of a virtual instructor and offers to the user learning sessions emulating different aspect of a real instructor, such as: 1) explores in a dynamic way all the information domain that is being taught to the user, 2) is capable of changing automatically the presentation of the information in order to make the user learning under different contexts, 3) modulates is a gradual form the addition of information to the knowledge set already leaned by the user, 4) Identifies information forgotten by the user and integrate it to the learning session, and 5) offers to the user the option of reinforcing the learned knowledge by presenting in a concentrated form all information from the learning set.

This document is organized as follows: chapter one, first presents an introduction to the topic and gives the motivations for the project. Then, presents the problem statement and specifies the general and specific aims accomplished in this work. Chapter two presents a brief description of learning theories, analyzes the factors that intervene in the social integration of deaf people, and gives an introduction to Intelligent Tutoring Systems, Genetic Algorithms, and the Mexican Signed Language. Chapter three describes the learning paradigm adopted in this work, presents the design of the genetic algorithm, and describes the strategies implemented for teaching the Mexican Signed Language. Chapter four presents the implementation of the genetic algorithm, the testing process, and the analysis of results. Finally, chapter five presents the conclusion.

Keywords: Genetic Algorithms, Intelligent Tutoring Systems, Mexican Sign Language, Human Computer Interaction.

Dedicatoria

En el andar de la vida, con los fracasos o exitos que uno tiene a lo largo de ella, no tendrían un significado o carecerían de valor sin ustedes, mi familia, por todo, gracias totales.

A MIS PADRES:

José Barraza Gamez

María Guadalupe Beltrán Calderón

Por su gran amor y apoyo incondicional en cada paso que doy

A MI ESPOSA E HIJOS

Carmen (peque) y mis dos grandes amores, mis hijos Tania
Guadalupe y José Pedro

A MIS HERMANOS:

Reyna, Magui, Candy, Miguel y Vanne

A MIS SOBRINOS:

Ale, Ariel, Dany, Cinthia, Jeny, Justin, Jimena, José Manuel
y Angeline

Agradecimientos

Deseo agradecer de manera especial a mi asesor M.C. Rafael Villa, por compartir su conocimiento en esas platicas largas llenas de ideas y razonamientos, por tener la paciencia, por su apoyo absoluto, y palabras de aliento en cada fase de este trabajo, pero aún más importante por no permitir que me diera por vencido, en verdad gracias.

A los miembros del comité Dra. Sara Ojeda Benitez, M.C. Brenda Leticia Flores Rios, M.C. Gabriela Lozano Olvera y M.I. Moisés Galindo Duarte, por su aportación a este trabajo así como por su valioso tiempo destinado a la revisión del mismo.

A mis compañeros de generación Liz, Rolando, Isabel, Gildardo y a cada uno de mis maestros por dedicar su tiempo y empeño puesto en cada una de sus clases.

En la vida hay personas que de alguna u otra manera en los tiempos difíciles de mi formación siempre estuvieron apoyandome, gracias, Dr. José Gómez Valdés, M.C. José Luis Briseño y M.C. Reynaldo Alanís Cantú por sus enseñanzas y palabras de aliento.

A todos y cada uno de mis amigos por todos los momentos compartidos.

Finalmente y no menos importante a mis cuñadas (July, Carmen) y cuñado (Benito Perez de la O) por todos esos instantes llenos de alegría y buena charla.

Al Instituto de Ingeniería de la Universidad Autónoma de Baja California, por darme el tiempo y espacio para la culminación de este trabajo.

TABLA DE CONTENIDO

Resumen	II
Abstract	III
Dedicatoria	IV
Agradecimientos	V
Lista de figuras	VII
Lista de tablas	IX
1. Introducción	1
1.1. Antecedentes	1
1.2. Planteamiento del problema	3
1.3. Justificación	3
1.4. Objetivos general y específicos	4
1.4.1. Objetivo general	4
1.4.2. Objetivos específicos	5
1.5. Organización de la tesis	5
2. Fundamentos teóricos	7
2.1. Teorías de aprendizaje	7
2.2. Discapacidad auditiva y del habla	8
2.3. Subsistemas de las tecnologías de interacción humano computadora	10
2.4. Sistemas tutores inteligentes	15
2.4.1. Módulo experto	16
2.4.2. Módulo del estudiante	16
2.4.3. Módulo pedagógico	17
2.4.4. Módulo de interfaz	17
2.5. Algoritmos genéticos	17
2.5.1. Codificación	18
2.5.2. Selección	19
2.5.3. Cruzamiento	21
2.5.4. Mutación	22
2.5.5. Funcionamiento de un algoritmo genético simple	23

2.6.	Lenguaje signado mexicano	25
2.6.1.	Dactilología	26
2.6.2.	Ideogramas	27
3.	Diseño e implementación del algoritmo genético	29
3.1.	Paradigma de aprendizaje	30
3.2.	Creación de cromosomas	32
3.3.	Evaluación de aptitud	33
3.4.	Criterio de solución	35
3.5.	Operador de cruzamiento	36
3.5.1.	Selección	37
3.5.2.	Cruzamiento	38
3.6.	Cruzamiento y mutación	38
4.	Evaluación y análisis de resultados	42
4.1.	Implementación del algoritmo genético	42
4.2.	Evaluación del AG implementado	46
4.3.	Resultados y discusión	47
5.	Conclusiones y trabajo futuro	52
5.1.	Conclusiones y aportaciones	52
5.2.	Limitaciones	52
5.3.	Trabajo futuro	53
	Referencias	54

LISTA DE FIGURAS

1.	Sistema de integración social de las personas con discapacidad auditiva y del habla	9
2.	Esquema de desarrollo de sistema de Tecnologías Interacción Humano Computadora (tomado de Hewett <i>et al.</i> , 1996).	12
3.	Subsistemas que influyen en el desarrollo de la herramienta de software. .	14
4.	Componentes de un Sistema Tutor Inteligente.	16
5.	Componentes de un algoritmo genético.	18
6.	Cruzamiento de punto simple (tomado de Beasley <i>et al.</i> , 1993).	21
7.	Cromosoma vista como un ciclo (tomado de Beasley <i>et al.</i> , 1993a).	22
8.	Cruzamiento uniforme (tomado de Beasley <i>et al.</i> , 1993a).	23
9.	Mutación de inversión de <i>bit</i> (tomado de Beasley <i>et al.</i> , 1993a).	23
10.	Etapas de un algoritmo genético (tomado de Ribero Filho <i>et al.</i> , 1994). .	24
11.	Dactilología del alfabeto manual mexicano. (tomado de Fleischmann, 1996)	26
12.	Ejemplo de los ideogramas de las palabras: todo, tienda y comunicación total. (tomado de Fleischmann, 1996)	27
13.	Población de cromosomas de tres individuos.	32
14.	Representación binaria de los símbolos del Alfabeto Signado Mexicano. .	34
15.	Proceso para generar una nueva población.	40
16.	Diagrama de flujo del algoritmo genético utilizado.	41
17.	Vista del inicio de la sesión de aprendizaje del Alfabeto Signado Mexicano.	43
18.	Vista que muestra el primer ciclo de la sesión de aprendizaje.	43
19.	Vista que muestra el segundo ciclo de la sesión de aprendizaje.	44
20.	Vista que muestra el tercer ciclo de la sesión de aprendizaje.	45
21.	Vista que muestra los símbolos del Alfabeto Signado Mexicano.	46
22.	Gráfica de asociación de ademanes en una sesión de aprendizaje de un alumno sin conocimiento previo del LSM.	49
23.	Gráfica de asociación de ademanes en una sesión de aprendizaje de un alumno con conocimiento previo del LSM.	50
24.	Gráfica de aprendizaje de dos usuarios.	51

LISTA DE TABLAS

I.	Muestra del resultado de la función aptitud de una población de tres individuos.	35
II.	Selección de individuos utilizando elitismo.	38
III.	Resultado del proceso de selección de individuos utilizando elitismo.	38
IV.	Población actual.	39
V.	Resultado de la sesión de aprendizaje para un estudiante sin conocimiento previo del LSM.	47
VI.	Resultado de la sesión de aprendizaje para un estudiante con conocimiento previo del LSM.	48

Capítulo 1

Introducción

1.1. Antecedentes

La idea de máquinas inteligentes para propósitos de enseñanza se remonta a 1926 cuando Sydney L. Pressey construyó una máquina con preguntas y respuestas de opción múltiple. Esta máquina desplegaba preguntas y proporcionaba una retroalimentación inmediata al usuario (Thomas, 2000).

Las computadoras han sido utilizadas en la educación por más de veinte años. El Entrenamiento Basado en Computadoras (Computer-Based Training, CBT) y la Instrucción Asistida por Computadora (Computer Aided Instruction, CAI) fueron los primeros sistemas desarrollados como un intento de enseñar utilizando computadoras. En estos tipos de sistemas, la instrucción no fue individualizada a las necesidades de los aprendices. En cambio, las decisiones alrededor de cómo guiar a un estudiante a través del material fueron estilo guión, tales como: “si la pregunta 21 es contestada correctamente proceda a la pregunta 54; de otra forma vaya a la pregunta 32”. las habilidades de los aprendices no eran tomadas en cuenta (Beck *et al.*, 1996).

Aún cuando los sistemas CBT y CAI pueden ser algo efectivos en ayudar a los aprendices, estos no proveen el mismo tipo de atención individualizada que un estudiante recibiría de un tutor humano. Para que un sistema educacional basado en computadora proporcione tal atención, este debe moderar alrededor del dominio y del aprendiz. Esto ha incitado la investigación en el campo de los Sistemas Tutores (Instrucción) Inteligentes

(Intelligent Tutoring Systems, ITS). Los ITS brindan una flexibilidad considerable en la presentación del material y una gran habilidad para responder a las necesidades de la idiosincrasia del estudiante. Estos sistemas logran su “inteligencia” al representar, de manera pedagógica, las decisiones sobre como enseñar, además de generar informes acerca del aprendiz, lo cual permite una gran versatilidad para modificar la interacción del sistema con el estudiante (Beck *et al.*, 1996).

En la última década, con el influjo masivo de las computadoras en la sociedad, la Interacción Humano-Computadora (Human Computer Interaction, HCI) se ha convertido en parte importante de nuestra vida diaria. El objetivo general en la construcción de sistemas HCI, ha sido el migrar las formas naturales que utilizamos los humanos para comunicarnos hacia estos sistemas de interacción entre el humano y la computadora (Pavlovic *et al.*, 1997). Con esta motivación, en los últimos años se ha incrementado el interés por introducir algunas modalidades de comunicación humano-humano, a estos sistemas de comunicación humano-computadora, tal es el caso de los lenguajes signados, utilizados por discapacitados del habla y del oído.

Al implementar el uso de lenguajes signados en un sistema HCI es necesario proveer a la computadora del significado con el cual interpretará dichos ademanes. Para la interpretación de éstos, es necesario que la computadora sea capaz de medir configuraciones dinámicas y/o estáticas de la mano, brazo u otras partes del cuerpo. Estos problemas de medición y reconocimiento han sido abordados en dos líneas diferentes que podemos agrupar como sigue: los trabajos de reconocimiento utilizando técnicas basadas en visión por computadora (Moghaddam y Pentland, 1997; Zhao *et al.*, 1998; Lee y Kim, 1999; Wilson y Bobick, 1999) y los trabajos de reconocimiento utilizando dispositivos basados en guantes (Kramer y Leifer, 1989; Quam, 1990; Fels y Hinton, 1993 y 1998; Villa-Angulo, 2001).

Las técnicas basadas en visión sugieren el uso de un conjunto de videocámaras y técnicas de visión por computadora para reconocer e interpretar los ademanes, mientras que las técnicas utilizando dispositivos basados en guantes, requieren que el usuario porte

un dispositivo que cubra su mano (en este caso un guante). Este dispositivo generalmente está conectado a un conjunto de cables que llevan señales eléctricas u ópticas a la computadora.

1.2. Planteamiento del problema

El trabajo de Villa Angulo (Villa-Angulo, 2001) consistió en desarrollar una interfaz humano-computadora, que basada en un guante de datos, es capaz de capturar los ademanes de las 27 letras del alfabeto en español (Villa-Angulo e Hidalgo, 2001 y Villa-Angulo e Hidalgo-Silva, 2005.). Este trabajo se centró principalmente en el diseño y la construcción de las herramientas físicas (hardware) de la interfaz y la utilización de técnicas de inteligencia artificial para el procesamiento de la información generada por éstas. El sistema, en su estado actual, no está soportado en una herramienta de software estructurada y no contiene los elementos necesarios para ofrecerle a un usuario un ambiente de aprendizaje y práctica del alfabeto signado mexicano.

El primer paso para desarrollar un sistema interactivo de aprendizaje es implementar un algoritmo capaz de emular aspectos de enseñanza de un maestro real, que sea capaz de proveer al usuario de conocimiento y evaluar el grado de aprendizaje para así regular la información que se provee al mismo. Es por esto que en este trabajo se plantea probar la viabilidad de un algoritmo genético para emular funciones de enseñanza de un maestro real.

1.3. Justificación

Actualmente, el proceso de enseñanza del Lenguaje Signado Mexicano (LSM) está basado y soportado por el uso de video cassettes, manuales y profesores de aula, así como sistemas multimediales, en el mejor de los casos existen cursos especializados sobre el tema más estos tienen un alto costo que en consecuencia no es accesible a toda la población que necesita aprenderlo.

Por otro lado, cuando hablamos de la instrucción que se le da a la persona esta se encuentra supeditada a el tiempo y practicas que se destinan a la misma, por lo que el

estudiante se ve limitado para explorar y practicar cuando tiempo sea necesario el LSM. En casos más críticos, las personas no tienen los medios necesarios para poder acceder a clases de este tipo o bien desconocen que existe un medio por el cual pueden ampliar su forma de comunicación con otras personas.

Además de la enseñanza que deben tener los sordomudos es importante que personas que no padecen este problema tengan la forma de poder practicar el LSM sin la necesidad de asistir a un curso formal del mismo. Por ello, se considera necesario sentar las bases para el desarrollo de un tutor inteligente que ayude a las personas interesadas en aprender el LSM a practicarlo cuantas veces lo consideren necesario.

Tanto el trabajo de Villa-Angulo 2001, como los trabajos similares, se han centrado en cuestiones de evaluación técnica de los algoritmos de reconocimiento y clasificación de los lenguajes signados y no en la evaluación del nivel de aprendizaje del usuario final de tales sistemas, por ejemplo, en los sistemas basados en visión por computadora, el enfoque principal ha sido en la optimización del procesamiento de imágenes para que los sistemas trabajen en tiempo real, por otro lado, en los sistemas basados en guantes de datos el enfoque principal ha sido en la evaluación de las técnicas de procesamiento de la información que los instrumentos proveen. Ambos enfoques, se centran en aspectos puramente técnicos del funcionamiento de las herramientas, más no en aspectos de utilidad como herramientas de aprendizaje de los lenguajes signados que puede tener el usuario final.

1.4. Objetivos general y específicos

1.4.1. Objetivo general

El objetivo principal de este trabajo es probar la viabilidad de un algoritmo genético para emular funciones de enseñanza de un maestro real.

