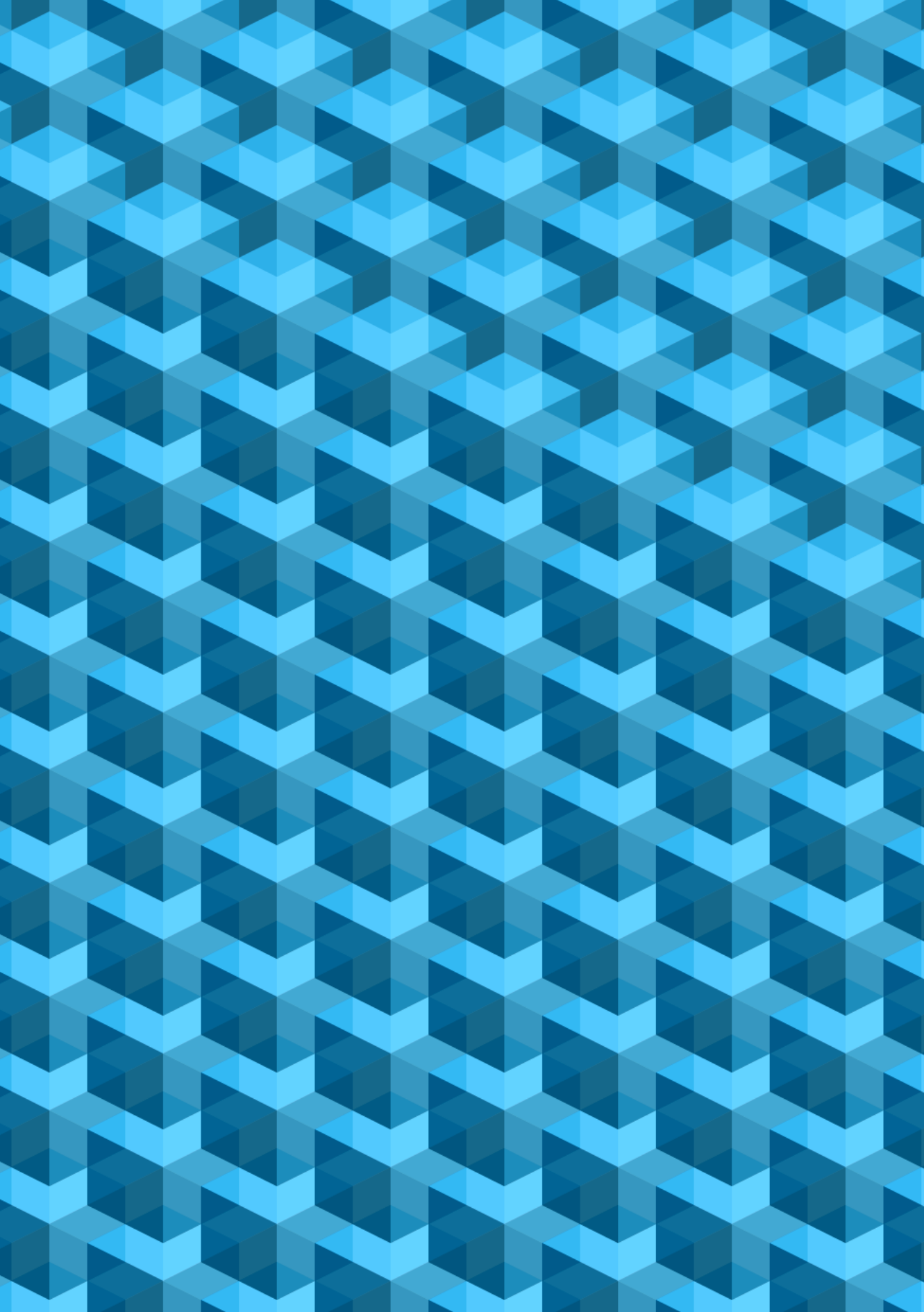
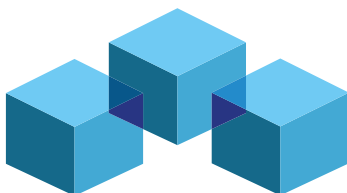


Diseño instruccional: teoría y práctica



Diseño instruccional: teoría y práctica

Luis Eduardo Otero Sotomayor



Corporación Universitaria

Unitec

MIEMBRO DE LA RED

ILUMNO

303.4
O73d

Otero Sotomayor, Luis Eduardo.
Diseño instruccional: teoría y práctica / Luis Eduardo Otero
Sotomayor ; Bogotá : Corporación Universitaria Unitec, 2016.
252 p.

1. INNOVACIONES EDUCATIVAS 2. MATERIALES DE
ENSEÑANZA 3. TECNOLOGÍA EDUCATIVA. 4. Tit.

© Corporación Universitaria Unitec®
2016. Todos los derechos reservados.

*Ninguna parte de este libro puede ser reproducida o utilizada de ninguna forma o por
ningún medio, sea electrónico o mecánico, sin permiso escrito por parte del editor.*

ISBN (obra impresa): 978-958-58198-6-3
ISBN (obra digital): 978-958-58198-7-0
Primera edición 2016

Edición

Centro de Publicaciones Académicas
Corporación Universitaria Unitec®
Calle 76 #12-61
Bogotá, D.C. Colombia
Correo electrónico: publicaciones@unitec.edu.co

Producción editorial

David Arturo Acosta Silva

Corrección de estilo

Marcela Garzón Gualteros

Fotografía cubierta

Ice Cubes

por Alexandre Hamada Possi

Licencia Creative Commons BY-2.0

<https://www.flickr.com/photos/grandexandi/>

Impresa con permiso del autor

Diseño y diagramación

Taller de Edición • Rocca® S. A.

Carrera 4A No. 26A-91, of. 203

Tel./fax: 243 2862 - 243 8561

Preprensa e impresión

Molher Impresores Ltda.

Calle 94A No. 58-13 Tel.: 256 8080

Impreso en Colombia/ Printed in Colombia

Contenido

Prólogo	11
<i>Jorge Humberto Torres Zea.</i>	
Introducción	15
Capítulo 1.	
Manejo de errores	19
1.1 El error como herramienta de aprendizaje	19
1.2 Tipos de errores	20
Capítulo 2.	
Retroalimentación	25
2.1 Definiciones de retroalimentación	25
2.2 Niveles de especificidad de la retroalimentación	26
2.3 Formas de la retroalimentación	27
2.4 Tiempo para la facilitación de la retroalimentación	28
2.5 Investigaciones sobre la retroalimentación	29
Capítulo 3.	
El control de la retroalimentación	43
3.1 Influencia sobre el control de la retroalimentación en el aprendizaje	44
3.2 Influencia sobre el control del material en el aprendizaje	46

Capítulo 4.

Resolución de problemas	49
4.1 Factores que inciden en la resolución de problemas	50
4.2 Estructuración	51
4.3 Complejidad de un problema	51
4.4 La representación del problema	52
4.5 Las diferencias individuales	52
4.6 Tipos de problemas	53

Capítulo 5.

La teoría de la carga cognitiva	59
5.1 La arquitectura de la cognición humana	60
5.2 Tipos de cargas cognitivas	63
5.3 Técnicas de la carga cognitiva que facilitan el aprendizaje	65

Capítulo 6.

Los ambientes de aprendizaje computacionales	69
6.1 Sistemas de ejercitación y práctica	69
6.2 Programas de simulación	70
6.3 Micromundos	70
6.4 Juegos	70
6.5 Sistemas expertos	70
6.6 Tutores para el aprendizaje	71
6.7 Multimedia	72
6.8 Sistemas de tutores inteligentes	73
6.8.1 Módulo experto	74
6.8.2 Métodos para la recolección de la información	75
6.8.3 Representación del conocimiento	76
6.8.4 Requerimiento de la representación del conocimiento	77
6.9 Generadores de tareas y bancos de tareas	77
6.10 Módulo del estudiante	78
6.10.1 Modelo de diagnóstico	79
6.10.2 Técnicas de diagnóstico	80

6.10.3 Funciones del módulo del estudiante	81
6.11 Módulo de enseñanza	82
6.11.1 Estrategias del tutor	82
6.11.2 Componentes del tutor	83
6.11.3 Funciones del tutor	84
6.11.4 Reglas de la planeación de la enseñanza	84
6.12 Módulo de comunicación	85
6.13 Redes semánticas	87
6.14 La interactividad	89
6.15 Objetos de aprendizaje	94

Capítulo 7.

Aprendizaje de conceptos	95
7.1 Definición de concepto	96
7.2 Taxonomía de los conceptos	96
7.3 Adquisición de conceptos	97
7.4 Fases del aprendizaje de conceptos	98

Capítulo 8.

La transferencia del aprendizaje	105
8.1 Base de la transferencia	105
8.2 Tipos de transferencia	107
8.3 La transferencia orientada al actor	107

Capítulo 9.

Diseño de un objeto de aprendizaje	113
9.1 Diseño	114
9.2 Componentes	117
9.3 Resolución de problemas	132
9.3.1 Definición de variables	135
9.3.2 Retroalimentación	137
9.4 Modelación matemática	138

Capítulo 10.

Programación en HTML, JavaScript y PHP	149
10.1 El concepto de programa	149
10.2 HTML	150
10.2.1 Las páginas de HTML y sus elementos	151
10.2.2 Tipos de datos en JavaScript	155
10.3 PHP	173
10.3.1 Tipos de datos	174
10.3.2 Un programa en PHP	176
10.3.3 Ejemplo de aplicación en aprendizaje de conceptos	205
10.4 Base de datos con MySQL	220

Referencias	225
--------------------	-----

Anexos	233
---------------	-----

Anexo A. CSS	233
---------------------	-----

Anexo B. Una clase en PHP y en JavaScript	240
--	-----

Anexo C. Estructuras de control	242
--	-----

Prólogo

Más que en el pasado, el papel del docente está lleno de retos, exigencias y compromisos que diariamente ponen a prueba esta hermosa vocación, cuya misión siempre está encaminada a lograr la transferencia del conocimiento a partir de procesos de enseñanza-aprendizaje; en estos últimos las estrategias y didácticas deben favorecer un ejercicio significativo en los estudiantes. Esto cobra una notoria relevancia frente al impacto que las tecnologías de la información y la comunicación (TIC) han alcanzado en todos los ámbitos de la vida y, sin duda, especialmente en la educación. Nos hemos transformado en una sociedad donde la democratización de la información —presentada esta de manera multimedial— favorece el desarrollo del conocimiento. Frente a ello, los docentes tienen la responsabilidad de adoptar nuevos roles a partir de su reinención y de su reaprender para propiciar nuevos espacios de aprendizaje diseñados desde la gestión del conocimiento a través de las TIC —sin que estas se conviertan en protagonistas—, buscando que sean un medio que facilite el aprendizaje a partir de la estructuración de recursos didácticos que tengan una adecuada fundamentación pedagógica y permitan el logro de los objetivos del aprendizaje.

En este sentido, el presente libro aborda en cada uno de sus capítulos aspectos basados en diversas teorías de los procesos cognitivos, los cuales deberían ser tenidos en cuenta al momento de diseñar ambientes de aprendizaje computacionales. De manera práctica estos se implementan y adecuan por medio del desarrollo de recursos educativos. La síntesis de estos procesos y teorías del aprendizaje es el resultado de las investigaciones y experiencias del profesor Luis Eduardo Otero, las cuales se convierten en los pilares que fundamentan la estructuración de los recursos de aprendizaje denominados objetos de aprendizaje.

En el primer capítulo, «Manejo de errores», se presenta el tema desde diferentes teorías, así como también la forma en la que este se evidencia en el desarrollo cognitivo del ser humano; de tal manera, este se establece como un elemento causal en el aprendizaje, ya que posibilita la adecuada formulación de la retroalimentación, propiciando la estructuración de nuevos esquemas en diferentes contextos que permitan la resolución de problemas.

En el segundo capítulo, «Retroalimentación», se exponen las posiciones teóricas de diversos actores frente al tema, con lo que se busca inicialmente identificar por qué se presenta el error en el ejercicio académico para, a partir de allí, encontrar la forma adecuada de reorientar el proceso formativo; lo anterior con el objetivo de que el estudiante haga una comprobación de lo asimilado de forma autónoma.

Posteriormente, en el siguiente capítulo, «Control de la retroalimentación», se indica cómo se pueden incorporar las TIC en el diseño de ambientes de aprendizaje computacionales como recurso de mediación pedagógica. En este caso el control de la retroalimentación podrá ser ejercido directamente por los estudiantes que interactúan (por medio de instrucciones con un recurso tecnológico de aprendizaje), o bien será ejecutado automáticamente por el sistema.

El cuarto capítulo, «Resolución de problemas», plantea la posibilidad dar solución a un problema de forma efectiva y rápida gracias a la capacidad de

identificar las situaciones vividas y relacionarlas con las nuevas. Esto se presenta cuando los modelos mentales de una persona son mayores a los que podría tener un neófito; mientras que este último, precisamente debido a que posee poca información, se encuentra limitado al momento de hallar la solución adecuada. Con base en lo anterior se plantea la importancia de formular problemas bien estructurados en el ambiente de aprendizaje computacional, donde se integren los elementos necesarios que den respuesta a reglas y directrices específicas. Así como en los anteriores capítulos, en este se realiza una mirada a cada uno de los planteamientos teóricos expuestos por diferentes autores, haciendo énfasis especialmente en los tipos de problemas, los cuales son expuestos de una manera didáctica y aplicada al ámbito educativo.

Cómo se percibe, se almacena y se clasifica la información en los procesos cognitivos son los temas más relevantes que se tratan en el quinto capítulo, «Teoría de la carga cognitiva». En él se detalla cómo la dimensión sensorial es la puerta de entrada a los datos que se procesan y que activan la memoria de trabajo y la memoria a largo plazo. Esto resulta relevante al momento de establecer estrategias que propicien el aprendizaje a través de ambientes de aprendizaje computacionales, basados en la forma adecuada en que se debe presentar la información.

Los ambientes de aprendizaje computacionales, descritos en el sexto capítulo, muestran el panorama de las posibilidades multimediales existentes para su aplicación y cómo cada uno de estos medios se ajusta a situaciones o necesidades particulares en el momento de propiciar un aprendizaje específico a través de ellos.

El séptimo capítulo, «Aprendizaje de conceptos», presenta una amplia perspectiva teórica sobre el tema a partir de su definición, taxonomía, proceso de adquisición y fases de aprendizaje y cómo dichos conceptos pueden ser identificados en un proceso de aprendizaje a partir de la formulación de ejemplos y contraejemplos.

El tema de transferencia del aprendizaje, tratado en el octavo capítulo, se refiere a la capacidad de representar analógicamente una situación, la cual puede ser aplicada a diferentes contextos en los cuales se identifica la misma estructura profunda; de esta manera, una persona estaría en capacidad de realizar inferencias en una nueva situación o problema que se le presente.

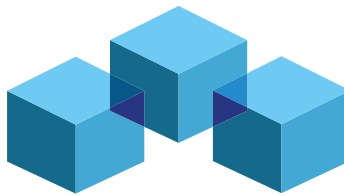
Por último, nos encontramos con los capítulos noveno y décimo, en donde la teoría es implementada a través de la creación o desarrollo de un objeto de aprendizaje, el cual propicie el aprendizaje autónomo, y esté cimentado en una adecuada fundamentación pedagógica resultado de las investigaciones y planteamientos expuestos en este libro.

JORGE HUMBERTO TORRES ZEA.

Publicista.

Magister en Tecnología de la Información y la

Comunicación Aplicada a la Educación.



Introducción

Esta obra se origina a partir de algunas de las investigaciones que he desarrollado como miembro del grupo de investigación *Ciencia, tecnología y sociedad*, línea de investigación en tecnología educativa, de la Corporación Universitaria Unitec®. De esta manera, deseo ofrecer en ella una perspectiva científica para la construcción de ambientes de aprendizaje y objetos de aprendizaje, basados en la psicología del aprendizaje. Así, no se trata de poner a disposición de los estudiantes programas informáticos vistosos que nacen del capricho de quien los realiza o libros electrónicos que ofrecen escasa o ninguna ayuda en el momento que se requiere enfrentar alguna dificultad cognitiva. Este libro está basado en los desarrollos investigativos que se han realizado sobre tema.

Teniendo en cuenta las potencialidades del computador, se pueden construir programas o sistemas interactivos que tienen en cuenta el estado de conocimiento del estudiante. Al introducir la información a través de cualquier medio conveniente, el sistema puede responder con una retroalimentación adecuada al tipo de información proveída. Así las cosas, si existe error, responderá con un mensaje dependiente de la naturaleza de dicho error.

Por otro lado, se debe conocer la influencia que tiene la retroalimentación sobre la información que posee el estudiante. Puede suceder que dicha

retroalimentación actúe con una función de confirmación, de corrección o que no se cumpla ninguna función. En el último caso, se debe averiguar las causas para que la retroalimentación ejerza su función: corregir errores cuando se produce una respuesta o se resuelve un problema. Por tanto, las diferentes formas de retroalimentación brindan una ayuda teniendo en cuenta las condiciones de aprendizaje.

En esta obra se ofrece una herramienta propuesta por Narciss (2001), que muestra paso a paso el análisis de conocimientos y habilidades, los errores que se pueden cometer y la retroalimentación para corregirlos. Asimismo, se ofrecen también criterios para determinar el tipo de control sobre la retroalimentación. El control lo puede tener el estudiante o el sistema; cuando lo tiene el estudiante, es él quien decide cuándo consultar la retroalimentación, mientras que cuando lo tiene el sistema, la retroalimentación se produce automáticamente al generar un mensaje de error o al informar que la respuesta es correcta.

En materia de resolución de problemas, se debe tener en cuenta la complejidad, la manera de representarlos, la forma como se estructuran y los tipos de problemas que se enfrentan. Tanto Sweller (1988) como Gick y Holyoak (1983) consideran que la competencia de resolución de problemas depende de los esquemas construidos en la resolución de otros problemas del mismo tipo.

Por su parte, la inclusión de la teoría de la carga cognitiva en nuestro discurso nos permite tener en cuenta la arquitectura de la cognición humana. El principio fundamental es que, en el contexto del aprendizaje, no se deben sobrepasar los recursos mentales que posee un individuo ya que el aprendizaje podría no producirse. Por lo tanto, se debería considerar la carga cognitiva total, la cual está compuesta de la carga cognitiva intrínseca y la carga cognitiva extrínseca. La primera se refiere a la dificultad inherente del tema; la segunda es manejable y es propia del material.

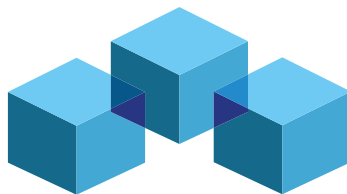
En consecuencia, una de las premisas claves de este libro es que los hallazgos encontrados por la psicología del aprendizaje deberían plasmarse en los

ambientes computacionales de aprendizaje. Es por ello que en las páginas que siguen describiremos su aplicación, entre otros, a los sistemas de ejercitación y práctica, los programas de simulación y los sistemas expertos.

Por otro lado, también deberíamos tener en cuenta la manera como se facilitan los contenidos. Es así como para el desarrollo de la competencia de resolución de problemas ofrecemos los resultados de las investigaciones alrededor de la teoría de la transferencia. Esta consiste en reconocer, en un contexto problemático, la estructura de otro problema que se ha resuelto previamente, para poder aplicar las mismas estrategias de resolución. La resolución de problemas está ligada a un contexto disciplinar y, para aprender los conceptos que lo componen, se puede utilizar la teoría del aprendizaje de conceptos. De esta forma, se puede considerar que un concepto realmente se ha aprendido cuando el estudiante puede discernir entre los atributos del concepto y evaluar nuevos conceptos con base en la membrecía de la categoría del concepto.

En los capítulos finales trataremos de materializar toda la teoría expuesta al presentar el diseño de un objeto de aprendizaje que se construyó teniendo en cuenta el análisis de tarea propuesto por Narciss (2001). Asimismo, expon-dremos conocimientos básicos de programación en HTML, JavaScript y PHP, esperando que sirva de estímulo a aquellos que les gusta traspasar fronteras.

Luis Eduardo Otero Sotomayor



Capítulo 1.

Manejo de errores

Los errores son una oportunidad muy importante para clarificar los malos entendidos en el aprendizaje; por tanto, constituyen un factor crítico de la retroalimentación, dado que según sea su naturaleza, así será la forma de retroalimentación que se puede llegar a facilitar. De tal forma, los procedimientos que tienen ver con el análisis de errores comunes pueden ser más útiles incluso que la tradicional retroalimentación dada a las respuestas correctas. En este capítulo mostraremos su importancia y algunas investigaciones en torno al tema.

1.1 El error como herramienta de aprendizaje

De acuerdo con Mory, los errores cometidos en el aprendizaje memorístico tienden a persistir (2004); de igual manera, Skinner (1958) encontró que en la instrucción programada sucede lo mismo; por su parte, Elley (1966, citada en Mory, 2004) probó la hipótesis de que el error desempeña un papel diferente en el aprendizaje memorístico frente a aquel del aprendizaje de tareas significativas. Lo importante de estos resultados es que demuestran que hay más retención en el aprendizaje memorístico cuando hay menos errores. También verificaron que el error persiste después de la retroalimentación inmediata.

Es así como Elley demostró que, cuando el aprendiz enfrenta problemas significativos, la incidencia del error no tiene relación con el desempeño.

De acuerdo con Mory (2004), la perspectiva actual considera al error como una oportunidad importante para clarificar los malos entendidos en el aprendizaje, convirtiéndolo en un factor relevante en la retroalimentación. La afirmación de que la función de la retroalimentación consiste en corregir errores hace que sea una función crítica en el proceso correctivo.

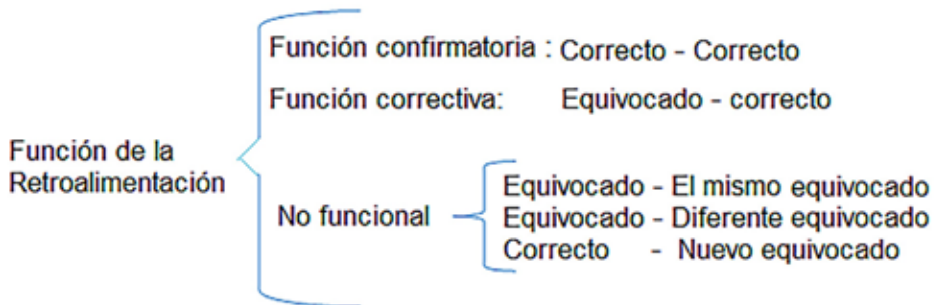
1.2 Tipos de errores

Phye y Andre (1989, citados en Mory 2004) dividen los eventos que pueden ocurrir cuando se usa retroalimentación informativa en diseños experimentales pre-postest, en tres funciones: una función confirmatoria, donde se confirma una respuesta correcta del pretest (una combinación de correcto-correcto); una correctiva, la cual corrige errores del pretest (una secuencia de equivocado-correcto); y una no funcional, cuando no cumple ninguna función (en este caso aparecen errores en postest).

Hay tres tipos de errores en los que se considera que la retroalimentación no es funcional: el primer tipo es aquel error recurrente, lo cual lo hace perseverante en su naturaleza; sucede cuando una respuesta inicial errónea se presenta en el postest, sin que la retroalimentación proveída produzca algún efecto; el segundo tipo ocurre cuando no se tiene en cuenta algún elemento en el pretest y el error es diferente a través de los ensayos; por su parte, el último tipo ocurre cuando la respuesta fue buena en pretest, pero cambió a errónea en el postest, tal vez porque en el pretest se adivinó la respuesta.

De esta manera, las combinaciones posibles de respuestas pretest-postest serían: 1) correcto-correcto; 2) error-correcto; 3) error-el mismo error; 4) error-diferente error; y 5) correcto-nuevo equivocado (figura 1).

FIGURA 1. Combinación de respuestas según los tipos de errores.



Por otra parte, Mory (2004) clasifica los errores de acuerdo con el resultado específico, los cuales pueden ser serios y no serios. Esta seriedad del error se mide por la distancia entre la regla equivocada y la correcta. La medida de la seriedad del tipo de error indica en qué forma una regla incorrecta difiere de la regla correcta.

En el aprendizaje de conceptos los errores se clasifican en generalización *por arriba*, generalización *por abajo* y *concepción errónea* (Tennyson & Cocchiarella, 1986, citados en Mory, 2004). Cuando los estudiantes están aprendiendo a clasificar instancias de un concepto deben discriminar entre ejemplos y contraejemplos. Algunos contraejemplos son difíciles de discriminar al compararlos con ejemplos del concepto; a estos se les llama *contraejemplos cercanos* (Dempsey, 1988, citado en Mory, 2004).

Cuando un estudiante está cometiendo un error de generalización por arriba, ello implica que clasifica instancias que no pertenecen al concepto como ejemplo; de esta manera, es probable que tenga problemas con la generalización fina del concepto. Pero si, por otra parte, el estudiante regularmente clasifica contraejemplos que son lejanos como verdaderos ejemplos, esto significa que tiene problemas de generalización por debajo, resultando en una discriminación gruesa.

Debido a que la generalización fina es más difícil de discriminar, se espera que ocurra con más frecuencia. Dempsey encontró que los estudiantes que cometían menos errores de discriminación fina durante la instrucción obtenían mejores resultados en un test de retención. Por su parte, Mory (2004) afirma que estos hallazgos estimulan el análisis de contraejemplos cercanos y lejanos asociados con la discriminación fina y gruesa cuando se trabaja en tareas de aprendizaje de conceptos.

Otro trabajo citado por Mory es el de Meyer (1986), quien identifica cuatro errores durante la revisión de una investigación sobre corrección de errores por parte de los estudiantes: 1) errores por falta de información o de conocimiento; 2) errores motores, que se originan cuando el estudiante conoce la información pero no puede expresarla; 3) confusiones, las cuales se presentan cuando el estudiante no puede discernir correctamente entre conceptos e ideas, y 4) errores de aplicación de reglas, que resultan de utilizar incorrectamente las reglas en situaciones de resolución de problemas.

De acuerdo con Mory, en la corrección de errores es donde la retroalimentación tiene los efectos más promisorios. Por tanto, los investigadores deben examinar maneras para manipular la retroalimentación con el fin de maximizar los eventos de aprendizaje. Asimismo, los procedimientos que tienen que ver con el análisis de errores comunes pueden ser más útiles que la tradicional retroalimentación de respuestas correctas, y la información de la retroalimentación adaptativa se puede prestar para trabajar en ambientes basados en computador, donde este puede analizar los tipos de errores que se cometen y, a partir de allí, generar la retroalimentación pertinente.

Otero (2012) llevó a cabo una investigación para medir el efecto del manejo de errores en la comprensión de un dominio de conocimiento y en la competencia de resolución de problemas en estudiantes de primer semestre de ingeniería de una institución de educación superior bogotana. El estudio fue de tipo cuasi experimental, con diseño pretest-postest, unifactorial con dos valores y con un grupo control y uno experimental. El grupo de control se sometió a clases magistrales y el grupo experimental a un objeto de

aprendizaje con manejo de error y retroalimentación del tipo de conocimiento de la respuesta correcta. La variable dependiente la constituyó el desempeño mostrado por los estudiantes en un examen sobre los temas aprendidos en el laboratorio; asimismo, a través del análisis de error, se determinó la función que prestó la retroalimentación.

Para el estudio se suministró a los participantes una prueba de entrada para evaluar los conocimientos sobre la línea recta, lo cual permitió controlar las diferencias frente a sus conocimientos previos. Luego se aplicó el tutor del objeto de aprendizaje, para luego realizar una evaluación sobre los aspectos teóricos; seguidamente, se sometió a los estudiantes a la resolución de problemas y, finalmente, se hizo otra prueba, para verificar el efecto del objeto aplicado.

Se conformaron dos grupos (A y B); los estudiantes del grupo A se enfrentaron al objeto de aprendizaje, el cual ofrecía la posibilidad de adquirir conocimientos y fortalecer la competencia de resolución de problemas en relación con la línea recta. Para favorecer la construcción de esquemas y la formación de conocimiento estructural en los estudiantes, se utilizó el manejo de errores que, a través de mensajes, facilitaba la adquisición de conocimiento sobre el tema y la competencia de utilización de estos conocimientos en la resolución de problemas.

