

TEORÍA DEL MARCO RELACIONAL: UNA TEORÍA DEL APRENDIZAJE PARA EL DISEÑO DE SOFTWARE EDUCATIVO

Mañas Mañas I.¹, Mesa Manjón H.², Silva González R.³

Universidad de Almería, España,

¹imanas@alboran.ual.es

²hilarionmesa@supercable.es

³rsg_ual@yahoo.es

Resumen. Presentamos a la Relational Frame Theory (RFT: Hayes, Barnes-Holmes y Roche, 2001) o Teoría del Marco Relacional (TMR) como una teoría del aprendizaje para el desarrollo de software educativo. La TMR es una teoría que estudia el lenguaje y la cognición humana desde una perspectiva monista. Comentamos algunas de las limitaciones de los actuales software educativos y presentamos las propiedades del lenguaje como la principal aportación de la TMR al diseño de software educativos.

1 Introducción.

Presentamos a la Relational Frame Theory (RFT), en español, la Teoría del Marco Relacional (TMR), como una teoría empírica, que estudia el lenguaje y la cognición humana desde una perspectiva monista. Proponemos a la TMR como una alternativa valiosa para el desarrollo de software educativo.

La psicología ha contribuido al desarrollo de software educativo, principalmente, a través de las aportaciones de las teorías del aprendizaje. Según Urbina (1999), dichas teorías han influido e influyen en los procesos de diseño y en las situaciones de aplicación de los programas informáticos educativos. De forma similar, Gros (1997) ha señalado que las concepciones teóricas afectan al desarrollo de software educativos, y lo hacen modificando los contenidos, las estrategias de enseñanza, el diseño de pantalla, seleccionando a los usuarios, etc.

Hoy día no se conoce qué teoría del aprendizaje es la más idónea para el desarrollo de software educativo. Al contrario, el panorama actual refleja ambigüedad en cuanto a la teoría que habría de tenerse en cuenta, y lo más importante, por qué se habría de tener en cuenta una teoría determinada y no otra. Actualmente, no existe una teoría que pueda solventar muchos de los problemas a los que se enfrentan los diseñadores de software educativos.

Dos de las clasificaciones que se han realizado de software educativos han sido, por un lado, las de Gros (1997), y por otro lado, las de Colom, Sureda y Salinas (1988). Gros

clasifica los software educativos en cuatro categorías: Tutorial; práctica y ejercitación; simulación y por último; hipertexto o hipermedia. La clasificación que realizan Colom, Sureda y Salinas se basa en: Considerar el aprendizaje *a través* del ordenador y, por otro lado, considerar el aprendizaje *con* el ordenador. En el primer caso, la utilidad del ordenador es la de proporcionar ayuda para adquirir determinados conocimientos. Un ejemplo de esto serían los programas de “Enseñanza Asistida por Ordenador” (EAO). La segunda clasificación de los autores invita a considerar al ordenador como un instrumento cuya finalidad es el desarrollo de procesos cognitivos. Un ejemplo de este tipo de software educativo es el LOGO.

Primero citaremos, varios conceptos y premisas de algunos autores y su repercusión en el diseño de software educativo. Comentaremos varias limitaciones que presentan los actuales software educativos aportando algunas soluciones. Haremos hincapié en la importancia de una teoría del aprendizaje válida para el diseño de software educativo y, por último, proponemos a la TMR como una teoría alternativa para el diseño de software educativo solventando algunas de las limitaciones de las actuales teorías.

2 Teorías del Aprendizaje y Software Educativo: EAO y LOGO

Un autor que ha reunido las aportaciones de algunas teorías del aprendizaje en el campo del software educativo ha sido Urbina (1999). Una de las teorías citadas es la de Skinner, pionero en lo que se ha denominado “Enseñanza Programada”. Enseñanza, que sería la base de los primeros usos del ordenador como herramienta en el campo educativo (o del aprendizaje). Daría como resultado lo que se conoce como la EAO (Enseñanza Asistida por Ordenador). Algunos de los inconvenientes destacados de la EAO son la rigidez en la secuencia de los contenidos, la escasa o nula participación del usuario y su carácter mecánico y repetitivo, entre otros.

Ausubel (Ausubel, Novak y Hanesian, 1989) autor de la Teoría del Aprendizaje Significativo, se opone a la EAO por su carácter mecánico y por el hecho de que la EAO no facilita la interacción, tanto de los alumnos entre sí, como de éstos con el tutor. Esto quedaría justificado atendiendo a uno de los conceptos centrales de su teoría: “Aprendizaje por Recepción”, donde la figura del tutor es fundamental, ya que es éste quien organiza el contenido del material para que el alumno lo “reciba”.