1.4.2. Objetivos específicos

- a) Tener la capacidad de explorar y presentar al usuario el dominio de la información que se desea aprenda.
- b) Proveer la misma información al usuario solo variando el contexto cuando este no la aprende.
- c) Aumentar de manera gradual la información que se provee al usuario cuando este halla aprendido la información previa.
- d) Identificar cuando el usuario a olvidado información previamente aprendida e integrar esta en los futuros ciclos de aprendizaje.
- e) Presentar información ya aprendida para reforzamiento.

1.5. Organización de la tesis

Este documento de tesis esta compuesto por cinco capítulos y organizado de la siguiente manera.

En este primer capítulo, se presenta una introducción general del uso de las computadoras en la educación así como la importancia de la interacción humano-computadora. De igual forma, se describe el planteamiento del problema y la justificación del mismo, el objetivo general y específicos que se siguieron para lograr los resultados esperados.

Dentro del capítulo dos, se realiza una descripción general de la teorías de aprendizaje, se analiza el ambiente que rodea la integración social de las personas con discapacidad auditiva y del habla y como esta influenciada por diferentes subsistemas. Se incluye una introducción a los sistemas tutores inteligentes, algoritmos genéticos y al lenguaje signado mexicano.

El capítulo tres, contiene el paradigma de aprendizaje a utilizar y la estrategia que se aplica para que el alumno aprenda el lenguaje signado mexicano. Se plantea el diseño,

evaluación y los criterios de solución del algoritmo genético.

Dentro del capítulo cuatro, se detalla la implementación y evaluación del algoritmo genético propuesto, posteriormente se lleva a cabo una visualización y discusión de los resultados obtenidos.

En el capítulo cinco, se muestran las conclusiones y aportaciones del trabajo realizado así como las limitaciones del mismo. Además, en el trabajo futuro se enuncian algunas propuestas para expandir el trabajo realizado.

Capítulo 2

Fundamentos teóricos

2.1. Teorías de aprendizaje

La teoría del conductismo se concentra en el estudio “caja negra” en el sentido de que las respuestas a estímulos se pueden observar cuantitativamente ignorando totalmente la posibilidad de todo proceso que pueda darse en el interior de la mente. Según Schuman (1996) esta teoría se basa en los cambios observables en el comportamiento del sujeto, centrandose sobre un nuevo patrón de conducta el cuál es repetido hasta que este se realiza de manera automática.

En contraparte, el cognoscitivismo se basa en el proceso de pensamiento que tiene lugar detrás de la conducta. Los cambios se observan en el comportamiento, pero sólo como un indicador de lo que está sucediendo en la mente del alumno (Schuman, 1996). Los teóricos del cognoscitivismo reconocen que una buena cantidad de aprendizaje involucra las asociaciones que se establecen mediante la proximidad con otras personas y la repetición. También reconocen la importancia del reforzamiento, pero resaltan su papel como elemento retroalimentador para corrección de respuestas y sobre su función como un motivador. Sin embargo, aceptando tales conceptos conductistas, los teóricos del cognoscitivismo ven el proceso de aprendizaje como la adquisición o reorganización de las estructuras cognitivas a través de las cuales las personas procesan y almacenan información (Good y Brophy, 1990). Uno de los principales protagonistas en el desarrollo del cognoscitivismo fue Jean Piaget, quién planteó los principales aspectos de esta teoría durante los años veinte (Mergel, 1998).

Finalmente, el constructivismo se sustenta en la premisa de que cada persona construye su propia perspectiva del mundo que le rodea a través de sus propias experiencias y esquemas mentales desarrollados. Este se enfoca en la preparación del que aprende para resolver problemas en condiciones ambiguas (Schuman, 1996).

El constructivismo realista considera la cognición como el proceso mediante el cual el aprendiz eventualmente construye estructuras mentales que corresponden o se acoplan a las estructuras externas de su entorno. En contraparte, el constructivismo radical concibe a la cognición como el proceso que sirve para que el que aprende organice las experiencias del mundo que le rodea en lugar de descubrir la realidad ontológicamente (Mergel, 1998).

El constructivismo se construye sobre el conductismo y el cognoscitivismo en el sentido de que acepta múltiples perspectivas y sostiene que el aprendizaje es una interpretación personal del mundo. En el conductismo y en el cognoscitivismo la evaluación se basa en alcanzar determinados objetivos, mientras que en el constructivismo la evaluación es mucho más subjetiva (Mergel, 1998).

2.2. Discapacidad auditiva y del habla

El ambiente que rodea la integración social de las personas con discapacidad auditiva y del habla a la sociedad es complejo. Desde esta perspectiva, una manera conveniente de atacar este problema es identificando el ambiente en el que se encuentra inmerso, los sistemas con lo que interactúa y la forma que éstos inciden sobre la problemática a resolver.

En consecuencia, la Figura 1 muestra el ambiente del sistema de integración social de las personas con discapacidad auditiva y del habla. Esta integración está influenciada por los siguientes subsistemas:

1. Interacción humana.
2. Entorno social.

3. Gobierno.
4. Economía.
5. Tecnologías de Interacción Humano Computadora.

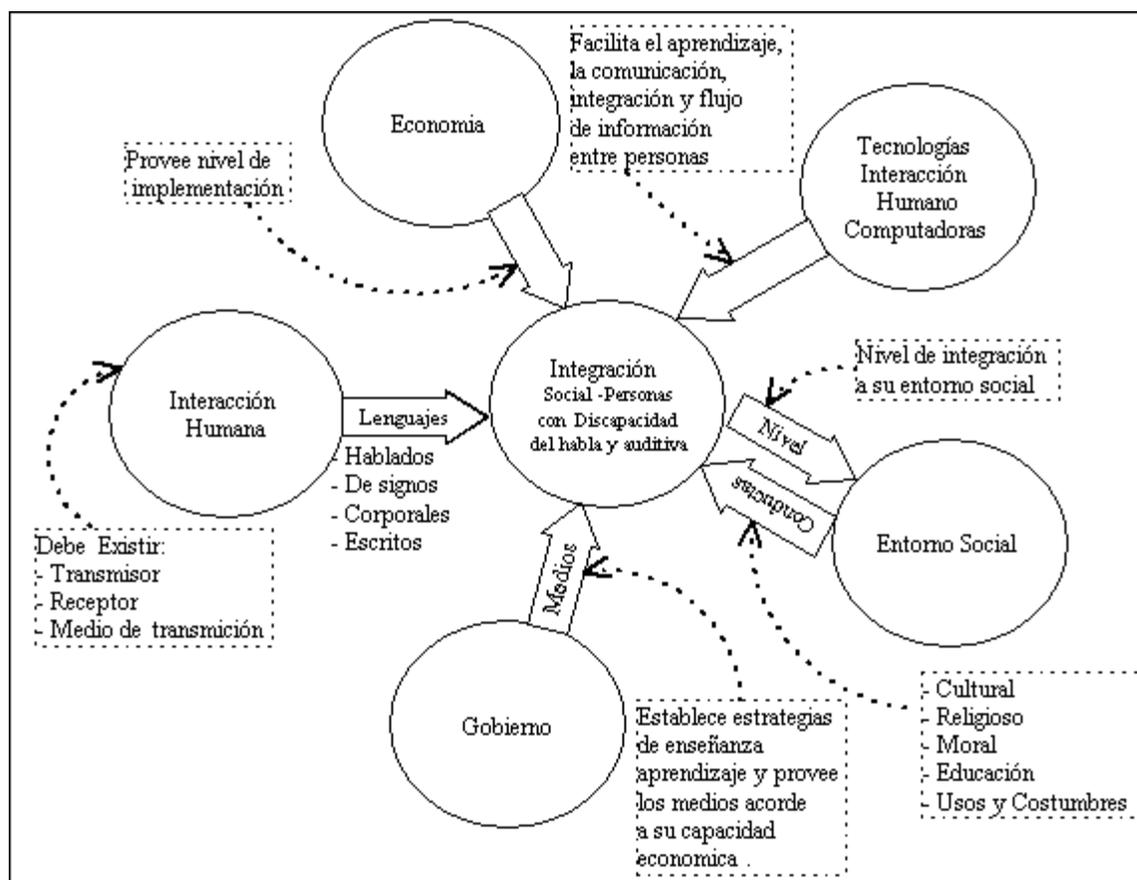


Figura 1: Sistema de integración social de las personas con discapacidad auditiva y del habla

El subsistema de interacción humana está dado por las distintas formas en que los humanos se pueden comunicar, esta comunicación puede ser de manera oral, de signos o manuales, corporales y escritos por mencionar algunos. Para que esta se lleve a cabo, es necesario que exista un transmisor, un receptor y un medio de transmisión del mensaje. Este subsistema influye al sistema de integración social de las personas con discapacidad auditiva y del habla en la forma del nivel de comunicación e interacción que puede existir entre las personas con esta discapacidad con su entorno social.

El subsistema del entorno social, esta integrado por cuestiones culturales, religiosas, morales, de educación, usos y costumbres de la región geográfica donde habita la persona con esta discapacidad. En su totalidad este subsistema forma la conducta de las personas lo cual a su vez impacta de manera positiva o negativa en el sistema de integración social de las personas con discapacidad auditiva y del habla a su entorno social.

El subsistema de gobierno, está integrado por las políticas, estrategias, programas y medios para fomentar y ayudar en la integración de las personas con discapacidad auditiva y del habla a su entorno social.

El subsistema económico, esta formado por la actividad comercial y nivel de riqueza del entorno social donde reside la persona con discapacidad auditiva y del habla. Este subsistema influye en la capacidad de implementación y acceso a los distintos programas o desarrollos tecnológicos que ayuden en la integración social de estas personas.

El subsistema de tecnologías de interacción humano computadora es una disciplina concerniente al diseño, implementación y evaluación de sistemas computacionales interactivos para uso humano, incluyendo el estudio de los fenómenos alrededor de estos. Este subsistema aporta al sistema de integración social de personas con discapacidad auditiva y del habla, herramientas y avances tecnológicos que facilitan la comunicación, integración y flujo de información entre las personas. Este trabajo de investigación, se enfocará en el subsistema de tecnologías de interacción humano computadora, el cual se describe en la siguiente sección.

2.3. Subsistemas de las tecnologías de interacción humano computadora

Partiendo del hecho de que la computadora fué inventada por los seres humanos para realizar de manera automatizada y con cierta autonomía, operaciones para el procesamiento y almacenamiento de información importante para el hombre, se puede decir que implícitamente y de manera natural nació el área de la Interacción Humano Compu-

tadora (Human Computer Interaction, HCI).

Desde una perspectiva de Ciencias de la Computación, el enfoque es en la interacción, específicamente entre uno o más humanos y una o más computadoras. Con el fin de dar una caracterización más rígida sobre la interacción humano-computadora como un campo de estudio, la ACM (Association for Computing Machinery) ofrece una lista de intereses especiales de ésta:

- La interacción humano-computadora se interesa en el rendimiento de la unión de tareas entre humanos y máquinas,
- En la estructura de la comunicación entre humanos y máquinas,
- La capacidad del hombre para el uso de las máquinas (incluyendo el aprendizaje y la adaptación a las interfaces),
- Los algoritmos y la programación de las interfaces,
- La ingeniería referente al diseño y construcción de éstas interfaces, y
- El proceso de especificación, diseño e implementación de las interfaces.

Por lo tanto, la interacción humano-computadora, involucra ciencia, ingeniería y aspectos de diseño. Por tanto, la HCI estudia al hombre y a la computadora en comunicación, ésta debe estar soportada por conocimiento de ambos lados. Por el lado de la computadora, las técnicas de graficación, los sistemas operativos, los lenguajes de programación y los ambientes de desarrollo son relevantes. Por el lado del humano, la teoría de comunicaciones, las disciplinas de diseño industrial y gráfico, la lingüística, las ciencias sociales, la psicología cognoscitiva y el rendimiento humano son relevantes. Y, por supuesto, la ingeniería y los métodos de diseño son igualmente relevantes.

La Figura 2 presenta la interrelación que existe entre algunos de los tópicos que encierra un ambiente HCI. Para hacer una descripción de la Figura 2 puede tomarse como ejemplo que: los sistemas de computadoras existen en un entorno grande de trabajo

social y organizacional (U1). Dentro de éste contexto existen aplicaciones para las cuales se desea emplear sistemas de cómputo (U2). El proceso de poner las computadoras a trabajar significa que los aspectos humanos, técnicos y de trabajo de la aplicación deben ser unidos entre sí de tal manera que el sistema provea formas de adaptabilidad, que ofrezca vías de aprendizaje al humano, u otro tipo de estrategias (U3).

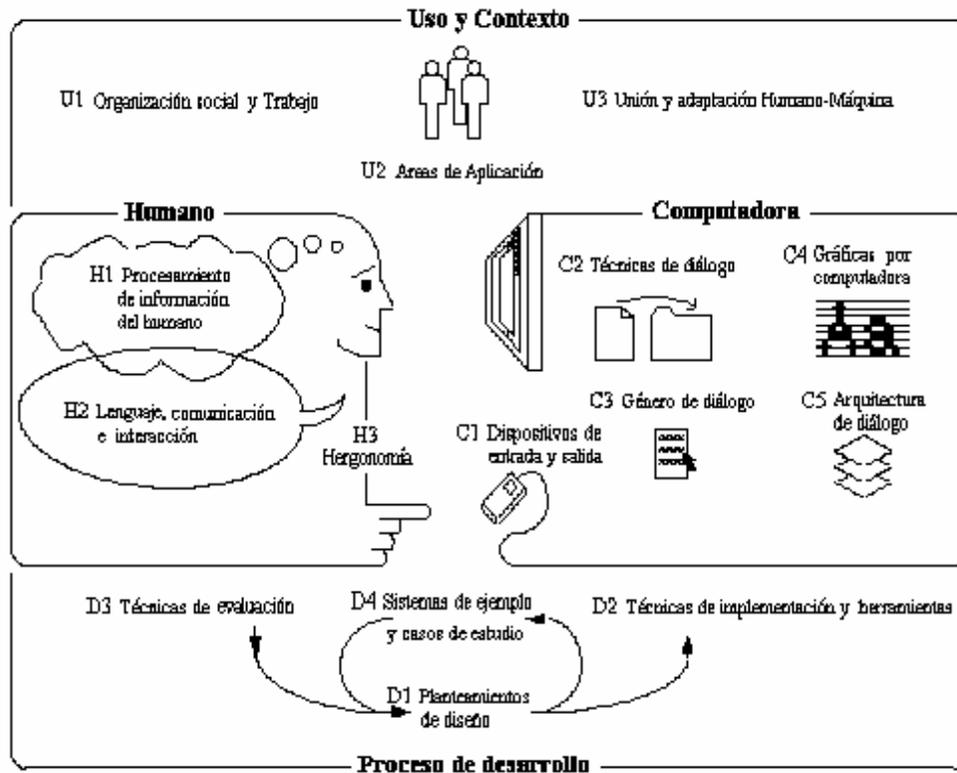


Figura 2: Esquema de desarrollo de sistema de Tecnologías Interacción Humano Computadora (tomado de Hewett *et al.*, 1996).