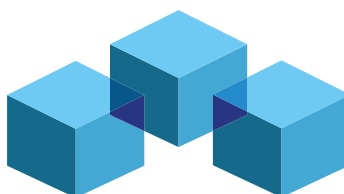
Los estudiantes del grupo B se sometieron a una clase magistral. Se comparó con este grupo precisamente porque se deseaba brindar alternativas de formación basadas en las características tecnológicas que ofrece el mundo actual. El experimento terminó con una prueba en la que cada pregunta tenía como propósito inducir en el estudiante la construcción de esquemas necesarios para la resolución de problemas. Para contrastar la hipótesis de la no existencia de diferencia significativa entre los grupos (es decir, si entre los grupos existió diferencia en el desempeño) se utilizó el paquete estadístico SPSS™ 15.0 (IBM Corporation) para Windows. En este caso, se buscó determinar si existía una diferencia significativa entre las medias, observando el valor de la significancia. La significancia representa la probabilidad de que la diferencia en el estadístico observado se deba al azar. El estadístico es significativo

cuando el valor de la significancia es menor que la probabilidad escogida para el error tipo I, alfa, que comúnmente se fija en 0.05. De acuerdo con el examen de entrada realizado, no hubo diferencia significativa en el conocimiento que poseían los estudiantes antes de comenzar la experimentación, pues los dos grupos mostraron el mismo desempeño (significancia: 0.866).

De acuerdo con la prueba de los rangos con signo de Wilcoxon y tomando la totalidad de los estudiantes, se pudo observar que hubo diferencia significativa entre las notas del examen de entrada y las del examen de salida, a favor del desempeño mostrado en el segundo de ellos (significancia: 0.004).

En esta prueba se verificó que en el 35 % de los casos tuvo una función confirmativa, siendo que la respuesta fue correcta en el pretest y en el postest. Por su parte, en el 46.1 % de los casos la función fue correctiva: de incorrecta, la respuesta pasó a correcta; en el 7.7 %, de las respuestas correctas se cambiaron a incorrectas. Como se evidencia, en el 81.1 % de los casos la retroalimentación fue funcional; esto quiere decir que los estudiantes, en general, sin tener en cuenta el grupo, aprovecharon las clases sobre la línea recta.

La comparación de desempeño entre los grupos mostró que no hubo diferencia significativa entre estos, lo que demuestra que la utilización de objetos de aprendizaje en la formación es una buena alternativa. La investigación de Otero (2012) puso a prueba la posibilidad de adquirir conocimiento y la competencia de resolución de problemas utilizando el análisis de error como estrategia para el desarrollo de objetos de aprendizaje, como lo sustentan el valor de significación y el porcentaje de los casos funcionales de la retroalimentación en el análisis de error.



Capítulo 2.

Retroalimentación

En este capítulo revisaremos la definición del concepto de retroalimentación, sus formas y el tiempo en el cual se presenta. Se traen a colación diferentes investigaciones en las que se ha evaluado su efecto sobre el aprendizaje.

2.1 Definiciones de retroalimentación

En el contexto del proceso enseñanza-aprendizaje han surgido diversas definiciones de la retroalimentación que reflejan su uso en dicho contexto. Se tiene entonces, por ejemplo, que la retroalimentación ha sido entendida como cualquier mensaje generado en respuesta a la acción de un aprendiz (Mason & Bruning, 2001) y, en este sentido, ayuda a identificar errores y a generar consciencia acerca de concepciones erróneas. Para Lemley (2005), por su parte, la retroalimentación es uno de los muchos procedimientos que le informan al estudiante si su respuesta ha sido correcta o incorrecta, dándole a conocer el porqué. Asimismo, puede ser entendida como cualquier comunicación o procedimiento dado para informar a un aprendiz acerca de la exactitud de su respuesta (Mory, 2004).

Como se deduce de las definiciones anteriores, la retroalimentación consiste en darle a conocer a una persona el desempeño que tuvo al responder una pregunta o al ejecutar una tarea. Esta interacción entre quien aprende y quien enseña –o lo que enseña– se puede utilizar para identificar en el proceso de enseñanza-aprendizaje tanto la fuente de éxito como la fuente de error de quien emite la respuesta. La respuesta correcta no tiene problema, pues eso es precisamente lo que se busca. Por otra parte, la respuesta errónea se puede generar por varias razones: porque el aprendiz no conoce la respuesta, porque se presentan errores no intencionales en la entrada, porque no se comprende la pregunta o por el manejo de conceptos erróneos.

Por tanto, la retroalimentación debe estar acorde con la naturaleza del error: si el estudiante no conoce la respuesta, se le deben brindar tanto el conocimiento teórico como la práctica necesaria para que pueda enfrentar las preguntas o problemas propuestos; si se trata de un error no intencional, es necesario hacerle saber que la respuesta no es correcta, brindando pautas para que pueda corregir la entrada; cuando no comprende la pregunta, se le debe explicar de otras maneras para facilitar su comprensión y, cuando se presenta por concepción errónea, la retroalimentación debe ser más elaborada porque el estudiante está convencido de la exactitud de lo que respondió.

2.2 Niveles de especificidad de la retroalimentación

Cuando se habla de los niveles de especificidad se hace referencia al nivel de información presente en los mensajes de retroalimentación (Goodman & Wood, 2004). Según estos autores, a medida que aumenta la especificidad de la retroalimentación, los mensajes se van enfocando más hacia la respuesta correcta; lo anterior trae como consecuencia que, a corto plazo, mejore el desempeño; pero a largo plazo, el aprendizaje se ve afectado debido a que cuando la especificidad es alta, disminuyen los procesos de inferencia que conectan la acción con los resultados, dado que el proceso de inferencia lo ejecuta la retroalimentación.

Cuando la retroalimentación es menos frecuente y el nivel de especificidad es bajo, el estudiante se compromete con una mayor exploración y con el procesamiento de la información en búsqueda de la respuesta correcta. Como veremos más adelante, lo que han concluido algunos investigadores (Goodman & Wood, 2004; Otero, 2010a) es que el nivel de especificidad depende de las condiciones del aprendizaje que se esté efectuando.

2.3 Formas de la retroalimentación

La retroalimentación asume diversas formas según su nivel de especificidad (Otero, 2010a). Kulhavy y Stock (1989, citados en Mason & Bruning, 2001) catalogan dos formas de retroalimentación: verificación y elaboración; la primera le informa al estudiante si su respuesta es correcta o no; la segunda conduce al estudiante hacia la respuesta correcta, guiándolo y mostrándole sus concepciones erróneas. Al mismo tiempo, estos autores consideran que la inclusión de la verificación y de la elaboración produce una gran ganancia en el aprendizaje.

Las formas de retroalimentación establecidas en la literatura son las de conocimiento de resultados (KR, del inglés *knowledge of results*), conocimiento de la respuesta correcta (KCR, *knowledge of the correct response*), responder hasta llegar a la respuesta correcta o múltiples intentos (AUC, *answer until correct*) y retroalimentación elaborada (EF, *elaborated feedback*). Mason y Bruning (2001) mencionan otras formas de retroalimentación, pero estas se pueden incluir en la retroalimentación elaborada. Por su parte, Narciss (2001) manifiesta que las formas KR, KCR y AUC pueden servir para tareas simples, pero no proveen la información necesaria para corregir errores o para enfrentar tareas de aprendizaje más complejas.

La retroalimentación elaborada puede existir de muchas formas, más en lo que sigue expondremos las propuestas por Mason y Bruning (2001):

Contingente al tema. Este tipo de retroalimentación provee información de verificación sobre la pregunta e información de elaboración general relacionada

con el tema que se está tratando. Cuando se produce el error, a los estudiantes se les presenta material de aprendizaje en el que aparece la respuesta y, alternativamente, se les da información adicional sobre dónde pueden localizar la respuesta. La retroalimentación contingente al tópico pone a disposición gran cantidad de información, pero al estudiante le corresponde encontrar la respuesta correcta dentro del material. En este caso, la especificidad de la retroalimentación es baja.

Contingente a la respuesta. También conocida como retroalimentación extra-instruccional, facilita tanto la retroalimentación de verificación como la de elaboración específica a la pregunta. Además de proveer la respuesta correcta, da a conocer por qué la respuesta es correcta o incorrecta. Aquí la especificidad de la retroalimentación es alta.

Retroalimentación relacionada con el error. Provee verificación y se enfoca en el error específico. Se basa en librerías de errores o en reglas que se generan para identificar y corregir una gran cantidad de errores de los estudiantes. Aunque no facilita la respuesta correcta, ayuda a identificar los errores ocurridos en los procedimientos para que se haga una autocorrección.

Aislamiento de atributos. Provee verificación y resalta los atributos centrales del concepto principal. Se concentra en los componentes principales del concepto a fin de mejorar la comprensión del fenómeno.

2.4 Tiempo para la facilitación de la retroalimentación

Existen dos formas para facilitar la retroalimentación según el tiempo transcurrido en que se da después de haberse producido la respuesta: la retroalimentación inmediata y la retroalimentación diferida.

Retroalimentación inmediata. Se facilita justo después de haberse producido la respuesta. Con los medios tecnológicos modernos, se puede dar retroalimentación después de cada pregunta.

Retroalimentación diferida. Se deja transcurrir un tiempo y después se dan a conocer los resultados, los cuales se acompañan con alguna de las formas de retroalimentación descritas. Tradicionalmente, los estudiantes hacen los exámenes y, un tiempo después, el profesor trae las evaluaciones calificadas y corregidas, con su respectiva retroalimentación.

Analicemos ahora algunas investigaciones sobre el tema de la retroalimentación y el impacto que han tenido en el aprendizaje a partir de los anteriores conceptos.

2.5 Investigaciones sobre la retroalimentación

Otero (2010a) trabajó los niveles de especificidad y los efectos que estos tienen en la competencia de resolución de problemas, en una investigación de tipo cuasi experimental, unifactorial, univariable, con un diseño pretest-postest con dos grupos. Se hizo el pretest para establecer el conocimiento inicial de los grupos, seguida de la aplicación de dos objetos de aprendizaje —uno para cada grupo— y, finalmente, el postest para verificar la influencia de los objetos aplicados.

La variable independiente la constituyeron los objetos de aprendizaje, uno caracterizado por facilitar el aprendizaje con retroalimentación con especificidad alta y otro con especificidad baja. Como variable dependiente se tomó el desempeño de los participantes. Otras variables que se contemplaron en el experimento fueron la cantidad de ejercicios efectuados por participantes, el nivel de complejidad alcanzado, la cantidad de ayudas consultadas y el puntaje en el curso regular de Álgebra Lineal.

Los objetos de aprendizaje utilizaron problemas de álgebra lineal que fueron clasificados según su complejidad en cuatro niveles, cada uno de ellos constó, a su vez, de dos etapas. En general, la etapa 1 involucró la definición y la descripción de las variables, y la etapa 2 la modelación matemática. En el nivel 1 se trataron problemas cuya solución abarcaba una ecuación con relación lineal entre las variables; en el nivel 2 se obtuvo dos ecuaciones para

luego despejar una variable en común y dejar una sola ecuación. En el nivel 3 las ecuaciones se formaron con matrices y en el nivel 4, se obtuvieron dos ecuaciones matriciales, de la cual se despejaba una, para dejar el modelo con una sola ecuación.

A los estudiantes que emplearon el objeto de aprendizaje con niveles altos de especificidad se les proveía tanto retroalimentación de conocimiento de resultados como la de conocimiento de la respuesta correcta (que son formas de retroalimentación de verificación) y, además, se les proporcionaba retroalimentación elaborada, mediante la cual se informaba si la respuesta era correcta o no, así como la razón de ello; además se les permitían múltiples intentos para encontrar la solución a la pregunta.

Con los estudiantes que trabajaron en el grupo del objeto de aprendizaje con retroalimentación de especificidad baja se utilizó retroalimentación de conocimiento de resultados y de retroalimentación elaborada. Pero la retroalimentación elaborada consistió en retroalimentación contingente al tópico, relacionada con el error y el aislamiento de atributos, donde el estudiante tenía como herramienta para corregir sus errores el análisis para llegar a la respuesta correcta, ello a través del procesamiento de información. También se utilizó retroalimentación de múltiples intentos para cada pregunta.

Los objetos de aprendizaje orientaron el desarrollo de la competencia de resolución de problemas tomando como base el planteamiento de estos, los cuales se presentaron en forma de historias para ser resueltos por ambos grupos. El objeto de aprendizaje de especificidad baja condujo al estudiante a construir la competencia de resolución de problemas, en los temas correspondientes, a través de la realización de actividades de procesamiento de información. Fue requisito resolver correctamente el paso actual para que se presentara otro o para seguir con el siguiente problema. En el proceso de obtención de principios, los estudiantes desplegaron actividades de comprensión y utilizaron esquemas previos que, combinados con el proceso de metacognición, conducen a la construcción del conocimiento.

El uso del objeto de aprendizaje de alta especificidad de retroalimentación se sustentó también en los esquemas previos formados en el estudiante para desarrollar la competencia de resolución de problemas. Cuando este cometía un error en el paso que estaba enfrentando, se le permitía a conocer la solución para que pudiera aplicar este conocimiento en los problemas venideros. En los dos casos, el estudiante debía recurrir a sus competencias interpretativas y propositivas, de lectura comprensiva y estrategias de resolución de problemas de reglas de tres que le permitiera apropiarse del conocimiento que se iba exponiendo y que le ayudaran a entender los ejemplos que se explicaban para profundizar en el dominio de los conceptos.

La prueba final puso al descubierto que se logró construir esquemas, al encontrarse diferencia significativa entre el desempeño mostrado entre las pruebas de entrada y salida. De acuerdo con los planteamientos teóricos, lo que pudo causar la diferencia entre los dos momentos fueron los esquemas construidos por la influencia de los objetos de aprendizaje. Esto indica que el uso de objetos de aprendizaje es una buena alternativa para brindar experiencias en la resolución de problemas. El hecho de que un estudiante pudiera resolver un problema después de habersele suministrado retroalimentación confirma los hallazgos de Sassenrath y Garverick (1965), quienes afirmaron que es mejor suministrar algún tipo de retroalimentación, que dejar de entregarla por completo.

Comparando los niveles de complejidad alcanzados, Otero (2010a) encontró que los integrantes del grupo de especificidad alta lograron niveles más altos en la complejidad, por lo que tuvieron un desempeño significativamente mejor que los miembros del otro grupo. Los estudiantes que conformaron el grupo de especificidad baja se vieron forzados a experimentar diferentes acciones para descubrir cuál de las muchas posibilidades de respuesta era la correcta. Esta experimentación condujo a una serie de acciones no exitosas que se tradujeron en la disminución del desempeño. En el nivel de más alta especificidad la guía proveída minimizaba la frecuencia de los errores y las

acciones no exitosas, lo cual demuestra el impacto que tiene la especificidad alta en el desempeño inmediato.

Se pudo observar también una relación positiva entre el nivel alcanzado y el resultado en la prueba: a más alto nivel de complejidad, mejor era el desempeño. De igual manera, se comprobaron las relaciones positivas significativas entre el puntaje y la cantidad de ejercicios efectuados, y entre el puntaje y el desempeño en el curso regular.

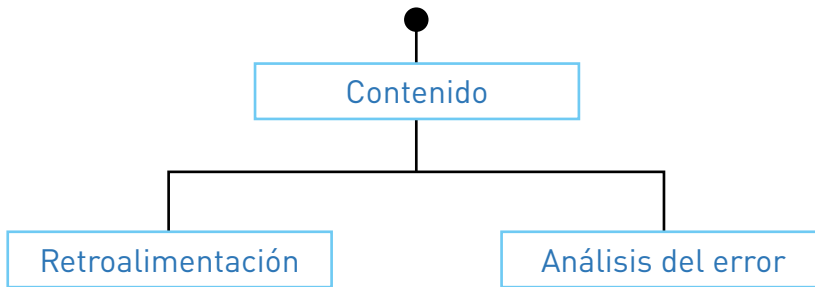
Por su parte, no hubo relación entre la cantidad de ayudas utilizadas y el puntaje obtenido. Esto se explica ya que después de que se entiende el principio expuesto en la ayuda, lo que viene a continuación es tener la competencia para aplicar esa información en el contexto específico. El conocimiento del dominio, aunque pueda pronosticar en parte la competencia de resolución de problemas, necesita estar acompañado de los esquemas de aplicación; por eso solucionar muchos problemas muestra una relación positiva con el desempeño, pues el hecho de estudiar muchas veces una ayuda no da la experiencia para resolver los problemas.

Lemley (2005), entre otros autores, manifiestan que si se comparan escenarios de aplicación de las formas de retroalimentación con escenarios donde no se la recibe, la retroalimentación de conocimiento de resultado es la menos efectiva, seguida de la de conocimiento de la respuesta correcta; mientras que cuando estas formas se acompañan de retroalimentación elaborada, aumenta la efectividad (Bangert-Drowns, Kulik, Kulik, & Morgan, 1991; Clariana, Ross, & Morrison, 1991; Kulhavy, 1977; Lee-Sammons & Wollen, 1989; Pridemore & Klein, 1991, citados en Lemley). Aunque la forma de retroalimentación elaborada requiere mayor esfuerzo en el diseño, vale la pena su empleo porque facilita al estudiante un ambiente más rico de experiencias.

Narciss (2001), por su parte, hace un trabajo con la finalidad de ofrecer lineamientos para el diseño de formas de retroalimentación y examinar el impacto que estas tienen en el aprendizaje y la motivación, en un experimento basado en el aprendizaje por computador.

Según este autor, desde la perspectiva del aprendizaje, la retroalimentación elaborada se considera una fuente de información necesaria, especialmente si el estudiante encuentra obstáculos o procede incorrectamente. Para convertir mensajes de retroalimentación elaborada en aprendizajes complejos es esencial analizar el conocimiento y las habilidades que tienen que ver con la tarea de aprendizaje, describir los errores típicos y los pasos incorrectos al solucionar una tarea y entender las concepciones erróneas o las estrategias ineficientes que se pueden atribuir a los errores descritos. En la figura 2 se hace un resumen de los elementos que se debe tener en cuenta

FIGURA 2. *Análisis de la tarea.*



Para analizar el conocimiento y las habilidades que tienen que ver con la tarea de aprendizaje propone la siguiente serie de pasos:

1. Seleccionar y especificar objetivos para el dominio dado de conocimientos en términos de eventos de aprendizajes concretos.
2. Identificar y clasificar los elementos específicos del contenido y las unidades de conocimiento relacionadas con el elemento del contenido (por ejemplo, hechos, conceptos, eventos, reglas, principios, modelos teorías).
3. Seleccionar y especificar las operaciones cognitivas que se deben asociar con la unidad de conocimiento (por ejemplo, recordar, transformar, clasificar, argüir, inferir).

4. Construir una matriz en la que un eje que contenga las unidades de conocimiento y el otro contenga las operaciones cognitivas.
5. Seleccionar y especificar una tarea de aprendizaje para cada celda de la matriz (por ejemplo, recordar hechos, identificar o definir conceptos, formular reglas, clasificar evento, argüir con bases en los modelos, inferir de las teorías).
6. Identificar y clasificar factores posibles, concepciones erróneas o fuentes que puedan causar un error por estrategias eficientes para cada tipo de tarea (por ejemplo, ignorar instancias negativas en la formación de conceptos).
7. Seleccionar y especificar la información que puede ser facilitada como retroalimentación si el error identificado ocurre o se encuentran dificultades (por ejemplo, señales para recuperar datos, analogías sugerencias sobre posibles fuentes de información y sugerencias sobre estrategias de éxitos).

Para el examen y el impacto que estas formas tienen en el aprendizaje y la motivación, Narciss se propuso contestar las siguientes preguntas: ¿el valor informativo de los mensajes de retroalimentación realzan el desempeño?; ¿el valor informativo de los mensajes de retroalimentación tienen impacto en la motivación del aprendiz?; ¿las características individuales de motivación interactúan con variables situacionales (el valor informacional de la retroalimentación)?

Para evaluar los principios de diseño, Narciss tomó una tarea de aprendizaje propuesta por Bruner, Goodnow y Austin (1956), en la cual se conocen los errores y las estrategias típicas. Este material consta de diferentes figuras (que aparecen en unas cartas) para formar conceptos. De acuerdo con este autor, los objetivos de aprendizaje de una tarea de formación de conceptos son la adquisición y la transferencia de estrategias eficientes de identificación de conceptos. El dominio de la tarea de formación de conceptos implica la identificación de características relevantes y no relevantes al comparar las

características de las cartas; además, implica una generación de hipótesis, que tiene que ver con las reglas que relacionan las características.

En este trabajo, las condiciones de retroalimentación fueron: *de baja especificidad*, que consistió solamente en el conocimiento de resultados; a los sujetos se les informaba si la hipótesis sobre el concepto estaba bien o equivocada; *de especificidad de la información media*, que consistió en retroalimentación de resultados y localización del error; a los sujetos se les hacía saber si la hipótesis estaba bien o mal y, en este último caso, se les informaba cuál de las tarjetas no era compatible con el concepto supuesto; *de especificidad de retroalimentación alta*, en la cual se brindó conocimiento acerca de cómo proceder; además del conocimiento de la respuesta y la localización del error, recibieron sugerencias sobre la atención que se debía prestar a las cartas con instancias negativas de los conceptos dados. Los participantes se asignaron aleatoriamente a cada una de las tres condiciones.

Por otra parte, se midió la autoeficacia pidiéndole a los participantes calificar tres elementos que tenían que ver con su confianza al resolver tareas. La variable dependiente la constituyó el desempeño en la formación de conceptos y la persistencia. Para medir el desempeño se tuvo en cuenta la cantidad de cartas requeridas para identificar un concepto y la cantidad de hipótesis falsas para este concepto. Para medir la persistencia, a todos los conceptos resueltos se les calificaba con 3; si el participante cancelaba un concepto y resolvía los otros, obtenía un 2, y si el participante cancelaba el concepto y renunciaba a su identificación, se le otorgaba un punto para este concepto. La suma de los puntos representó una medida de la persistencia. Los datos fueron analizados usando los procedimientos generales de los modelos lineales con el sistema SPSS™ 10.0 (IBM Corporation). El nivel de significancia se fijó en 0.05.

El resultado del experimento mostró que el valor informacional de la retroalimentación (la especificidad) tiene impacto sobre el aprendizaje y la motivación: un mejor desempeño se relaciona con una especificidad más alta, especialmente en las tareas más difíciles. Los efectos motivacionales de la retroalimentación informativa son moderados por la autoeficacia en la tarea.

La alta persistencia se relaciona con la especificidad de retroalimentación más alta. En general, manifiesta Narciss (2001), vale la pena desarrollar y facilitar formas de retroalimentación bien fundamentadas en situaciones de aprendizaje basadas en el computador. Futuras investigaciones sobre este tema son importantes en relación con la pregunta sobre cómo promover aprendizaje autorregulado con las tecnologías modernas de información.

Por otro lado, Goodman y Wood (2004) trabajaron en la especificidad de la retroalimentación, para lo cual estudiaron los efectos de la especificidad en las oportunidades de aprendizaje y directamente en el aprendizaje; en cada caso se establecieron condiciones de ejecución buena y condiciones pobres de la tarea. En las condiciones buenas, el ambiente es apropiado para realizar la tarea y no se presentan dificultades (adicionales a las normales para esa tarea). En las condiciones pobres, por el contrario, el ambiente o las herramientas no son las adecuadas para hacer la tarea y en su ejecución se presentan dificultades no previstas.

Cada condición de tarea se trabajó con tres niveles de especificidad: bajo, medio y alto. Las oportunidades de aprendizaje se midieron en un tiempo llamado periodo de práctica y el aprendizaje se evaluó en un tiempo dos, llamado periodo de examen. En este último tiempo se facilitó poca guía y se evaluaron las destrezas y el conocimiento necesarios para hacer las tareas de manera independiente.

Goodman y Wood propusieron cuatro hipótesis para su trabajo. La hipótesis I expresa que la especificidad de la retroalimentación afecta las oportunidades de aprendizaje, de tal manera que entre más alta la especificidad, más alto el porcentaje de instancias que se responden bien en las condiciones buenas de la tarea, y más bajo el porcentaje de las instancias que se responden incorrectamente en las condiciones pobres de la tarea.

La hipótesis II señala que las oportunidades de aprendizaje están relacionadas con los eventos de aprendizaje. De este modo, el porcentaje de las instancias a las cuales se responde bien durante la práctica corresponden

positivamente con el aprendizaje de reglas para responder bien en condiciones buenas de la tarea, y negativamente con el aprendizaje de reglas para responder bien en condiciones pobres de la tarea.

La hipótesis III afirma que la especificidad de la retroalimentación está relacionada positivamente con el aprendizaje de reglas para responder bien a las condiciones buenas de la tarea, y relacionada negativamente con el aprendizaje de reglas para responder en las condiciones pobres de la tarea.

La hipótesis IV contempla que las relaciones entre la especificidad de la retroalimentación y el aprendizaje de reglas para responder correctamente en condiciones buenas y pobres de la tarea están mediadas por las oportunidades de aprendizaje, específicamente:

- a. La especificidad de la retroalimentación está relacionada estrechamente con el porcentaje de instancias para responder bien en las condiciones buenas de la tarea durante la práctica, y subsecuentemente con el aprendizaje de respuestas correctas en dichas condiciones.
- b. La especificidad de la retroalimentación está ligada negativamente con el porcentaje de instancias para responder en las condiciones pobres de la tarea durante la práctica, y subsecuentemente al aprendizaje de respuestas correctas en dichas condiciones.

Goodman y Wood manipularon la especificidad de la retroalimentación durante una fase de práctica que no era familiar para los participantes, los cuales completaron 19 tareas de ensayos en la práctica en cada uno de los tres niveles de especificidad, que variaron de retroalimentación de resultados solamente en la especificidad baja, a retroalimentación de resultado más información sobre el proceso en todas las decisiones, en la especificidad alta. Luego se examinaron los efectos de la retroalimentación facilitada durante la práctica en el desempeño de dos tareas de transferencia (14 ensayos), que se realizaron dos días más tarde, sin brindar retroalimentación específica.

El aprendizaje se midió como la exactitud de las respuestas a las diferentes condiciones de la tarea que se enfrentaron durante la tarea de transferencia. Estas medidas permitieron examinar la hipótesis dada sobre la existencia de diferencias en el aprendizaje de reglas para respuestas correctas en condiciones buenas contra condiciones pobres de la tarea, a través de las condiciones de especificidad de la retroalimentación.

Los resultados mostraron que la hipótesis I se cumplió: la especificidad de la retroalimentación afectó las oportunidades de aprendizaje durante la práctica. Entre más alta la especificidad, mayor el porcentaje de instancias en las que los participantes respondieron a las condiciones buenas de la tarea y más bajo el porcentaje de instancias en que respondieron a las condiciones pobres de la tarea durante la práctica.

La hipótesis II se cumplió parcialmente: el porcentaje de instancias para responder a las condiciones buenas de la tarea contra las condiciones pobres durante la práctica se relacionaron positivamente con el aprendizaje de reglas para responder bien a las condiciones buenas de la tarea durante la práctica. Sin embargo, la relación negativa entre el porcentaje de instancias para aprender reglas para contestar bien en las condiciones pobres de la tarea no se pudo comprobar.

La hipótesis III se verificó: la especificidad de la retroalimentación afecta el aprendizaje de reglas para responder a las condiciones buenas y pobres de la tarea, como se evidenció en la interacción entre la especificidad de la retroalimentación y el tipo de reglas. El grupo que conformó la condición alta de especificidad aprendió lo que tenía que hacer en las condiciones buenas de la tarea; lo mismo sucedió con los de especificidad moderada y especificidad baja. La diferencia entre la condición moderada y baja no fue significativa: quienes estaban en las condiciones bajas de la tarea aprendieron mejor qué hacer en las condiciones pobres de la tarea.

La hipótesis IV se cumplió para el aprendizaje de reglas para responder a las condiciones buenas, pero no para las condiciones pobres: la retroalimentación más específica conduce a un porcentaje más grande a responder a las

buenas condiciones durante la práctica, lo cual permite aprender más reglas asociadas con dichas condiciones.