Autores como Vygotski, Piaget, Bruner, Gagné y Papert, han tenido y siguen teniendo mucha influencia en el desarrollo de software educativo. Vygotsky con el concepto de “Zona de Desarrollo Próxima”; Piaget con los estadios del desarrollo (sensoriomotor, pre-operacional, operaciones concretas y operaciones formales) y los conceptos de “Adaptación” y “Acomodación”. Piaget ha sido, y continúa siendo, uno de los autores que mayor repercusión ha tenido en el desarrollo de teorías del aprendizaje, que a su vez, han influido en los diseños de software educativos. Por ejemplo, Bruner, autor que pretende, mediante el entrenamiento en operaciones lógicas, estimular cognitivamente al alumno; Algunas de las aportaciones de Gagné se derivan del principio de no castigar las respuestas del alumno sustituyendo la sanción por feedback con el propósito de motivarlos intrínsecamente. Papert

(1995) es el creador del primer lenguaje de programación para niños, el LOGO. Programa cuyo propósito inicial era que los niños adquirieran conceptos básicos de geometría. Papert tiene una visión constructivista del conocimiento y fácilmente se pueden observar las influencias de Piaget en su teoría y, por ende, en el LOGO.

Se podría decir que primero surgió la EAO, y que ésta se basa en los principios de teorías denominadas “conductistas”. Más tarde, intentando solventar los problemas e inconvenientes de la EAO, surge el LOGO, un lenguaje de programación basado en asunciones o teorías que podríamos denominar “cognitivas o constructivistas”.

Autores como Martí (1992) han indicado las ventajas e inconvenientes de la EAO. Este mismo autor junto a Delval (1986), han señalado algunas de las limitaciones del LOGO. Según estos autores, algunos de los inconvenientes de la EAO son: Rigidez en la secuencia de los contenidos; el rol pasivo del alumno; que impide la interacción con el tutor; no provee información al usuario sobre su ejecución en términos de correcto/incorrecto, etc. Por otro lado, algunas de las limitaciones que se le han hecho al LOGO son las siguientes: Según Delval (1986), los resultados de las investigaciones llevadas a cabo sobre el efecto que produce el LOGO aplicado en las escuelas muestran efectos prácticamente inapreciables. Delval apunta que los ejercicios que se realizan en las escuelas con los ordenadores son tareas de tipo repetitivas careciendo de interés. Por otro lado, Martí (1992) señala otra carencia del LOGO, la de no tener en cuenta las diferencias individuales, por lo que, los resultados obtenidos pueden ser muy variables. También se han indicado limitaciones del LOGO por no ofrecer soluciones a los errores que pueda cometer el alumno, o por ejemplo, por no contemplar el papel del profesor.

3 Limitaciones de los actuales software educativos.

La mayoría de los software educativos no contienen de forma explícita los objetivos que deben alcanzarse, o lo que es lo mismo, según nuestro punto de vista: No indican la utilidad del propio software, es decir, para qué sirven. En otras ocasiones, algunos software, sí ofrecen información sobre los objetivos que persiguen. Por regla general, los objetivos pueden quedar expresados en términos tales como: “Mejorar las actividades metacognitivas” o, por ejemplo, “estimular los procesos cognitivos” Esto ocurre, por ejemplo, en el caso del LOGO.

Según nuestro punto de vista, un software debe contener los objetivos que persigue de forma explícita ya que éstos representan la finalidad del software y, por tanto, la utilidad de éste. Pero un software, además de contener de forma explícita los objetivos, debe de permitir poder evaluar si dichos objetivos se han alcanzado, y si es así, hasta que punto. Sin embargo, la mayoría de los software educativos no posibilitan el hecho de poder evaluar sus efectos en el usuario. Esta es otra de las mayores limitaciones que presentan los software educativos en la actualidad. Esto quizás se deba, entre otros motivos, al modo en que quedan expresados los objetivos que se pretenden conseguir. Como en el ejemplo anterior (en el caso del LOGO), los objetivos quedan definidos de forma tal, que en la mayoría de los casos, impiden el hecho de poder comprobar si éstos se han alcanzado o no. Es decir, un

software determinado puede pretender “mejorar las actividades metacognitivas” en el usuario, sin embargo, en ocasiones, no ofrecen modo alguno de poder comprobar la consecución de tal objetivo.