En adición al uso y al contexto social de las computadoras, por el lado humano, debemos tomar en cuenta los aspectos de procesamiento de información (H1), de comunicación (H2), y características físicas de los usuarios (H3). Por el lado de la computadora, una gran variedad de tecnologías han sido desarrolladas para el soporte de la interacción con los humanos: los dispositivos de entrada y de salida conectan al humano con la computadora (C1). El diálogo entre el humano y la computadora se organiza mediante varias técnicas (C2). Estas técnicas son usadas a su vez para implementar diseños más grandes,

como la metáfora de la interfaz (C3). Profundizando aún más en los elementos básicos que soportan el diálogo, ésta puede hacer uso extensivo de técnicas de graficación por computadora (C4). Los sistemas de diálogos complejos involucran la consideración de una arquitectura necesaria para soportar programas de aplicaciones interconectables, sistemas de ventanas, respuestas en tiempo-real, comunicaciones en redes, multi-usuarios e interfaces cooperativas y multitareas de diálogos de objetos (C5). Finalmente, está el proceso de desarrollo, el cual incorpora diseño de diálogos humano-computadora (D1), técnicas y herramientas para la implementación de éstos (D2), sus respectivas técnicas de evaluación (D3), y un determinado número de diseños clásicos para su estudio (D4). Cada uno de estos componentes del proceso de desarrollo esta ligado con los demás de una forma recíproca, de tal forma que cada decisión hecha en una área impacta en las opciones de decisión de las otras.

La Figura 3 muestra los subsistemas que influyen en el desarrollo de esta herramienta. A continuación se explica como cada uno de los subsistemas interactua.

El subsistema de personas con discapacidad auditiva y del habla, es quien manda los símbolos a través del subsistema de Dispositivos de Comunicación al subsistema de Herramienta de Software. Este subsistema, por su naturaleza introduce cierto grado de entropía al sistema en general, lo cuál es originado por la cantidad y diversidad de símbolos tanto válidos (lenguaje signado mexicano) como no válidos (símbolos distintos al lenguaje signado mexicano) que la persona con discapacidad auditiva y del habla puede enviar al subsistema de Herramienta de Software.

Por su parte el subsistema de Dispositivo de Comunicación, es el encargado de recibir lo enviado por el subsistema de personas con discapacidad auditiva y del habla, procesarlo y enviarlo al subsistema de Herramienta de software.

El subsistema de estrategias de enseñanza-aprendizaje, es donde se va a tomar el esquema bajo el cual se va a enseñar el lenguaje signado en español a la persona con

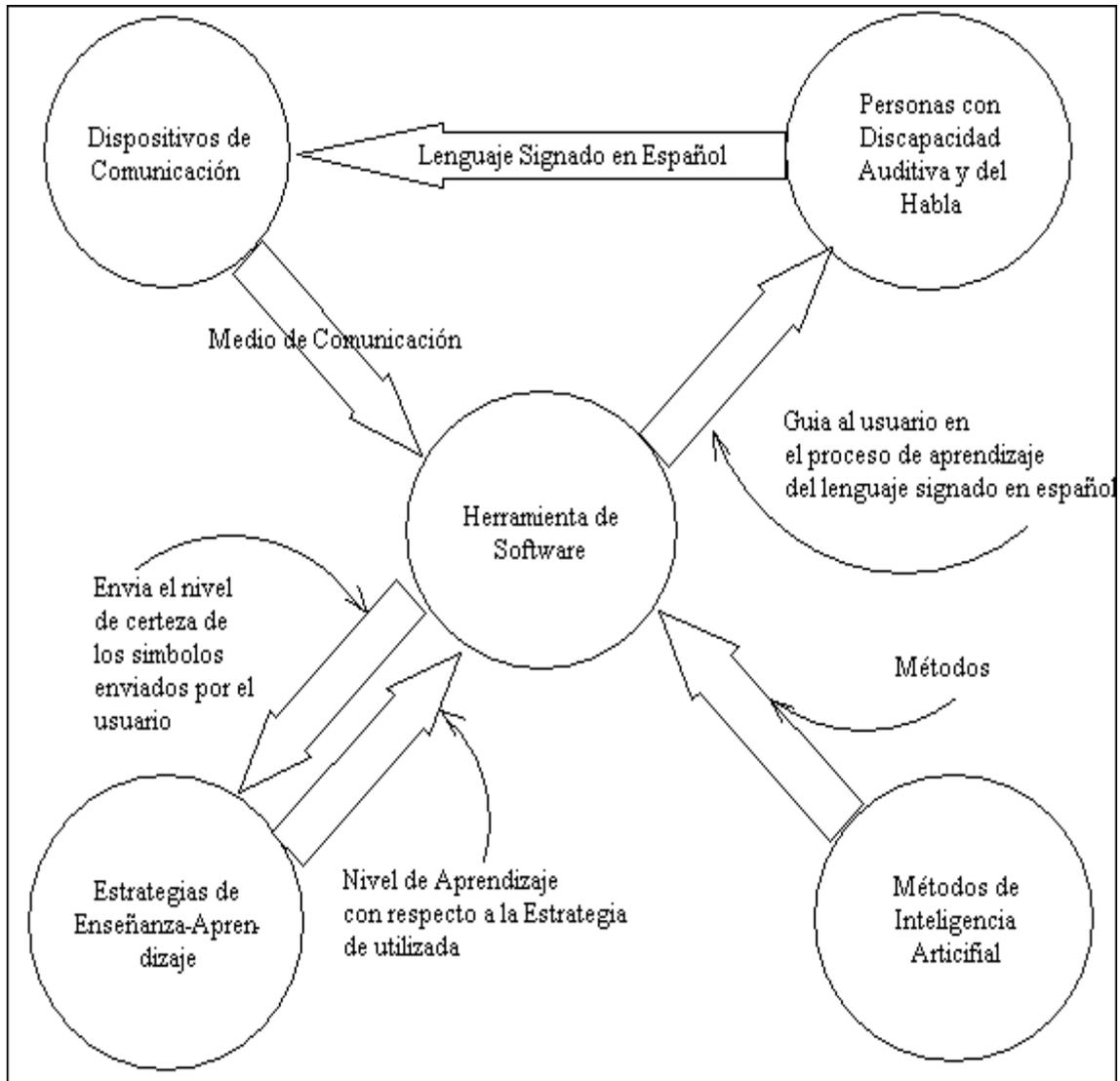


Figura 3: Subsistemas que influyen en el desarrollo de la herramienta de software.

discapacidad auditiva y del habla, por tanto este subsistema recibe del subsistema de herramienta de software el éxito o fracaso de los símbolos enviados por el usuario, posteriormente aplica reglas de evaluación del nivel de aprendizaje y retroalimenta al subsistema de herramienta de software. Se puede decir que este subsistema ayuda en la estabilización del sistema, ya que este es de donde se toman las reglas bajo las cuales funcionará el sistema en general.

El subsistema de métodos de inteligencia artificial, como su nombre lo dice, nos proveerá de los métodos necesarios para desarrollar los algoritmos que le den a la he-

herramienta de software un comportamiento dinámico con capacidad de adaptación a los distintos usuarios del sistema así como la capacidad de “aprender” las reglas de enseñanza-aprendizaje que se adopten y a partir de estas evaluar el nivel de aprendizaje del usuario.

Por último, tenemos al subsistema de herramienta de software, este es el encargado en primera instancia de recibir lo enviado por el usuario, posteriormente con la ayuda de los subsistemas de estrategias de enseñanza-aprendizaje y el de métodos de inteligencia artificial, tomar una decisión respecto al símbolo recibido, una vez tomada la decisión, se retroalimenta al usuario si es correcto o incorrecto el símbolo enviado.

2.4. Sistemas tutores inteligentes

El término Sistemas Tutores Inteligentes (Intelligent Tutoring Systems, ITS) se refiere a cualquier programa de computadora (*software*) que puede ser utilizado en aprendizaje y que contiene “inteligencia” (Freedman *et al.*, 2000), destinado a proveer una instrucción efectiva, apropiada y flexible a través de la aplicación de técnicas de Inteligencia Artificial (Artificial Intelligence, AI) y representación del conocimiento (Hanh *et al.*, 1997).

Un ITS, al ser un *software* educativo que contiene componentes de IA, le da seguimiento al trabajo de estudiante adaptando la retroalimentación y haciendo sugerencias a lo largo del proceso de enseñanza, de igual forma, mediante la recolección de información sobre el desempeño del estudiante, puede hacer inferencias sobre las fortalezas y debilidades y sugerir trabajo adicional al mismo (Hafner, 2004).

Los ITS pueden parecer sistemas monolíticos¹, pero para propósitos de conceptualización y diseño, frecuentemente es más sencillo pensar que están formados de varios componentes interdependientes (Beck *et al.*, 1996). Existen cuatro módulos en un ITS (ver Figura 4): un módulo experto (o dominio), un módulo del estudiante y un módulo pedagógico. Además, un módulo de interfaz necesario para comunicarse con los estudiantes (Hanh *et al.*, 1997).

¹Que están hechos de sólo una pieza, que no contiene módulos o componentes

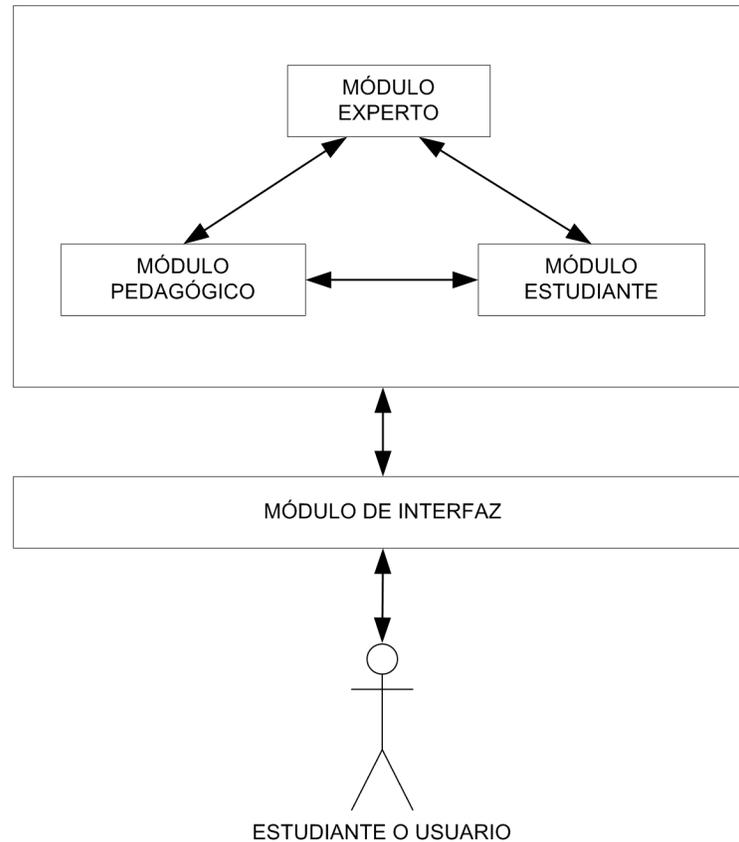


Figura 4: Componentes de un Sistema Tutor Inteligente.

2.4.1. Módulo experto

El módulo experto es el componente del ITS que provee el dominio del conocimiento, el cual representa la información que se va a enseñar al estudiante (Burns y Capps, 1988 y Beck *et al.*, 1996).

2.4.2. Módulo del estudiante

El módulo del estudiante almacena información que es específica para cada aprendiz. Como mínimo, el modelo debe dar seguimiento al desempeño que el estudiante tiene sobre el material que le está siendo enseñado. Una posible adición a esto sería llevar un control de los errores.

2.4.3. Módulo pedagógico

El módulo pedagógico utiliza la información generada por el módulo del estudiante para determinar que aspectos del dominio del conocimiento se debe mostrar al aprendiz. Esta información, por ejemplo puede ser un nuevo material, una revisión de temas anteriores o bien una retroalimentación del tópico actual.

2.4.4. Módulo de interfaz

El módulo de interfaz es el medio por el cual el estudiante interactúa con el ITS, este puede incluir desde diálogos hasta diseño de pantallas que usualmente son presentadas en forma de interfaces gráficas.

2.5. Algoritmos genéticos

Los algoritmos genéticos (Genetic Algorithms, GA) fueron inventados por John Holland en los años sesentas, mismos que desarrollo junto a sus estudiantes y colegas en la Universidad de Michigan durante los años sesentas y setentas (Mitchell, 1998 y Holland, 1992). Los GA son métodos adaptables que pueden ser utilizados para resolver problemas de búsqueda y optimización. Estos están basados sobre los procesos genético de los organismos biológicos (Beasley *et al.*, 1993 y Forrest, 1996) y combinan la supervivencia del más apto dentro de estructuras de cadenas que tienen un intercambio aleatorio de información. En cada generación, un nuevo conjunto de individuos artificiales (cadenas) son creados haciendo uso de la información de sus antecesores más capaces (Goldberg, 1989).

Los GA manipulan una población de soluciones potenciales para problemas de optimización (o búsqueda). Ellos operan específicamente sobre representaciones codificadas de las soluciones, equivalentes al material genético de los individuos en la naturaleza, y no directamente sobre las soluciones en sí mismas. Cada solución es asociada con un *valor de aptitud* que refleja cuán buena es comparada con otras soluciones en la población. Entre más alto es el valor aptitud de un individuo, se incrementa su posibilidad de sobrevivir y reproducirse en las subsecuentes generaciones (Srinivas y Patnaik, 1994).

Los GA tienen los siguientes elementos en común: una población de *cromosomas*, selección de acuerdo a su aptitud, cruzamiento para producir nuevas generaciones y la mutación aleatoria de las nuevas generaciones (Mitchell, 1998).

El término *cromosoma* o *individuo* (ver Figura 5) se utiliza en GA para representar o hacer referencia a una solución candidata de un problema, frecuentemente codificada como una cadena de dígito binario (*bit*). El *cromosoma* esta formado por *genes* que son ya sean simples *bits* o bien pequeños bloques de *bits* adyacentes que codifican un elemento particular de la solución cadidata (Mitchell, 1998). Los *genes* codifican rasgos distintivos (por ejemplo el color de los ojos), y los posibles valores que estos pueden tener se les da el nombre de *alele*. Cada *gene* tiene su propia posición en el *cromosoma* a la cual se le denomina *locus* (Obitko, 1998).

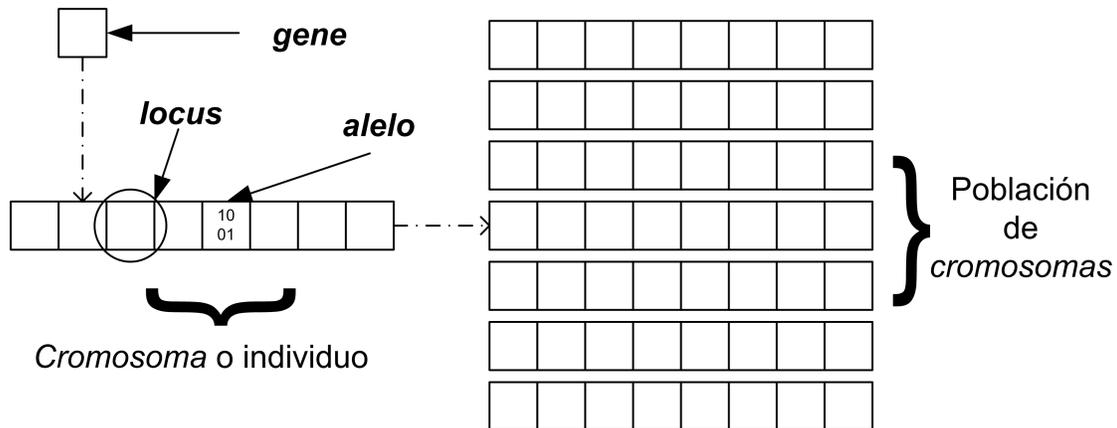


Figura 5: Componentes de un algoritmo genético.