Asimismo, aunque la especificidad de la retroalimentación se asocia negativamente con el aprendizaje de reglas para responder a las condiciones pobres de la tarea, las oportunidades de aprendizaje no afectaron, durante la práctica, el aprendizaje de reglas para responder bien a las condiciones pobres. Por tanto, no se encontró el soporte para establecer la relación especificidad de la retroalimentación-condiciones pobres de la tarea.

El resultado del experimento demostró que al incrementar la especificidad de la retroalimentación durante la práctica, se afecta la oportunidad de aprendizaje durante dicha práctica, y que los efectos de la especificidad de la retroalimentación en el aprendizaje dependen de los aspectos de la tarea que se va a aprender.

El incremento de la especificidad beneficia el aprendizaje de respuestas en condiciones buenas y perjudica su aprendizaje en las condiciones pobres de la tarea. En capítulos posteriores veremos la relación que tiene esto con la transferencia de conocimiento de un contexto a otro. Si los contextos tienen características superficiales parecidas (condiciones buenas de la tarea) existe una gran posibilidad de que ocurra la transferencia.

Para la discusión acerca de la retroalimentación inmediata y diferida se trae a colación la investigación de Lemley (2005), quien trabajó sobre el impacto de la retroalimentación inmediata y diferida en el desempeño de estudiantes de nivel secundario en la Brigham Young University. Los estudiantes se dividieron en dos grupos: el primero estaba integrado por los estudiantes inscritos que recibían y entregan los trabajos vía correo; el segundo por estudiantes que trabajan con una versión web, quienes recibían y enviaban su asignación digitalmente. El primer grupo se sometió a la retroalimentación diferida y el segundo la retroalimentación inmediata.

Las dos hipótesis que guiaron su trabajo fueron: 1) los estudiantes que recibían retroalimentación inmediata obtendrán mejores promedios que los que

están en la condición de retroalimentación diferida, y 2) el tiempo transcurrido desde la asignación del primer curso hasta su terminación será significativamente menor para los estudiantes que reciban retroalimentación inmediata, en comparación con los que reciben retroalimentación diferida.

La nota final, tomada como un indicador del desempeño del estudiante en un curso, fue la variable dependiente. Los resultados mostraron que los estudiantes que recibieron retroalimentación inmediata obtuvieron mejores notas en el examen final que los que estaban en el grupo tradicional. En cuanto a la segunda hipótesis, se pudo verificar que a los estudiantes que estaban en el grupo de retroalimentación diferida les llevó menor tiempo completar el curso que los que estaban en la otra condición.

Para Mason y Bruning (2001) históricamente el tiempo de entrega de la retroalimentación ha tenido resultados mezclados. Manifiestan que algunos investigadores argumentan que la retroalimentación inmediata es necesaria para corregir los errores de los estudiantes antes de que la información se codifique en la memoria; según ellos, otros indican que la retroalimentación diferida reduce la interferencia proactiva, lo cual facilita que se olvide el error inicial y se corrija la información que se va a codificar sin ninguna interferencia.

Kulik y Kulik (1988, citados en Mason & Bruning, 2001) mostraron que la retroalimentación inmediata es más efectiva que la diferida en estudios aplicados y aprendizajes de lista, pero es menos efectiva para la adquisición de contenidos de pruebas. Otros investigadores (Jonassen & Hannum, 1987, citados en Mason & Bruning) manifiestan que la retroalimentación inmediata puede ser más efectiva para la toma de decisiones, para tareas de información nuevas y para tareas de bajo nivel basadas en conocimiento.

Para tareas de alto nivel, como conceptos abstractos y habilidades de comprensión y aplicación, la retroalimentación diferida ha dado mejores resultados (Gaynor, 1981; Jonassen & Hannum, 1987, citados en Mason & Bruning, 2001). Roper (1977, citado en Mason & Bruning), señala que la adquisición de conceptos se facilita a través de la retroalimentación inmediata, mientras que la

retención a largo plazo se realiza con la retroalimentación diferida. Este autor recomienda la combinación de la retroalimentación inmediata de verificación con la retroalimentación diferida elaborada.

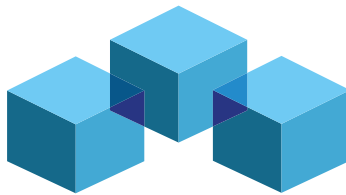
Según Kulhavy (1977, citado en Mory, 2004), la retroalimentación diferida incrementa significativamente la retención de los estudiantes, como se demostró en las pruebas finales de su investigación. Este fenómeno se conoce como *efecto de retención diferida* y se encontró principalmente en estudios que tienen que ver con exámenes de preguntas múltiples; esto se explica por la transferencia proactiva que se produce en la retroalimentación inmediata debido al error en las respuestas iniciales. En otras investigaciones (Peck *et al.*, 1985; Phiem *et al.*, 1976, citados en Mory, 2004) se encontró que la retroalimentación diferida no produce una diferencia significativa.

De este capítulo se puede concluir que la retroalimentación es parte importante del aprendizaje y le permite a los estudiantes conocer su desempeño y los errores cometidos. La retroalimentación de alta especificidad produce buenos resultados para problemas que tienen características superficiales parecidas, pero no prepara al estudiante para enfrentar los problemas en condiciones pobres de la tarea. La forma de retroalimentación que se utilice depende de los tipos de problemas que se vayan a solucionar (*véase* capítulo 4).

Para diseñar mensajes basados en retroalimentación elaborada, es decir, para aprendizajes complejos, es esencial analizar el conocimiento y las habilidades que tienen que ver con la tarea de aprendizaje; describir los errores típicos y los pasos incorrectos al solucionar una tarea, y entender las concepciones erróneas o las estrategias ineficientes que se pueden atribuir a los errores descritos.

Finalmente, Otero (2010a) demostró que la retroalimentación de especificidad alta ejerce una influencia positiva sobre el desempeño inmediato. En cuanto a la especificidad baja de la retroalimentación, se encontró que para que sea efectiva se necesita más tiempo de aplicación, porque los estudiantes se ven forzados a probar diferentes acciones para llegar a las soluciones de

los problemas; además, se pueden enfrentar a callejones sin salida donde se requieran orientaciones específicas que los ayuden a superar las dificultades. Otero también encontró que existen variables como cantidad de ejercicios efectuados, el nivel de complejidad de los problemas y la formación previa, que pueden influenciar la competencia de resolución de problemas.



Capítulo 3.

El control de la retroalimentación

De acuerdo con Pridemore y Klein (1991), aunque se han efectuado muchas investigaciones sobre el control del aprendizaje por parte del estudiante, son pocas las que se han conducido acerca del control de la retroalimentación. Afirman estos autores que los computadores permiten controlar la cantidad de retroalimentación en la instrucción, pero que no es claro si otorgarlo es beneficioso para el aprendizaje.

Estos mismos autores, además, reportan trabajos de otros investigadores acerca de este tema: Newkirk (1973) determinó que el control influencia positivamente la retención de información y el interés de quien aprende. Por otra parte, Kinzie, Sullivan y Berdel (1988), en relación a las opciones de repaso, y Ross y Morrison (1989), en lo relativo a las propiedades contextuales en lecciones impartidas por *instrucciones apoyadas en el computador* (CAI, por sus siglas en inglés), indicaron que estas lecciones incrementan el desempeño (según resultados de un postest). Otro de los autores citados es Hansen (1974), quien encontró que el control del estudiante sobre la retroalimentación en las lecciones a través de las CAI disminuye su ansiedad acerca del aprendizaje, mientras otros como Schloss, Wisniewski y Cartwright (1988) reportaron que la retroalimentación y el control del estudiante en las CAI aumentan la actitud y el desempeño.

De acuerdo con Corbalan, Kester y Van Merriënboer (2009), las investigaciones se han centrado más en el control por parte del estudiante de los temas de aprendizaje. El control óptimo del aprendizaje le permite a este hacer selecciones acordes con su estado actual de conocimiento, intereses y preferencias, lo cual se supone actúa positivamente en el aprendizaje y la motivación (Flowerday & Schraw, 2000, citados en Corbalan, Kester, & Van Merriënboer, 2009). Siguiendo Corbalan, Kester, & Van Merriënboer (2000), hay estudios que afirman efectos positivos y negativos del efecto que tiene el control del estudiante sobre su aprendizaje, pues parece ser que la efectividad depende de los elementos que controlan el paso, las características de presentación o de la tarea, así como si el estudiante reconoce el control que se le otorga.

3.1 Influencia sobre el control de la retroalimentación en el aprendizaje

Pridemore y Klein (1991) estudiaron el efecto que tiene en el estudiante el control de la retroalimentación en las instrucciones asistidas por computador. La variable independiente fue el tipo de control (ya fuera del programa o del estudiante) y el nivel de retroalimentación (de verificación o elaboración). Las variables dependientes fueron el desempeño mostrado en un postest, la actitud hacia el programa y el tiempo para estudiar la retroalimentación.

El material utilizado en este estudio fueron cuatro lecciones presentadas por programas de instrucciones apoyadas en el computador, un postest y un cuestionario de actitud. Las lecciones se basaron en el concepto de fiabilidad y validez, cuyas diferencias se dieron por el tipo de control y el nivel de retroalimentación. Bajo el control del programa, el computador administró automáticamente alguna de las condiciones de retroalimentación. En la condición de verificación al estudiante se le indicaba si su respuesta era correcta o no; en la condición de elaboración se le informaba si la respuesta era correcta y se le daba una explicación somera; además, se le preguntaba si deseaba usar la retroalimentación o no.

Los análisis de los resultados indicaron que el nivel de retroalimentación tiene un efecto significativo en el desempeño. Los estudiantes que recibieron retroalimentación elaborada se desarrollaron mejor que aquellos que recibieron retroalimentación de verificación, sin importar el tipo de control; es más, la diferencia del tipo de control no fue significativa. En cuanto a la actitud, los estudiantes que conformaron el grupo de verificación mostraron un mayor deseo de que se les brindara más retroalimentación de la que recibieron.

En cuanto al tiempo de estudio, los datos revelaron un efecto significativo según el nivel de retroalimentación. Los estudiantes del grupo de elaboración gastaron más tiempo que los del grupo de verificación. El tipo de control no tuvo un efecto significativo en el tiempo de estudio y, en general, los estudiantes que recibieron retroalimentación elaborada durante la instrucción tuvieron mejor desempeño que los del grupo de verificación. El tipo de control no tuvo incidencia en los resultados.

Otero (2012) realizó una investigación con el objetivo de medir el efecto del control sobre la retroalimentación en el aprendizaje de conceptos y su aplicación a la resolución de problemas con una variable dependiente: los efectos de cada uno de los valores de la variable independiente sobre el aprendizaje de conceptos y su aplicación a la resolución de problemas sobre la línea recta, que se midió a través del desempeño mostrado en un examen de salida. Por su parte, la variable independiente la constituyeron objetos de aprendizaje con dos valores: la retroalimentación con control del programa y la retroalimentación con control del estudiante. El primero brindaba un ambiente interactivo, con retroalimentación elaborada, donde se adaptaba al error del estudiante y se establecía un diálogo para hacerle entender su error y la diferencia de este con la respuesta correcta. El segundo dejaba a potestad del estudiante si leía la retroalimentación o no.

La investigación pretendió determinar los efectos de cada uno de los niveles de la variable independiente sobre el aprendizaje de conceptos y su aplicación a la resolución de problemas sobre la línea recta, la cual se constituyó en la variable dependiente.

A los estudiantes participantes de los dos grupos se les suministró una prueba de entrada para constatar su conocimiento sobre la línea recta y su aplicación a la resolución de problemas, lo cual permitió establecer la equivalencia inicial de los grupos. Al final del experimento se aplicó un posttest para establecer el desempeño alcanzado por los grupos.

De acuerdo con los contrastes de hipótesis efectuados con el paquete estadístico SPSS™ 15 (IBM Corporation) para Windows, tanto los estudiantes que trabajaron con el control de retroalimentación por parte del programa, como los que tenían control sobre la retroalimentación tuvieron un desempeño significativo (ello al hacer un test de significación de diferencia de medias entre las pruebas de pretest y posttest).

Al realizar el contraste de hipótesis para probar la diferencia de medias del desempeño entre los dos grupos se encontró que no hubo diferencia significativa entre estos. Es decir, los grupos mostraron el mismo desempeño: el tipo de control no tuvo una influencia significativa en este. Dichos resultados concuerdan con los de otros autores (Pridemore & Klein, 1991). Estos investigadores encontraron que los sujetos que reciben retroalimentación elaborada durante la instrucción, tienen mejor desempeño que aquellos que reciben retroalimentación de verificación. El tipo de control no tiene una influencia significativa en el desempeño.

3.2 Influencia sobre el control del material en el aprendizaje

En cuanto a trabajos que han estudiado la influencia del control de material instruccional sobre el aprendizaje de los estudiantes, podemos analizar la investigación de Corbalan, Kester y Van Merriënboer (2009), quienes condujeron un experimento para estudiar el efecto de la retroalimentación conocimiento de la respuesta correcta (KCR) y el control sobre la selección de tareas de aprendizaje. Según estos autores, la retroalimentación KCR ayuda al estudiante a tener una eficiencia más alta y mayor motivación al enfrentar las tareas.

Las características estructurales son relevantes para la solución de problemas, pero no son percibidas por estudiantes novatos, de manera que se espera que la retroalimentación de conocimiento de la respuesta correcta los ayude a reconocer las características estructurales y a que puedan escoger las tareas de aprendizaje de manera acertada.

Según Corbalan, Kester y Merriënboer las características estructurales se refieren a aspectos de la tarea necesarios para llegar a una solución; por ejemplo, los procedimientos en matemática. Pero los estudiantes novatos no distinguen tareas que tienen diferentes características estructurales, lo que puede tener un impacto negativo sobre el aprendizaje y la motivación.

Si una dificultad para lograr la transferencia es reconocer la similitud estructural (Bassok, 1990, citado en Corbalan, Kester, & Merriënboer, 2009), la inducción para hacer las características estructurales de la tarea explícitas durante el entrenamiento puede promover la transferencia.

La motivación se presenta cuando la forma de retroalimentación de conocimiento de la respuesta correcta ayuda al estudiante a concentrarse en las características estructurales de la tarea o en los pasos para llegar a una solución, promoviendo así la percepción de relevancia del material instruccional, puesto que le permite al estudiante ver la conexión entre lo que necesita para aprender y las oportunidades de aprendizaje que se le presentan. La relevancia incrementa el uso de estrategias cognitivas que mejoran el aprendizaje (Corbalan, Kester, & Merriënboer, 2009).

Dicha eficiencia se mide en relación con el desempeño. Cuando el desempeño es más alto de lo esperado, tomando como base el esfuerzo mental percibido, la condición instruccional es más eficiente; por el contrario, cuando el desempeño es más bajo de lo esperado, la condición instruccional se cataloga como menos eficiente. El esfuerzo mental refleja la cantidad de capacidad cognitiva utilizada en la solución y se empleó como índice para la carga cognitiva. Representa el esfuerzo que se requiere para resolver un problema y fue medido con la escala de Likert: 1 para un esfuerzo muy pequeño y 7 para uno muy alto.

Los autores mencionados propusieron cuatro condiciones de investigación: control del aprendizaje por parte del programa con y sin retroalimentación de conocimiento de la respuesta correcta (KCR), y control del aprendizaje por parte del estudiante con y sin retroalimentación KCR. Los resultados evidenciaron que a quienes se les facilitó la retroalimentación KCR mostraron más eficiencia y mayor motivación que aquellos a quienes no se les brindó ningún tipo de retroalimentación, sin importar la condición de control. Comparando las condiciones de control del material instruccional se pudo verificar que no hubo diferencia significativa entre los grupos.

En general, lo que muestran las investigaciones es que el control de la retroalimentación o el control del material de estudio producen un mismo resultado, sin importar si es el sistema o el estudiante quien lo aplica. De manera semejante a Pridemore y Klein (1995), Otero (2014a) verificó que, sin importar que el control de la retroalimentación lo tenga el sistema o los estudiantes, estos últimos obtienen un desempeño significativo. La diferencia se puede dar por el tipo de retroalimentación proveída; por su parte, Pridemore y Klein demostraron que la retroalimentación elaborada produce mejores resultados.



Capítulo 4.

Resolución de problemas

«Un problema constituye (...) una situación incierta que provoca en quien la padece una conducta —*resolución del problema*— tendiente a hallar la solución —*resultado esperado*—» (Perales, *et al.*, 2000, p. 11, las cursivas están en el texto original). En esta definición se puede observar que existen dos elementos importantes: el problema en sí y el sujeto que está expuesto a su influjo. Según Jonassen (2006), estos son los aspectos a considerar en la creación de ambientes de aprendizaje.

De acuerdo con Sweller (1988) y Gick y Holyoak (1983), las competencias para resolver problemas dependen de los esquemas construidos en la resolución de problemas particulares de un determinado tipo, ya que la representación de un nuevo problema consiste en aplicar un esquema construido al que se está tratando, lo que le facilita el reconocimiento de diferentes estados del problema que conducen a la solución.

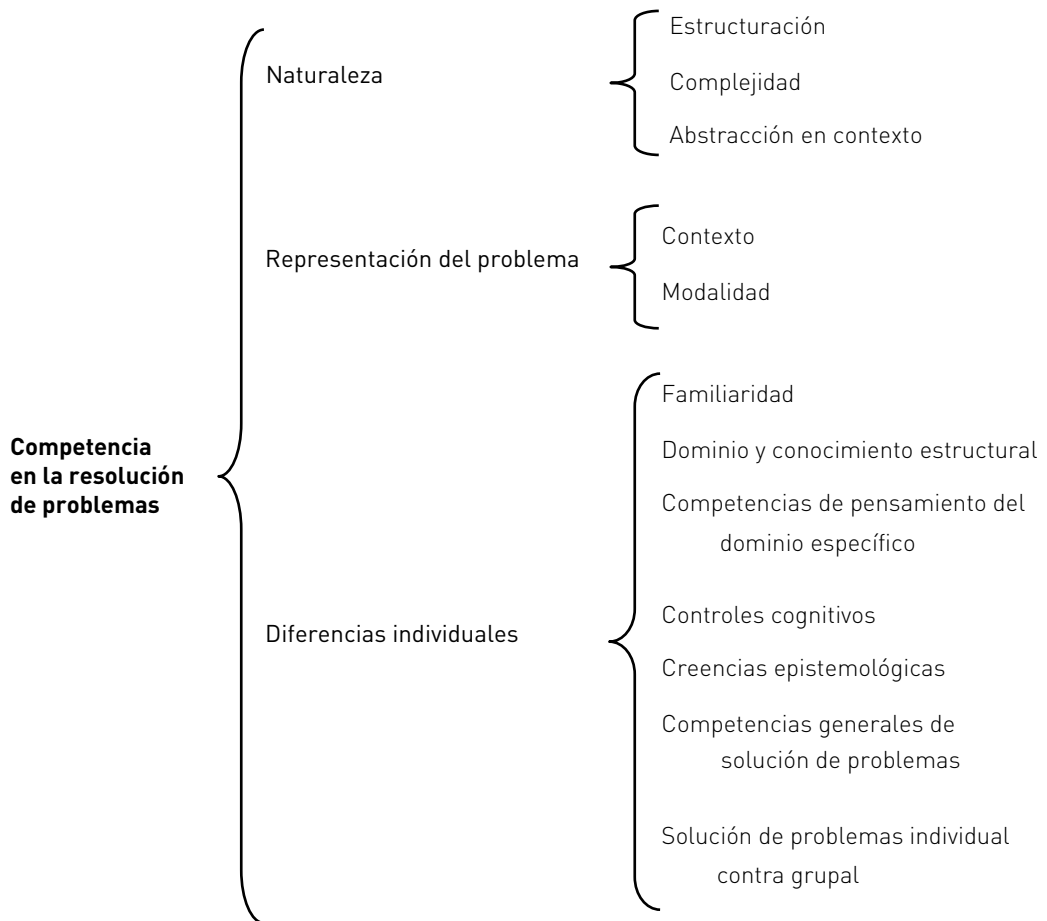
Sweller (1988) considera que los expertos poseen mayor facilidad para resolver problemas porque, al transitar por los diferentes estadios de un problema, evocan soluciones efectuadas previamente. Los principiantes, que no poseen estos esquemas, no pueden reconocer ningún tipo de problema, por lo que deben recurrir a estrategias generales de solución de problemas, como

el procesamiento de información mediante ensayo y error (la cual es una estrategia débil).

4.1 Factores que inciden en la resolución de problemas

Según Jonassen (2006) la competencia de resolución de problemas es un resultado de la naturaleza del problema, su representación y el conjunto de diferencias individuales de quienes lo resuelven (figura 3).

FIGURA 3. Competencia en la resolución de problemas



En el estudio de la competencia en la resolución de problemas, respecto a la naturaleza del problema, Jonassen (2006) señala que es necesario distinguir en él su estructuración, complejidad y abstracción en contexto.

4.2 Estructuración

De acuerdo con Jonassen (2006), los problemas pueden presentarse como bien estructurados o mal estructurados. Los problemas bien estructurados constan de un estado inicial bien definido, una meta conocida y un conjunto de operadores lógicos. Estos se caracterizan por poseer todos los elementos para su solución, realizándose al aplicar un número dado de reglas definidas concernientes a un dominio dado de conocimiento y tienen respuestas correctas y convergentes.

Los problemas mal estructurados no están limitados a un dominio de conocimiento; no se conocen todos sus elementos con certidumbre; tienen metas vagas o imprecisas; pueden poseer múltiples o ninguna solución; no poseen prototipos; no hay seguridad sobre los conceptos y reglas que se tienen que aplicar, y requieren de quien los resuelve hacer juicios y defenderlos.

4.3 Complejidad de un problema

Esta se entiende como la cantidad de elementos, funciones o variables que definen un problema, junto con las interrelaciones y la predicción de comportamientos de estos componentes (Jonassen, 2006). La complejidad es una variable que tiene que ver con la memoria de trabajo. De Jong *et al.* (1998) definen la complejidad como la cantidad de información presente; esta determina la facilidad con la cual se puede resolver un problema.

Jonassen (2006) afirma que la competencia de resolución de problemas es específica al dominio y al contexto, a lo cual denomina *abstracción en contexto*. Se refiere al dominio porque las estrategias cognitivas para resolver problemas dentro de un dominio dependen de este y de su naturaleza, así como

también del contexto, porque se trata de actividades situadas y empotradas uno particular. Estas estrategias se pueden generalizar a través de todo el dominio.

4.4 La representación del problema

Jonassen (2006) menciona dos modalidades de la representación del problema: la *modalidad natural*, donde el individuo debe distinguir los hechos relevantes de los que no lo son y crear un espacio para generar la solución, y la *modalidad simulada*, donde el diseñador decide los componentes y partes que se deben incluir, así como la amplitud de las pistas que se den para resolver el problema y determinan su dificultad. A la hora de representar un problema se debe tomar en consideración la fidelidad con la realidad: si se incluyen el tiempo y las presiones sociales; si se resuelve en tiempo real, o si se realiza en forma cooperativa o competitiva, lo cual hace referencia a elementos del contexto.

4.5 Las diferencias individuales

Entre las diferencias individuales Jonassen (2006) menciona la *familiaridad del individuo con el problema*. Bransford y Stein (1993, citados en Jonassen, 2006) la identifican como problemas de rutina y no rutina. Un individuo que resuelva rutinariamente cierto tipo de problema, está más familiarizado con su solución. Las personas que tienen más experiencia en la resolución de problemas se diferencian de los novatos por los esquemas que poseen. Los esquemas, por su parte, son estructuras cognitivas que constituyen la base del conocimiento (Sweller, 1988) y se adquieren en el transcurso de la vida.

Para que haya aprendizaje se requiere un cambio de estructuras esquemáticas de la memoria a largo plazo y, a su vez, de los esquemas; ello se logra, según Cooper (2003), gracias a la reducción de carga de la memoria de trabajo. La familiaridad facilita la solución de tipos de problemas parecidos, pero, según Gick y Holyoak (1983), rara vez este conocimiento se transfiere a problemas de otro tipo. Los problemas no rutinarios exigen de quien los soluciona una alta capacidad de transferencia basada en el esfuerzo y en la acción consciente.

Dentro de las diferencias individuales, Jonassen (2006) también menciona *el conocimiento del dominio*, el cual sirve para comprender el problema y generar soluciones; pero para que esto ocurra se debe poseer un conocimiento estructurado, es decir, un conocimiento de la manera como se interrelacionan los conceptos de un dominio. El conocimiento estructurado se conoce también como estructura cognitiva u organización de relaciones entre los conceptos de la memoria.

Otras dimensiones que ayudan a dilucidar la competencia en resolución de problemas son: la *habilidad en el dominio*; el *control cognitivo*, que representa los patrones de pensamiento que controlan las formas como el individuo procesa y razona con la información; las *creencias epistemológicas*, especialmente importantes en los problemas mal estructurados donde, quien los resuelve, debe considerar la veracidad de las ideas y múltiples perspectivas en la evaluación del problema o de la solución. Estas creencias epistemológicas influyen en la forma de solucionar los problemas; así, los problemas más complejos y mal estructurados requieren niveles más altos de desarrollo intelectual. Asimismo, el proceso de resolución de problemas demanda procesos cognitivos y metas cognitivas y, adicionalmente, procesos conativos y afectivos, así como también perseverancia.

Por último, se considera dentro de las diferencias individuales el *trabajo individual contrapuesto al trabajo en grupo*. Para este último caso, es menester que el grupo comparta modelos mentales similares, para, de esa forma, poder trabajar sobre concepciones, estados y soluciones parecidas.

4.6 Tipos de problemas

De acuerdo con la tipología de problemas propuestos por Jonassen (2006), estos van desde bien estructurados hasta aquellos mal estructurados, lo mismo que de lo concreto a lo abstracto. Cada tipo de problema, por tanto, requiere procesos cognitivos que varían según su naturaleza. Así, los problemas pueden ser:

Problemas lógicos. Son pruebas lógicas abstractas que confunden al aprendiz y se usan para evaluar su agudeza mental, claridad y razonamiento lógico. En este tipo de problemas se deben encontrar las secuencias más eficientes de acción (mínimo número de movidas). Para cada problema hay un método específico de solución que conduce a la solución más eficiente, la cual el aprendiz deberá encontrar. Ejemplo de este tipo de problemas son: la torre de Hanoi, el juego de los caníbales y los misioneros.

Problemas algorítmicos. Son los más comunes en los centros educativos. En los cursos de matemáticas se enseña a solucionarlos usando un conjunto de procedimientos limitados y rígidos, con decisiones que se pueden predecir.

Para resolver ecuaciones se requiere aplicar una secuencia determinada de operaciones. Smith (1991) señala que este tipo de problemas no deben considerarse como tales, a menos que se busque una modificación del algoritmo para solucionarlos. Una limitante de esta clase de problemas es que se apoyan en estructuras de conocimientos procedimentales y en la ausencia de entendimiento conceptual del procedimiento. Asimismo, los contenidos aprendidos raramente pueden transferirse por la falta comprensión de los procesos subyacentes.

Problemas en forma de historia. Sirven para contextualizar algoritmos y son usados por los autores de textos y profesores. Contienen en una narrativa breve los valores necesarios para resolver el algoritmo; los aprendices deben extraer los valores y aplicar las fórmulas o procedimientos adecuados para encontrar las variables desconocidas. Requieren procesos cognitivos más complejos, pero resultan ser problemas poco llamativos para los estudiantes, quienes fallan a la hora de entender los principios y las aplicaciones conceptuales que subyacen en el desempeño; por esta razón, los estudiantes son incapaces de transferir la habilidad a problemas que posean la misma estructura profunda pero que difieran en sus características superficiales.

Problemas basados en reglas. Son aquellos de los que se tiene claro el propósito o la meta, pero hay muchas maneras de llegar a la solución y no existe un

método específico para alcanzarla. Las soluciones dependen de las reglas que se utilizan (las cuales pueden ser múltiples) e involucran la toma de decisiones.