Este inconveniente podría solventarse diseñando un software que permitiera poder evaluar los efectos que produce. Para lograr esto, es necesario expresar los objetivos de forma que permitan ser medidos, y para ello, se han de definir operacionalmente. Siguiendo con el ejemplo anterior, si se pretende “mejorar las actividades metacognitivas”, debemos suponer que: Por un lado, el software entraña una serie de “tareas” que el usuario ha de realizar y que éstas se crearon con un fin, el de “mejorar las actividades metacognitivas”; y por otro lado, el software debe permitir evaluar el efecto de dichas “tareas”, es decir, “la mejora de las actividades metacognitivas”.

Para poder comprobar el efecto (“la mejora de las actividades metacognitivas”) del software, primero se deberían de obtener medidas del usuario, mediante la ejecución por parte de éste, en “tareas” que midan “actividad metacognitiva”. Dichas medidas han de ser objetivas, por ejemplo, número de respuestas correctas, latencia de respuesta, nivel de ejecución expresado en tantos por ciento, etc. Segundo, se llevarían a cabo las “tareas” que se espera “mejoren las actividades metacognitivas”. Estas “tareas” son las que representarían el motivo de existencia del propio software. Tercero y último, se volvería a obtener otra medida similar a la primera. Es decir, se ha de evaluar la ejecución del sujeto en “tareas” que midan de forma equivalente “actividad metacognitiva” pero utilizando ítems diferentes. La diferencia entre ambas medidas podría representar la “mejora” de las “actividades metacognitivas”. Es decir, una medida expresada cuantitativamente, representaría el efecto de las “tareas”, cuyo objetivo era el de “mejorar las actividades metacognitivas”. De esta forma, podría evaluarse el efecto de un determinado software educativo y conocer si este es efectivo, y si es así, cuánto de efectivo es. Podría decirse que esta secuencia es la descrita en la literatura experimental como un diseño ABA (Barlow y Hersen, 1988).

Continuando con el ejemplo anterior, cabría de esperar que el software educativo contuviera tanto las tareas que miden “aquello” que se pretende “mejorar”, así como también, las “tareas” con las que se pretenden “mejorar”. Son tareas diferentes. Las tareas que miden “aquello” que se pretende “mejorar” corresponden a las Fases A, y las tareas que se utilizan con el fin de “mejorar” corresponden a la Fase B. La primera Fase A sería la Línea Base; la Fase B es la intervención (el propósito de la aplicación del software educativo); la segunda Fase A, es la segunda medida de la conducta y es donde puede observarse el efecto de la intervención (en este caso, el efecto del software educativo).

Sin embargo, la mayoría de los software educativos no tienen en cuenta las tareas mínimas necesarias, es decir, los diseños de software no contienen elementos que le permitan validarse a sí mismos demostrando sus efectos, y por tanto, su utilidad. Es más, incluso las tareas que utilizan los software para alcanzar los objetivos perseguidos, no suelen estar validadas, siendo esto otra de las principales limitaciones de los actuales software educativos. Necesariamente, dichas tareas han de estar en estrecha relación con una teoría del aprendizaje o un modelo teórico determinado. Por lo que, en función del tipo de asunciones teóricas o principios generales del modelo, se diseñarán las tareas que contiene el software educativo. Dichas teorías, afectan también al proceso de aplicación del software, a los obje-

tivos perseguidos, al modo en que se evalúan los efectos producidos, a la hora de explicar los resultados obtenidos, etc.

¿Cómo es posible conocer qué teoría del aprendizaje es mejor que otra, si no es posible comprobar el efecto que producen sus “principios” en el usuario de software educativo? Si un software no delimita los objetivos que persigue, o lo hace, pero no evalúa si dichos objetivos se han alcanzado, ¿cómo podemos conocer su efectividad?, ¿y su utilidad?

Otras de las críticas que se pueden realizar a los software educativos, es el hecho de no atender a la competencia verbal de los usuarios como un criterio a tener en cuenta. Y, en ocasiones, cuando lo hacen, lo que tienen en cuenta no son las propiedades del lenguaje. Esto podría solucionarse con una teoría que ofrezca garantías de la implicación de la conducta verbal en función de la edad. O dicho de otro modo, una teoría que explique qué es la conducta verbal y describa sus propiedades pudiendo demostrarlas empíricamente, confiando a la variable “edad” el lugar que le corresponda.