2.5.1. Codificación

Al igual que otros métodos de búsqueda y aprendizaje, la forma como se codifica(n) la(s) solución(es) de un problema es un factor central en el éxito de un GA (Mitchell, 1998) y esta depende de la naturaleza de las variables del problema (Srinivas y Patnaik, 1994). Las cadenas de bits de longitud y orden fijo son las más comúnmente utilizadas. Sin embargo en los últimos años se han realizado muchos experimentos con otros tipos de codificación entre las cuales se encuentra el uso de valores reales y letras (Coello Coello,

1995).

De igual forma, se requiere de una *función aptitud*, que asigne una figura de mérito para cada solución codificada, la cual se supone debe ser proporcional a la “utilidad” o “habilidad” del individuo que el *cromosoma* representa. Durante la corrida, los antecesores (padres) deben ser seleccionados para la reproducción y recombinación para generar nuevas generaciones (Beasley *et al.*, 1993).

La función objetivo, la función que va ser optimizada provee del mecanismo para evaluar cada cadena (cromosoma). Sin embargo, sus rangos de valores varían de un problema a otro. Para mantener la uniformidad sobre varios dominios de problemas, se hace uso de la función aptitud para normalizar la función objetivo a un rango conveniente de 0 a 1. El valor normalizado de la función objetivo es la aptitud de la cadena, la cual el mecanismo de selección usará para evaluar las cadenas de la población (Srinivas y Patnaik, 1994).

2.5.2. Selección

El proceso de selección representa la oportunidad que tienen los individuos de la generación actual, de ser parte de la siguiente generación.

La selección de los padres es la tarea de asignación de oportunidades de reproducción para cada individuo. En principio, se copia a los individuos de la población a una “piscina de acoplamiento”, donde se encuentran tanto los individuos altamente aptos con más probabilidad de recibir más de una copia y los no aptos que son más probables de no recibir ninguna copia. Después, los individuos son tomados en pares de manera aleatoria y alineados. Esto se repite hasta que la “piscina de acoplamiento” es agotada (Beasley *et al.*, 1993).

El comportamiento de los GA depende en mucho de cómo son seleccionados los individuos para ir dentro de la “piscina de acoplamiento”. Existen dos procedimientos para realizar lo anterior: remarcado explícito de aptitud y remarcado implícito de aptitud. En el primero, se toma el registro de aptitud de cada individuo, se marca sobre una nueva

escala, y se utiliza este valor como el número de copias que van a ir dentro de la “piscina de aclopiamiento” (el número de pruebas reproductivas). El otro método logra un efecto similar, pero sin ir al paso intermedio de calcular una aptitud modificada (Beasley *et al.*, 1993).

Dentro del remarcado explícito de aptitud los principales métodos son la aptitud escalada y la aptitud de ventana, en el primero, al número máximo de ensayos reproductivos asignados a un individuo se le asigna un cierto valor, normalmente 2.0. Esto se logra sustrayendo un valor apropiado desde el registro real de aptitud, después se divide por el promedio de los valores ajustados de la aptitud. Restando una cantidad fija incrementa la proporción de las aptitudes máximas a las aptitudes promedio. Es importante prevenir que se generen aptitudes con valores negativos. El segundo método es similar al de la aptitud escalada, excepto porque la cantidad que se va a sustraer se elige de manera diferente. La aptitud mínima de cada generación es registrada, y la cantidad restada es la aptitud mínima observada durante las n generaciones anteriores, donde n es típicamente igual a 10. Con este esquema la presión de selección varía durante una corrida, así como desde un problema a otro (Beasley *et al.*, 1993).

Otro método comúnmente empleado en el remarcado explícito es la aptitud de *ranking*, el cual supera la confianza sobre un individuo excepcional. Los individuos son ordenados en función de sus aptitudes reales y posteriormente son asignadas aptitudes reproductivas según su rango (Beasley *et al.*, 1993).

En el remarcado implícito de aptitud, la selección por torneo es una de las técnicas utilizadas. Existen varias variantes. La forma más simple es la selección de torneo binaria, donde pares de individuos son escogidos aleatoriamente de la población. Cualquiera que tenga una aptitud alta es copiado en la “piscina de aclopiamiento” (después ambos son reemplazados en la población original). Esto se repite hasta que la “piscina de aclopiamiento” se llena (Beasley *et al.*, 1993).

Este operador puede o no incluir *elitismo*, que consiste en buscar en cada generación

7. De aquí el cruzamiento de doble punto realiza la misma tarea que el cruzamiento de punto simple (i.e. intercambiar un solo segmento), pero de forma más general (Beasley *et al.*, 1993a).



Figura 7: Cromosoma vista como un ciclo (tomado de Beasley *et al.*, 1993a).

En contraste, el cruzamiento uniforme es radicalmente diferente al cruzamiento de punto simple. Cada gene en la generación es creado al copiar el gene correspondiente de uno u otro padre, se escogen de acuerdo a la mascara de cruzamiento que se genera de manera aleatoria. Donde hay un 1 en la mascara de cruzamiento, el gene es copiado del primer padre, y donde existe un 0 el gene es copiado del segundo padre, como se muestra en la Figura 8. Este proceso se repite con el intercambio de los padres hasta producir la segunda generación. Una nueva mascara de cruzamiento es aleatoriamente generada para cada par de padres (Beasley *et al.*, 1993a).

2.5.4. Mutación

La mutación es el operador secundario de los GA para explorar el espacio de solución (Anh *et al.*, 2005), responsable de reintroducir valores perdidos inadvertidamente del gen (aleles), previniendo la deriva genética, y proporcionando un pequeño elemento de búsqueda aleatoria en la vecindad de la población cuando esta ha convergido en gran parte (Beasley *et al.*, 1993a).

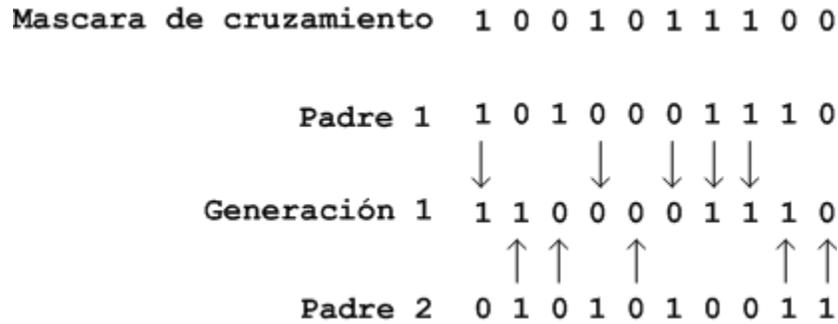


Figura 8: Cruzamiento uniforme (tomado de Beasley *et al.*, 1993a).

Para lograr lo anterior, este operador altera ya sea uno o más valores del gen de un *cromosoma* a partir de su estado inicial, lo cual resulta en valores enteramente nuevos que son agregados a la población de genes (NeuroDimension, 2006). Una amplia gama de operadores de mutación han sido propuestos, que se extienden desde alteraciones completamente aleatorias hasta operadores de búsqueda local heurísticamente motivados (Grefenstette, 1993). La Figura 9 muestra la mutación de inversión de *bit* que es utilizada cuando la representación de los individuos es binaria y donde el valor del gen seleccionado aleatoriamente se sustituye por su complemento (cero cambia a uno y uno cambia a cero) (Liao y Sun, 2001).

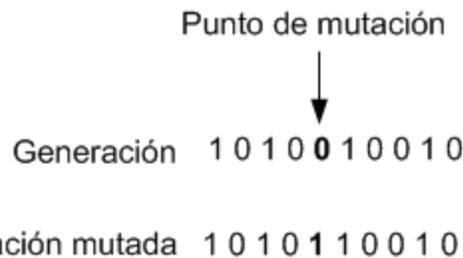


Figura 9: Mutación de inversión de *bit* (tomado de Beasley *et al.*, 1993a).

2.5.5. Funcionamiento de un algoritmo genético simple

Según Ribero Filho *et al.* (1994) en términos de computación, un GA mapea un problema dentro de un conjunto (típicamente binarias) de cadenas, donde cada una de ellas representa una solución potencial. Posteriormente el GA manipula las cadenas más prometedoras en su búsqueda por mejorar las soluciones. Un GA funciona a través de

etapas cíclicas simples:

1. Creación de una “población” de cadenas.
2. Evaluación de cada cadena.
3. Selección de las “mejores” cadenas, y
4. Manipulación genética para crear la nueva población de cadenas.

La Figura 10 muestra las cuatro etapas. En la primera etapa, la población inicial se inicializa haciendo uso de cuanto conocimiento este disponible sobre las posibles soluciones. En ausencia de este, la población inicial debe representar una muestra aleatoria del espacio de búsqueda (Grefenstette, 1993) que simbolice el punto inicial de la misma. Cada elemento de la población se codifica dentro de una cadena (*el cromosoma*) para ser manipulada por los operadores genéticos. En la siguiente fase, el desempeño (o aptitud) de cada individuo es evaluado con respecto a las restricciones impuestas por el problema. Basado en cada una de las aptitudes de los individuos, un mecanismo de selección escoge “parejas” para el proceso de manipulación genética. La combinación del proceso de evaluación y selección se le da el nombre de reproducción (Ribero Filho *et al.*, 1994).

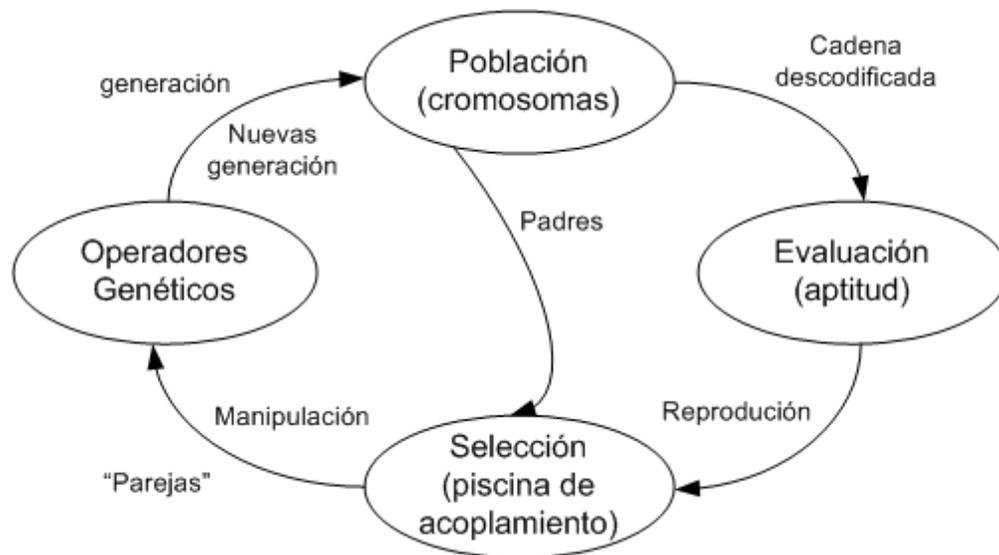


Figura 10: Etapas de un algoritmo genético (tomado de Ribero Filho *et al.*, 1994).

El proceso de manipulación utiliza los operadores genéticos para producir una nueva población de individuos manipulando la “información genética” referida como genes, poseídos por los miembros (padres) de la población actual. Este consta de dos operaciones: cruzamiento y mutación. El operador de cruzamiento recombina el material genético de una población, tomando dos cromosomas (padres) e intercambiando parte de su información genética para generar nuevos cromosomas. El operador de mutación, introduce nuevas estructuras genéticas en la población modificando aleatoriamente algunos de los componentes de la estructura genética (*building blocks*) (Ribeiro Filho *et al.*, 1994).

La generación producida por el proceso de manipulación genética es la siguiente prole que va ser evaluada. El GA puede reemplazar a toda la población (método generacional) o solamente a sus miembros menos aptos (método *steady-state*). El ciclo creación-evaluación-selección-manipulación se repite hasta que una solución satisfactoria para el problema es encontrado o se cumple con algún otro criterio de terminación (Ribeiro Filho *et al.*, 1994).

2.6. Lenguaje signado mexicano

Los lenguajes manuales o signados son concebidos como un tipo de comunicación no verbal entre personas. Estos pueden ser desde acciones simples de usar nuestra mano para señalar algo, hasta movimientos complejos para expresar sentimientos o establecer conversaciones estructuradas con otras personas.

Estos lenguajes han sido desarrollados de forma natural por personas sordas y sordomudas, con el fin de resolver sus necesidades de comunicación tanto con personas que poseen las mismas discapacidades como con personas normales (que no poseen estas discapacidades). Y, aún cuando pueden encontrarse modalidades regionales de estos lenguajes, en cada país existe un concentrado de ideogramas que son utilizados como el lenguaje signado de ese país en particular.

Con propósitos educativos, los lenguajes signados han sido complementados añadién-

doles un alfabeto manual, llamado también dactilología, con la cual se representa cada una de las letras del alfabeto con un ademán específico. Por lo que, el lenguaje signado de cada país se compone de un conjunto de ideogramas y una dactilología asociada. Esto los hace ser lenguajes muy ricos, con amplios vocabularios y gramáticas sofisticadas que tienen muy pocas diferencias con los lenguajes hablados de los países donde han sido desarrollados. Una persona discapacitada puede expresar cualquier palabra o idea por complicada que ésta sea, puede deletrear una palabra nueva y con el tiempo inventar y asignar un ideograma a ésta.

El lenguaje signado mexicano tiene sus inicios en el año de 1905 con el padre Camilo Torrente (Fleischmann, 1996), y éste, al igual que el de todos los países que han desarrollado lenguajes signados análogos, se divide en dos grandes ramas: la dactilología y los ideogramas.

2.6.1. Dactilología

La dactilología es la representación manual de cada una de las veintisiete letras que componen el alfabeto en español. A través de ésta se puede transmitir a la persona sorda, cualquier palabra que se desee comunicar, por complicada que ésta sea. En la Figura 11 se muestra la dactilología del español signado mexicano.

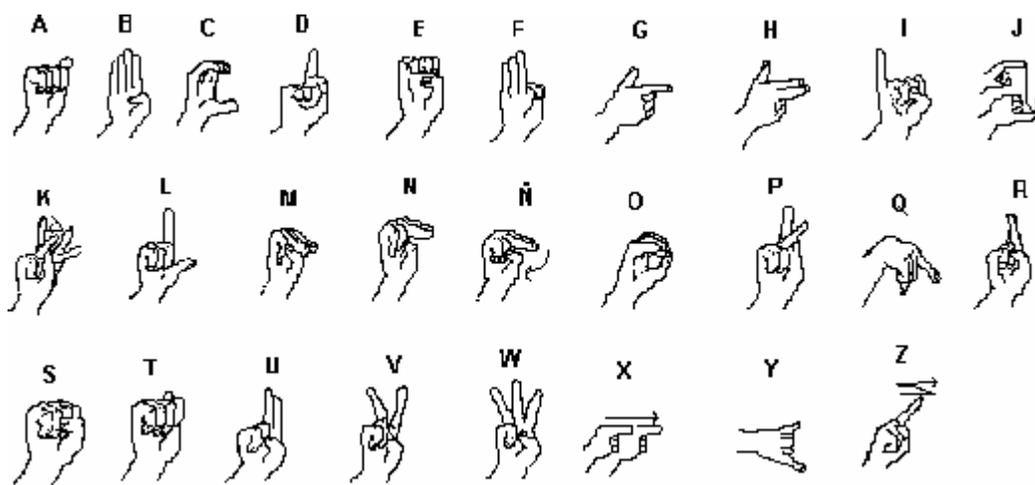


Figura 11: Dactilología del alfabeto manual mexicano. (tomado de Fleischmann, 1996)

2.6.2. Ideogramas

Un ideograma es la representación de una idea o una palabra, a través del movimiento de las manos; esto es, a través de un ademán específico. A diferencia de la dactilología, que hace uso sólo de una mano para ser representada, los ideogramas en el español signado mexicano pueden ser representados con una o ambas manos, debido a que en muchos de los casos se trata de representar el significado de una palabra o hecho imitando los movimientos reales (o lo mas similar posible) del concepto a representar. En la Figura 12 se muestran algunos ejemplos de ideogramas que ilustran lo dicho anteriormente.