Problemas de toma de decisiones. Se restringen a decisiones con un número limitado de soluciones. El número de factores a considerar entre las soluciones puede ser muy complejo; usualmente, requieren la comparación y el contraste entre las ventajas y desventajas de soluciones alternativas, y las decisiones se justifican en términos de la ponderación de esos factores.

Problemas de reparación de daños. Se trata de una de las formas más comunes de resolución de problemas, cuyo propósito principal es el diagnóstico del estado de error. Un sistema no funciona apropiadamente, lo que se manifiesta a través de una serie de síntomas que se deben identificar y comparar con el conocimiento del usuario respecto a una serie de estados defectuosos que este conoce. Los síntomas se utilizan para establecer y probar hipótesis sobre la razón del desperfecto.

La competencia para estos tipos de problemas demanda una combinación de conocimiento del sistema y del dominio (modelo conceptual del sistema). Se emplean estrategias de búsqueda y reemplazo, eliminación en serie, división de espacio y experiencia. Las estrategias se orientan a aplicar un conocimiento que permita generar hipótesis y trabajar planes.

Dicha competencia debe integrar modelos conceptuales, así como conocimientos funcionales y declarativos; incluye además conocimiento de interacción y conocimiento del sistema, flujo de control, estados defectuosos y procedimientos de prueba de error.

Problemas de diagnóstico. Son parecidos a los de reparación de daño ya que también comienzan con un estado defectuoso, pero se diferencian de aquellos en que parten de una serie de síntomas que particularizan el caso. Las estrategias que se emplean son más limitadas y los caminos para llegar a la solución del problema son ambiguos. Se puede citar como ejemplo la situación del médico que le diagnostica una determinada enfermedad a un paciente.

Problemas de casos situados. Son difíciles de resolver porque no siempre se sabe en qué consiste el problema; tienden a ser mal estructurados y se parecen mucho a los que se encuentran en la vida real. Los problemas de caso requieren de quien los resuelve el articular la naturaleza del problema y las diferentes perspectivas que lo impactan antes de sugerir una solución.

Estos problemas están mucho más ligados al contexto que cualquiera de los otros tipos mencionados; por tanto, su solución se basa en el análisis de factores contextuales. Para este tipo de problemas la justificación de la solución hace parte de los componentes más importantes del proceso de respuesta.

Problemas de diseño. Son uno de los tipos de problema peor estructurados y, dada su complejidad, requieren un balance entre las necesidades y las restricciones, además de una gran aplicación de conocimiento del dominio y de estrategias que conduzcan a un diseño original. Van más allá del concepto de transferencia específico a un problema, para llegar a la generalización de un conjunto de competencias.

Normalmente, se requiere que el diseñador haga una evaluación de necesidades y use el dominio de conocimiento para generar un diseño que funcione dentro de las restricciones. Por lo general, estos problemas poseen múltiples soluciones y, dado que los criterios para juzgar la mejor solución no son obvios, suele requerirse una gran competencia en argumentación y justificación.

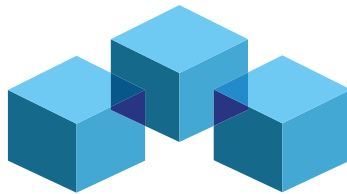
Problemas que consisten en dilemas. Son los peor estructurados e impredecibles, pues no hay solución que sea totalmente aceptable para la totalidad de problema. Estos representan situaciones sociales complejas, con perspectivas conflictivas.

De todos los anteriores tipos, el más utilizado en el medio educativo colombiano son los de forma de historia, empleados en las pruebas de Evaluación de la Calidad de la Educación Superior (ECAES) y en las pruebas de Estado para verificar la calidad de la educación.

Estos tipos de problemas tienen la ventaja, frente a los algorítmicos, de brindar un espacio para fortalecer las competencias interpretativas y propositivas,

así como contextualizar los aspectos teóricos de una asignatura en un problema que simula una situación real.

A pesar de que la literatura al respecto considera que este tipo de problemas son poco interesantes y señala también la incapacidad del estudiante para emplear las competencias desarrolladas en problemas similares, estas desventajas se pueden controlar a través de la mediación pedagógica, los temas utilizados en los problemas y las estrategias pedagógicas que convierten cada paso del problema en un reto para el estudiante.



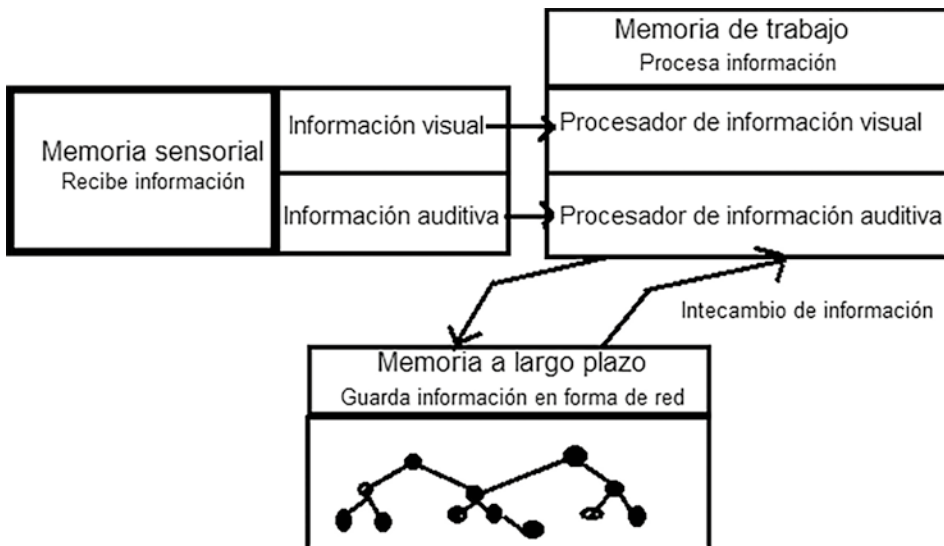
Capítulo 5.

La teoría de la carga cognitiva

En el marco de las ciencias cognitivas (las cuales, según Cooper, 2003, abordan principalmente los procesos mentales del aprendizaje, la memoria y la resolución de problemas), cobran especial relevancia los temas de la educación y el entrenamiento, así como la efectividad y la eficacia de estrategias variadas del diseño instruccional. Dentro de estas ciencias encontramos a la *teoría de la carga cognitiva* (Sweller & Van Merriënboer, 1998); según estos autores el aprendizaje tiene más éxito cuando sucede bajo condiciones que se alinean con la arquitectura de la cognición humana.

Dicha teoría considera que la arquitectura cognitiva del ser humano está compuesta de un sistema de procesamiento de información que involucra la interacción de tres componentes: la memoria sensorial, la memoria a largo plazo y la memoria de trabajo. La figura 4 ilustra el modelo del proceso de información de la arquitectura cognitiva humana, propuesto Sweller y Van Merriënboer (citados en Cooper, 2003).

FIGURA 4. Modelo de proceso de información de la arquitectura cognitiva humana



5.1 La arquitectura de la cognición humana

De acuerdo con Sweller (1988), el conocimiento de la arquitectura de la cognición humana es esencial para el diseño instruccional y la cognición visual es un aspecto central de dicha cognición. Por esta razón, existen diseños de efectos instruccionales que se apoyan en la manera en la que los humanos procesan visualmente la información.

El primer contacto de un individuo con el ambiente lo facilita la memoria sensorial (figura 4), encargada de los estímulos que proveen los sentidos; a cada uno de ellos le corresponde un segmento de esta memoria. Por su parte, en la memoria a largo plazo se almacenan los conocimientos y las habilidades de una manera permanente; su activación ocurre como resultado directo de un proceso de búsqueda de la memoria de trabajo sobre hechos reales a través de la conciencia.

Los conocimientos y las habilidades que se utilizan con regularidad se activan automáticamente sin la aplicación de niveles altos de conciencia (Cooper, 2003). Por otro lado, Sweller (1998) manifiesta que la memoria a largo plazo

no tiene como único objetivo el reconocer o recordar información, sino que es un componente integral de toda la actividad cognitiva incluyendo actividades tales como la resolución de problemas.

La memoria de trabajo provee la conciencia, permitiéndonos pensar lógica y creativamente, ser expresivos y resolver problemas. Su gran limitación es la cantidad de información que procesa, la cual se cree que varía entre siete y nueve elementos simultáneamente; esta capacidad se puede aumentar si se mezclan los modos de presentación, por ejemplo, una parte visual y otra auditiva, dado que, como se puede observar en la figura 4, cada uno de estos modos de presentación son procesados por segmentos diferentes, de forma tal que se pueden complementar.

Sweller (1998) señala que una consecuencia de la limitación de la memoria de trabajo es que, cuando se trabaja con elementos de alta interactividad, estos no pueden procesarse adecuadamente. Para entender dicho material se deben utilizar otros mecanismos u otras estructuras, como el concurso de la memoria a largo plazo, en donde se codifica y guarda el conocimiento.

Cooper (2003) define al aprendizaje como la codificación y el *almacenamiento* de conocimientos y habilidades en la memoria de largo plazo, de tal manera que ese conocimiento o habilidad se pueda recordar o aplicar más tarde cuando se demande. Para que esto ocurra, la memoria de trabajo debe atender y procesar la información; en palabras de Piaget (1972), se requiere asimilar la información antes de acomodarla o, como se expresa en términos de modelos mentales, si no se comprende la información, no se la puede anexar a la red de modelos mentales existentes.

En este sentido, la atención que se le preste a la información entrante tiene que ver con la carga cognitiva que, a su vez, está asociada con la cantidad de actividad mental que se le impone a la memoria de trabajo en un momento específico. En este proceso se analiza la cantidad de elementos a los que tiene que atender la memoria y las relaciones que se establecen entre ellos. Esto último define la *interactividad entre elementos*, descrita por Cooper

(2003) como el grado en el cual los elementos de la información que se van a aprender se pueden o no comprender aisladamente. Así, la información que se vaya a aprender y que posea alto nivel de interactividad en los elementos impondrá una mayor carga a la memoria de trabajo en comparación a los elementos individuales que la componen, porque también se debe atender a la relación entre ellos.

De otro lado, la consideración del elemento varía de acuerdo con el individuo: lo que para un novato pueden ser varios elementos, para un experto es uno solo. Los elementos contenidos en la memoria de largo plazo son los esquemas,¹ los cuales permiten categorizar elementos de información, dependiendo de la manera en que se vayan a utilizar. Muchas veces un esquema puede contener varios elementos y sus relaciones, si bien para un novato este no sea el caso. Por el contrario, para los expertos, los cuales poseen altos niveles de experiencia en el contenido de un área determinada, los elementos con los que se ocupa la memoria de trabajo son grandes redes de conocimiento complejo, es decir, esquemas de alto nivel, por lo que la memoria solo necesita prestarle atención a unos pocos elementos para retener la información que se va a aprender.

Cuando los aprendices tienen poca experticia se trabaja con elementos simples, por lo que la memoria de trabajo necesita prestarle atención a más elementos para retener la información que se va a aprender. En su caso, los recursos cognitivos se utilizan más allá de su capacidad, quedando pocos para el proceso de aprendizaje. Así las cosas, para los aprendices la manipulación del diseño instruccional será efectivo si se reduce la carga cognitiva dentro de los límites de la capacidad de su memoria de trabajo.

Según Sweller (1988) las diferencias entre un novato y un experto se pueden dividir en tres categorías: 1) memoria de la configuración de los estados de los problemas; 2) estrategias de solución de problemas; y 3) características

1 Sweller (1998) describe los esquemas como las estructuras cognitivas que conforman la base de conocimiento de los individuos.

usadas en la categorización de los problemas. Lo que se ha encontrado en relación con la memoria de la configuración de los estados de los problemas es que la diferencia entre un experto y un novato no radica en factores de memoria a corto plazo, sino en el tamaño de los *chunks* (segmentos de información).

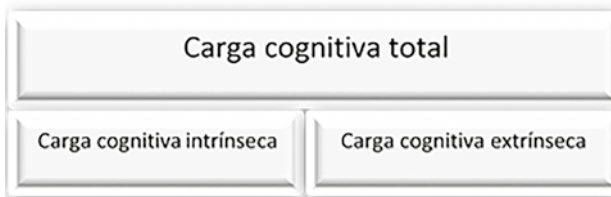
En lo que respecta a las estrategias de solución, los expertos utilizan el conocimiento que tienen de la configuración de los estados de los problemas, contenidos en sus esquemas, que también contienen los pasos de solución; los novatos, por el contrario, no poseen los esquemas apropiados y emplean otras estrategias de tipo general, lo cual complejiza la búsqueda de la solución.

Para categorizar los problemas, los expertos utilizan las estructuras profundas de estos para así poder aplicar las estrategias comunes de solución, evocando la configuración de estados de problemas conocidos cuando se presentan problemas similares. Por el contrario, los novatos se guían por la similitud de las características superficiales que, muchas veces, requieren de diferentes principios para resolver el problema.

Sweller (1998) concluye que el mayor factor que diferencia a los novatos de los expertos es el conocimiento del dominio en forma de esquemas; la diferencia en la memoria de los estados de los problemas, las estrategias de solución y la categorización se explican asumiendo que los expertos han adquirido esquemas que influyen la manera como ven y solucionan los problemas.

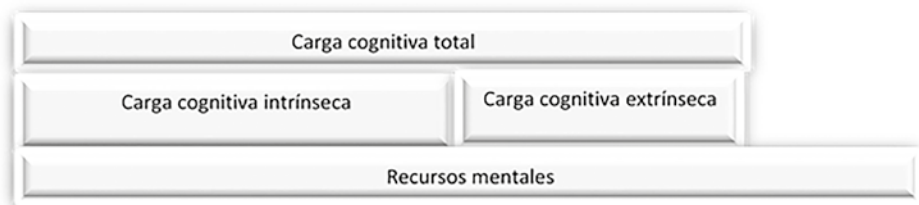
5.2 Tipos de cargas cognitivas

Existen tres tipos de carga cognitiva: la *intrínseca* corresponde a la dificultad inherente al tema y no se puede modificar mediante el diseño instruccional; por su parte, la *extrínseca* es propia del material que se prepara para los estudiantes; el aprendizaje se facilita o dificulta dependiendo del manejo que se le dé a esta. Finalmente, la carga cognitiva *total* es la suma de las dos anteriores, y no debe sobrepasar los recursos mentales poseídos. En la figura 5 se establece una relación entre los componentes de la carga cognitiva.

FIGURA 5. *Relación entre los componentes de la carga cognitiva*

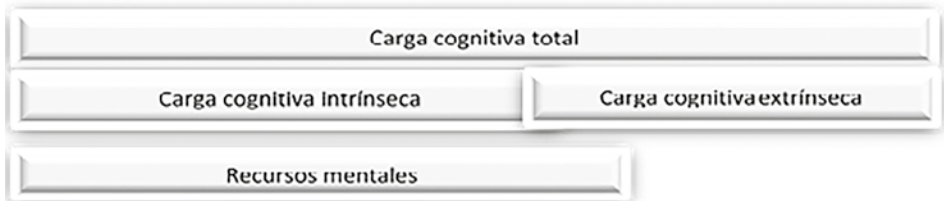
Las figuras 6 y 7 ilustran las situaciones que se pueden presentar en el manejo de la carga cognitiva total. Es necesario tener en cuenta, como se mencionó, que la carga cognitiva intrínseca no es manejable; pero la carga extrínseca se puede reducir a niveles más bajos, para lograr ajustar la carga total a los recursos mentales.

Cuando la carga cognitiva intrínseca es baja se dispone de suficientes recursos para que el aprendiz asimile cualquier clase de material instruccional, aunque se imponga una carga cognitiva extrínseca alta (figura 6).

FIGURA 6. *Carga cognitiva intrínseca baja*

Si las cargas cognitivas intrínseca y extrínseca son altas, posiblemente se superarán los recursos mentales, impidiéndose de esta manera el aprendizaje (figura 7).

FIGURA 7. Carga cognitiva intrínseca alta



En este caso se pueden modificar los materiales instruccionales para bajar la carga cognitiva, de tal manera que no se sobrepasen los recursos mentales y se logre el aprendizaje (figura 5).

5.3 Técnicas de la carga cognitiva que facilitan el aprendizaje

Cooper (2003) describe varias técnicas (a las que denomina «efectos») las cuales permiten reducir la carga cognitiva extrínseca al crear materiales de aprendizaje. Entre estas se encuentran: el efecto de la meta libre, el efecto de completar problemas, el efecto de la división de la atención, el efecto de la redundancia y el efecto de la modalidad; estas técnicas conducen a mantener la carga cognitiva dentro de los límites de la memoria de trabajo.

Efecto de la meta libre. En esta técnica no se define un estado final de los problemas sino que se invita al estudiante a encontrar lo que pueda con los datos proveídos, lo que induce a los aprendices a trabajar hacia adelante. Esta estrategia impone niveles bajos de carga cognitiva y facilita el aprendizaje.

Efecto de completar problemas y el ejemplo trabajado. El estudiante resuelve secuencialmente los problemas, siguiendo los pasos señalados de otro problema resuelto. Impone un nivel bajo de carga cognitiva y solamente se le presta atención al estado que se está resolviendo, así como a los operadores y las reglas que permiten la transformación.

Efecto de la división de la atención. Cuando en un material instruccional se requieren texto e imágenes es conveniente integrarlos para conformar una

sola fuente de información instruccional, de manera que el estudiante no tenga que dirigir la atención a otro lugar para tener ambas informaciones; esto daría lugar a la división de la atención, con el aumento correspondiente de carga cognitiva.

Efecto de redundancia. La información textual y gráfica se deben complementar; cuando sean redundantes, se debe eliminar una de estas. La aplicación de la atención en dos fuentes con la misma información impone al aprendiz una mayor carga cognitiva.

Efecto de la modalidad. De acuerdo con Pavio (1990) y Baddeley (1992, citados en Cooper, 2003), bajo ciertas condiciones se puede expandir la memoria de trabajo debido a que esta atiende la información visual y oral por diferentes canales; luego, la segmentación de la información (utilizando los dos sentidos correspondientes) incrementa la posibilidad de que aprendizaje efectivamente ocurra y expande la capacidad de la memoria de trabajo.

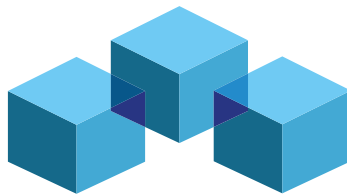
Como vemos, la teoría de la carga cognitiva tiene importantes aplicaciones en el diseño de los ambientes de aprendizaje computacionales: para bajar la carga cognitiva se puede manejar la carga cognitiva extrínseca, por lo que el material de aprendizaje a elaborarse puede dividirse en unidades más pequeñas que son evaluadas y retroalimentadas hasta que el estudiante las domine. Lo anterior contribuye a evitar la utilización de estrategias como el análisis de medios y fines, que consiste en reducir la diferencia entre el estado actual del problema y la meta, donde el aprendiz trabaja hacia atrás, a partir de la meta hasta el problema dado.

Con la subdivisión del material de aprendizaje en unidades más pequeñas se logra emular la técnica del efecto libre de meta, pues el estudiante trabaja hacia adelante, concentrándose en cada paso, cada uno de los cuales deberá resolver para poder avanzar hasta llegar a la meta planteada en el problema. En la medida en que se avanza en la construcción de esquemas, se va aumentando el nivel de complejidad de los problemas; esto asegura la comprensión

del tema e incrementa la posibilidad del almacenamiento de la información en la memoria a largo plazo.

La teoría de la carga cognitiva se pone en evidencia en la medida en que se deben construir esquemas solucionando problemas del mismo tipo, para poder poseer la competencia de resolución de problemas. Al evaluar el desempeño de los estudiantes, por tanto, lo que se hace es verificar si poseen los esquemas para resolver los problemas en cuestión. Los ambientes computacionales de aprendizaje, por su parte, deben proponer suficientes problemas, de tal manera que se logre automatizar el proceso y facilitar la conversión del novato en experto.

Estas técnicas contrastan con las que usan los llamados *problemas convencionales* (Sweller, 1988), que emplean la estrategia de medios y fines para su solución. Se trata de una técnica utilizada cuando no existe familiaridad con el tipo de problema específico, y se basa en la reducción de la diferencia entre el estado actual del problema y la meta trabajando para ello hacia atrás; es decir, de la meta hacia el problema dado. La anterior es una estrategia muy efectiva para lograr obtener la solución a un problema, pero conlleva altos niveles de carga cognitiva, puesto que la atención se debe dirigir al estado actual, la meta, diferencia entre la meta y el estado actual, el procedimiento para reducir esa diferencia y algunas submetas posibles que puedan conducir a la solución (Sweller, 1988).



Capítulo 6.

Los ambientes de aprendizaje computacionales

Un ambiente de aprendizaje computacional es un entorno integrado que propone situaciones para que el estudiante desarrolle su conocimiento y lo aplique en el dominio disciplinar que corresponda. Para el desarrollo de estos ambientes se deben tener en cuenta las investigaciones de las ciencias que tienen que ver con el aprendizaje y es aquí en donde se materializa la aplicación de los temas vistos hasta ahora. A continuación se exponen los diferentes tipos de aprendizaje señalados por Groner y Kersten (2001).

6.1 Sistemas de ejercitación y práctica

Los programas de ejercitación y práctica buscan alcanzar los objetivos de aprendizaje a través de ejercicios, cuya secuencia es: el ejercicio, la entrada de una respuesta y la retroalimentación del sistema. Estos sistemas son utilizados como medios para reforzar la enseñanza que se da de manera usual; vienen sin un tutor que ofrezca ayuda didáctica y la retroalimentación se limita solamente a la corrección, en caso de error, de la entrada generada por el estudiante.

6.2 Programas de simulación

Mediante los programas de simulación se representan modelos de sistemas con el fin de que el estudiante haga descubrimientos y, además, pueden combinarse con sistemas inteligentes para incrementar su utilidad. Usualmente con estos programas se ilustran complejos sistemas científicos, técnicos, sociales, económicos o ecológicos, los cuales se caracterizan por tener una flexibilidad e iniciativa media del estudiante. El alumno tiene control de la secuencia, puede cambiar parámetros y observar el resultado de sus acciones. La retroalimentación aparece de manera implícita, pues no se emiten explicaciones y se acepta la introducción de parámetros ilógicos.

6.3 Micromundos

Los micromundos tienen la intención de profundizar temas conocidos; por tanto, es muy poco lo que estos sistemas pueden llegar a beneficiar a personas que no tengan la habilidad para formular experimentos, confirmar o negar hipótesis, o hacer automonitoreo. La retroalimentación es implícita y no se dan explicaciones sobre la acción del estudiante, quien actúa utilizando el principio de ensayo y error. Estos ambientes ofrecen la posibilidad de interactuar libremente con situaciones que en la vida real son peligrosas o muy costosas.

6.4 Juegos

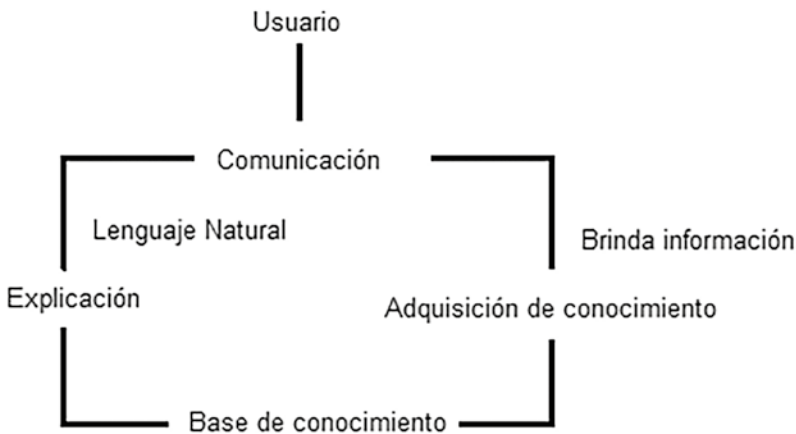
Son programas que ofrecen el contenido de aprendizaje en forma de juego, siendo usados especialmente en la escuela básica. A través de estos juegos se ejercitan determinadas competencias de forma amena.

6.5 Sistemas expertos

Los sistemas expertos, basados en conocimiento, son posibles gracias a la tecnología de la inteligencia artificial y buscan establecer un diálogo natural con el sistema. En relación con los sistemas de aprendizaje, la flexibilidad es

alta, mientras que es baja con respecto al usuario, dado que este tiene poca libertad y control sobre la secuencia de la presentación de los ejemplos, el origen de las tareas o sobre su grado de dificultad. En la figura 8 se observa que todas las funciones se construyen a partir de la base de conocimiento, y las explicaciones del sistema se dan a través de un lenguaje natural y, mediante el módulo de adquisición de conocimiento, el usuario adquiere información a partir de la base de conocimiento.

FIGURA 8. *Arquitectura de un sistema experto basado en conocimiento*



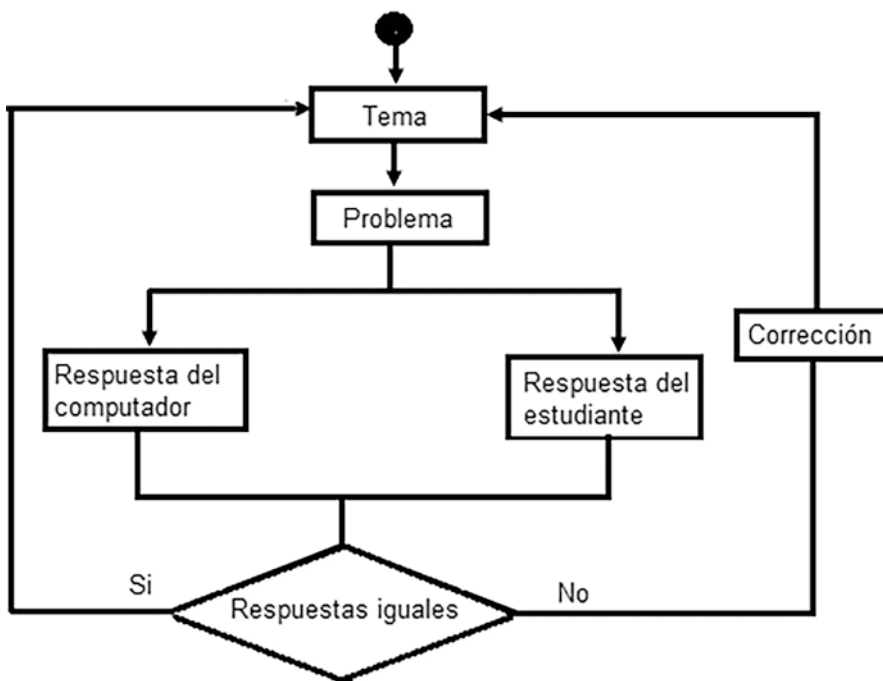
6.6 Tutores para el aprendizaje

Los programas tutores buscan ajustarse a situaciones de aprendizaje individualizado, donde los temas son introducidos a través de ejemplos, preguntas y tareas que comprueban la comprensión que se haya tenido del tema; las instrucciones que siguen estos programas dependen del adelanto que los usuarios tengan en los temas precedentes. Los tutores pueden ser activos o pasivos: los activos, implementados mediante las *instrucciones asistidas por computador* (CAI, por sus siglas en inglés), se manifiestan cuando se presenta la necesidad de alguna ayuda; los pasivos, por su parte, son implementados mediante los *entrenamientos basados en el computador* (CBT, por sus siglas en inglés).

Los programas CAI ilustrados en la figura 9 se basan en la teoría del estímulo-respuesta de Skinner y evalúan en cada punto si la respuesta es correcta. Cuando se presentan errores el programa trata de corregirlos, proponiendo nuevo material o presentando tareas más fáciles. Si la respuesta es correcta, se proponen nuevos problemas para ser solucionados.

Debido a que los programas CAI no evalúan la fuente del error del estudiante, se desarrollaron los programas de *instrucciones asistidas por computador inteligente* (ICAI, por sus siglas en inglés), cuya característica principal consiste en la habilidad de diagnosticar los errores y enmendarlos de acuerdo con este diagnóstico.