Cuando hay ciertas evidencias de los efectos que producen algunos software educativos, las explicaciones que se realizan del fenómeno suelen ser diversas y, por lo general, suelen depender, en gran medida, de la interpretación (según una teoría) de la persona que aplica el software. Si un software es efectivo, debería de poder demostrarse sin lugar a dudas que así lo es. Para ello, el software debe poder evaluar a posteriori de forma objetiva el aprendizaje que precisamente éste espera producir. Una teoría del aprendizaje que no pueda demostrar que tras ciertas “tareas” el usuario ha “aprendido”, es una teoría del aprendizaje que no puede explicar, por otra parte, lo supuestamente “aprendido”. La explicación de un fenómeno implica la predicción y el control del mismo.

4 Una alternativa para el diseño de software educativo: Teoría del Marco Relacional.

El lenguaje es actividad relacional. Supóngase el siguiente ejemplo donde se llevan a cabo dos entrenamientos de forma directa. A un niño se le entrena, por un lado, a responder a la *bandera* de un país (B) en presencia del *nombre de ese país* (A), y por otro lado, se le entrena a responder al *nombre de la capital* del país (C) en presencia de la *bandera* de ese país (B). Es decir, se llevan a cabo dos entrenamientos: $A \rightarrow B$ y $B \rightarrow C$. No obstante, el niño aprenderá a responder no sólo a las dos relaciones entrenadas directamente, sino a seis relaciones, de las cuales cuatro de ellas son relaciones derivadas. El niño responderá a las siguientes relaciones:

- A la bandera de un país en presencia del nombre de ese país: Relación $A \rightarrow B$ (la primera relación entrenada de forma directa);
- A la capital del país en presencia de la bandera de ese país: Relación $B \rightarrow C$ (la segunda relación entrenada de forma directa);
- Al nombre del país en presencia de la bandera de ese país: Relación $B \rightarrow A$. Este tipo de relación es llamada relación Simétrica o Vínculo Mutuo.

- A la bandera en presencia del nombre de la capital de ese país: Relación $C \rightarrow B$, también denominada Simétrica o Vínculo Mutuo.
- Al nombre de la capital del país en presencia del nombre del país. Relación $A \rightarrow C$, llamada Transitiva o Vínculo Combinatorio.
- Al nombre del país en presencia del nombre de la capital de ese país. Relación $C \rightarrow A$, denominada relación de Equivalencia o Vínculo Combinatorio.

La principal característica de este fenómeno, radica en que a partir de un conjunto ordenado de aprendizajes discriminativos entre varios estímulos emergen sin entrenamiento explícito y también de forma ordenada, nuevas relaciones de control discriminativo entre ellas. Estas relaciones entre los estímulos siguen la lógica de la matemática de conjuntos, en la nomenclatura utilizada por Sidman (1971, 1994): Reflexividad ($A=A$), simetría (Si $A=B$, $B=A$), y transitiva (Si $A=B$ y $B=C$, entonces $A=C$).

La emergencia de relaciones nuevas no entrenadas explícitamente le confiere a las relaciones de equivalencia un gran interés aplicado ya que permite una considerable aceleración del aprendizaje (García, A., Gómez, J., Gutiérrez, M. y Puche, A., 2001). Se han hecho explícitas las importantes y diversas aplicaciones educativas que giran en torno a este fenómeno. Puede verse una síntesis teórica de las relaciones de equivalencia en Valero y Luciano (1992) y de la derivación de funciones en Gómez (2001).

Diferentes investigaciones indican que la derivación de relaciones de estímulos, tales como la equivalencia acontecen en individuos con competencia verbal (Barnes, McCullagh y Keenan, 1990; Devany, Hayes y Nelson, 1986), y que el procedimiento de la equivalencia puede ser usado para el establecimiento de habilidades de lectura (de Rose, de Souza, Rossito y de Rose, 1992). Se han señalado también aplicaciones en el aprendizaje de palabras en otros idiomas (Valero, 1990), de habilidades musicales (Hayes, Thompson y Hayes, 1989) o en la formación de conceptos (Luciano y Ferro, 1991), entre otras.

Para algunos analistas del comportamiento la equivalencia constituye una analogía empírica de las propiedades simbólicas del lenguaje natural (Barnes, 1994; Barnes, Browne, Smeets y Roche, 1995). Hayes (1991) considera el fenómeno de la equivalencia “en términos de un tipo de respuesta de orden superior, interpretado como un caso especial de respuestas relacionales aplicables arbitrariamente. La explicación de esta clase de respuesta será dada en términos de la historia que la hace posible”. Por lo que, la derivación de relaciones de estímulo (o aprendizaje relacional) es conducta aprendida.