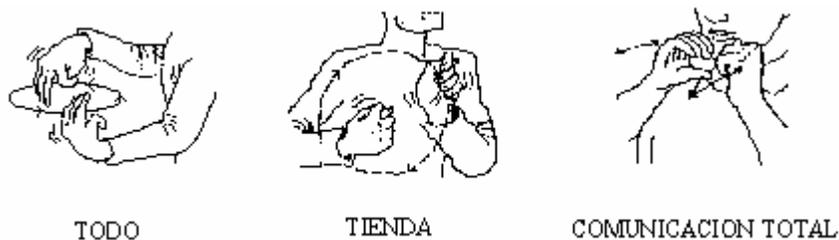


Figura 12: Ejemplo de los ideogramas de las palabras: todo, tienda y comunicación total. (tomado de Fleischmann, 1996)

La mayor parte de los ideogramas se realizan representando dactilológicamente la primera letra con que se escribe la palabra, y haciendo algún movimiento determinado. Por lo que, el aprendizaje del abecedario (alfabeto manual) es determinante para el buen manejo del lenguaje manual. Para una referencia completa de los ideogramas mas comunes del español signado mexicano, véase (Fleischmann, 1996).

Para las personas discapacitadas, es una ventaja que los lenguajes signados sean estructurados, puesto que puede generarse cualquier equivalente, en signo, a una palabra o concepto del lenguaje hablado. Sin embargo, para las personas que no son discapacitadas del habla y del oído, aprender estos lenguajes representa todo un reto, y más aún, si no interactúan de manera habitual con discapacitados. De hecho, son muy pocas las personas que conocen algún lenguaje manual o signado. Esto aunado a otros factores, hace que en la vida diaria de los discapacitados, su interacción con su entorno sea muy restringida y no exista un flujo libre de información. En la mayoría de estas interacciones, el contexto

está sujeto a la solución de necesidades básicas del discapacitado, y no a una comunicación más amplia de intercambio de ideas.

Capítulo 3

Diseño e implementación del algoritmo genético

Un aspecto fundamental en el sistema a desarrollar es que el algoritmo genético (Genetic Algorithms, GA) sea capaz de emular aspectos de un maestro real. Esto implica que la computadora debe ser capaz de ofrecer al usuario el conocimiento que este debe aprender, y, al igual que lo hace un maestro real, no solo mostrarlo, si no también verificar de alguna forma si el usuario ya domina este conocimiento; de ser así, mostrar conocimiento nuevo, junto con el que ya se domina, para que el usuario lo aprenda. De lo contrario modificar la presentación de este conocimiento para que el usuario lo aprenda, con otra variante de presentación.

Para lograr esta función usaremos el GA, con el fin de aprovechar su naturaleza de ser un proceso estocástico de búsqueda. Una ventaja que aprovecharemos de estos es que son capaces de medir la eficacia (o aptitud) del conocimiento existente para seleccionar aquel que cumple cierto criterio de evaluación. Son capaces de generar o direccionar información nueva, de forma aleatoria, partiendo de información existente. Otra ventaja importante de estos algoritmos es el hecho de utilizar funciones análogas a los operadores genéticos de mutación y cruzamiento para generar la información nueva, estas funciones como método de búsqueda hacen al algoritmo muy robusto porque involucra dos factores importantes en un problema de búsqueda dentro de un espacio paramétrico grande: la explotación y la exploración. Con la explotación se toma una región del espacio paramétrico, o un conjunto pequeño del conocimiento, en nuestro caso, y se evalúa la eficacia o aptitud de

este conocimiento, que en nuestro caso es equivalente a tomar un poco del conocimiento y evaluar el aprendizaje, por parte del usuario, de este subconjunto de conocimiento. Por otra lado, con la exploración: se conmuta, de manera estocástica, a otra región del espacio paramétrico, con el fin de realizar funciones de explotación y verificar la aptitud de esta nueva región, que en nuestro caso sería tomar otro subconjunto de conocimiento para mostrarlo al usuario con el fin de que lo aprenda (sabemos que el hecho de intentar que lo aprenda es el proceso de explotación de esta nueva región de conocimiento).

3.1. Paradigma de aprendizaje

Esté paradigma se aborda bajo un esquema de conductivismo. El objetivo es enseñar al usuario a representar el alfabeto signado mexicano bajo un esquema de prueba y error donde el sistema presenta un conjunto de símbolos y evalúa los aciertos y errores del usuario y en función de esto decide si muestra nuevos símbolos o bien deja en la sesión los símbolos aún no aprendidos por el mismo. Para lograr lo anterior se hace uso de los GA.

Un aspecto importante para diseñar el GA es definir o visualizar el espacio paramétrico de búsqueda, en nuestro caso serán todas la posibles variantes en la presentación del conocimiento que se desea aprenda el usuario.

Como parte de la estrategia para que el usuario aprenda la información (los signos de las veintisiete letras del alfabeto, ver Figura 11), se tomó la decisión de trabajar con conjuntos de símbolos (letras signadas), esto es: no presentar al usuario letra por letra para verificar si sabe su ademán, sino, presentarle grupos de letras y el usuario trate de representar cada una de ellas o las que se sepa. Con esto se pretende que exista un contexto distinto cada vez que interactúa con parte del conocimiento, que no sea un sólo símbolo sino un conjunto de ellos, que cada vez será distinto (de forma aleatoria).

Puede tomarse cualquier número de símbolos para realizar esta tarea. Dependiendo de este número será el tamaño del espacio paramétrico de búsqueda, por ejemplo: si

tomamos conjuntos de cinco símbolos, de entre los veintisiete símbolos debemos seleccionar conjuntos de cinco para presentárselos al usuario. Para obtener el tamaño del espacio de posibilidades (espacio paramétrico de búsqueda para el algoritmo) tenemos dos opciones: que en los conjuntos no importe el orden y los símbolos se puedan repetir, o que los símbolos no se puedan repetir y un grupo determinado de símbolos no pueda volver a aparecer no importa el orden en que hayan estado dentro del conjunto.

Para la primera opción, que especifica que no importa el orden de aparición y los símbolos pueden repetirse e incluso un conjunto puede volver a aparecer con el mismo orden de símbolos que ya haya aparecido, el número total de posibles conjuntos tomando cinco a la vez de entre un conjunto de veintisiete, corresponde al número de permutaciones, que se calcula como sigue:

$${}_n P_r = \frac{n!}{[(n-r)]!} = {}_{32} P_5 = \frac{32!}{[(32-5)]!} = 24,165,120 \quad (1)$$

posibles conjuntos. Donde P representa el número de permutaciones de n elementos que representan los símbolos del LSM considerados en grupos de r elementos.

Para la segunda opción, que especifica que los símbolos no se pueden repetir dentro de un conjunto, y que un conjunto no puede ser a su vez repetido, el número total de posibles conjuntos tomando cinco a la vez de entre un conjunto de veintisiete, corresponde al número de combinaciones, que se calcula como sigue:

$${}_n C_r = \frac{n!}{[r!(n-r)]!} = {}_{32} C_5 = \frac{32!}{[5!(32-5)]!} = 201,376 \quad (2)$$

posibles conjuntos. Donde C representa el número de combinaciones de n elementos que representan los símbolos del LSM considerados en grupos de r elementos.

Para este trabajo tomamos la primera opción, la razón es la siguiente, una persona en promedio aprende mejor cuando la información le es presentada de una forma gradual¹. Ahora, si observamos la primera opción, que es elegir los conjuntos (por permutaciones)

¹De manera progresiva.

trae implícito este hecho. Por la razón de que los símbolos pueden ser repetidos en un subconjunto, y aun más, un subconjunto puede volver a ser presentado idéntico al usuario. Esta libertad de presentación de información nos ofrece la ventaja de que el usuario seguirá viendo y practicando con ademanes que ya aprendió o practicó con anterioridad, de tal forma que un conocimiento ya aprendido seguirá recordándolo, garantizando con esto que no lo olvide.

3.2. Creación de cromosomas

El primer paso para diseñar un GA es adaptar la forma de la solución a un cromosoma. Esto debido a que el GA trabaja generando y evaluando cromosomas que tienen ya una forma predeterminada para el problema a resolver. En nuestro caso, elegimos utilizar subconjuntos de cinco símbolos por lo que podemos generar un cromosoma de cinco aleles, para representar la solución. Esto queda de la siguiente forma:

$$CROMOSOMA : X_1, X_2, X_3, X_4, X_5 \quad (3)$$

La Figura 13, muestra un ejemplo de una población de cromosomas de tres individuos



Figura 13: Población de cromosomas de tres individuos.

Ahora, con esta representación podemos aplicar sin problema el operador de cruce, por lo que podemos hacer explotación en el espacio cubierto por el subconjunto

de símbolos de la población inicial. Sin embargo no podemos aplicar el operador de mutación. Para esto debemos tener una representación mas granular de cada símbolo. Para resolver este detalle optamos por dar una representación binaria a cada uno de dichos símbolos. De igual forma, se asigno una representación decimal a cada símbolo. Esto con el fin de resolver a priori la necesidad de asignarle una dirección, dentro de un arreglo, a cada símbolo. La Figura 14 muestra como queda la representación.

3.3. Evaluación de aptitud

El siguiente paso al diseñar el algoritmo es definir la forma en que será evaluada cada generación. En este paso lo importante es definir, de acuerdo al problema, como se va a obtener de manera numérica, la aptitud de cada individuo. Aquí es donde hay que definir la función aptitud, de tal forma que pueda ser calculada una aptitud para cada individuo y una aptitud total de la población. En nuestro caso, esta aptitud total nos indicará si el subconjunto de símbolos que forman la población ha sido aprendido por el usuario o no.

En este punto del algoritmo es donde insertamos el aspecto de interactividad del programa con el usuario. De tal forma que la función aptitud la compone la respuesta del usuario, que para cada símbolo que le sea presentado se obtendrá como respuesta si lo sabe (1) o no lo sabe (0). Aquí lo que se ejecuta es lo siguiente:

1. Se toma al primer individuo de la población, y se presenta al usuario (en pantalla) el primer alele de este cromosoma (la letra en español correspondiente al símbolo, de los cinco que lo conforman). El usuario ejecutará, utilizando el teclado de la computadora, la representación manual de la letra que le ha sido mostrada. Una vez capturado el ademán, se anotará si es el correcto o no. De la misma forma se le presentara cada uno de los aleles de cada individuo que conforma la población actual.
2. La aptitud de cada individuo es el número de símbolos que fueron bien representados por el usuario. De esta forma se obtiene la aptitud de cada miembro de la población.

LSM	ALFABETO	DECIMAL	BINARIO	LSM	ALFABETO	DECIMAL	BINARIO
	A	0	00000		P	16	10000
	B	1	00001		Q	17	10001
	C	2	00010		R	18	10010
	D	3	00011		S	19	10011
	E	4	00100		T	20	10100
	F	5	00101		U	21	10101
	G	6	00110		V	22	10110
	H	7	00111		W	23	10111
	I	8	01000		X	24	11000
	J	9	01001		Y	25	11001
	K	10	01010		Z	26	11010
	L	11	01011		A	27	11011
	M	12	01100		E	28	11100
	N	13	01101		I	29	11101
	Ñ	14	01110		O	30	11110
	O	15	01111		U	31	11111

Figura 14: Representación binaria de los símbolos del Alfabeto Signado Mexicano.

$$\text{Aptitud del individuo } i = A_i = \text{número de aciertos} \quad (4)$$

3. La aptitud total se obtiene sumando todas la aptitudes de los individuos y dividiéndola por el número de individuos. Esto lo representamos como sigue:

$$\text{Aptitud total} = \frac{(i = 1nA_i)}{(\text{total de individuos})} \quad (5)$$

La Tabla I muestra un ejemplo de lo anterior, donde partiendo de una población de tres individuos donde cada uno de ellos esta compuesto por un conjunto de aleles (alelos), mismos que una vez presentados al usuario se obtuvieron los aciertos que se presentan en la columna número tres.

Tabla I: Muestra del resultado de la función aptitud de una población de tres individuos.

Individuo	Alleles	Aciertos	Aptitud del individuo
$I_1=$	V Y J R D	5	5
$I_2=$	M K T G D	3	3
$I_3=$	S E D A B	1	1

La aptitud total de la población es el promedio de las aptitudes de los individuos, que la obtenemos de la siguiente forma:

$$\text{AptitudTotal} = (5 + 3 + 1)/3 = 3 \quad (6)$$

3.4. Criterio de solución

Una vez que se tiene tanto la aptitud de cada individuo como la total de la población, el siguiente paso es evaluar si se cumple el criterio de solución al problema. Normalmente, en un problema de búsqueda, si se cumple el criterio, el algoritmo se detiene y da por solucionado el problema. De lo contrario, realiza el proceso de selección y genera una nueva población para ser evaluada.

En el caso particular del GA implementado, no es necesario que este se detenga. Esto debido a que una aptitud alta o adecuada de una generación determinada sólo indica que el subconjunto de símbolos de esa generación ya se aprendió o bien se recordó (según el caso) por el usuario. En este punto, es momento de utilizar otro subconjunto (que puede o no contener símbolos que estaban en el subconjunto anterior). Este proceso de evaluar subconjuntos de símbolos y estarlos variando, puede en un momento dado (dependiendo de la respuesta del usuario) regresar a un subconjunto que ya se había tenido antes. Esto debido a que el usuario puede olvidar ademanes, y al no reconocerlos hace que el algoritmo los integre de nuevo al proceso de aprendizaje. De esto que, el algoritmo debe estar en un ciclo de moverse en todo el espacio paramétrico, variando de un subconjunto de conocimiento a otro, y el usuario decide en que momento terminar la sesión del programa.

Tomando en cuenta este requerimiento, se definió el criterio de evaluación como sigue: El valor a tomar en cuenta es la aptitud total de la población. Si el valor promedio de efectividad esta del 80% hacia arriba asumimos que la generación ya es apta. Lo que significa que: si la aptitud tiene un valor alto (entre 4 y 5) entonces sabemos que el conocimiento (los símbolos del subconjunto de la población actual) ya fué, la mayoría, aprendido o recordado por el usuario. Si la aptitud total es menor de 4 entonces sabemos que el conocimiento no ha sido satisfactoriamente aprendido o recordado por el usuario. Al hacer esta evaluación de la aptitud total de la población tenemos dos posibles respuestas: que no se cumpla el criterio de evaluación, es decir que la aptitud total sea menor de 4, o que si se cumpla. Lo que el algoritmo debe hacer al tener el resultado de esta evaluación, se presenta en la siguiente apartado.