FIGURA 9. Montaje de un programa de instrucciones asistidas por computador



6.7 Multimedia

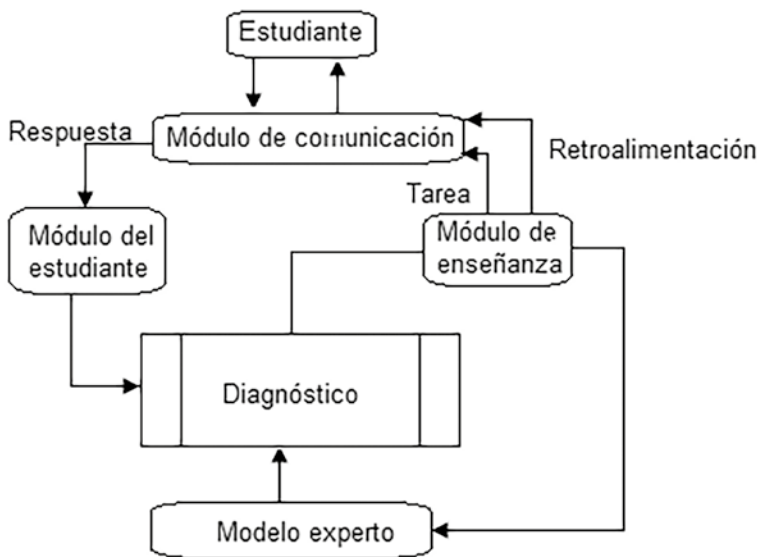
Se entiende por multimedia a la unión de múltiples medios: texto, gráficos, dibujos, vídeos y secuencias de audio en programas cuya aplicación se

extiende a diferentes ámbitos. La hipermedia designa la estructura no lineal de la información en la red; es por ello que la *world wide web* se constituye como un medio de aprendizaje debido a que tiene una estructura capaz de manejar hipermedia y multimedia. Los programas multimedia exigen una alta motivación e iniciativa propia, razón por la cual las personas no motivadas o que no conocen el sistema sacan poco provecho de este medio ya que se pierden fácilmente en el hiperespacio. Este medio permite el aprendizaje colaborativo a través de estrategias como el correo electrónico, los foros de discusión o el chat.

6.8 Sistemas de tutores inteligentes

Los sistemas de tutores inteligentes (ITS, por sus siglas en inglés) constan de cuatro partes: a) la base de conocimiento o módulo experto; b) la información sobre el estado de conocimiento del aprendiz o módulo del estudiante; c) las estrategias de enseñanza o módulo de enseñanza, y d) el módulo de comunicación. En la figura 10 se ilustra la interacción entre estos componentes.

FIGURA 10. Secuencia de un sistema de tutor inteligente ideal



El módulo de enseñanza selecciona una tarea y se la comunica al módulo experto y también, a través del módulo de comunicación, al estudiante. Este primer módulo determina el contenido de la interacción, mientras que el módulo de comunicación determina la forma. El estudiante realiza la tarea y establece la solución, o de acuerdo con la situación, los pasos intermedios; por tanto, la solución es transmitida a través del módulo de comunicación al módulo del estudiante, donde es comparada con la solución del módulo experto.

El sistema emite un diagnóstico y lo comunica al módulo de enseñanza, lo que le facilita a este producir una retroalimentación. Se pueden generar tareas que concuerden con el conocimiento del estudiante u otros ejemplos que sirvan como ilustración cuando se cometen errores. Entonces, la siguiente unidad de enseñanza dependerá del estado de conocimiento del estudiante.

6.8.1 Módulo experto

El módulo experto es la espina dorsal de los sistemas de tutores inteligentes y busca emular las competencias de un buen profesor. Posee varios procedimientos de solución y está en capacidad de explicar los argumentos sobre los cuales descansan los caminos de solución. Este módulo, además de tener un componente de solución de problemas, tiene también un componente de explicación; este último representa el conocimiento mediante listas, diagramas, hechos y reglas.

En este módulo se pueden guardar diferentes clases de conocimiento: declarativo, procedimental, cualitativo y heurístico, los cuales se describen a continuación:

Conocimiento declarativo. Trata conocimientos fácticos. El módulo experto proporciona información sobre hechos generales y responde preguntas del tipo ¿qué está bien? Por ejemplo, $4 + 6 = 10$; Bogotá es la capital de Colombia.

Conocimiento procedimental. Designa los conocimientos de procesos y describe cómo se lleva a cabo una tarea. Puede contestar preguntas del tipo ¿cómo

se hace esto? Por ejemplo, cómo se envía un SMS o cómo se instala un componente de *hardware*.

Conocimiento cualitativo. Busca explicar el conocimiento observado a partir de modelos mentales. Trata de contestar preguntas de tipo ¿por qué pasa esto?

Conocimiento heurístico. Consta de la experiencia y el conocimiento de resolución de problemas de un experto y no está ligado a un contenido. Este tipo de conocimiento es importante para que un tutor pueda orientar al estudiante en su proceso de aprendizaje y en su conducta de resolución de problemas.

El tipo de conocimiento más fácil de comunicar y aplicar es el declarativo; el conocimiento procedimental constituye todavía un gran reto para desarrolladores de ambientes de aprendizaje y es aplicado en ambientes de simulación. Finalmente, el conocimiento cualitativo todavía no ha sido posible implementarlo.

6.8.2 Métodos para la recolección de la información

La adquisición de conocimiento en un módulo experto hace referencia a la recolección y estructuración del saber disciplinar; en esto trabajan los programadores y expertos en el campo. Existen varios métodos para su recolección:

Entrevista o búsqueda de campo. A través de este método se reúne información abundante y no estructurada; no obstante, es costoso y la información recolectada es difícil de analizar. Se recomienda su uso en el aprendizaje del conocimiento procedimental, para el cual es importante tomar en consideración el proceder de los expertos en la resolución de un problema.

Protocolo o monitoreo. En lugar de preguntar al experto, se puede observar y analizar su proceder; de esta manera es posible captar comportamientos que el experto no considera importantes, pero que son de mucha ayuda para el aprendiz.

Introspección o pensar en voz alta o cuestionario. Este método busca capturar la manera de reflexionar de un experto.

Repaso. El programador le explica al experto lo que entendió de una explicación previa para que este pueda precisar confusiones y vacíos en la base de conocimiento.

Método del sabio de Oz. En este caso una persona toma la función de un computador; el experto se sienta ante una pantalla y resuelve un problema. En el otro terminal no se encuentra ningún computador sino el programador y, este último, puede mirar las reacciones del primero para tenerlas en cuenta en el desarrollo del módulo experto. Esta misma metodología se utiliza para los estudiantes; en este caso se comparan los resultados con los del experto y se utiliza la diferencia para la parte didáctica.

6.8.3 Representación del conocimiento

Después de que se haya determinado el tipo de conocimiento a manejar y el método para la adquisición de conocimiento, se debe proceder a determinar cómo se va a representar el conocimiento, para lo cual existen tres modelos:

Modelo de la caja negra. Aquí se provee la respuesta correcta pero el programa no tiene la capacidad de dar una explicación de la solución diferenciada o de diagnosticar la causa del error. Se dice que tiene una solución impenetrable y la retroalimentación se reduce a contestar si la respuesta es correcta o incorrecta.

Modelo glassbox (caja de vidrio). En este sistema se puede ver la solución, así que se dice que es un modelo transparente. Requiere que el conocimiento en la base de conocimiento esté dado en forma de reglas y en forma lógica. En el caso de conocimiento declarativo, se dice que es un sistema basado en reglas; en el caso del conocimiento procedimental se trata de un sistema lógico, que puede estar representado a través de redes semánticas o marcos. Se pueden observar los caminos de solución pero estos no corresponden al comportamiento de un ser humano, por lo que las explicaciones pueden ser difíciles de entender.

Modelo cognoscitivo. Se considera una extensión del modelo *glassbox*, ya que busca descomponer el conocimiento en partes y simular el proceso de solución de problemas del ser humano. Las explicaciones se hacen así comprensibles, pero es más intenso en el uso de reglas y es el más dispendioso en la recolección y estructuración del conocimiento.

6.8.4 Requerimiento de la representación del conocimiento

A continuación se presentan las exigencias que el conocimiento debe cumplir para conformar la base de conocimiento funcional:

Íntegro. La representación del conocimiento debe ser suficientemente amplia, de tal manera que el sistema sea autosuficiente en el campo disciplinar.

Comprensible. Debe ser fácilmente visible, para que posibilite el diseño del componente explicativo.

Modular. Facilidad de cambio y ampliación.

Asequible. Aprehensible, sobre todo que conduzca a hacer deducciones que se puedan almacenar.

6.9 Generadores de tareas y bancos de tareas

Después que se tenga una forma de representación del conocimiento, el siguiente paso es pensar de qué manera se van a elaborar las tareas. Existen dos formas: generadores de tareas y bancos de tareas.

Los generadores de tarea funcionan durante los tiempos de ejecución y se deben ajustar a las capacidades y motivaciones del estudiante. Esta labor resulta poco costosa cuando las tareas están parametrizadas. Los bancos de tareas se utilizan para tareas más complejas; por ejemplo, en caso de diagnósticos médicos. En este caso, el módulo de enseñanza elige una tarea que está predefinida en los bancos de tarea, la cual debe concordar con el estado de conocimiento del estudiante.

6.10 Módulo del estudiante

La función de este módulo es generar un modelo exacto del estudiante, el cual debe ser una imagen del criterio, las habilidades, la motivación, los conocimientos previos y las preferencias del usuario. Para ello se tienen en cuenta características emocionales y sociales, el estilo de aprendizaje, el tiempo de trabajo y la capacidad de concentración.

Una función importante de un sistema de tutor inteligente (ITS) es la personalización de los procesos de aprendizaje, la cual se logra a través de la adaptabilidad del sistema. Para esto, el módulo del estudiante debe poseer funciones de diagnóstico, el cual es utilizado por el módulo de enseñanza para adaptar la tarea seleccionada al usuario; también sirve para determinar la nueva unidad de estudio.

Cuando interviene el programa, este le genera al estudiante una explicación de acuerdo con su nivel de formación. Una mejor ayuda se puede lograr cuando el programa reconoce si el usuario es un novato o un experto; no obstante, se debe tener cuidado al hacer esta clasificación, porque si se coloca una tarea dirigida a un experto a un novato, puede producir desalientos en los estudiantes, al no poder desarrollar esta tarea.

Adquisición de conocimiento. El desarrollador de un ITS debe determinar cómo producir el modelo del estudiante a partir del conocimiento de este; asimismo, se debe decidir cuándo crearlo.

Se puede adquirir el conocimiento antes del tiempo de ejecución. Mediante este proceso se modela el estudiante ideal; en el desarrollo del sistema se puede observar la conducta de resolución de problema del usuario a través del análisis de protocolo en trabajos realizados. Adicionalmente, se le pide al estudiante que piense en voz alta; de esta manera se puede construir una biblioteca de errores que contendrá todas las fuentes de error del estudiante que se puedan pronosticar.

El conocimiento también se puede adquirir durante el tiempo de ejecución. Aquí se establece una diferencia entre la adquisición de conocimiento

implícito y explícito: con el conocimiento implícito se puede observar la conducta de resolución de problema del usuario y se analiza, por ejemplo, la forma y el contenido de la respuesta del estudiante; con el conocimiento explícito se hacen preguntas explícitas. No obstante lo anterior, también se pueden proponer tareas adicionales para facilitar el diagnóstico y, asimismo, el conocimiento del camino de resolución puede comprender un solo paso, pasos intermedios o la solución final.

Modelos de representación del conocimiento. Después de que se hace la adquisición del conocimiento por los métodos descritos, este se puede representar de diferentes maneras:

Modelo de la lógica clásica: este método comprende las reglas y los hechos del modelo del usuario y las relaciones entre sus propiedades; una vez que se deriven conclusiones, estas no se pueden cambiar.

Modelo dinámico del usuario: se adapta a los cambios de las competencias del usuario; cuando una respuesta es catalogada como incorrecta, el sistema la corrige y en el modelo del estudiante se reflejan estos cambios.

Modelo del usuario alternativo: este modelo trabaja con estereotipos. Las características de un estudiante se comparan y se ajustan con estereotipos previamente dados, de manera tal que el sistema reaccionará en concordancia con estos.

6.10.1 Modelo de diagnóstico

Sirve para encontrar la causa de los errores cometidos por el usuario. El modelo busca la discrepancia entre las respuestas de los estudiantes y el módulo experto. Existen dos clases:

El modelo de superposición. Señala que el estudiante comete errores porque, a pesar de que su conocimiento es correcto, no está presentado de forma completa.

Modelo del error. Supone que el estudiante posee el conocimiento correcto, pero comete errores en la aplicación de este.

6.10.2 Técnicas de diagnóstico

Se pueden considerar varias técnicas para establecer el estado del conocimiento; este se determina a partir de la manera en que se trabaja una tarea y la frecuencia con la que se cometen los errores. Las técnicas de diagnóstico conocidas son:

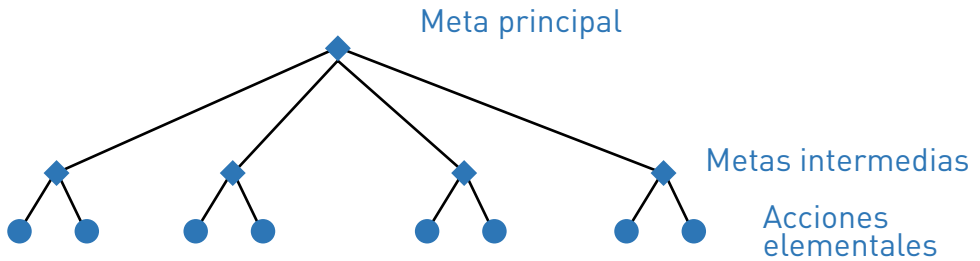
Protocolo. Se puede protocolizar las actuaciones del estudiante cuando se conocen todos los pasos para la solución. A través del protocolo se puede determinar qué tan bien resuelve el estudiante una tarea o dónde ha cometido un error.

Búsqueda del camino. Cuando no se conocen los pasos de solución efectuados por el estudiante, se pueden utilizar los pasos intermedios para generar el camino completo de la solución. A través de preguntas al usuario se puede adquirir información adicional. En general, se pueden construir varios caminos, por lo que la escogencia del más posible se hace difícil.

Inducción por regla. Busca reconstruir las respuestas del estudiante a través de un sistema de reglas compuesto de acciones y condiciones. Su ventaja es que no necesita de una biblioteca de errores, mientras que su dificultad estriba en que tienen que darse muchas situaciones problemáticas para reconocer las reglas subyacentes; un problema adicional es que muchas veces no es clara la escogencia de las acciones y condiciones relevantes.

Reconocimiento de un plan. La condición para este diagnóstico es que se haya construido un sistema experto jerárquico con conocimientos de tipo procedimental. Cuando se conocen los pasos de solución de problemas del estudiante se puede modelar un árbol de solución del problema (figura 11), cuyas hojas son las acciones elementales, los nodos intermedios describen submetas y los nodos de la raíz la meta principal. El camino conocido a través del árbol conduce a un modelo del estudiante, que se construye con las técnicas del protocolo.

FIGURA 11. Árbol de solución de un problema



6.10.3 Funciones del módulo del estudiante

Existen veinte funciones, agrupadas en seis categorías:

Función correctiva. El sistema debe tener la capacidad de corregir los conocimientos incorrectos del estudiante.

Función elaborativa. El sistema debe intervenir cuando el conocimiento del estudiante es correcto pero incompleto.

Función estratégica. El sistema debe adaptarse al plan de aprendizaje del usuario: este puede cambiar la metodología, utilizar otra estrategia o ajustar el grado de dificultad.

Función de diagnóstico. El sistema debe descubrir las preferencias del aprendiz para poder analizar el modelo del estudiante.

Función predicativa. El sistema debe tener la capacidad de poder simular al aprendiz como modelo y determinar la dirección del proceso de aprendizaje; así, deberá predecir la conducta del estudiante.

Función evaluativa. El sistema debe estar en capacidad de reconstruir los procesos de aprendizaje del estudiante, para lo cual es importante conocer la historia completa de su aprendizaje.

6.11 Módulo de enseñanza

Este módulo representa a un experto en didáctica y metódica. La didáctica es entendida como el método que, según Schulmeister (1996), ayuda a ordenar el proceso y el ambiente de aprendizaje, además de apoyar las situaciones y los medios de aprendizaje. El módulo de enseñanza determina la interacción (donde el contenido de la enseñanza seleccionado concuerda con el modelo de estudiante), escoge la meta de aprendizaje y decide cuándo el tutor debe interrumpir al usuario; de esta manera, se ve afectada la propia iniciativa del estudiante.

6.11.1 Estrategias del tutor

Van desde la utilización de estrategias receptivas, en las cuales el estudiante está limitado en su espacio de libertad, hasta estrategias de descubrimiento, donde el estudiante puede investigar la disciplina por sí mismo. Los tutores pueden ser de varios tipos:

Receptivo y deductivo. Se presentan primero las bases y luego se ilustran mediante ejemplos.

Receptivo e inductivo. Primero se muestran los ejemplos y luego se dan los principios teóricos subyacentes.

Orientado al descubrimiento. Se dan ejemplos para que el estudiante descubra los principios teóricos relacionados.

Orientado al descubrimiento por adaptación. Se da retroalimentación al estudiante, la cual puede ser inmediata o cuando este se encuentre un callejón sin salida. La inmediata se provee una vez que el estudiante comete un error; la segunda, cuando puede seguir por sí solo.

Descubrimiento conducido. El material de apoyo y las metas de aprendizaje se dan con antelación. El estudiante determina las metas y el tiempo.

Descubrimiento libre. El proceso de aprendizaje es asumido totalmente por el estudiante.

6.11.2 Componentes del tutor

La función principal de un tutor es la selección, secuenciación y presentación del contenido de enseñanza; por tanto, se debe distinguir entre el contenido del currículo y el componente de ayuda.

Componente curricular o tutor global. Facilita decisiones sobre selección, secuenciación y presentación del contenido de enseñanza a largo plazo, y es responsable de la planeación de los objetivos de aprendizaje. Se construye con redes semánticas y sistemas de producción. Para la construcción del plan de enseñanza se organiza el conocimiento de la base de conocimiento en metas de aprendizaje y se establecen relaciones semánticas entre las unidades de aprendizaje que se producen (cuyo orden y selección dependen del estudiante). Cada tarea debe ser realizable por el estudiante y las unidades de aprendizaje deberán reflejar el conocimiento disciplinar.

Componente de ayuda o tutor local. Maneja el contenido de enseñanza a corto plazo y determina tanto el contenido de la tarea siguiente, como la intervención del tutor dentro una lección. Se deben analizar la clase, el contenido, el tiempo, el momento y el nivel con el que interviene el componente de ayuda para que concuerden con los sucesos de enseñanza.

Control de intervención. Permite decidir si la intervención es realizada por el estudiante o si es ofrecida por el sistema. *Al momento de la intervención* se debe definir si la intervención se hace una vez que se comete un error, después de repetir los errores, después de que el estudiante se aleja de la solución, cuando el sistema prevé que en el siguiente paso se va a cometer un error o, lo óptimo, cuando se encuentra en un callejón sin salida. El *nivel de intervención* corresponde a la intervención en el comportamiento del individuo o del conocimiento: en el primer caso, se trata de dirigir la conducta de solución del problema del estudiante con sugerencias y consejos; en el segundo caso, el sistema corrige estructuras cognitivas a través de explicaciones.

6.11.3 Funciones del tutor

Un tutor debe tener la capacidad de asignar tareas y reaccionar ante los errores y las preguntas de los estudiantes; estas pueden ser de dos tipos: cerradas o abiertas. En las preguntas cerradas se ofrecen varias alternativas de las que se escoge la que se considera como correcta; en las preguntas abiertas, el estudiante puede formular su respuesta.

Aquí es importante identificar dónde interviene el programa y el grado de dificultad de la tarea, pues ante una respuesta incorrecta el modelo de enseñanza puede reaccionar: a) repitiendo la información o remitiendo a tareas previamente resueltas; b) llamando parte o la solución completa; c) presentando el protocolo de error o de solución total o parcialmente; d) mediante contraejemplos, mostrando que la solución está en contradicción a la información presentada; o, finalmente, e) asignando una tarea más fácil.

6.11.4 Reglas de la planeación de la enseñanza

Para el diseño del tutor y de la enseñanza en general, Lusti (1992) presenta, a partir de fuentes provenientes de las teorías de la psicología del aprendizaje —e incluso del sentido común—, las siguientes reglas:

Contenido de la presentación. Formular metas de aprendizaje detalladamente; estructurar el material de aprendizaje jerárquicamente o por redes; ordenar en temas y en lecciones; organizar la situación de aprendizaje cercana a la realidad; motivar a través de sorpresa, duda, contradicción, juego y simulación; visualizar e ilustrar con ejemplos y material confiable; personalizar; llamar al aprendiz por su nombre; adaptar los pasos al nivel de competencia; incentivar el pensamiento en voz alta; minimizar la carga de la memoria a corto plazo; promover el aprendizaje firme a través de la repetición.

Colocación de trabajo. Adaptar los trabajos a las metas de aprendizaje y al modelo del estudiante; otorgar libertad de selección; ofrecer suficientes y variadas oportunidades de práctica.

Reacción ante errores y preguntas. Mejorar cuando se presentan errores; reforzar conductas positivas y consistentes, apoyado en criterios objetivos intrínsecamente relacionados con el futuro; variar el refuerzo; mantener la retroalimentación negativa como información; integrar las respuestas coherentemente; comparar con el desempeño propio y no con el desempeño de otros.

Secuenciación y puntos de intervención. Secuenciar primero lógicamente y después según grado de dificultad; suministrar un vistazo general del contenido de aprendizaje, antes de que se profundicen las metas; favorecer intervenciones cortas; proporcionar refuerzo inmediato.

6.12 Módulo de comunicación

Facilita la comunicación entre el sistema y el aprendiz, determinando así la forma de interacción adecuada del diálogo entre el hombre y la máquina. Algunos de los modos de comunicación conocidos son: cajas de textos, lista de selección y menús; según el grado de libertad que ofrece el módulo, estos modos pueden variar desde estilos de diálogo totalmente controlados hasta formas de comunicación donde el aprendiz tiene libertad de selección.

La *interactividad* se puede construir de una manera incremental, mediante el aprovechamiento de cierta información, selecciones o cambios de hojas; ofrece la posibilidad de respuesta de solución múltiple y sí/no, y de remisión a información adicional; posibilita la marcación de determinados sitios de interacción y la activación de información adicional correspondiente; permite el registro libre de respuestas complejas sobre planteamientos complejos con retroalimentación inteligente de tutoriales (diálogo socrático), así como el diálogo libre con un tutor o con un compañero con ayuda de híper o multimedia.

El *diseño del diálogo*, por su parte, representa la interacción hombre-computador y ofrece diferentes estilos. Janetzko (1999) describe tres clases de diálogos:

El usuario pregunta-el sistema responde: en este diálogo el usuario tiene la iniciativa. Generalmente, se trata de un conjunto de preguntas definidas previamente y disponibles para el usuario; en esta categoría se encuentran los menús.

En los sistemas de aprendizaje que poseen este estilo de interacción el aprendiz puede moverse o manipular libremente por el programa; el computador reacciona a las entradas, realiza determinadas funciones y da información de retroalimentación. Dentro de esta categoría de *software* se encuentran sistemas de aprendizaje de baja flexibilidad y alta iniciativa de aprendizaje. Aquí se pueden nombrar como ejemplo los sistemas hipermediales, que se distinguen por poseer diálogos conducidos por el aprendiz, cuya parte esencial es la navegación libre en una red de información que le permite al usuario reunir la información de manera individual.

El sistema pregunta-el usuario responde: este tipo de diálogo no es simplemente contrario al anterior, sino que incluye otros aspectos importantes: el sistema debe disponer de mecanismos que le posibiliten al usuario tomar la iniciativa del diálogo, así como determinar el cuándo y la clase de diálogo. Este tipo de diálogo supone el dominio de estrategias de dirección y debe tener la capacidad de actuar contextualmente. El aprendiz tiene un rol pasivo y actúa ante las preguntas y propuestas del sistema. Estos sistemas de aprendizaje ostentan alta flexibilidad del sistema y baja iniciativa del aprendiz.

El usuario o el sistema pregunta-el usuario el sistema responde: Esta implica técnicas de desarrollo más exigentes. Debe poseer opciones para el usuario, como también mecanismos de conducción de diálogo por parte del sistema. Después de la iniciativa del usuario, el sistema debe poseer una capacidad de conducción que haga reflexionar al usuario y lo conduzca a nuevas iniciativas. El sistema debe ser capaz de interpretar diferentes formas de comunicación y reaccionar de acuerdo con estas. Se caracteriza por tener alta flexibilidad del sistema y alta iniciativa del usuario.

Con el *tratamiento de lenguaje natural*, los diálogos son eficientes cuando son naturales, consistentes, poco redundantes, amigables y flexibles. El diálogo es natural cuando ocurre en el orden acostumbrado de lenguaje y es consistente cuando, por ejemplo, se pide una ayuda y el texto siempre se llama con la misma tecla, o cuando en caso de error se muestra un texto y este el texto siempre es intermitente; asimismo, se considera amigable cuando ofrece textos de ayuda y sistemas de mensajes eficientes, y es flexible cuando se acomoda a las capacidades e intereses del usuario (Lustig, 1992).

La enseñanza asistida por computador produce buenos resultados cuando el estudiante tiene acceso directo al contenido de aprendizaje y no debe aprender el manejo del *software* o del *hardware*. Idealmente, el usuario debería poder comunicarse con el sistema a través del lenguaje natural, de la misma manera como se comunicaría con un profesor; así, lo óptimo sería que el estudiante pudiese hacer las preguntas y presentar sus soluciones en lenguaje natural.

6.13 Redes semánticas

Las redes semánticas son un grafo dirigido compuesto por varios nodos que representan objetos o conceptos, como también por arcos que representan las relaciones entre estos objetos. Normalmente a estos nodos y arcos se les asigna un nombre para identificarlos; los nodos pueden ser descriptores que suministran información adicional sobre los objetos, mientras que los arcos representan relaciones que unen a los objetos y los descriptores. Los más conocidos son:

Es_un, con el cual se representa la relación entre clases y casos particulares, por ejemplo: «Nerón es un dóberman». También se puede expresar una relación con una subcategoría, por ejemplo: «un dóberman es un perro».

Tiene, denota una relación entre partes y elementos de la parte, por ejemplo: «Un perro tiene una cola».

Herencia, denota el hecho de que un nodo hereda las características de otros nodos con los cuales está unido. La herencia de propiedades es una consecuencia de la relación *Es_un* y significa que todos los casos particulares toman todas las propiedades de las clases supraordenadas a las cuales pertenecen:

Instancia de (Nerón, dóberman)

Es_un (dóberman, perro)

Tiene (perro, cola)

Tiene (X, atributo): Es_un (X, Y), tiene (Y, atributo)

Tiene (X, atributo): Instancia de (X, Y), tiene (Y, atributo)

Tripleta objeto, atributo, valor es otro método común para representar el conocimiento; se trata de un caso especial de representación utilizando redes semánticas. Los objetos pueden ser entidades físicas o unidades conceptuales; por su parte, los atributos son características generales o propiedades que son asociadas con los objetos (grosor, forma, color son atributos típicos de los objetos físicos), cuyo valor indica la condición específica de un atributo en determinada situación (tabla 1).

TABLA 1. Ejemplos tripleta objeto, atributo, valor

Objeto	Atributo	Valor
Manzana	Procedencia	Israel
Manzana	Consistencia	Buena
Uva	Color	Azul
Uva	Procedencia	Italia
Niño	Categoría	Nombre
Niño	Género	Masculino
Niño	Número	Singular
Niño	Persona	3
Canta	Categoría	Verbo
	Concordancia	Persona 3
		Número singular
	Tiempo	Presente
	Modo	Indicativo

Los *marcos* están compuestos por un conjunto de nodos y pares de atributo-valor en una red semántica que en su totalidad describe objetos, actos o acontecimientos. Se les puede comprender como una vista parcial de una red semántica; tienen valores por defecto y conexiones procedimentales.