La Teoría del Marco Relacional (TMR) es una aproximación analítica conductual al estudio del lenguaje humano y la cognición. Se trata, de una aproximación monista al estudio de la cognición. Un aspecto central de la TMR es que las relaciones de estímulos derivadas es conducta aprendida y explica el fenómeno de la equivalencia y las respuestas relacionales derivadas, en términos de una “operante generalizada” (Barnes-Holmes y Barnes-Holmes, 2000). La característica principal de esta operante generalizada es que es una respuesta relacional derivada aplicable arbitrariamente. Y, como operante: Se desarrolla a través del tiempo; es flexible y puede moldearse; está bajo control estimular y; estará controlada por sus consecuencias. Un marco relacional es una clase específica de operante controlada contextualmente y aplicada arbitrariamente. El comportamiento verbal se asien-

ta en marcos relacionales (Hayes y Wilson, 1993). Se diferencian, marcos de equivalencia (“igual a”) y marcos de no-equivalencia (por ejemplo, de comparación, superioridad, oposición, etc.).

Las propiedades o características de los marcos relacionales (del lenguaje), son las siguientes:

1. El Vínculo Mutuo o Bidirección (mutual entailment): Si A se relaciona con B entonces, B se relaciona con A. Esta propiedad describe la bidireccionalidad del lenguaje incluso cuando esta relación no es simétrica. Por ejemplo, si A es igual a B entonces, B es igual a A (simetría). Pero si A es mayor que B, entonces, B no es mayor que A sino menor (no simétrica).
2. El Vínculo Combinatorio o Vínculo Combinado (combinatorial entailment): Si A se relaciona con B y B se relaciona con C entonces, A se relaciona con C (transitividad) y C se relaciona con A (equivalencia). El Vínculo Combinatorio implica las relaciones de al menos tres estímulos.
3. La Transformación de Funciones (transformation of stimulus functions): Implica el cambio de funciones sin contingencias directas, tanto en marcos de equivalencia (se hablaría de Transferencia de Funciones) como en marcos de no-equivalencia (la Transformación de Funciones).

El aprendizaje relacional es la base del lenguaje. Por lo que, debemos entender que la naturaleza del lenguaje humano es de carácter bidireccional, combinatorio y sujeto a la transformación de funciones. Estas propiedades de los marcos relaciones se han demostrado empíricamente. Los métodos más utilizados son: Igualación a la Muestra, el Entrenamiento de tipo Respondiente, el Entrenamiento de Antes-Después y el Procedimiento de Evaluación Relacional.

La TMR es una teoría que puede explicar los procesos cognitivos atendiendo a las propiedades del lenguaje. Se han realizado estudios que permiten demostrar la mejora en diferentes áreas de la cognición, tales como el lenguaje y la inteligencia, entre otras (Barnes, Hegarty y Smeets, 1997; Healy, Barnes-Holmes y Smeets, 2000).

Quizás, la contribución más importante que puede realizar la TMR al diseño de software educativo es el hecho de aportar las bases empíricas de las propiedades que presenta el lenguaje humano. De esta forma, un software podría demostrar su validez y, por tanto, su utilidad. Si el software está diseñado atendiendo a las propiedades del lenguaje, éste podrá provocar y medir mediante entrenamientos y tareas de evaluación respectivamente el aprendizaje para el cual fue diseñado. Un software debe de estar diseñado para alcanzar unos objetivos explícitos y debe contener los tipos de entrenamientos o tareas que se requieren para lograr dichos objetivos. Esto es posible, cuando la teoría del aprendizaje que está a la base del software educativo puede explicar ese tipo de aprendizaje. La TMR es una teoría del aprendizaje que cumple tales criterios.