3.5. Operador de cruzamiento

Cuando no se cumple con el criterio de evaluación, significa que al presentar los símbolos del subconjunto actual al usuario, éste no los representó bien; en consecuencia, el algoritmo debe permanecer en el mismo subconjunto y presentarlo de nuevo al usuario para que este lo aprenda. En este punto, se requiere que el orden de los símbolos, al ser

presentados de nuevo al usuario, cambien con respecto al orden en que se presentaron la vez anterior.

Como se mencionó anteriormente, se desea que cambie el contexto de cada uno de los símbolos, al ser presentado cada cromosoma al usuario. Para lograr esto se aplica el operador de cruzamiento entre los individuos de la población actual. Esto con el fin de generar una nueva población pero con los mismos símbolos del subconjunto actual sin salirse de la región del espacio paramétrico que cubre la población existente. Esto arroja como resultado nuevos individuos pero generados con los mismos símbolos existentes. Con esto, al presentarse la información de nuevo al usuario, los símbolos serán los mismos que ya se le presentaron pero en orden distinto. En términos de algoritmos genéticos, esto significa que se realiza el proceso de explotación dentro de la región existente haciendo cruzamiento en los individuos a nivel alele.

3.5.1. Selección

Con el finalidad de que el usuario memorice la representación manual de los símbolos presentados, se hace uso de los símbolos que se hayan representado bien, junto con los símbolos en los que se hayan fallado. Para lograr esto se selecciona, para hacer el cruzamiento, individuos que tengan una aptitud alta, y se cruzan con individuos que tenga una aptitud baja. Con esto lo que se logra es que: al presentarle de nuevo los símbolos al usuario, los que no sabe vendrán acompañados de otros que si sabe, generando un contexto conocido alrededor de los símbolos desconocidos. La técnica que se utiliza y que es la que mas se ajusta a la necesidad de mostrar conocimiento desconocido junto con conocimiento conocido es la de elitismo. Con ésta, aquellos individuos que mas aptitud tengan serán los que más se crucen con otros.

La Tabla II muestra un ejemplo donde el individuo que tiene la aptitud más alta es cruzado con el resto de los individuos para generar una nueva población. La Tabla III muestra como quedarán las parejas seleccionadas para la reproducción:

Tabla II: Selección de individuos utilizando elitismo.

Individuo	Aptitud
I_1	5
I_2	3
I_3	2
I_4	2
I_5	1

Tabla III: Resultado del proceso de selección de individuos utilizando elitismo.

Individuo con (<i>aptitud más alta</i>)	Pareja con (<i>individuos con aptitud más baja</i>)
I_1	I_5
I_1	I_4
I_1	I_3
I_1	I_2

3.5.2. Cruzamiento

Una vez seleccionados los individuos que serán cruzados, se elige el punto de cruce, y este es el alele en el cual se partirá cada individuo. Para esto, se toma cada par de individuos y se genera un número aleatorio entre uno y la longitud del individuo, tomando este valor como el número de alele, de izquierda a derecha, en el cual se hará el cruce. Este procedimiento se realiza para cada par de individuos a cruzar, de tal forma que el punto de cruce puede ser distinto cada vez que sea ejecutado el proceso.

3.6. Cruzamiento y mutación

En caso de que sí se cumpla que la aptitud total sea mayor o igual a cuatro, que significa que el usuario si representó bien los símbolos mostrados, tenemos entonces que la información que se le presentó al usuario ya la aprendió. Por lo que es necesario hacer uso de nuevos símbolos para mostrarle al usuario, con el fin de que incremente su acervo de conocimiento. Esto implica que el algoritmo debe aplicar el proceso de exploración, saliendo de la región actual y posesionándose en otra región (que puede incluir parte de la actual). Para realizar este proceso es necesario hacer cruzamiento y

mutación entre los individuos, pero, a un nivel distinto del cruzamiento que se realiza si el criterio de evaluación no se cumple. Ahora necesitamos no solo realizar cambio de símbolos en cada individuo, si no realizar cambio de símbolos en toda la población actual. Cosa que no produce el cruzamiento a nivel alele que vimos anteriormente, que solo intercambia símbolos entre individuos.

Para lograr este cambio de símbolos en la población, utilizamos los operadores de cruzamiento y mutación pero a nivel sub-alele, esto es: tomamos la representación binaria de cada alele (se definió en el paso 1) y aplicamos los operadores genéticos en estas cadenas binarias. Esto provoca que la población resultante sea diferente a la población actual, en cuanto a cadenas binarias, que como vimos en al paso 1 de este trabajo, corresponde a direccionar otros símbolos distintos a los que se tienen en la población actual. A continuación se muestra un ejemplo de este proceso:

La Tabla IV presenta la población actual, a partir de la cual se desea generar de manera aleatoria otros quince aleles donde pueden o no aparecer elementos ya existentes.

Tabla IV: Población actual.

Individuo	Aleles	Sub-aleles	Aptitud
I_1	V Y J R D	10110 11001 01001 10010 00011	5
I_2	M K T G D	01100 01010 10100 00110 00011	3
I_3	S E D A B	10011 00100 00011 00000 00001	1

Para la realizar lo anteriormente descrito, se toma el alele uno de cada individuo ($I_1 \dots I_n$) y se cruza entre sí para generar tres aleles que pertenecen a la nueva generación. De tal manera que los tres aleles existentes son 10110, 01100 y 10011, que al cruzarlos tomando un punto de cruce aleatorio y una determinada probabilidad de mutación se tiene lo siguiente (ver Figura 15):

1. El alele uno (10110) del individuo I_1 se cruza con el alele uno (01100) del individuo I_2 . De este proceso se generan dos nuevos aleles que son: (10100) y (01110). Del resultado se toma el primero de ellos que es (10100).

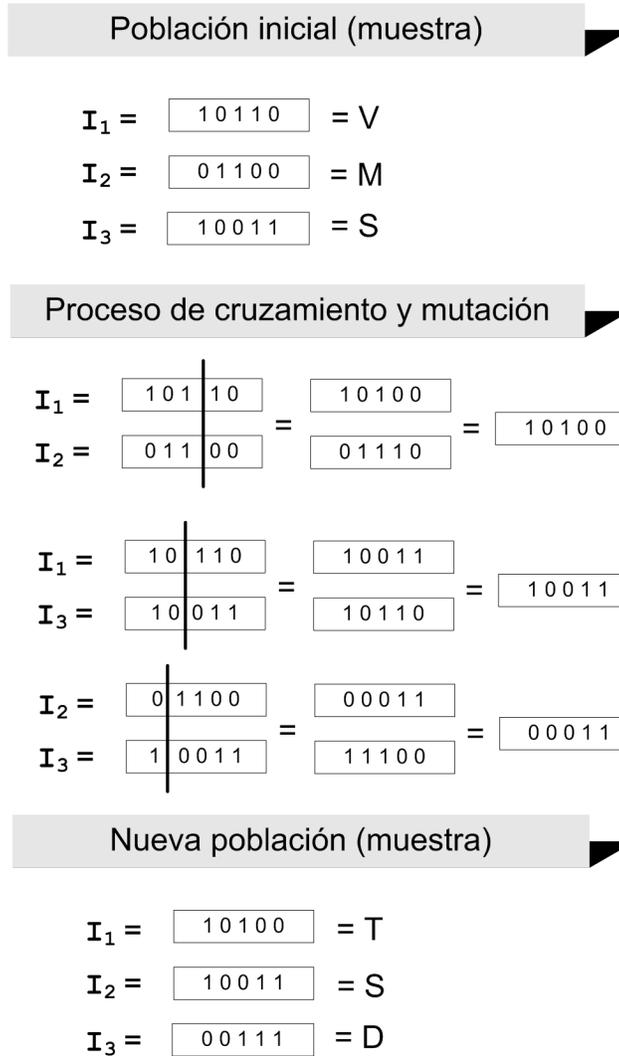


Figura 15: Proceso para generar una nueva población.

2. El alele uno (10110) del individuo I_1 se cruza con el alele uno (10011) del individuo I_3 . De este proceso se generan dos nuevos aleles que son: (10011) y (10110). Del resultado se toma el primero de ellos que es (10011).
3. El alele dos (01100) del individuo I_2 se cruza con el alele uno (10011) del individuo I_3 . De este proceso se generan dos nuevos aleles que son: (00011) y (11100). Del resultado se toma el primero de ellos que es (00011).

Los tres nuevos aleles de la nueva población son (10100), (10011) y (00011). Se puede observar en los valores de la Figura 15 que estos nuevos aleles corresponde a las letras T, S y D, mismas que no existían en la población inicial. De esto que, se valida el proceso de

exploración ya que el GA se mueve de la región actual a una región nueva en el espacio total de búsqueda. Este procedimiento se aplica a los aleles dos, tres, cuatro y cinco de los individuos para generar el resto de la población. La Figura 16 muestra el diagrama de flujo del GA descrito en este trabajo.

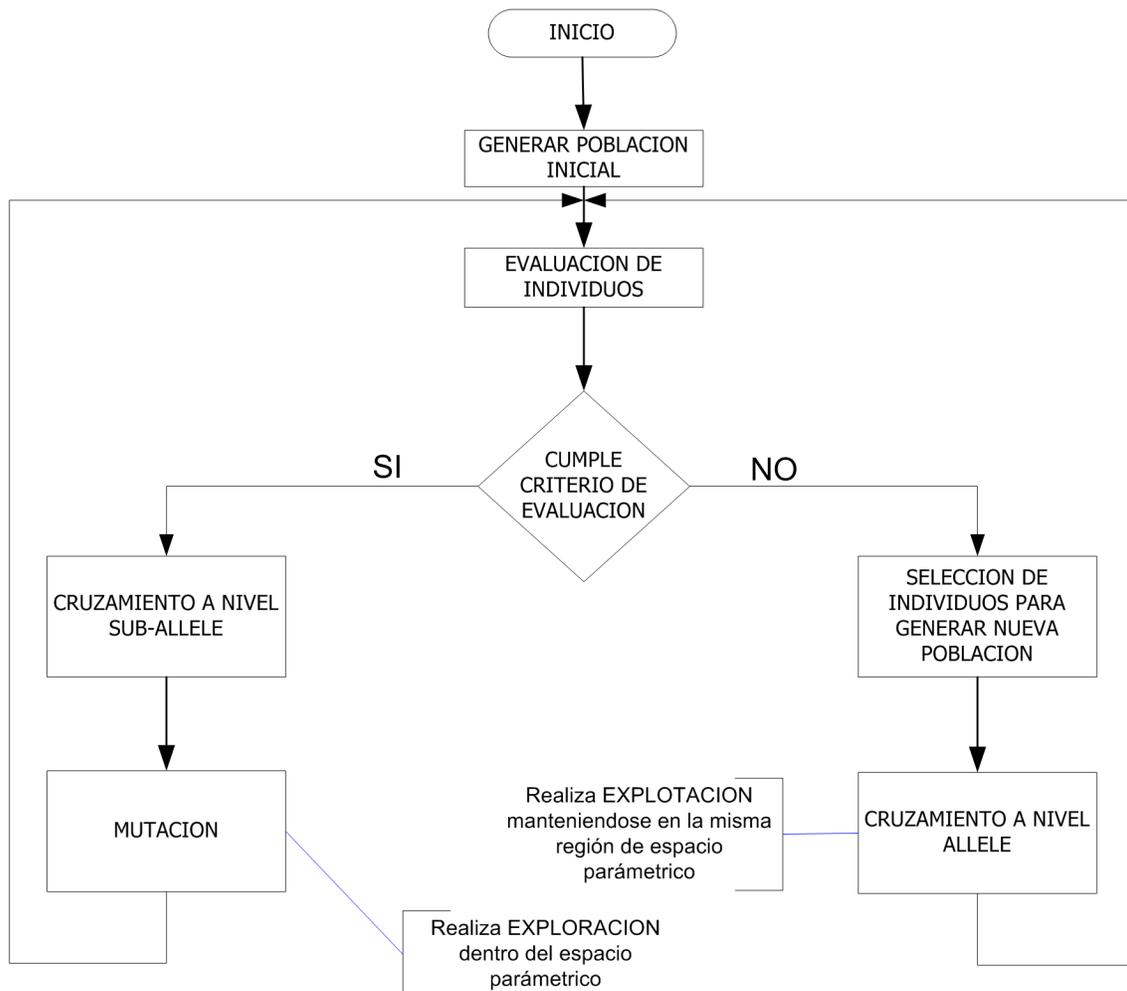


Figura 16: Diagrama de flujo del algoritmo genético utilizado.

Capítulo 4

Evaluación y análisis de resultados

4.1. Implementación del algoritmo genético

El algoritmo genético (Genetic Algorithms, GA) basa las sesiones de aprendizaje en ciclos, donde cada uno de ellos muestra al estudiante un subconjunto inicial de quince signos de los veintisiete que componen el conjunto total del conocimiento del alfabeto signado mexicano. A lo largo de la sesión de aprendizaje, el GA administra el contenido de lo que se le va a presentar al aprendiz en cada ciclo en función del avance del mismo. A continuación se presenta una ejecución del algoritmo donde se muestra el comportamiento del mismo a lo largo de una sesión de aprendizaje compuesta por tres ciclos.

En primera instancia el sistema muestra al estudiante una pantalla de inicio (ver Figura 17) que contiene el alfabeto signado mexicano, esto con la intención de que el alumno conozca y lleve a cabo una asociación de éste con su letra correspondiente antes de iniciar con la sesión de aprendizaje.

Una vez iniciada la sesión de aprendizaje, el algoritmo toma el subconjunto inicial de conocimiento y lo presenta al estudiante tal y como lo muestra la Figura 18. En este punto, se ven los primeros quince signos seleccionados por el sistema organizados en forma de una matriz de tres renglones por cinco columnas, donde cada uno de los renglones corresponden a un individuo dentro del GA. De tal forma que el primer renglón de símbolos compuesto por las letras Z,Q,I,O,V corresponden al primer individuo del GA; de igual forma los renglones dos y tres corresponden a los individuos dos y tres



Figura 17: Vista del inicio de la sesión de aprendizaje del Alfabeto Signado Mexicano.

respectivamente.

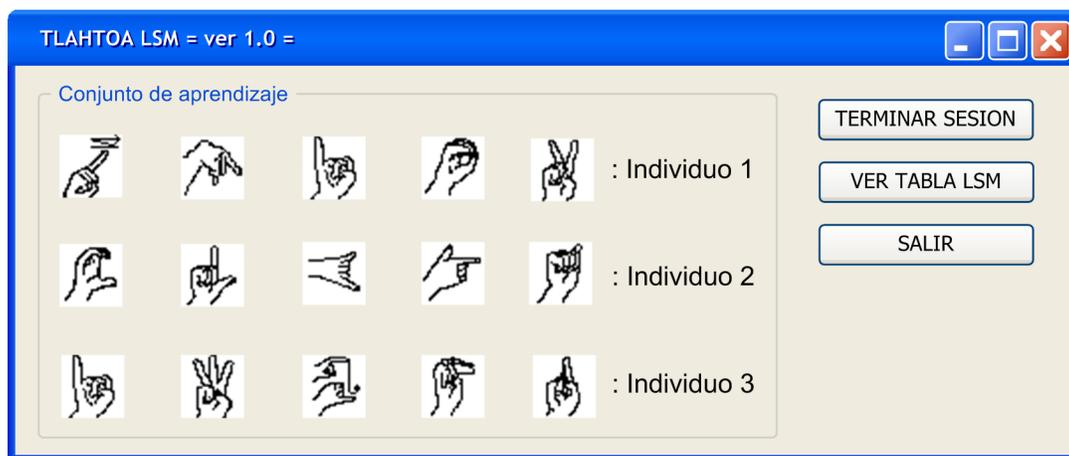


Figura 18: Vista que muestra el primer ciclo de la sesión de aprendizaje.