6.14 La interactividad

La interactividad se enfoca en dos sentidos: uno hace referencia a la participación activa del individuo en situaciones y la influencia que esta ejerce sobre el individuo; el otro, hace alusión a la relación del operador humano con la tecnología.

En términos generales, la interactividad significa actuar mutuamente. Para ejecutar actos recíprocos se requiere cooperación, coordinación por parte de los sujetos, así como el poder y la influencia que ejerce el uno sobre el otro, además de algún grado de negociación sobre lo que el otro hará, cuándo y cómo.

En el aprendizaje, según Barker (1994), la interactividad es un mecanismo necesario y fundamental para adquirir conocimiento y desarrollar habilidades cognitivas y físicas. La implementación de interactividad requiere de habilidades, del entendimiento por parte del estudiante, de un diseño instruccional riguroso y de la aplicación de interfaces gráficas apropiadas.

La interactividad facilita la interacción de un individuo con el medio ambiente, el cual ejerce influencia sobre sus estructuras mentales, cambiando así su percepción. Según el constructivismo (Harrop, 2003), tanto la participación activa del individuo en una situación como la situación misma afectan al crecimiento cognitivo. Tal actividad de construcción, manifiesta el autor, se basa en la noción de Piaget (1972) de asimilación y acomodación: nuevos procedimientos se pueden asimilar antes del aprendizaje, pero el aprendizaje significativo ocurre solamente después de que se acomodan los nuevos esquemas; de esta manera, los nuevos conceptos se agregan a la red de los que ya existen. Pero si no ocurre la comprensión, entonces un individuo asimilará la información pero no ocurrirá el crecimiento cognitivo.

La psicología cognitiva, dice Harrop, ha basado este análisis en el concepto de modelo mental, los cuales se adquieren con la interacción. Según Johnson-Laird (1983, citado en Harrop, 2003), en la teoría de comprensión del lenguaje las personas construyen modelos mentales a partir de entidades reales (personas, objetos, eventos), los cuales se manipulan y se transforman mentalmente para hacer inferencias concernientes al lenguaje.

Según señala Harrop (2003), la comunicación necesita del uso de representaciones externas. Se puede asumir que: a) las representaciones internas son influenciadas y limitadas por situaciones externas (cognición situada): el conocimiento es inseparable del hacer y está ligado a los contextos físicos; y b) las conexiones entre las representaciones internas se pueden lograr a través de actividades externas, facilitando de esta manera la construcción de redes de conocimiento. Por eso, cuando se quiere resolver un problema en el dominio conceptual de los números —por ejemplo—, el razonamiento de una persona es guiado por sus modelos mentales internalizados, formados a través de interacciones previas en situaciones externas concretas.

La relación del operador humano con la tecnología se puede observar en la multimedia, que como tecnología educativa, se refiere al conjunto de medios (texto, audio, video), la tecnología (computador) y los productos como la información, los juegos y los programas educativos. Al integrar estos elementos es posible decir que existe interactividad, la cual puede enfocarse en la relación entre el operador humano y la tecnología. Estas instrucciones tecnológicas tratan de hacer que la interacción sea significativa y atractiva para el usuario.

La interactividad es el elemento que puede distinguir lo que se hace como tecnólogos de la educación de otros productos interactivos. Según Damarin (citado en Sims, 1997), las características que hacen interactivo a un *software* instruccional incluye una serie de opciones como: observar, encontrar, hacer, usar, construir y crear. Por su parte, Jonassen (citado en Sims, 1997), describe a la interactividad como una actividad entre el estudiante y una aplicación basada en el computador, donde el estudiante logra establecer en un diálogo verdadero con este último; la calidad de la interacción dependerá de la

naturaleza de la respuesta del estudiante y la retroalimentación por parte del computador. Si la respuesta es consistente con las necesidades de procesamiento de información del estudiante, entonces la interactividad es significativa.

La interactividad se puede visualizar, por tanto, como la función de entrada requerida por el estudiante para responderle al computador y analizar las respuestas y la naturaleza de la acción por parte del computador.

Schwier y Misanchuk (citados en Sims, 1997) identificaron diferentes niveles de interactividad basados en tres dimensiones: niveles (reactivo, proactivo, mutuo), funciones (confirmación, mediación, navegación, indagación y elaboración) y transacciones (teclado, pantalla, ratón y voz). Las funciones asociadas incluyen: verificación del aprendizaje (confirmación), control del aprendiz (medición), indagación por parte del estudiante y soporte del desempeño (indagación), control instruccional (navegación) y construcción del conocimiento (elaboración).

Cuando se desarrollan aplicaciones de multimedia debe hacerse énfasis en las maneras en que los usuarios pueden acceder, manipular y navegar a través del contenido material. A partir de esto se construyen 10 conceptos de interactividad basados en la propuesta de Sims (1997), los cuales pueden ser usados como guía de los diferentes modos de comunicación entre un computador y una persona:

- De objeto.
- Lineal.
- De apoyo.
- De actualización.
- De construcción.
- De reflexión.
- De simulación.
- De hipervínculo.
- No inmersa en el contexto.
- Virtual inmersa.

La *interactividad de objeto* (búsqueda proactiva) hace alusión a la aplicación en la cual los objetos (botones, personas, cosas) son activadas usando el ratón o cualquier dispositivo apuntador; cuando el usuario hace clic sobre el objeto se genera alguna respuesta en forma audiovisual.

La *interactividad lineal* (espaciado reactivo) se refiere a las aplicaciones donde el usuario se puede mover (hacia delante o hacia atrás) a través de una secuencia lineal predeterminada de material instruccional. Esta clase de interacción no facilita retroalimentación a la acción del aprendiz, sino que simplemente provee acceso a la pantalla siguiente o anterior en la secuencia.

La *interactividad de apoyo* es la facilidad que recibe el usuario en el soporte del desempeño, la cual puede variar desde mensajes simples hasta sistemas tutoriales complejos.

La *interactividad de actualización* se refiere a los componentes individuales de la aplicación o a eventos en los cuales se establece el diálogo entre el aprendiz y el contenido generado por el computador. La aplicación genera o presenta problemas ya sea de una base de datos o como una función de un nivel de desempeño individual al cual debe responder el aprendiz; la retroalimentación proveída por el computador resulta del análisis de la respuesta.

La *interactividad de construcción* (elaboración proactiva) es una extensión para actualizar la interactividad y requiere de la creación de un ambiente instruccional en el que se le pide al aprendiz manipular objetos para alcanzar metas específicas. Esta clase de interacción puede proveer un enlace entre el aprendizaje no situado y los ambientes simulados, enfrentándolo a acciones del mundo real.

La *interactividad de reflexión* (elaboración proactiva) se incluye para afrontar muchas situaciones en las cuales se requiere la utilización de respuestas de manera textual.

La *interactividad de simulación* extiende el papel del estudiante a ser controlador u operador, según lo cual las selecciones individuales determinan la secuencia del entrenamiento.

Las *interactividades de simulación y de construcción* están muy relacionadas y pueden necesitar que el estudiante complete una secuencia específica de tareas antes de que se produzca una actualización adecuada.

En la *interactividad de hipervínculo* (navegación proactiva) el estudiante tiene acceso a una riqueza de información y puede viajar en la base de conocimiento. Puede facilitar medios para presentar problemas que se resuelven correctamente, navegando por el grueso de la información.

El concepto de *interactividad no inmersa en el contexto* combina y se extiende a varios niveles de interactividad en un ambiente de entrenamiento virtual completo, en el cual el entrenado puede trabajar en un contexto relacionado de trabajo significativo. A través de una serie de secuencias es transportado en un micromundo que modela el ambiente de trabajo, y la tarea que se emprende refleja la experiencia de trabajo.

La *interactividad virtual inmersa* provee un ambiente en el cual el estudiante es proyectado en un mundo generado por el computador que responde a cada movimiento y acción.

Basado en estos conceptos, Sims (1997) propone un modelo de tres dimensiones donde la primera se refiere a la interactividad de navegación en la que el usuario se mueve de un lado a otro, o instruccional, la cual involucra el contenido que facilita el aprendizaje. La segunda dimensión, de control, se refiere a la extensión en la cual el sistema (control de programa) o el usuario (control del aprendiz) toman las decisiones instruccionales y de navegación. La tercera dimensión tiene que ver con el concepto de interactividad, que facilita una indicación del tipo de interacción que se espera dentro de las condiciones de variabilidad definida por el modelo.

6.15 Objetos de aprendizaje

Los objetos de aprendizaje son parte de los ambientes de aprendizaje y para su diseño se necesita de las ideas expuestas en este capítulo. Según el

Ministerio de Educación Nacional (Pontificia Universidad Javeriana, 2009), un objeto de aprendizaje es un conjunto de recursos digitales, autocontenible y reutilizable, con un propósito educativo y constituido por al menos tres componentes internos: contenidos, actividades de aprendizaje y elementos de contextualización. Además, el objeto de aprendizaje debe tener una estructura de información externa (metadato) para facilitar su almacenamiento, identificación y recuperación. De acuerdo con Chan (2001), los objetos de aprendizaje deben constituirse en un acervo para ser utilizados por muchos docentes.

Chan manifiesta que la información no es la que le otorga la connotación de aprendizaje a un objeto, sino el contenido de la instrucción dirigido a la persona que aprende. En consecuencia, para que un objeto de conocimiento se convierta en un objeto de aprendizaje se debe integrar con una estrategia instruccional, y cobrará sentido en cuanto responda a las necesidades de los usuarios.

Finalicemos este capítulo recordando que en él analizamos las diferentes técnicas computacionales que se pueden utilizar para la construcción de ambientes computacionales u objetos de aprendizaje y que tienen en cuenta los trabajos que descansan en investigaciones científicas, teniendo así mayor probabilidad de brindar resultados de aprendizaje en personas que persiguen el mejoramiento de la calidad de la educación.

Capítulo 7.

Aprendizaje de conceptos

El aprendizaje de conceptos se puede utilizar con dos finalidades. La primera de ellas tiene ver con el dominio disciplinar. De acuerdo con Jonassen (2006), este es uno de los indicadores que aumenta la probabilidad de éxito en la resolución de un problema. La resolución de problemas depende de las estrategias cognitivas que derivan de la parte conceptual del dominio disciplinar. La otra finalidad apunta hacia la informática, en donde se utiliza para elaborar un tipo de dato llamado clase. Con estas clases se construyen objetos que tienen las mismas reglas que los conceptos.

Según Day y Goldstone (2012), el conocimiento humano está constituido por un conjunto de símbolos discretos que representan los conceptos. Estos últimos son la base del razonamiento humano y el manejo de formación de concepto es una herramienta que facilita el aprendizaje de un dominio disciplinar. De acuerdo con Gulmans (1998), los conceptos ayudan a la categorización de los objetos cuando cumplen las reglas correspondientes, y permiten hacer inferencias y resolver problemas; si se tiene un concepto claro es más fácil discernir las estructuras profundas de los problemas.

7.1 Definición de concepto

Markle y Tiemann (1970 y 1971, citados en Dempsey, 1990) definen el concepto como una categoría que agrupa objetos, eventos, símbolos o relaciones que comparten características o propiedades, llamadas atributos. Los atributos describen las dimensiones que diferencian los objetos y los eventos; por tanto, la membrecía a una categoría se basa en los atributos físicos percibidos o por la definición. Klausmeier y Feldman (1975), por otra parte, consideran que un concepto se ha aprendido cuando el estudiante puede discriminar entre los atributos del concepto y evaluar nuevos ejemplos basados en la membrecía de la categoría del concepto.

Para Gulmans (1998) un concepto se define por uno o más atributos relacionados con una regla. La estructura mental mediante la cual un sujeto representa una categoría se llama concepto. De esta manera, un concepto es la representación mental de una categoría, que le permite a una persona asignar objetos a dicha categoría, si estos cumplen con la regla. El aprendizaje de un concepto implica separar los atributos relevantes de los que no lo son. Debido a que los conceptos están definidos tanto por los atributos relevantes como por las reglas de combinación, el aprendizaje de conceptos requiere el descubrimiento de los atributos y la regla.

7.2 Taxonomía de los conceptos

Para describir la relación entre los conceptos se utiliza una taxonomía: un concepto con un rango elevado en la taxonomía es el más general, mientras que los conceptos que están en los niveles más bajos son más específicos (Gulmans, 1998).

Rosch (1978, citado en Gulmans, 1998) afirma que para la categorización de objetos naturales se distinguen tres niveles: básico, superordinado y subordinado. Un concepto subordinado tiene siempre los atributos de los conceptos superordinados y también otros atributos; de esta manera, hay una herencia

de atributos de conceptos de niveles más altos hacia los niveles más bajos. Los conceptos que están al mismo nivel se llaman coordinados.

La organización de los conceptos en superordinados, básicos y subordinados se conoce como dimensión vertical de las categorías. La dimensión horizontal está formada por los conceptos coordinados. Los conceptos sucesivos se caracterizan por una relación vertical, de tal manera que los conceptos superordinados contienen a los conceptos subordinados. Los conceptos coordinados tienen atributos comunes y al menos un atributo distintivo (Gulmans, 1998).

7.3 Adquisición de conceptos

Gulmans (1998) señala que, para aprender los conceptos, los sujetos deben descubrir los atributos relevantes y las reglas que los relacionan; esto se logra aplicando los siguientes procesos: 1) descubrir los atributos relevantes; 2) descubrir las reglas que relacionan los atributos; 3) probar hipótesis, y 4) usar estrategias para determinar los procedimientos que se utilizan para aprender un concepto. La tarea consiste en poner atención a los atributos relevantes y descartar los que no son.

Los conceptos especifican las reglas que se siguen para categorizar los objetos; tienen instancias y estas pueden ser ejemplos o contraejemplos de un concepto. Un estudiante ha aprendido el concepto cuando puede identificar correctamente que un objeto o evento es miembro de una clase determinada.

Merryl y Tennyson (1978, citados en Tennyson & Cocchiarella, 1986) especificaron una serie de guías para enfatizar la enseñanza de conceptos, reduciendo la aparición de tres errores de clasificación: sobregeneralización, generalización por debajo y concepción errónea. Estas guías están centradas en la organización de cuatro variables de diseño instruccional para construir una estrategia de enseñanza:

1. Definición: una regla o generalidad que expresa verbalmente la estructura de los atributos críticos.

2. Instancias expositivas: ejemplos y contraejemplos que sistemáticamente organizan y presentan la información de contenido en un formato declarativo.
3. Instancias interrogativas: ejemplos y contraejemplos que sistemáticamente presentan el contenido en una formación de preguntas.
4. Elaboración de atributos: análisis de los atributos críticos en las instancias expositivas y retroalimentación de los atributos críticos en las instancias interrogativas.

Dempsey (1990) define la *sobregeneralización* como aquel error que se produce cuando un estudiante clasifica instancias que no pertenecen al concepto. Por su parte, la *generalización por debajo* se presenta cuando los estudiantes clasifican ejemplos de un concepto como contraejemplos; este error se relaciona con los conceptos sucesivos; es decir, es un error de generalización. Finalmente, las concepciones de errores incluyen ambos tipos de errores.

7.4 Fases del aprendizaje de conceptos

Tennyson y Cocchiarella (1986) probaron un modelo de aprendizaje de conceptos que incluye dos procesos cognitivos: la formación de conocimiento conceptual y el desarrollo del conocimiento procedimental. El conocimiento conceptual se forma en la memoria por medio del almacenamiento integrado de dimensiones significativas, seleccionadas de ejemplos conocidos y conectando estas entidades a un dominio dado de información.

Según estos autores, el conocimiento procedimental se desarrolla empleando el conocimiento conceptual para resolver problemas de un dominio específico. Estos dos procesos interactúan de tal manera que, con el uso del conocimiento procedimental en la solución de problemas, se elabora el conocimiento conceptual.

El modelo *aprendizaje de conceptos* propuesto por Tennyson y Cocchiarella (1986) está constituido por dos componentes de diseño: la estructura de contenido de un dominio dado de información y la organización de las variables de diseño instruccional relacionadas con el uso de estructuras de contenidos específicos. Por su parte, la estructura del contenido del dominio se debe analizar de acuerdo con dos condiciones, a saber: la relación estructural entre los conceptos del dominio y las características de los atributos de cada concepto en ese dominio (Tennyson & Cocchiarella, 1986).

Merrill y Tennyson (1978, citados en Tennyson & Cocchiarella, 1986) establecieron que la relación estructural de conceptos se puede basar en sus reglas de producción. Cuando se hace un análisis de contenido, las relaciones sucesivas y coordinadas determinan la estructura del dominio. Estas relaciones se definen con referencia a los procesos cognitivos de clasificación de conceptos, asociados a dos habilidades: generalización y discriminación. Con las relaciones sucesivas, el aprendizaje se limita al desarrollo de la generalización dentro de una clase del concepto; por su parte, con la relación coordinada, el aprendizaje incluye el desarrollo de habilidades para generalizar dentro de la clase del concepto y discriminar entre los conceptos. Por tanto, las estructuras de contenido de un dominio de información (que incluyen relaciones coordinadas) desarrollan primordialmente conocimiento procedimental, el cual mejora la conexión entre los conceptos (Tennyson & Cocchiarella, 1986).

En el análisis de contexto, los atributos pueden permanecer estables a través de los contextos (así que son denominados *conceptos de dimensión constante*) o pueden variar entre los contextos (los *conceptos de dimensión variable*). Para clasificar los conceptos como de dimensión constante o variable se debe analizar su perceptibilidad, lo cual tiene que ver con la definición del concepto. Así, si la definición no cambia con relación al contexto en el cual se aprende, el concepto tiene una dimensión constante; mientras que un concepto tendrá una dimensión variable si la definición y los ejemplos de este cambian de un contexto a otro (Gulmans, 1998).

La interacción entre la estructura relacional de los conceptos y las características de los atributos determina unas estrategias de diseño instruccional, las cuales están compuestas por un conjunto de variables de diseño instruccional (tabla 2). Las variables de diseño instruccional de rótulo y definición, de contexto y recuperación de información guardada, se asocian con el proceso de hacer conexiones en la memoria entre el conocimiento existente y los conceptos que se van a aprender. La variable de recuperación de información guardada se usa para ayudar al estudiante a hacer conexiones con información requerida que ya existe en la memoria.

TABLA 2. Modelo de enseñanza de aprendizaje basados en los conceptos de Tennyson y Cochiarella.

Características del atributo	Estructura relacional	
	Sucesivo	Coordinado
Dimensiones constantes	<p>Estrategia 1</p> <p>Rótulo y definición</p> <p>Mejor ejemplo</p> <p>Ejemplos expositivos (presentación sucesiva)</p> <p>Ejemplos interrogativos (opcional)</p> <p>Recuperación de información en guardada.</p>	<p>Estrategia 2</p> <p>Rótulo y definición</p> <p>Mejor ejemplo</p> <p>Ejemplos expositivos (presentación simultánea)</p> <p>Ejemplos interrogativos</p> <p>Elaboración de atributos</p> <p>Recuperación de información en guardada.</p>
Dimensiones variables	<p>Estrategia 3</p> <p>Rótulo y definición</p> <p>Contexto (dominio de conocimiento)</p> <p>Mejor ejemplo</p> <p>Ejemplos expositivos (presentación sucesiva)</p> <p>Ejemplos interrogativos</p> <p>Información de estrategias</p> <p>Recuperación de información en guardada</p>	<p>Estrategia 4</p> <p>Rótulos y definiciones</p> <p>Contexto (dominio de conocimiento)</p> <p>Mejor ejemplo</p> <p>Ejemplos expositivos (presentación simultánea)</p> <p>Ejemplos interrogativos</p> <p>Elaboración de atributos</p> <p>Información de estrategia</p> <p>Recuperación de información en guardada</p>

Las variables «mejor ejemplo» y «ejemplos expositivos» influyen directamente la comprensión inicial que tiene el estudiante del concepto. Las variables «ejemplos interrogativos», «información de estrategias» y «elaboración de atributos» tratan directamente con el desarrollo de reglas de producción para la recuperación de información en la memoria en la solución de problemas al nivel de habilidades intelectuales del aprendizaje.

Tennyson y Cocchiarella (1986) conciben al proceso de aprendizaje de conceptos como compuesto por tres partes: 1) establecimiento de una conexión en la memoria entre los conceptos que se van a aprender y un conocimiento específico necesario; 2) mejoramiento de la formación de conocimiento conceptual y elaboración de estructuras esquemáticas adicionales de conceptos relacionales; y 3) mejoramiento del desarrollo de la habilidad de conocimiento procedimental.

Volviendo al contenido de la tabla 2, la clasificación de conceptos en cuatro estrategias de diseño instruccional proporciona un medio para seleccionar las variables apropiadas que se ajusten a la estructura y las características del concepto que se da.

La primera estrategia, *sucesivo-constante*, sirve para enseñar conceptos sucesivos que tienen características de dimensiones constantes; esta requiere de una instrucción mínima, con cantidades de práctica que dependen de la experiencia del estudiante en los dominios dados de información.

La segunda estrategia, *coordinado-constante*, se utiliza para enseñar conceptos coordinados de características de dimensiones constantes, y requiere la presentación de secuencias simultáneas. Dado el proceso de discriminación adicional, se requieren ejemplos interrogativos con retroalimentación de atributos.

La tercera estrategia sugiere que los conceptos *sucesivos-variables* se aprenden mejor en contexto. Para ayudar al estudiante a entender la variabilidad de sus atributos críticos, la información de estrategias en ejemplos expositivos

e interrogativos puede incrementar la generalización del concepto a través del contexto.

En la cuarta estrategia, los conceptos presentados como *coordinados-variables*, requieren mayor instrucción para mejorar el aprendizaje porque son conceptos muy demandantes, lo cual se evidencia en las asignaturas que incluyen atributos críticos que se son variables y que se solapan, presentándole dificultades a los estudiantes en el desarrollo de la habilidad de la discriminación. Tennyson y Cocchiarella (1986) manifiestan que, por ejemplo, en el aprendizaje de la estadística podría ser más efectivo comenzar con un contexto específico para el cual el estudiante tenga un conocimiento de base adecuado. De allí, se le podría ayudar a desarrollar los conceptos estadísticos más complicados en la medida que van encontrando respuesta a las preguntas del dominio disciplinar.

Por su parte, Tennyson *et al.* (1981, citados en Tennyson & Cocchiarella, 1986) propusieron en un estudio la hipótesis de que el aprendizaje de conceptos incluye, además de la formación de un conocimiento conceptual en la memoria por medio de la exposición del mejor ejemplo, el desarrollo de un conocimiento procedimental a través de la práctica de comparación y el contraste del conocimiento conceptual utilizando nuevas instancias interrogativas. Dicha hipótesis se probó contrastando tres maneras de presentación: una forma expositiva, en forma interrogativa y en una combinación de forma expositiva-interrogativa (figuras 12 y 13). En el experimento los estudiantes asignados a cada una de las tres formas recibieron el mismo entrenamiento y la definición del concepto de polígono.

Los sujetos en la forma expositiva recibieron una exposición con la presentación de ejemplos y contraejemplos, con la expectativa de formación de conocimiento conceptual y un mínimo de conocimiento procedimental. En la forma interrogativa recibieron una presentación de ejemplos y contraejemplos con la expectativa de desarrollar conocimiento procedimental, pero mínimo conocimiento conceptual. Finalmente, en la tercera condición, con ejemplos

y contraejemplos, se esperaba la formación de conocimiento conceptual y procedimental.

La adquisición del concepto se midió con cuatro niveles de desarrollo: concreto, recuerdo de los atributos físicos (identidad, recuerdo de ejemplos aprendidos); clasificatorio (generalización a nuevos ejemplos) y formal (discriminación entre nuevas instancias encontradas) (Klausmeier *et al.*, 1974, citados en Tennyson y Cocchiarella, 1986). En el postest se encontró que los tres grupos lograron formar el concepto al nivel concreto y de identidad; sin embargo, el tercer grupo obtuvo mejores notas que los otros grupos en los niveles formales y de clasificación.

FIGURA 12. Ejemplos de presentación de formas expositivas

Mire cuidadosamente todas estas figuras.

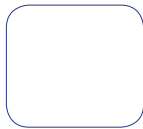


Figura A. No es un círculo porque no todos los puntos del exterior de la figura equidistan del centro.

Un círculo es una curva cerrada cuyos puntos equidistan del centro.

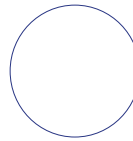


Figura B. Es un círculo.

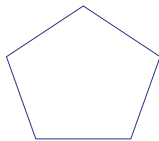


Figura C. No es un círculo porque tiene 5 lados. Los círculos tienen infinitos lados.

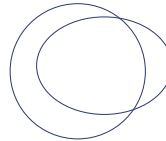
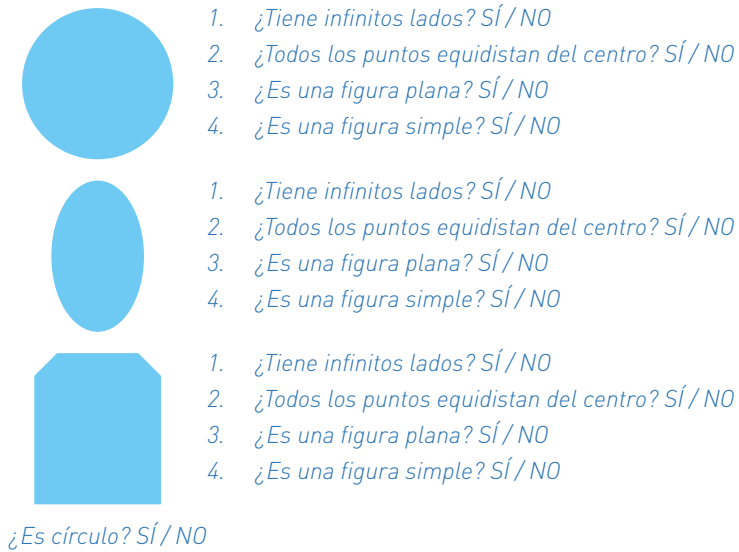
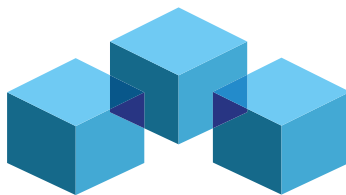


Figura D. No es un círculo porque es una figura compleja. Los círculos son simples

Verá ahora algunas figuras. Su trabajo consistirá en decir si la figura es un círculo. Cada vez que vea una figura, hágase las siguientes cuatro preguntas. Si su respuesta a todas las preguntas es sí, la figura es un círculo. Si la respuesta es no a algunas de las preguntas, la figura no círculo.

FIGURA 13. Ejemplo de presentación de una forma interrogativa

En este capítulo analizamos la definición de concepto, su formación y atributos, así como algunas estrategias para su enseñanza. Este es un tema importante tanto para su utilización en el aprendizaje de un dominio disciplinar, como para la comprensión del concepto de clase que se utiliza en programación.



Capítulo 8.

La transferencia del aprendizaje

De acuerdo con Perkins y Salomon (1992), los fines de la educación no se logran a menos que ocurra la transferencia; esto es, cuando un aprendizaje adquirido en un determinado contexto logre influir en el desempeño mostrado en otro. Afirman estos autores que el contexto final de aplicación debe ser diferente del contexto de aprendizaje, lo que dificulta en gran medida la transferencia; no obstante, existen variables que se pueden tomar en cuenta para aumentar su ocurrencia. A continuación haremos un desarrollo de la idea de transferencia utilizada por Otero en su investigación (2014b).

8.1 Base de la transferencia

Según el enfoque cognitivo tradicional, manifiestan Day y Goldstone (2012), el conocimiento humano se representa por medio de un sistema de símbolos discretos que corresponden a los conceptos. Para representar una situación, estos símbolos se combinan través de una sintaxis estructural que define las relaciones entre los conceptos constituyentes. Se tiene entonces que cualquier situación está compuesta de objetos y características, así como de sistemas de relaciones.

Los seres humanos tienen la capacidad de representar los sistemas de relaciones independientemente de los objetos y de sus características, lo que permite pensar analógicamente; de acuerdo con Day y Goldstone (2012), la analogía es un apareamiento de los sistemas de relaciones entre dos situaciones (al comparar sus estructuras profundas), sin importar sus características superficiales, que son los objetos y características presentes (Day & Goldstone, 2012).