Referencias Bibliográficas

- Ausubel, D., Novak, J. y Hanesian, H. (1989). *Psicología cognitiva. Un punto de vista cognoscitivo*. Méjico. Trillas.
- Barlow, D. y Hersen, M. (1988). *Diseños experimentales de caso único*. Barcelona, España. Edt. Martínez Roca.
- Barnes, D., McCullagh, P.D., y Keenan, M. (1990). Equivalence class formation in non-hearing impaired children and hearing impaired children. *The Analysis of Verbal Behavior*, 8, 19-30.
- Barnes, D. (1994). Stimulus equivalence and relational frame theory. *The Psychological Record*, 44, 91-124.
- Barnes, D., Browne, M., Smeets, P. M., y Roche, B. (1995). A transfer of functions and a conditional transfer of functions through equivalence relations in three- to six-year old children. *The Psychological Record*, 45, 405-430.
- Barnes, D., Hegarty, N., y Smeets, P. M. (1997). Relating equivalence relations to equivalence relations: A relational framing model of complex human functioning. *The Analysis of Verbal Behaviour*, 14, 57-83.
- Barnes-Holmes, D. (2000). Behavioral pragmatism: No place for reality and truth. *The Behavior Analyst*, 23, 191-202.
- Barnes-Holmes, D. y Barnes-Holmes, Y. (2000). Explaining complex behavior: Two perspectives on the concept of generalizad operant classes. *The Psychological Record*, 50, 251-265.
- Colom, A., Sureda, J. y Salinas, J. (1988). *Tecnología y medios educativos*. Madrid. Cincel.
- Delval, J. (1986). *Niños y máquinas. Los ordenadores y la educación*. Madrid, Alianza.
- De Rose, J. T., de Souza, D. G., Rossito, A. L., y de Rose, T. M. S. (1992). Stimulus equivalence and generalization in reading after matching-to-sample by exclusion. S. C. Hayes y L. J. Hayes (Eds.), *Undertandign verbal relations* (pp. 69-82). Reno, NV: Context Press.
- Devany, J. M., Hayes, S. C., y Nelson, R. O. (1986) Equivalence class formation in language-able and language-disabled children. *Journal of the Experimental Analysis of Behavior*, 46, 243-25.
- García, A., Gómez, J., Gutiérrez, M.T. y Puche, A. (2001). Formación y ampliación de clases de equivalencia aplicadas al tratamiento de un niño autista. *Análisis y Modificación de Conducta*, 27, 114, pp. 649-699.
- Gómez, S. (2001). Derivación de relaciones de estímulos y comportamiento verbal. Algunos ejemplos de aplicación al contexto clínico. *Análisis y Modificación de Conducta*, 27, 111-136.
- Gros, B. (1997). *Diseños y programas educativos*. Barcelona. Ariel.
- Hayes, S., Barnes-Holmes, D. y Roche, B. (2001). *Relational Frame Theory: A post-skinnerian account of human language and cognition*. New York: kluwer academic.
- Hayes, L., Thompson, S. y Hayes, S. (1989). Stimulus equivalence and rule following. *Journal of the Experimental Analysis of Behavior*, 52, 275-291.
- Hayes, S. C. (1991). A relational control theory of stimulus equivalence. En L. J. Hayes & P. N. Chase (Eds.), *Dialogues on verbal behaviour* (pp. 19-40). Reno, NV: Context Press.
- Hayes, S. C. y Wilson, K. G. (1993). Some applied implications of a contemporary behaviour-analytic account of verbal events. *The Behaviour Analyst*, 16, 283-301.
- Healy, O., Barnes-Holmes, D. y Smeets, P.M. (2000). Derived relational responding as generalized operant behavior. *Journal of the Experimental Analysis of Behavior*, 74, 207-227.
- Luciano, M. C. y Ferro, R. (1991). Formación de reglas y formación de equivalencia en un estudio aplicado. Estudio preliminar. Universidad de Granada. Manuscrito no publicado.
- Martí, E. (1992). *Aprender con ordenadores en la escuela*. Barcelona, ICE-Horsori.
- Papert, S. (1995). *La máquina de los niños*. Barcelona. Paidós.
- Sidman, M. (1971). Reading and auditory-visual equivalences. *Journal of Speech and Hearing Research*, 14, 5-13.
- Sidman, M. (1994). *Stimulus Equivalence: A Rsearch Story*, Authors Cooperative, Boston.
- Urbina, S. (1999). *Informática y teorías del aprendizaje*. *Píxel-Bit*, N° 12, Enero.
- Valero, L. (1990). La emergencia de nuevas conductas a través de relaciones de equivalencia: análisis experimental de sus componentes básicos y aplicados. Tesis doctoral en microfichas. Universidad de Granada.
- Valero, L. y Luciano, M.C. (1992). Relaciones de Equivalencia: una síntesis teórica y los datos empíricos a nivel básico y aplicado. *Psicothema*, 4, 2, pp. 413-428.