Una vez que el estudiante ingreso las letras correspondientes a los signos mostrados en el primer ciclo de aprendizaje, el algoritmo en función de los aciertos o errores del mismo genera un segundo ciclo de aprendizaje donde puede o no agregar nuevos símbolos

del alfabeto signado mexicano. La Figura 19 en la parte inferior izquierda en la sección *Evaluación de la muestra anterior* se presentan los valores obtenidos para cada uno de los individuos, los cuales representan el nivel de aprendizaje de cada uno ellos. En la sección *Valores de la muestra anterior*, se tienen los aciertos y errores que ha tenido el estudiante en cada ciclo de aprendizaje y finalmente en la sección *Valores acumulados* se tienen el total de errores y aciertos durante toda la sesión de aprendizaje.

TLAHTOA LSM = ver 1.0 =

Conjunto de aprendizaje

Individuo 1

Individuo 2

Individuo 3

Evaluación de la muestra anterior

Aptitud individuo 1 : 0

Aptitud individuo 2 : 2

Aptitud individuo 3 : 1

Aptitud total : 1

Valores de la muestra anterior

Aciertos : 3

Errores : 12

Valores acumulados

Símbolos acertados : 3

Símbolos erróneos : 12

TERMINAR SESION

VER TABLA LSM

SALIR

Figura 19: Vista que muestra el segundo ciclo de la sesión de aprendizaje.

En la Figura 19 se puede observar en la sección *Evaluación de la muestra anterior*, que el estudiante no acertó ninguno de los primeros cinco ademanes (valores del individuo uno); acertando de manera correcta solamente dos y uno en los individuos dos y tres respectivamente. Por lo que la aptitud total es igual a 1, lo que significa que se aprendió menos del ochenta por ciento de los ademanes mostrados en la Figura 18. En consecuencia, los ademanes que se muestran para el segundo ciclo de aprendizaje (ver Figura 19) son los mismos que se presentaron en el primer ciclo de aprendizaje (ver Figura 18) pero con un orden distinto, de tal forma que el algoritmo simula lo que hace

un maestro real, al cambiarle el contexto de la información a un estudiante cuando este no la ha aprendido de la manera inicial como le fue mostrada.

Evaluado el ciclo de aprendizaje anterior, el AG genera un tercer ciclo (ver Figura 20), donde se observa que el estudiante acertó todos los ademanes, dando una aptitud total para cada individuo igual a 5. En consecuencia, para el tercer ciclo de aprendizaje se muestran ocho ademanes del ciclo anterior junto a siete nuevos ademanes que antes no existían. Con este algoritmo emula lo que hace un maestro real al incrementar la información que enseña al estudiante en función de los avances del mismo.

TLAHTOA LSM = ver 1.0 =

Conjunto de aprendizaje

Individuo 1

Individuo 2

Individuo 3

Evaluación de la muestra anterior

Aptitud individuo 1 : 5

Aptitud individuo 2 : 5

Aptitud individuo 3 : 5

Aptitud total : 5

Valores de la muestra anterior

Aciertos : 15

Errores : 0

Valores acumulados

Símbolos acertados : 3

Símbolos erróneos : 27

TERMINAR SESION

VER TABLA LSM

SALIR

Figura 20: Vista que muestra el tercer ciclo de la sesión de aprendizaje.

Finalmente, la Figura 21 muestra la Tabla del alfabeto signado, la cual se activa de manera automática en el cuarto ciclo de aprendizaje siempre y cuando el estudiante haya estado por debajo del ochenta por ciento de aciertos en todos los ciclos anteriores. En caso contrario, de que estudiante haya estado por arriba del ochenta por ciento de aciertos en uno o más de los ciclos anteriores, sólo se activa la opción y se deja al criterio

del estudiante si desea visualizarla o no. El objetivo de esta tabla es llevar a cabo un refuerzo en la asociación del alfabeto durante la sesión de aprendizaje.



Figura 21: Vista que muestra los símbolos del Alfabeto Signado Mexicano.

4.2. Evaluación del AG implementado

Se llevo a cabo un experimento de evaluación con la finalidad de estimar la aptitud que se gana o se pierde en cada ciclo de aprendizaje, en función del comportamiento de los aciertos y errores que el usuario comete durante una sesión de enseñanza. De igual forma, se deseaba conocer el impacto que tiene en el aprendizaje del usuario, el hecho de darle la opción de ver la tabla completa del alfabeto cada cierto número de ciclos.

Para realizar este experimento de evaluación se contó con la participación de dos personas, una de ellas con conocimientos previos de los signos del Lenguaje Signado Mexicano (LSM) y la otra sin conocimiento previo de los mismos. A cada una ellas se les mencionó que la prueba consistía en realizar un sesión de aprendizaje del alfabeto

signado de diez ciclos, posteriormente se les dio una introducción al sistema.

4.3. Resultados y discusión

La tabla V presenta los datos obtenidos para el usuario sin conocimientos previos del LSM y la tabla VI muestra los resultados para el usuario con conocimiento previo del LSM. Cada tabla contiene los valores arrojados durante una sesión de aprendizaje compuesta por diez ciclos (columna ciclo de aprendizaje), para cada ciclo se midió tanto la aptitud individual (columna aptitud individuo -uno, dos, tres) como total (columna aptitud total) lograda por el estudiante. De igual manera se contabilizó la cantidad de aciertos y errores (columna ademanes del LSM -aciertos, errores-) en la asociación de los ademanes del LSM, y finalmente las ocasiones en las que el algoritmo puso a disposición del estudiante el mecanismo de refuerzo (columna refuerzo) ya sea de forma automática o manual.

Tabla V: Resultado de la sesión de aprendizaje para un estudiante sin conocimiento previo del LSM.

Sesión de aprendizaje para un estudiante sin conocimiento previo del LSM							
Ciclo de aprendizaje	Aptitud individuo			Aptitud total	Ademanes del LSM		Refuerzo
	Uno	Dos	Tres		Aciertos	Errores	
1	3.00	2.00	2.00	2.33	7	8	
2	3.00	1.00	3.00	2.33	7	8	
3	1.00	1.00	3.00	1.67	5	10	
4	1.00	2.00	3.00	2.00	6	9	Automático
5	5.00	3.00	5.00	4.33	13	2	
6	3.00	4.00	4.00	3.67	11	4	
7	4.00	5.00	4.00	4.33	13	2	
8	5.00	5.00	5.00	5.00	15	0	Manual
9	5.00	3.00	5.00	4.33	13	2	
10	5.00	5.00	5.00	5.00	15	0	

En los resultados para el alumno sin conocimiento previo del LSM (ver Tabla V), se puede observar que la aptitud total para los primeros cuatro ciclos de aprendizaje es inferior a 4.0, esto indica que el nivel de aprendizaje o asociación de los ademanes del LSM esta por debajo del 80% que esta directamente relacionado con la cantidad de

aciertos y errores obtenidos (ver Figura 22) para cada serie.

Al identificar el algoritmo que la aptitud total del alumno para cada uno de los primeros cuatro ciclos esta por debajo de 4.0 este de manera automática activa y despliega (ver Figura 21) la tabla de símbolos del LSM con el objetivo de que el alumno identifique cuales son los ademanes que esta asociando de manera equivocada. Lo anterior, se ve reflejado en la aptitud total obtenida la cual es superior a 4.0 en la mayoría de los casos a partir del quinto ciclo con excepción del sexto, incluso existen series (octava y decima) donde se obtuvo una aptitud igual a 5.0 lo que indica que el alumno para estas series identificó todos los símbolos que se le presentaron de manera correcta (ver Figura 22).

En los resultados para el alumno con conocimiento previo del LSM (ver Tabla VI) se puede observar que la aptitud total para las primeras cuatro series son de un 50 % superior al 4.0 y otro 50% es inferior al 4.0. En este punto, el alumno para las series uno y dos obtuvo un aprendizaje por arriba del 80 %, posteriormente para las series tres y cuatro su nivel de aprendizaje estuvo por debajo del 80 % incidido por los nuevos símbolos del LSM agregados a cada una de las series de aprendizaje. La Tabla VI muestra el resultado de los aciertos y errores obtenidos por el alumno durante la sesión de aprendizaje.

Tabla VI: Resultado de la sesión de aprendizaje para un estudiante con conocimiento previo del LSM.

Sesión de aprendizaje para un estudiante con conocimiento previo del LSM							
Ciclo de aprendizaje	Aptitud individuo			Aptitud total	Ademanes del LSM		Refuerzo
	Uno	Dos	Tres		Aciertos	Errores	
1	4.00	5.00	4.00	4.33	13	2	
2	5.00	5.00	5.00	5.00	15	0	
3	3.00	3.00	5.00	3.67	11	4	
4	3.00	4.00	3.00	3.33	10	5	Manual
5	5.00	5.00	5.00	5.00	15	0	
6	5.00	4.00	4.00	4.33	13	2	
7	5.00	4.00	5.00	4.67	14	1	
8	5.00	5.00	5.00	5.00	15	0	Manual
9	5.00	4.00	5.00	4.67	14	1	
10	5.00	5.00	5.00	5.00	15	0	

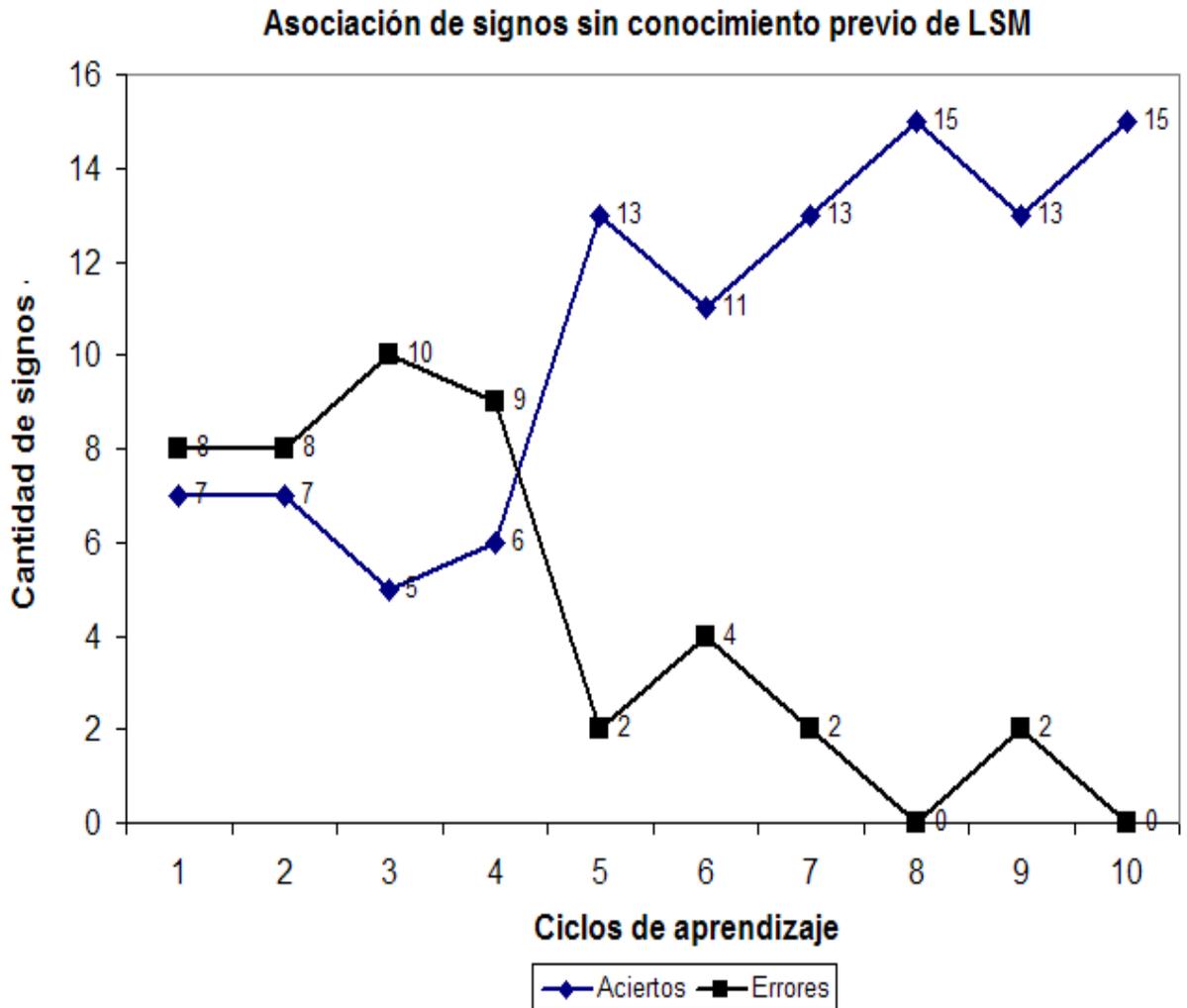


Figura 22: Gráfica de asociación de ademanes en una sesión de aprendizaje de un alumno sin conocimiento previo del LSM.

El alumno antes de iniciar la quinta serie toma la opción de refuerzo que le proporciona el algoritmo, en consecuencia se observa como la aptitud total posterior a la serie cuatro tiene niveles de aprendizaje por arriba del 80% reflejado directamente en la cantidad de aciertos y errores (ver Figura 23) obtenidos durante el resto de la sesión de aprendizaje.

Un resultado a observar, es como la actividad de refuerzo incidió en este usuario de manera directa en los porcentajes de aprendizaje posteriores a esta acción, los participantes en las pruebas coinciden que el uso del refuerzo fue determinante para identificar cuales símbolos del LSM tenían una asociación errónea. La Figura 24 muestra el comportamiento de aprendizaje para ambos alumnos durante la sesión.