El conocimiento de la similitud estructural representa la base para la transferencia significativa del conocimiento. Así, los procesos de transferencia requieren que el estudiante abstraiga o comprenda la estructura profunda en el primer problema, y luego que reconozca esta estructura profunda en el segundo problema, para que se pueda aplicar el mismo procedimiento (Chi & VanLehn, 2012).

Aunque algunos psicólogos cognitivos consideran que el componente crítico en la transferencia productiva del conocimiento es la similitud estructural, investigaciones han mostrado que las que ocurren a menudo son aquellas que recurren a revisar las características superficiales comunes. Por tanto, la gran dificultad cognitiva radica en notar la similitud estructural entre un caso nuevo y otro encontrado previamente (Day & Goldstone, 2012).

Existen algunas condiciones que pueden facilitar la identificación de las estructuras profundas de un problema a otro; entre ellas está el conocimiento anterior o conocimiento inicial que una persona posea, lo que Day y Goldstone (2012) explican en términos de la experticia. Estos autores traen a colación el trabajo de Chi, Feltovich y Glaser (1981), quienes examinaron la diferencia entre los expertos y los novatos en el dominio de la física. Al respecto, Day y Goldstone (2012) manifiestan que los expertos agrupan los problemas por los principios generales que subyacen a la solución y que, en cambio, los novatos los clasifican según sus características concretas; asimismo, señalan que de un ancho rango de dominios para evaluar similitudes, los expertos utilizan la similitud estructural y los novatos las características superficiales.

8.2 Tipos de transferencia

Perkins y Salomon (1998) plantean dos tipos de transferencia: la cercana y la lejana. La transferencia cercana se refiere a aquella dada entre contextos que tienen características superficiales similares. La transferencia lejana se refiere a la que se da entre contextos que parecen muy remotos o extraños entre sí y que, por ende, difieren mucho en sus características superficiales, aunque tengan estructuras profundas parecidas.

Para estos autores la transferencia ocurre por dos mecanismos: la transferencia reflexiva o de bajo camino y la transferencia de alto camino. La transferencia de bajo camino ocurre cuando las condiciones entre los contextos son tan similares que disparan respuestas automáticas bien desarrolladas, siendo así un proceso reflexivo que conduce a la transferencia cercana. La transferencia de alto camino, por su parte, depende de la abstracción consciente del contexto de aprendizaje y de una búsqueda deliberada de conexiones. Requiere tiempo para la exploración y la utilización de un esfuerzo mental, y puede conducir fácilmente a la transferencia lejana.

Perkins y Salomon (2012) discuten si debe considerarse una situación como un fracaso de transferencia o como un fracaso de aprendizaje inicial. Sobre el asunto citan a Chi y VanLehn (2012), quienes afirman que el fracaso de la transferencia se debe a que el estudiante no logra una comprensión lo suficientemente profunda del contexto. Perkins y Salomon, a su vez, comentan que los investigadores no se ponen de acuerdo al respecto y que se necesitan más investigaciones empíricas para clarificar este caso.

8.3 La transferencia orientada al actor

Otro punto de vista sobre la transferencia lo ofrece Lobato (2012), quien en su trabajo acerca de la perspectiva de la transferencia orientada al actor (AOT), manifiesta que la contribución aportada por la perspectiva cognitiva es la consideración de las estructuras cognitivas: la memoria de corto y largo plazo, junto con la memoria sensorial; las representaciones como estructuras de símbolos

mentales que codifican, procesan y almacenan las experiencias, y un mecanismo de control que supervisa la recuperación y la utilización de la información. La propuesta de Lobato emerge como respuesta a las críticas de la postura cognitiva sobre la transferencia, desafiando sus presupuestos epistemológicos.

Desde la perspectiva cognitiva tradicional, manifiesta Lobato, la transferencia se refiere a un conocimiento adquirido en una tarea o situación que puede ser aplicado a otra diferente. Por su parte, desde el punto de vista de la AOT, la transferencia se define como la generalización del aprendizaje, es decir, la influencia de las actividades previas del estudiante sobre otras nuevas. En su escrito, esta autora menciona cinco dimensiones de divergencia entre las perspectivas: la naturaleza del saber y la representación, los puntos de vista, lo que se transfiere, los métodos y las metas.

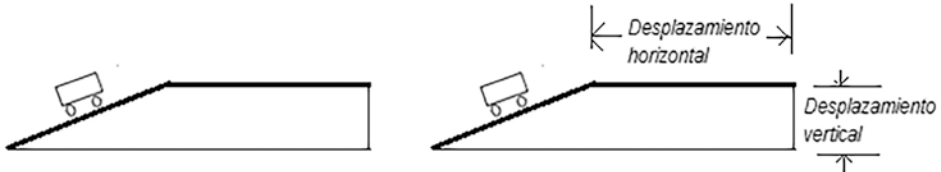
La perspectiva AOT hace mayor énfasis en la naturaleza interpretativa de las situaciones que la perspectiva cognitiva, pues en ella se buscan las maneras con las cuales el estudiante trata las situaciones de transferencia como instancias de algo que ya había pensado, basado en su interpretación y en la comprensión del significado de las actividades y eventos de la situación de aprendizaje inicial. Por el contrario, en la perspectiva cognitiva hay una correspondencia más cercana entre los objetos del mundo y la representación mental.

En cuanto al punto de vista, Lobato indica que hay una distinción entre el observador de la perspectiva cognitiva y el del actor de la perspectiva AOT. El primer punto de vista implica la predeterminación de una estrategia particular, principio o heurística que el estudiante necesita demostrar para que su trabajo sea reconocido como una transferencia a una situación nueva. En el punto de vista del actor, el investigador no mide la transferencia comparándola con una meta cognitiva particular, sino que investiga instancias en las cuales las experiencias anteriores de los estudiantes le dan forma a sus actividades en la situación de transferencia, sin importar que el resultado no sea normativo o su desempeño sea correcto.

Lobato trae a colación un estudio en el que los estudiantes debían, inicialmente, encontrar la pendiente de objetos como escaleras y líneas. La pendiente

se define como el cociente entre el desplazamiento vertical y el desplazamiento horizontal. Posteriormente, se les solicitó, como tarea de transferencia, encontrar la pendiente de un deslizadero. La figura 14 muestra una solución parecida a la que dieron los estudiantes.

FIGURA 14. *Determinación de la pendiente*



De acuerdo con Lobato, los estudiantes recordaron bien la fórmula en la nueva situación: desplazamiento horizontal sobre desplazamiento vertical; pero la interpretación de la situación no fue correcta. Así que lo que demuestra el ejercicio es la carencia de un conocimiento más exacto del concepto, pero evidencia también, en cierta medida, la transferencia de los conocimientos que estos poseen.

En la perspectiva cognitiva tradicional, según Lobato, no se utiliza en la transferencia facultades generales de solución de problemas; se transfieren conocimientos específicos relacionados con el contexto. Por otra parte, en la perspectiva AOT se transfieren acciones y estrategias bien definidas, mientras que en la perspectiva cognitiva se transfieren conceptualizaciones más holísticas.

Como ejemplo de lo anterior se tiene un caso presentado por Lobato, quien lo tomó a su vez de un estudio efectuado por Thompson (2011), en el que se le pide a un estudiante de sexto grado encontrar el tiempo que se demora una tortuga (una figura de un programa de computador) en recorrer 200 cm si va 25 cm/s. El estudiante comienza dibujando líneas sucesivas de 25 cm cada una hasta alcanzar los 200 cm; luego las cuenta y, finalmente, da su respuesta. Seguidamente, se le hace la siguiente pregunta: ¿a qué velocidad debe viajar un conejo de tal manera que recorra 200 cm en 7 segundos? Como el estudiante

no pudo dibujar segmentos optó por una estrategia de ensayo y error, probando diferentes velocidades hasta encontrar la solución correcta.

Según Thompson (citado en Lobato), el primer episodio se puede considerar como una instancia de transferencia desde el punto de vista de elementos comunes. Pero la segunda tarea no se puede contar como transferencia bajo la perspectiva mencionada, porque el estudiante no aplicó el mismo método de solución. No obstante, desde la AOT se considera que sí hubo transferencia completa, porque se transfirió el concepto que tenía el estudiante de velocidad, tiempo y distancia; el estudiante consideró que la situación era similar en ambos casos y, en el segundo episodio, llegó a la solución tomando en consideración la idea de segmento, hasta llegar al tamaño requerido.

En lo que se refiere a los métodos —manifiesta Lobato (2012)—, en la corriente cognitiva tradicional a los sujetos se les enseña una solución en la situación de conocimiento inicial y luego se les pide resolver una situación con estructuras profundas similares, pero con diferentes características superficiales. Luego, el desempeño del grupo experimental se compara con un grupo de control: si el grupo control obtiene un mejor desempeño que el grupo experimental se llega a la conclusión de que hubo transferencia.

La perspectiva AOT descansa en métodos cualitativos para identificar la naturaleza del razonamiento del estudiante en la situación de transferencia y de la comprensión de las actividades previas de aprendizaje. Lobato expone métodos en los que se utilizan entrevistas o configuraciones instruccionales donde el estudiante posee conocimientos limitados sobre el contenido que se va a aprender, para aislar la experiencia que influencia el razonamiento del participante en las tareas de transferencia.

En relación con el tema de las metas, Lobato señala que la perspectiva cognitiva busca documentar la ocurrencia de la transferencia, lo cual incluye el tipo de conocimiento que se transfiere mejor, las condiciones que impiden o promueven la transferencia y los métodos instruccionales que apoyan la transferencia. La meta en la perspectiva AOT no es obtener transferencia sino

considerar la naturaleza interpretativa de las conexiones que la gente construye entre el aprendizaje y la situación de transferencia, como también los procesos sociales situados que dan lugar a estas conexiones.

Con el anterior trabajo de Lobato hemos querido mostrar un camino más que se puede explorar a través de la investigación para determinar su utilidad en la construcción de material instruccional. La finalidad de esta obra no es presentar una postura frente a las diferentes posiciones que asuman los autores, sino de tomar sus ideas y materializarlas en la construcción de materiales instruccionales que aumenten la probabilidad de éxito del proceso enseñanza-aprendizaje.

Por otra parte, en la investigación realizada por Otero (2014b) sobre el efecto que tiene —en el proceso de transferencia— la visibilidad de las estructuras profundas de los problemas en la competencia de resolución de problemas, se formaron dos grupos experimentales: uno trabajó con un objeto de aprendizaje que mostraba gráficamente las estructuras profundas de los problemas y otro de control, el cual fue sometido al objeto de aprendizaje con retroalimentación del programa. El estudio se clasificó dentro de los diseños cuasiexperimentales unifactoriales, univariantes con un diseño pretest-posttest.

En la ejecución de las actividades se aplicó a los estudiantes participantes de los grupos una prueba de entrada para evaluar los conocimientos sobre resolución de problemas cuyas variables presentan una relación lineal, lo cual permitió establecer el conocimiento inicial de los grupos. Luego se aplicaron los objetos de aprendizaje y, finalmente, se hizo otra prueba, para verificar la influencia del objeto aplicado y establecer la diferencia de desempeño entre los grupos.

La variable independiente la constituyeron los dos objetos de aprendizaje, caracterizados uno por utilizar la visibilidad de las estructuras profundas para lograr la transferencia de conocimiento y, el otro, por el manejo del control de la retroalimentación por parte de estudiante. La variable dependiente la constituyó el desempeño inferido por la calificación obtenida en las evaluaciones.

La prueba final puso al descubierto que los participantes no lograron transferir las estructuras profundas de los problemas ejemplos a los problemas presentados en la evaluación final; dicha transferencia no se dio ni en el objeto de aprendizaje donde se visibilizan las estructuras profundas, ni tampoco en el objeto de aprendizaje con control de retroalimentación por parte del sistema.

De acuerdo con los resultados del contraste de hipótesis para pruebas no paramétricas se evidenció la falta de construcción de esquemas sobre la parte disciplinar, lo cual no les permitió identificar a los participantes las variables relevantes ni sus funciones en los problemas del postest y, como consecuencia, no pudieron construir los modelos matemáticos correspondientes, a pesar de haberse provisto dentro de los objetos de aprendizaje la teoría y los ejercicios de práctica sobre el tema utilizado en la investigación.

Una de las posibles causas del fracaso de transferencia (dentro de las muchas existentes) fue tal vez la falta de un conocimiento inicial bien cimentado, pues a pesar de haberse asimilado la información durante el experimento, en el momento de la evaluación una semana más tarde, este ya se había olvidado. Otra causa probable —relacionada con la anterior— fue la cantidad de información en el tiempo previsto para los talleres; posiblemente esta superó la capacidad de memoria a corto plazo de los participantes, impidiendo su comprensión y su posterior inclusión en la memoria a largo plazo.

De lo anterior se deriva que, por no tener un conocimiento inicial bien establecido para seguir los conceptos, se produjo la imposibilidad de su acomodación, lo cual impidió durante el experimento construir esquemas lo suficientemente expansivos para enfrentar los problemas propuestos.

De este trabajo se puede concluir que lograr la transferencia es una tarea difícil, y que para aumentar la probabilidad de éxito en su consecución quien aprende debe tener un conocimiento inicial bien cimentado. Asimismo, se debe utilizar una gran cantidad y variedad de problemas con características superficiales diferentes donde se evidencien sus estructuras profundas comunes.

Capítulo 9.

Diseño de un objeto de aprendizaje

A continuación se presenta el diseño² de un objeto de aprendizaje efectuado por Otero (2013) y en él confluyen los temas que se han venido trabajando. De la psicología del aprendizaje se utiliza el concepto de transferencia para desarrollar el contenido del dominio disciplinar que corresponde a un tema de álgebra lineal. Primero se explican los conceptos y luego aparecen diferentes ejemplos con características superficiales diferentes; mediante una lista de texto y un gráfico se demuestra que tienen las mismas estructuras profundas y, por lo tanto, se pueden resolver con el mismo procedimiento. De la psicología del aprendizaje se utiliza también, en las actividades de aprendizaje, la retroalimentación y el análisis del error, lo cual permite que el estudiante tome conciencia sobre sus concepciones erróneas.

El otro tema utilizado fue el de los ambientes de aprendizaje. De aquí se tuvo en cuenta los fundamentos de diseño expuestos por Groner y Kersten (2001), por un lado, y el análisis de la tarea expuesto por Narciss (2001), por el otro. De los ambientes de aprendizaje se utilizó el fundamento de los tutores inteligentes que constan de la interfaz, el módulo de enseñanza, el módulo

2 La palabra 'diseño' en este contexto se utiliza en el sentido de la programación de computadores.

de aprendizaje y el módulo de enseñanza. El análisis de tarea de Narciss se empleó para hacer la fase de análisis del objeto de aprendizaje. El tercer tema que se puso en acción fue la programación en HTML, Javascript y PHP.

En consecuencia, y como podemos observar, la finalidad de este capítulo es mostrar cómo se aplican los principios anteriormente explorados en la elaboración de ambientes de aprendizaje o en la creación de objetos de aprendizaje. De esta forma se podrían generar objetos que obedezcan a principios evidenciados en investigaciones y no formulaciones fundamentadas en criterios personales, que desde luego no han sido probados científicamente. A continuación se muestra el diseño efectuado por Otero (2013).

9.1 Diseño

En lo que sigue aparece el análisis del objeto de aprendizaje, aplicando las ideas que propone Narciss para tal fin. La numeración que se observa corresponde a los pasos del análisis.

1. Dominio de conocimiento: Álgebra lineal.

Objetivo: Teoría y resolución de problemas que involucran las formas $y = mx + a$

2. Elementos específicos del contenido:

- Teoría
 - ~ Concepto de variables.
 - ~ Relación entre variables.
 - ~ Variable dependiente.
 - ~ Variable independiente.
 - ~ Proporcionalidad.
 - ~ Solución de problemas de la forma $y = mx + a$.
 - ~ Despeje de la variable independiente.

- ~ Solución de problemas de la forma $y = m_1 x_1 + m_2 x_2$
- ~ Solución de la forma:

$$y_1 = m_1 x_1 + m_2 x_2$$

$$y_2 = m_3 x_1 + m_4 x_2$$

3. Operaciones cognitivas asociadas con la unidad de conocimiento:

Tema: Variable.

Identificar en un ejemplo las variables relevantes de un problema.

Tema: Variables directamente proporcionales.

Comprender el concepto de variables directamente proporcionales.

Tema: Variable dependiente.

Identificar la variable dependiente.

Tema: Variable independiente.

Identificar la variable independiente.

Tema: Constante de proporcionalidad.

Inferir la constante de proporcionalidad.

4. Matriz de unidades de conocimiento versus operaciones cognitivas

TABLA 3. Matriz de unidades de conocimiento versus operaciones cognitivas.

Operaciones cognitivas \ Unidades de conocimiento	Var.	R en Var.	Dep.	Ind.	CP
Comprender		x			
Identificar	x		x	x	
Comparar					
Inferir					x

Nota: Var.: variable; R en Var.: relaciones entre variables; Dep.: dependiente; Ind.: independiente; CP: constante de proporcionalidad.

5. Actividad de aprendizaje para cada celda de la matriz:

- **Actividad de aprendizaje 1:** identificar en un contexto problemático las variables relevantes de un problema.
- **Actividad de aprendizaje 2:** identificar en un contexto problemático el tipo de relación entre las variables.
- **Actividad de aprendizaje 3:** identificar en un contexto problemático la variable dependiente de un problema.
- **Actividad de aprendizaje 4:** identificar en un contexto problemático la variable independiente de un problema.
- **Actividad de aprendizaje 5:** inferir, a partir de una situación problemática, las unidades de la constante de proporcionalidad.

6. Identificación y clasificación de factores posibles, concepciones erróneas o fuentes que pueden causar un error o estrategias ineficientes para cada tipo de actividad:

- **Actividad de aprendizaje 1:** falta un análisis más profundo del contenido de la pregunta; se identifican variables que no son relevantes en la definición del problema.
- **Actividad de aprendizaje 2:** no se identifican las variables tienen una relación directamente proporcional.
- **Actividad de aprendizaje 3:** no interpreta la función de las variables dentro del problema.
- **Actividad de aprendizaje 4:** no interpreta la función de las variables dentro del problema.
- No identifica las variables dependiente e independiente; por tanto, no puede formar la razón variable dependiente/variable independiente.

7. Información contenida en los mensajes de retroalimentación: En los errores de identificación se muestran mensajes que invitan al usuario a leer

detenidamente, de manera que se focalice donde se hace mención a la aparición de las variables dependientes e independientes de los problemas (la variable dependiente es el predicado de una oración, la variable independiente el sujeto). Ejemplo: en la oración «un trabajador fabrica veinte piezas en una hora», *trabajador* es la variable independiente y *pieza* es la variable dependiente. Para formar la constante de proporcionalidad se debe tener en cuenta que esta es una proporción que muestra la cantidad de variables dependientes por cada unidad de variables independientes.

9.2 Componentes

El objeto de aprendizaje consta de una base de datos y de las funciones que constituyen el cuerpo de objeto. La base de datos se presenta en la figura 15, mientras que las clases de funciones en la 16.

FIGURA 15. Tablas de la base de datos PLINKI.

Preguntas	Temas	Problemas
Codigo_preg	Codigo_tem	Codigo_pro
Nom_Var_preg	Nombre_tem	Contenido_pro
Opciones_preg	Contenido_tem	Nombre_pro
Retro_preg	Codigo_niv	DescripcionVasr_pro
Clave_preg		Tema_pro
Contenido_preg	Estudiantes	Codigo_niv
Codigo_act	Codigo_est	
Codigo_tem	Nombre_est	TeoProblema
Codigo_niv	Apellido_est	Codig_teo
	Objeto_est	Contenido_teo
	Codigo_niv	Codigo_niv

FIGURA 16. Clases de un objeto de aprendizaje

```

var tema = {
    pregunta:"",
    teoria:"",
    opciones:"",
    clave:"",
    nomVar:"",
    retro:""
};

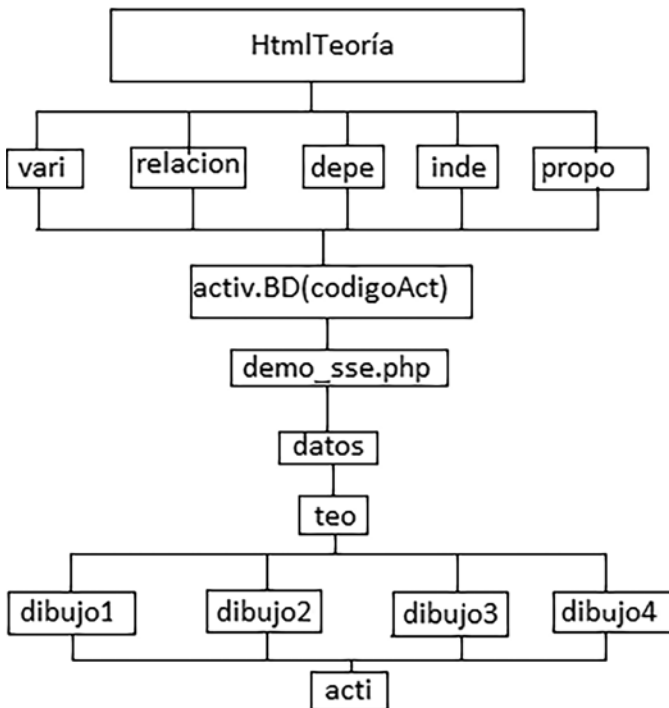
var opt = {
    opcion1: "",
    opcion2: "",
    opcion3: "",
    opcion4: "",
    .
    .
    .
    opcionn:""
};

```

La base de datos se maneja con funciones efectuadas en PHP y el resto del objeto de aprendizaje con jQuery y JavaScript. El objeto de aprendizaje en sí consta de dos partes: la fundamentación teórica (figura 17) —acompañada de actividades de aprendizaje para permitir la retención del conocimiento—, y su modelación matemática.

El diagrama de la sección teórica aparece en la figura 17. La función *Html-Teoria* llama a cada una de las funciones, que pasan como parámetro el tema a tratar a la función PHP, que carga la base de datos. Según el tema, se llaman las funciones que dibujan estructuras profundas de los conceptos teóricos, en los que se proporciona la definición del concepto, seguidos luego por las actividades de aprendizaje.

FIGURA 17. Diagrama de la sección teórica



En la figura 18 se ilustra el menú que conduce a cada uno de los temas.

FIGURA 18. Menú de la parte teórica

Variables	Relación entre Variables	Variable Dependiente	Variable Independiente	Proporcionalidad
-----------	--------------------------------	-------------------------	---------------------------	------------------

En la casilla «Dibujo 1» (variables) se maneja el tema de variables que se presenta a continuación (figura 19).

FIGURA 19. Concepto de variable

Contextualización	Base teórica	Modelación matemática
<p>Variables.</p> <p>Magnitud que puede recibir diferentes valores. En este objeto de aprendizaje las variables se representan con una sola letra.</p> <p>Ejemplo.</p> <p>En el problema, María compró cierta cantidad de manzanas, cada una a \$ 2000 y se desea saber cuánto gastó en las manzanas; cantidad de manzanas es una variable y el gasto en las manzanas es otra variable.</p> <div data-bbox="221 809 1108 1056" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> </div>	<p>Relación entre Variables</p> <p>Variable Dependiente</p>	<p>Variable Independiente</p> <p>Proporcionalidad</p>
<p>Haga clic hacer para actividad de aprendizaje</p>		

El objetivo de la actividad es mostrar que, independientemente del contexto del problema, todos los de este tipo se pueden representar de acuerdo con la gráfica. La figura 20 muestra unas palabras en los rectángulos grises, los cuales son cajas arrastrables que se pueden transportar hasta el rectángulo donde está la palabra ‘carpintero’. La retroalimentación se produce según la opción que elija el usuario.

FIGURA 20. Actividad de aprendizaje 1

En el siguiente problema: Un carpintero produce en su casa, 20 sillas en una hora, para una empresa ubicada en Bogotá y registrada en la cámara de comercio de Usaquén. Se desea saber cuántas sillas se pueden producir en un número determinado de horas. Determine cuál grupo de palabras constituyen las variables que definen el problema

Escoja con el ratón la opción correcta y llévela a la figura.

Carpintero

Variables

Carpintero, casa.

Sillas, horas.

Empresa, Bogotá.

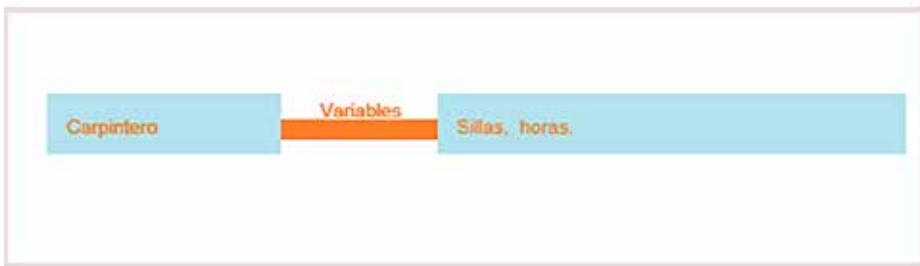
Cámara de comercio, Usaquén.

Retroalimentación actividad de aprendizaje 1. Para la retroalimentación se utilizan reglas de producción que comparan la respuesta dada con la respuesta correcta contenida en el programa.

Cuando se produce un error:

Sí [respuesta dada igual a Sillas, horas].

Perfecto, mire la figura (figura 21).

FIGURA 21. Respuesta correcta actividad de aprendizaje 1

Sí no:

{No es la opción correcta. Observe que m , n no están contenidas en la pregunta del problema. Lea con atención y localice la pregunta del problema, esta contiene las variables del problema.
}

Comentarios. Es una retroalimentación relacionada con el error, la cual posee una especificidad moderada; no da la respuesta pero sugiere donde la puede encontrar dentro del problema dado. Se utilizan los tipos de retroalimentación de la forma conocimiento de resultado y de múltiples intentos; por su parte, las letras m y n representan alguna de las otras opciones que no son correctas.

En la figura 22 aparece una explicación y un ejemplo de lo que es una relación directamente proporcional

FIGURA 22. Relación entre variables


<u>Contextualización</u>	<u>Base teórica</u>	<u>Modelación matemática</u>
	<p>Variables Relación Variable Variable Proporcionalidad entre Dependiente Independiente Variables</p>	
<p>Dos variables tienen una relación directamente proporcional cuando al aumentar una de ellas, la otra también se incrementa. Se representa por α. La expresión: $y = \alpha x$ Se lee y es directamente proporcional a x. Ejemplo. En el ejemplo anterior María compró cierta cantidad de manzanas, cada una a \$ 2000 y se desea saber cuánto gastó en las manzanas. Se puede observar que a medida que se compran más manzanas hay que pagar más dinero, por lo tanto, son variables directamente proporcionales. A más manzanas compradas más dinero se debe pagar: Relación directamente proporcional.</p>		
<p>entre más Manzanas — más — Pago</p> <p>6 1200</p>		
<p>Haga clic hacer para actividad de aprendizaje</p>		

Actividad de aprendizaje 2. Comprender el concepto de variables directamente proporcionales (figura 23).

FIGURA 23. Actividad de aprendizaje 2

Determine cuál de las variables tiene una relación directamente proporcional:

Escoja con el ratón la opción correcta y llévela a la figura.



entre más — más

4 — 800

Cantidad de trabajadores y tiempo para realizar una obra.

Cantidad de trabajadores y cantidad de obra realizada.

Tamaño de baldosas y cantidad de baldosas para embaldosar un piso.

Velocidad y tiempo de un tren para hacer el mismo recorrido.

Retroalimentación actividad de aprendizaje 2. Si se produce un error al escoger la opción que no es correcta:

Sí (respuesta correcta)

Muy bien, entre más trabajadores más cantidad de obra realizarán.