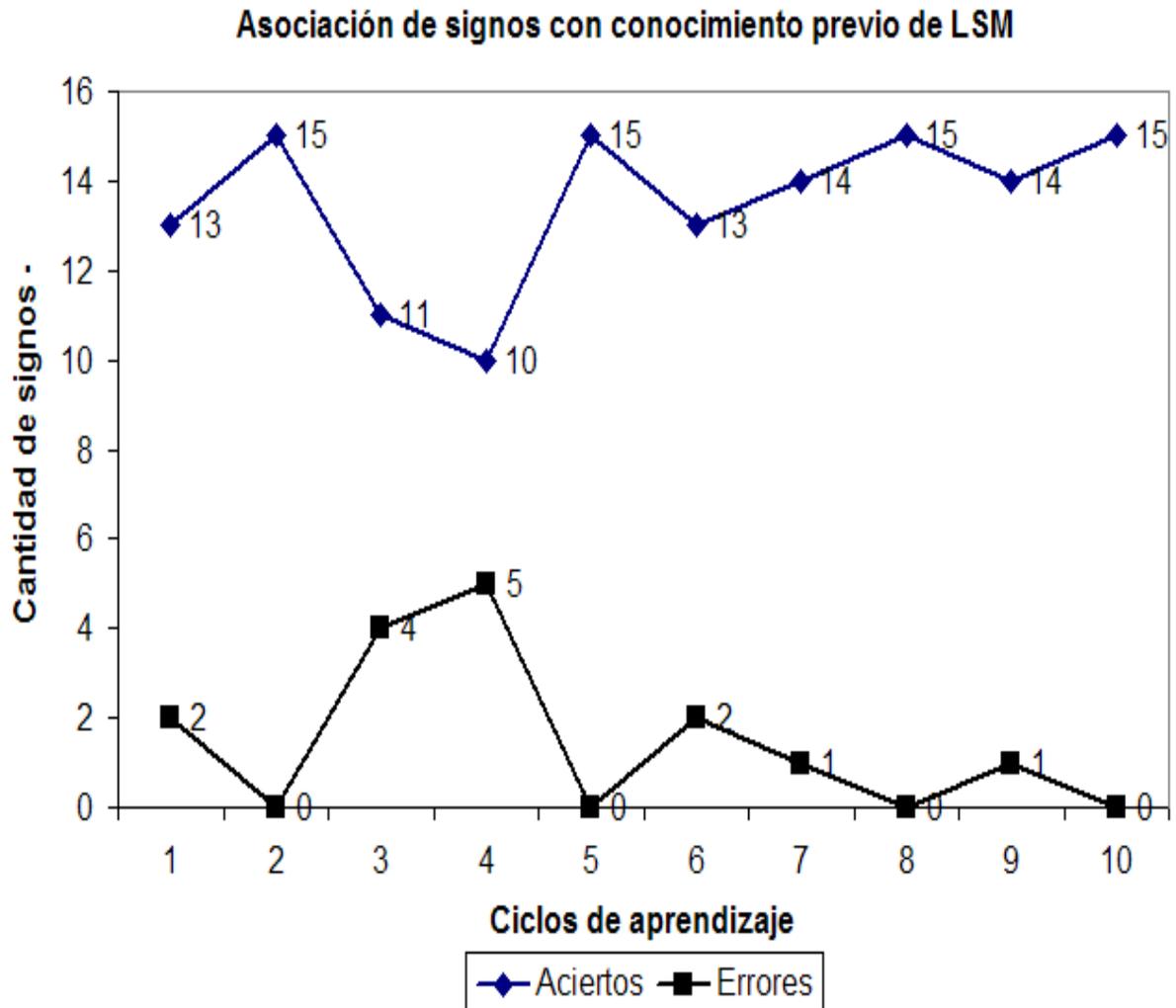


Figura 23: Gráfica de asociación de ademanes en una sesión de aprendizaje de un alumno con conocimiento previo del LSM.

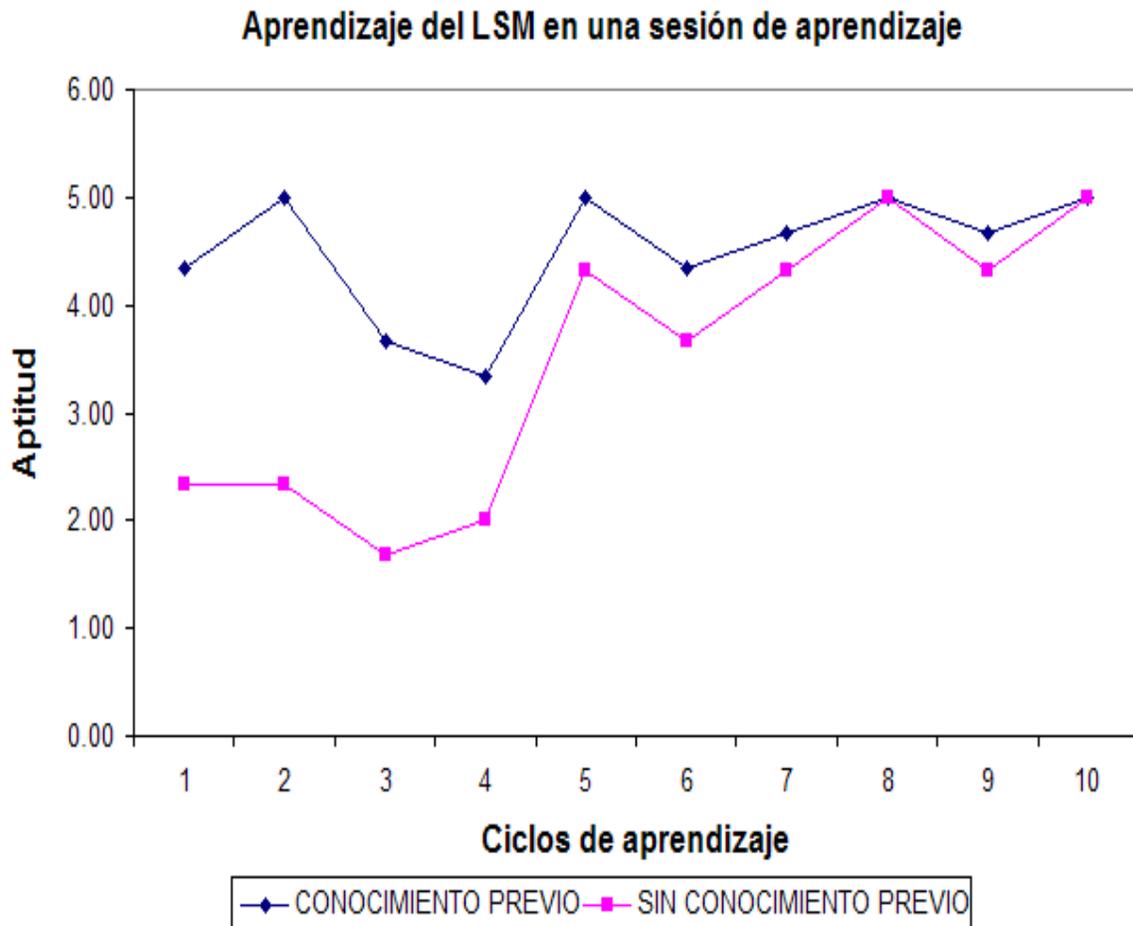


Figura 24: Gráfica de aprendizaje de dos usuarios.

Capítulo 5

Conclusiones y trabajo futuro

En este trabajo se propuso probar la viabilidad de un algoritmo genético para emular funciones de enseñanza de un maestro real. Para lograr anterior, se eligió el lenguaje signado mexicano como el dominio del conocimiento a aprender por los usuarios. Posteriormente, se diseñó y codificó el algoritmo genético. Y finalmente se probó el algoritmo con usuarios reales.

5.1. Conclusiones y aportaciones

El algoritmo genético implementado es capaz de generar sesiones de aprendizaje interactivas del alfabeto signado mexicano emulando algunos aspectos de un maestro real. A continuación se presenta una lista de las aportaciones logradas con este trabajo:

1. Diseño e implementación de un algoritmo genético que al igual que un maestro real logra que un estudiante aprenda de manera gradual y regulada del conocimiento que se le presenta.
2. La implementación del núcleo (kernel) para el desarrollo futuro de una plataforma interactiva para el aprendizaje del lenguaje signado mexicano.

5.2. Limitaciones

En la implementación actual del algoritmo se tienen limitaciones que inciden en el aprendizaje del alumno, a continuación se presenta una lista de ellas:

1. La interacción del algoritmo con el estudiante es por medio del teclado de la computadora (aún no esta integrado el guante de datos).
2. El conocimiento a aprender por el usuario esta limitado sólo a las veintisiete letras del alfabeto y no considera los ideogramas del lenguaje signado mexicano.

5.3. Trabajo futuro

La evolución del algoritmo implementado esta dentro del marco de agregar un guante de datos por medio del cual se puedan representar los símbolos del LSM de tal forma que la interacción entre el tutor y el estudiante sea mas rica y sensible al momento de las sesiones de aprendizaje.

Basados en el algoritmo implementado desarrollar un sistema tutor inteligente.

Referencias

- Ahn, C. W., S. Oh y R. S. Ramakrishna. 2005. On the practical genetic algorithms. Proceedings of the 2005 conference on genetic and evolutionay computation. ACM Press. New York, NY. 1583-1584 p.
- Beasley, D., D. R. Bull y R. R. Martin. 1993. An overview of genetic algorithms: Part 1, fundamentals. University Computing. 15(2): 58-69 p.
- Beasley, D., D. R. Bull y R. R. Martin. 1993a. An overview of genetic algorithms: Part 2, research topics. University Computing. 15(4): 170-181 p.
- Beck, J., M. Stern y E. Haugsjaa. 1996. Applications of AI in education. Crossroads The ACM Student Magazine. 3(1): 11-15 p.
- Burns, H. L. y C. G. Capps. 1988. Foundations of intelligent tutoring systems: An introduction. En: Polson, M. C. y J. J. Richardson (eds). Foundations of Intelligent Tutoring Systems. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1-19 p.
- Coello Coello, C. A. 1995. Introducción a los algoritmos genéticos. Soluciones Avanzadas, Tecnologías de Información y Estrategias de Negocios. 3(17): 5-11 p.
- Fels, S. S. y G. E. Hinton. 1993. Glove-Talk: A neural network interface between a Data-Glove and a speech synthesizer. IEEE Transactions on Neural Networks. 4(1): 2-8 p.
- Fels, S. S. y G. E. Hinton. 1998. Glove-TalkII: A neural-network interface which Maps gestures to parallel formant speech synthesizer controls. IEEE Transactions on Neural Networks. 9(1): 205-212 p.
- Fleischmann M. E. 1996. Aprendizaje de español signado de México. ISBN 970-91523-1-9, SEP 41227.
- Forrest, S. 1996. Genetic algorithms. ACM Computing Surveys. 28(1): 77-80 p.
- Freedman, R., Ali S. S. Ali y S. McRoy. 2000. What is an Intelligent Tutoring System? ACM Intelligence Magazine. 11(3): 15-16 p.
- García Nájera, A. 2005. Algoritmos genéticos para problemas de ensamble de tarjetas de circuitos impresos. Tesis de Maestría. Centro de Investigación Científica y de Educación Superior de Ensenada. Ensenada, B.C., México. 114 p.

- Grefenstette, J. J. 1993. Genetic algorithms. *IEEE Expert*. 8(5): 5-8 p.
- Goldberg, D. E. 1989. Genetic algorithms in search, optimization & machine learning. Addison Wesley Longman, Inc. First Edition. Boston. 412 p.
- Good, T. L. y J. E. Brophy. 1990. Educational psychology: A realistic approach, fourth Edition. Longman Inc. New York. 836 p.
- Kramer, J. y L. Leifer. 1989. The talking glove a speaking aid for nonvocal deaf and deaf-blind individuals. In Proceedings of RESNA 12th Annual Conference. RESNA Press. New Orleans, Louisiana. 471-472 p.
- Hafner, K. 2004. Software tutors offer help and customize hints. *The New York Times*. [En red]. Disponible en: <http://www.nytimes.com/2004/09/16/technology/circuits/16tuto.htm>. Consultado en febrero 2006.
- Hanh, S. H., J. S. Song, K. Y. Tak y J. H. Kim. 1997. An intelligent tutoring system for introductory C language course. *Computer and Education*. 28(2): 65-143 p.
- Hewett, T. T., R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong y W. Verplank. 1996. ACM SIGCHI Curricula for Human-Computer Interaction. The Association for Computing Machinery, Copyright ©1996. [En red]. Disponible en: <http://www.sigchi.org/cdg/cdg2.html#D4>. Consultado en octubre 2003.
- Holland, J. H. 1992. Genetic algorithms. *Scientific American*. 267(1): 66-72 p.
- Lee, H. K. y J. H. Kim. 1999. An HMM-Based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 21(10): 961-973 p.
- Liao, Y. H. y C. T. Sun. 2001. An educational genetic algorithms learning tool. *IEEE Transactions On Education*. 44(2): 1-22 p.
- Mergel, B. 1998. Instructional design & learning theory. [En red]. Disponible en: <http://www.usask.ca/education/coursework/802papers/mergel/brenda.htm>. Consultado en abril 2006.
- Mitchell, M. 1998. An introduction to genetic algorithms. The MIT Press. First Edition. Cambridge, Massachusetts. 209 p.
- Moghaddam, B. y A. Pentland. 1997. Probabilistic visual learning for objects representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19(7): 696-710 p.
- NeuroDimension. 2006. Genetic algorithms. [En red]. Disponible en: <http://www.nd.com/genetic/>. Consultado en septiembre 2007.
- Obitko, M. 1998. Introduction to genetic algorithms. [En red]. Disponible en: <http://cs.felk.cvut.cz/~xobitko/ga/>. Consultado en septiembre 2007.

- Pavlovic, V. I., S. Rajeev y S. H. Thomas. 1997. Visual interpretation of hand gestures for human-computer interaction: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19(7): 677-695 p.
- Quam, D. L. 1990. Gesture recognition with a DataGlove. *Proceedings of the IEEE 1990 National Aerospace and Electronics Conference*, vol 2. NAECON 1990. 755-760 p.
- Ribeiro Filho, J. L., P. C. Trelevan y C. Alippi. 1994. Genetic-algorithm programming environments. *IEEE Computer*. 27(6): 28-43 p.
- Schuman, L. 1996. Perspectives on instruction. San Diego State University, Department of Educational Technology. San Diego CA. [En red]. Disponible en: <http://edweb.sdsu.edu/courses/EDTEC540/Perspectives/Perspectives.html>. Consultado en febrero de 2006.
- Srinivas, M. y L. M. Patnaik. 1994. Genetic algorithms: A survey. *IEEE Computer*. 27(6): 17-26 p.
- Thomas, E. 2000. Intelligent tutoring systems. [En red]. Disponible en: <http://coe.sdsu.edu/eet/articles/tutoringsystem/start.htm>. En: Bob Hoffman (ed.). *The Encyclopedia of Educational Technology*. San Diego State University, Department of Educational Technology. San Diego CA. [En red]. Disponible en: <http://coe.sdsu.edu/eet/>. Consultado en marzo de 2006.
- Villa-Angulo, R. 2001. Diseño e implementación de un sistema de comunicación oral y gráfica para personas con discapacidad auditiva y del habla, utilizando redes neuronales y autómatas programables. Tesis de Maestría. Centro de Investigación Científica y de Educación Superior de Ensenada. Ensenada, B.C., México. 158 p.
- Villa-Angulo, R. y H. Hidalgo. 2001. Una interfaz adaptable basada en redes neuronales para la captura y reconocimiento del alfabeto signado en español, utilizando un guante de datos. *Memoria del 3er Encuentro Internacional de Ciencias de la Computación*. Aguascalientes, México. 147-156 p.
- Villa-Angulo e Hidalgo-Silva. 2005. A wearable Neural Interfaca for Real Time Translation of Spanish Deaf Sign Language to Voice and Writing. *Journal of Applied Research and Technology*. 3(3): 169-186 p.
- Wilson, A. D. y A. F. Bobick. 1999. Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 21(9): 884-900 p.
- Zhao, M., F. K. H. Quek. y X. Wu. 1998. RIEVL: Recursive induction learning in hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 20(11): 1174-1185 p.