Mire la figura 24:

FIGURA 24. Respuesta correcta actividad de aprendizaje 2

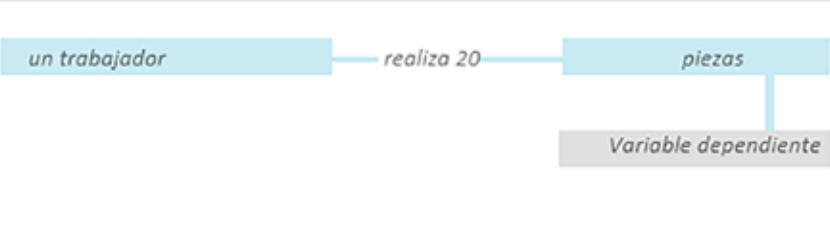


Si no:

No es la opción correcta. Observe que entre más m, menos n, lo cual no define una relación directamente proporcional. Observe la figura a mano izquierda.

En la figura 25 aparece la definición de la variable dependiente y un ejemplo que señala cómo se puede identificar la variable dependiente dentro de un problema.

FIGURA 25. Variable dependiente


<u>Contextualización</u>	<u>Base teórica</u>	<u>Modelación matemática</u>		
Variables	Relación entre Variables	Variable Dependiente	Variable Independiente	Proporcionalidad
Variable dependiente				
Es aquella cuyo valor depende del valor que asuma la variable independiente. Es consecuencia de un fenómeno o proceso.				
Ejemplo 1.				
Un trabajador realiza 20 piezas, un determinado número de trabajadores ¿Cuántas piezas producirán?				
Solución:				
La variable dependiente son las piezas a producir. Estas son consecuencia de la actividad del trabajador.				
 <pre> graph LR A[un trabajador] --- B[realiza 20] B --- C[piezas] C --- D[Variable dependiente] </pre>				
Haga clic hacer para actividad de aprendizaje				

Actividad de aprendizaje 3. Variable dependiente (figura 26). En esta actividad se debe identificar la variable dependiente por parte del estudiante.

FIGURA 26. Actividad de aprendizaje 3

En el siguiente problema: La densidad del aceite es de 920 kilogramos por cada metro cúbico. Se desea saber la masa de un volumen dado de aceite, la variable dependiente es:

Escoja con el ratón la opción correcta y llévela a la figura.



La masa.

El volumen.

La densidad.

Masa / volumen.

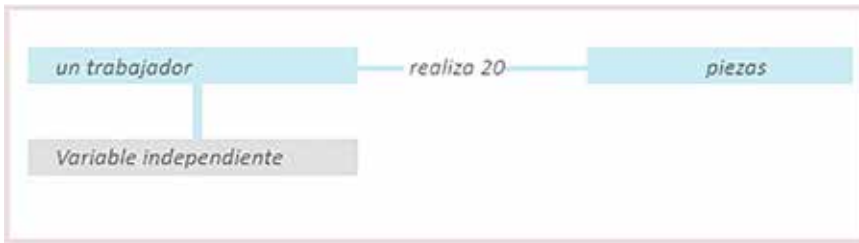
Retroalimentación actividad de aprendizaje 3. Si se produce el error:

La variable m no es la opción correcta. La variable cuyo valor se quiere saber aparece en la pregunta. Construya una frase con ella que involucre la otra variable con la cual se relaciona. El predicado de la oración (lo que se dice del sujeto) es la variable dependiente. Mire el recuadro del ejemplo que aparece a la izquierda:

Si no

Muy bien, observe la figura 27.

FIGURA 27. Respuesta correcta actividad de aprendizaje 3



En la figura 28 se muestra el concepto de variable independiente y un ejemplo de cómo identificarla en un contexto problemático.

FIGURA 28. Variable independiente


<u>Contextualización</u>	<u>Base teórica</u>	<u>Modelación matemática</u>
Variables	Relación entre Variables	Variable Dependiente Variable Independiente Proporcionalidad
<p>Variable independiente</p> <p>Es la que hace posible la consecución del valor de la variable dependiente. Es la que causa el fenómeno. Cuando se reemplaza el valor de la variable independiente en la ecuación se obtiene la variable dependiente.</p> <p>Ejemplo 2.</p> <p>En el problema del trabajador y las piezas, la variable independiente es el trabajador, es quien elabora las piezas, Su definición y descripción está dada por:</p> <p>x: Cantidad de trabajadores empleados para producir las piezas.</p>		
<p>Haga clic hacer para actividad de aprendizaje</p>		

Actividad de aprendizaje 4. Identificar la variable independiente en un contexto problemático (figura 29).

FIGURA 29. La variable independiente en un contexto problemático

En el problema siguiente: Si la rebaja en un artículo es de \$200.000. ¿Cual será la rebaja en cualquier cantidad de artículos comprados? Identifique cuál de las opciones contiene la variable independiente y dependiente.

Escoja con el ratón la opción correcta y llévela a la figura.



Variable independiente

La variable independiente es la cantidad de artículos y la dependiente es la rebaja.

La variable independiente es la rebaja y la dependiente es la cantidad de artículos.

La variable independiente es el descuento y la variable dependiente es la rebaja.

La variable independiente es la rebaja y el descuento es la variable dependiente.

Retroalimentación actividad de aprendizaje 4. Si existe un error:

La variable m no es la opción correcta. La variable cuyo valor se quiere conocer aparece en la pregunta. Construya una frase con ella que involucre la otra variable

con la cual se relaciona. El sujeto de la oración contiene la variable dependiente. Mire el recuadro del ejemplo que aparece a la izquierda:

Si no (respuesta correcta)


Muy bien, observe la figura 30.

FIGURA 30. *Identificación de la variable independiente*



La figura 31 muestra la definición de la constante de proporcionalidad y, adicionalmente, expone gráficamente cómo se construye.

FIGURA 31. Constante de proporcionalidad

Contextualización	Base teórica	Modelación matemática		
Variables	Relación entre Variables	Variable Dependiente	Variable Independiente	Proporcionalidad
<p>Constante de proporcionalidad.</p> <p>Hace posible definir la igualdad en una ecuación. La cual indica la cantidad de variable dependiente por cada unidad variable independiente. Representa la pendiente en un modelo matemático.</p> <p>Ejemplo3.</p> <p>En el problema: Un trabajador realiza 20 piezas, un determinado número de trabajadores ¿Cuántas piezas producirán?, La constante de proporcionalidad es</p> <p>20 Piezas / trabajador</p> <p>Su definición y descripción se da con el valor que aparezca en el problema.</p>				
 <pre> graph TD A[un trabajador] -- realiza 20 --> B[piezas] A --> C[piezas/trabajador] B --> C C -- Constante de proporcionalidad --> D[Constante de proporcionalidad] </pre>				
<p>Haga clic hacer para actividad de aprendizaje</p>				

Retroalimentación actividad de aprendizaje 5. Si no es la opción correcta:

La respuesta dada equivale a decir: una empresa produce tres empresas en una hora. Fíjese que esa frase no se encuentra en el texto del problema.

9.3 Resolución de problemas

La resolución de problemas se divide en dos partes: una en la que se definen las variables y otra donde se formula el modelo matemático. La sección de definición de variables hace énfasis en la identificación de las variables y su función en el contexto problemático, lo mismo que en su definición y la descripción correspondiente. La inferencia de las funciones de las variables parte de la constante de proporcionalidad; por tanto, lo primero que se debe hacer es inferir la constante de proporcionalidad del contexto del problema, la cual indica la cantidad de variables dependientes por cada unidad de variables independientes.

La modelación matemática está compuesta por cuatro niveles: en el primero se tratan problemas de la forma $y = mx$; en el segundo se despeja la variable independiente de la ecuación dada; en el tercero se estudian ecuaciones de la forma $y = mx_1 + mx_2$, y en el cuarto son dos ecuaciones de la forma:

$$y = mx_1 + mx_2$$

En la tabla 4 se observa las operaciones cognitivas para el análisis de la resolución de problemas.

TABLA 4. Matriz de unidades de conocimiento versus operaciones cognitivas para los problemas.

Operaciones cognitivas \ Unidades de conocimiento	Definición de variables	Modelación matemática
Comprender		
identificar	x	
Comparar		x
Inferir	x	x

A continuación se detalla la identificación y clasificación de los factores posibles, concepciones erróneas o fuentes que pueden causar un error, estos, estrategias ineficientes para cada tipo de actividad:

- No puede inferir la constante de proporcionalidad.
- No identifica las variables ni sus funciones dentro del contexto problemático.
- No identifica el valor de la constante de proporcionalidad dentro del problema.
- No aplica la comparación para derivar la ecuación.

En la figura 32 aparece de la parte de clase de la modelación matemática.

FIGURA 32. Constante de proporcionalidad

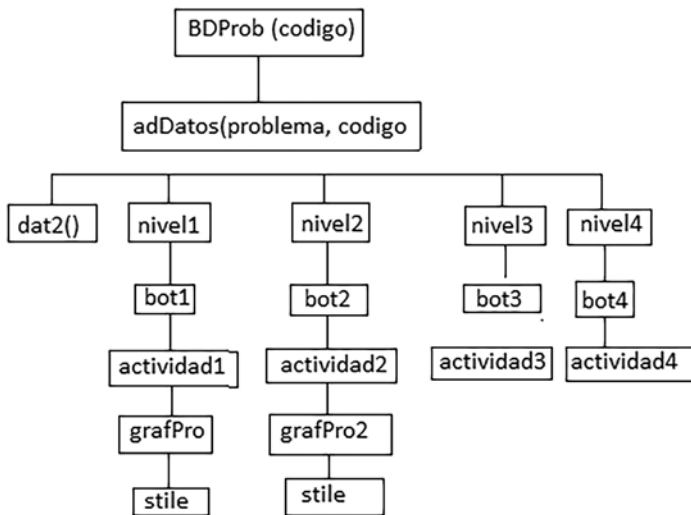
```
var problema ={          var teoProblema {
  Codigo_pro:"",        CodigoTeo: "",
  Contenido: "",        ContenidoTeo:"",
  NomVar: "",           CodigoNiv: ""
  Descrpcion: "",      };
  CodigoNiv: ""
};
```

El atributo «nomVar» es convertido por el programa en una matriz que contiene las funciones de las variables dentro del problema; «descripción» es otra matriz que contiene las descripciones de las variables. En la base de datos aparece, por ejemplo: «Número de días_día_Cantidad de televisores_televisor_televisores/día_7» y mediante la función *Split* de JavaScript se convierte en una matriz:

```
descripción=problema.split("_");
descripción[0]=Número de días;
descripción[1]=día;
descripción[2]=Cantidad de televisores;
descripción[3]=televisor;
descripción[4]=televisores/día;
descripción[5]=7;
```

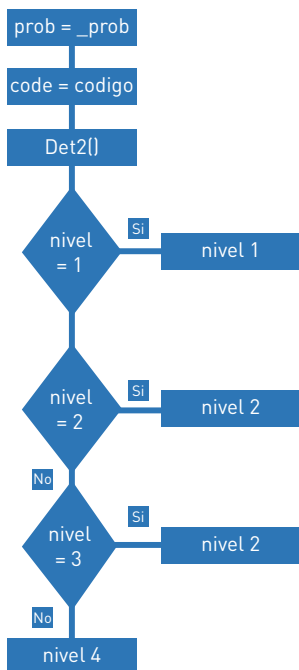
El diagrama de la parte de modelación matemática se presenta en la figura 33.

FIGURA 33. Diagrama de modelación matemática



La función BDPProb recibe los datos de la base de datos y se los pasa a la función dat2, la cual a su vez asigna los valores a los objetos. Luego, según el nivel (figura 34), se les asignan problemas a los estudiantes.

FIGURA 34. Algoritmo para escoger nivel



9.3.1 Definición de variables

Con la teoría y los ejercicios de definición de variables se busca que el estudiante identifique las variables y sus funciones dentro del contexto de los problemas. Se trata de una verificación de conocimientos de los temas ya estudiados en la parte teórica. En la figura 35 aparece una descripción de cómo realizar el procedimiento; se puede notar que si no hay problemas para identificar la constante de proporcionalidad, se pueden descubrir las variables fácilmente.

FIGURA 35. Definición de variables

Contextualización	Base teórica	Modelación matemática
Definición de Variables		Modelo matemático
<p>Definición de variables</p> <p>Para definir las variables se puede partir de la constante de proporcionalidad. En ésta, el numerador representa la variable dependiente y el denominador la variable independiente.</p> <p>Ejemplo</p> <p>Defina las variables del siguiente problema: Un vendedor de artículos en cuero desea fabricar bolsos para la venta. Para su fabricación, un bolso requiere de 2 pies de cuero ¿Qué cantidad de pie de cueros se necesita para fabricar cierta cantidad de bolsos?</p> <p>Solución</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Constante de proporcionalidad</p> <p>Variable dependiente</p> <p>y pies de cuero</p> </div> <div style="text-align: center;"> <p>pies de cuero / bolso</p> <p>Variable independiente</p> <p>x bolso</p> </div> </div>		
<p>Pulse para realizar actividad</p>		

Actividad de aprendizaje 1. Para realizar la actividad se espera que el estudiante identifique la constante de proporcionalidad para que, a partir de esta y con ayuda del ejemplo, arrastre las variables a su posición de acuerdo con su función dentro del problema y, posteriormente, dé su definición (figura 36).

FIGURA 36. *Actividad de identificación de variables*

Construya el modelo matemático del siguiente problema: En un día 10 trabajadores ensamblan 7 televisores. En un número dado de días, trabajando al mismo ritmo, ¿Cuántos televisores ensamblarán?

Siguiendo el ejemplo parte izquierda arrastre a las casillas correspondiente la descripción dada y en las cajas de textos (color gris), introduzca la variable que desee (una sola letra)

Número de días

Cantidad de televisores

televisores/día

Constante de proporcionalidad

Variable dependiente

Variable independiente

La figura 37 muestra el ejercicio resuelto.

FIGURA 37. Variables ya definidas

Construya el modelo matemático del siguiente problema: En un día 10 trabajadores ensamblan 7 televisores. En un número dado de días, trabajando al mismo ritmo, ¿Cuántos televisores ensamblarán?

Siguiendo el ejemplo parte izquierda arrastre a las casillas correspondiente la descripción dada y en las cajas de textos (color gris), introduzca la variable que desee (una sola letra)

El diagrama muestra la siguiente estructura de variables:

- Constante de proporcionalidad: **televisores/día**
- Variable dependiente: **t** Cantidad de televisores
- Variable independiente: **d** Número de días

9.3.2 Retroalimentación

De acuerdo con la respuesta dada se generará la retroalimentación correspondiente. Se utiliza el espacio de la constante de proporcionalidad para introducir otra variable y se devolverá la caja a su sitio previo; lo mismo sucede con cada uno de los sitios reservados para el concepto apropiado. Si se introduce la variable dependiente en el sitio de la variable independiente, se le avisa al usuario que la variable dependiente es la que queda en la parte superior de la constante de proporcionalidad; o lo contrario, la parte inferior es la variable independiente.

9.4 Modelación matemática

La modelación matemática tiene una sección teórica para cada nivel, así como otra de modelación matemática en sí. A continuación se muestran ambas partes para ambos conceptos.

Modelación matemática nivel 1. La figura 38 muestra la parte teórica y el ejemplo de la modelación matemática del primer nivel.

FIGURA 38. Ejemplo modelo matemático nivel 1

Definición de Variables	Modelo matemático
Modelo matemático (nivel 1)	
<i>Para el caso en estudio, un modelo matemático de primer nivel tiene la forma.</i>	
$y = m x$	
Variable dependiente	Constante de proporcionalidad Variable independiente
Ejemplo	
<i>Construya el modelo matemático del siguiente problema: En una empresa se producen 90 carros en una hora, ¿Qué cantidad de carros se producen en un número dado de horas?</i>	
Constante de proporcionalidad 90	carros/hora c t Modelo Matemático $c = 90t$
Pulse para realizar actividad	

El objetivo de la actividad es lograr expresar modelos matemáticos de la forma $y = mx$. En el ejemplo (figura 38) se muestra la estructura de cómo se deriva el modelo matemático: se parte de la constante de proporcionalidad, de la cual se derivan las variables independiente y dependiente.

Actividad de aprendizaje nivel 1. La figura 39 representa la estructura para ayudar al estudiante a obtener el modelo matemático.

FIGURA 39. Actividad para el nivel 1

Construya el modelo matemático del siguiente problema: En un día 10 trabajadores ensamblan 7 televisores. En un número dado de días, trabajando al mismo ritmo, ¿Cuántos televisores ensamblarán?

Siguiendo el ejemplo de la izquierda, arrastre la descripción de variables de acuerdo con el orden de aparición en el problema, y defina las variables y construya las ecuaciones dentro de las cajas de texto.

Constante de proporcionalidad

Modelo Matemático

Ecuación

En la figura 39 se puede observar que hay cinco cajas de texto para introducir datos, cada una de las cuales controla la entrada del usuario brindándole retroalimentación. Se interpreta el tipo de error y, de acuerdo con este, se provee el mensaje de retroalimentación correspondiente.

En la caja de texto para el valor de la constante de proporcionalidad pueden suceder los siguientes eventos.

- Introducir un valor que no corresponde a la constante de proporcionalidad.

- Introducir un valor no numérico.
- Oprimir ENTER sin introducir ningún valor.
- Introducir el valor correcto.

Para cada caso, en su orden, se dan los siguientes mensajes de retroalimentación:

- El valor de la constante de proporcionalidad no se relaciona con la pregunta del problema. Lea la pregunta, la cual lo ayudará a identificar el valor correcto.
- Se esperaba un dato numérico: el valor de la constante de proporcionalidad.
- Introduzca algún valor. No se admiten cajas de textos vacías.
- Perfecto, ubicó muy bien el valor de la constante de proporcionalidad.

Para la constante de proporcionalidad puede suceder:

- Que se introduzca la variable dependiente.
- Que se introduzca la variable independiente.
- Que la parte superior no sea la variable dependiente.
 - ~ Se confunde con la variable independiente.
 - ~ Se da otro valor.
 - ~ La variable no relevante.
- Que la variable inferior no sea la variable independiente.
 - ~ Se confunde con la variable dependiente.
 - ~ Se da otro valor.
 - ~ La variable no es relevante.
- Se olvidó escribir la barra o *slash* (/).

Los mensajes de retroalimentación respectivos son:

- Se esperaba una razón: variable dependiente/variable independiente.
- Se esperaba una razón: variable dependiente/variable independiente.
- En la parte superior va la variable dependiente. Escribió la variable independiente.
- Esa variable no aparece en el problema.
- Esa variable no es relevante en el contexto del problema.
- En la parte superior va la variable dependiente. Escribió la variable dependiente.
- Esa variable no aparece en el problema.
- Esa variable no es relevante en el contexto del problema.

Cajas para escribir la definición de las variables. En este espacio se puede escribir cualquier letra. Si se escribe el número, se pide una letra; si se escribe más de una letra, se solicita que las definiciones de variables se hagan solamente con una letra.

Caja del modelo matemático. Si se escribe la variable dependiente donde va la independiente o viceversa se emite el mensaje: «La variable dependiente o independiente (según el caso) no corresponde a la definición que se hizo en el paso anterior».

Problemas de nivel 2. Los problemas de este nivel tienen como objetivo la búsqueda de la variable independiente; por tanto, después de expresado el modelo, esta se debe despejar. La sección teórica aparece en la figura 40.

FIGURA 40. Teoría de problemas nivel 2

Definición de Variables	Modelo matemático
<p>Modelo matemático (nivel 2)</p> <p>El modelo matemático de segundo nivel contempla el siguiente procedimiento:</p> <p>$t = m \times x$ $mx = t$ $x = t / m$</p> <p>Ejemplo</p> <p>Construya el modelo matemático y después, despeje la variable independiente del siguiente problema: Un agricultor desea sembrar árboles en su finca. Si cada árbol requiere de 4 metros cuadrados de terreno, ¿Cuántos árboles se pueden sembrar</p> <p>Solución:</p> <p>Constante de proporcionalidad $\xrightarrow{4}$ metros cuadrados/árbol</p> <pre> m a v v Modelo Matemático --- m = 4a v 4a = m a = m/4 </pre>	
<p>Pulse para realizar actividad</p>	

Modelación matemática nivel 2. En la figura 41 se observa la estructura propuesta para que los estudiantes aprendan a despejar la variable. Es parecida a la del nivel 1, pero incluye el paso del despeje.

FIGURA 41. Actividad nivel 2

Un archivo de texto contiene un número determinado de caracteres, incluyendo espacios, si una hoja de un tamaño determinado se llena con 2374 caracteres, ¿Cuántas hojas se necesitan para imprimir el archivo?

Siguiendo el ejemplo de la izquierda, arrastre la descripción de variables de acuerdo con el orden de aparición en el problema, y defina las variables y construya las ecuaciones dentro de las cajas de texto.

El diagrama muestra un flujo de información. Un recuadro etiquetado como 'Modelo Matemático' tiene una línea que apunta a un recuadro etiquetado como 'Ecuación'. Desde 'Ecuación', una línea apunta hacia arriba a un recuadro etiquetado como 'Constante de proporcionalidad'. Desde 'Ecuación', una línea apunta hacia abajo a una caja de texto. Desde 'Ecuación', una línea también apunta hacia la derecha a otra caja de texto. Hay un recuadro adicional que apunta a 'Ecuación' desde arriba, pero no está etiquetado.

El error más frecuente en este nivel es el de confundir la variable dependiente con la independiente; este error se genera por escribir la constante de proporcionalidad invertida. Los mensajes de retroalimentación conducen al estudiante a percibir este error. Lo mismo que en el despeje, estos escriben el valor de la constante de proporcionalidad arriba en lugar de ubicarlo abajo:

$$h = c/2374: \text{correcto.}$$

$$h = 2374/c: \text{incorrecto.}$$

Problemas de nivel 3. Los problemas de nivel 3 se construyen con lo aprendido en los niveles anteriores. Incluyen ecuaciones de la forma $y = m_1x_1 + m_2x_2$; es decir, la suma de las expresiones obtenidas anteriormente (figura 42).

FIGURA 42. Teoría de resolución de problemas nivel 3

Definición de	Modelo matemático	
Variables		
En el tercer nivel están implicados dos términos de la forma $y = m x$, y se quiere hallar la suma de éstos dos términos. Si $y = m x$, $w = d f$; el total, t es:		
$t = m x + d f$		
Ejemplo		
Para fabricar una silla se requieren 2 pies de madera y para una mesa se requieren 3 pies de madera. Si se van a producir cierta cantidad de sillas y mesas, proponga un modelo matemático que facilite conocer la cantidad de madera que se necesita para estos dos productos.		
x : Cantidad de sillas	$y = 2x$	
y : Madera para las sillas		
f : Cantidad de mesas	$w = 3f$	
w : Madera para las mesas		
		t : total madera
		$t = 2x + 3f$

Pulse para realizar actividad

Como en los casos anteriores, lo fundamental es mostrar la estructura del problema para aumentar la probabilidad de transferencia de los procedimientos de solución a problemas con estructuras profundas parecidas. En la figura 42 se observa que el nivel 3 tiene como submetas los niveles anteriores: se muestra la constante de proporcionalidad de las ecuaciones porque se supone un dominio de esta técnica.

Modelación matemática nivel 3. En la figura 43 se puede observar que la actividad consiste en arrastrar las definiciones a sus cajas respectivas; cuando no corresponden, se devuelven al lugar de origen. Se validan las entradas de manera parecida a como sucede en los anteriores casos.

FIGURA 43. Actividad nivel 3

Un agricultor desea sembrar mangos y naranjos en su finca. Para producir 1 árbol de mango necesita 5 m² de terreno y para un naranjo 3 m². Proponga un modelo matemático que facilite conocer la cantidad de terreno que se necesita para sembrar cierta cantidad de estos dos clases de árboles.

Siguiendo el ejemplo de la izquierda arrastre la descripción, de las variables de acuerdo con el orden de aparición en el problema; y defina las variables y construya las ecuaciones en las cajas de texto.

terreno para naranjo

terreno para mango

total terreno

Cantidad de mangos

Cantidad de naranjos

Problemas de nivel 4. El nivel 4 corresponde a dos ecuaciones del nivel 3 (figura 44):

$$y_1 = x_1 m_1 + x_2 m_2$$

$$y_2 = x_1 m_3 + x_2 m_4$$

Las submetas son las ecuaciones estudiadas en el nivel 3.

FIGURA 44. Parte de los problemas nivel 4

Definición de Variables	Modelo matemático
Problemas de nivel 4.	
<p>El objetivo de este nivel es construir modelos matemáticos que se expresen mediante dos ecuaciones.</p> $m_1 x + m_2 y = t$ $m_3 x + m_4 y = h$	
<p>Ejemplo</p> <p>Un vendedor de artículos en cuero desea fabricar bolsos y chaquetas para la venta. Un bolso se fabrica con 2 pies de cuero y 4 cremallera. Una chaqueta con 3 pies de cuero y 2 cremalleras, ¿Qué cantidad de cuero y cremallera se necesitan para producir cualquier cantidad de bolsos y cualquier cantidad de chaquetas.</p>	
<p>t: Cantidad de cuero</p> <p>x: Cantidad de bolsos</p> <p>y: cantidad de chaquetas</p> <p>h: cantidad de cremalleras</p>	<p>2x + 3y = t</p> <p>4x + 2y = h</p> <p>Sistema de ecuaciones</p> $\begin{cases} 2x + 3y = t \\ 4x + 2y = h \end{cases}$
<p>Pulse para realizar actividad</p>	

En la figura 45 aparece la actividad de aprendizaje para el nivel. En él se busca la identificación de las variables con sus respectivas descripciones; luego, la formación de ecuaciones parecidas a las formas en el nivel 3 y, finalmente, el sistema de ecuaciones perteneciente al problema que se está resolviendo.

FIGURA 45. Actividad para los problemas nivel 4

Una mueblería produce sillas y mesas. Para producir una silla se utilizan 2 pies de madera y 5 horas - hombre. Para producir una mesa se necesitan 3 pies de madera y 4 horas - hombre. Si se posee cierta cantidad de madera y determinada cantidad de horas - hombre, haga un modelo matemático que permita encontrar la cantidad de sillas y mesas que se pueden producir con esa cantidad de materia prima.

Siguiendo el ejemplo de la izquierda arrastre la descripción, de las variables de acuerdo con el orden de aparición en el problema; y defina las variables y construya las ecuaciones en las cajas de texto.

Cantidad de mesas

Cantidad de madera

total madera

Cantidad de sillas

Cantidad de horas-hombre

En este capítulo se describió el diseño de un objeto de aprendizaje efectuado por Otero (2013), que muestra las estructuras profundas de los problemas; se expone aquí para ilustrar cómo se pueden utilizar las diferentes teorías de la psicología del aprendizaje, con los principios de programación de objetos de aprendizaje, para producir un material de apoyo que refuerce la formación de los estudiantes en las áreas disciplinares. Se diseñó con la finalidad de ser utilizado en una investigación posterior (Otero, 2014b), dirigida a fomentar en los estudiantes la identificación de estructuras profundas de los problemas de un contexto a otro.

Capítulo 10.

Programación en HTML, JavaScript y PHP

Este capítulo busca brindar al lector algunos conocimientos básicos acerca de distintos *lenguajes de programación* cuya finalidad es diseñar programas para Internet; esperamos que esta concisa presentación motive a que el lector siga investigando para profundizar estos conocimientos —en Internet existe abundante información al respecto—. Por tanto, se revisarán algunos conceptos de programación, para orientar a quienes carezcan de conocimientos sobre el tema.

10.1 El concepto de programa

Un programa es una secuencia de instrucciones para realizar una tarea. Las tareas que se van a explicar están encaminadas a la realización de objetos de aprendizaje que sustentan su diseño en principios de la psicología del aprendizaje. Los programas se escriben mediante un código fuente, el cual es un grupo de instrucciones que puede entender un computador. Existen programas de sistemas y programas de aplicación; en este caso, se escribirán programas de aplicación orientados a web, los cuales se escriben en HTML, JavaScript o su forma más sencilla, JQuery y PHP. Con HTML se pueden crear sitios web a los que se les da mayor interactividad utilizando JavaScript y PHP.

El HTML es un lenguaje estático que sirve para mostrar información; cuando se requiere mayor interactividad se necesitaría emplear lenguajes como JavaScript, PHP, JSP o ASP. Es este capítulo se explica el uso de JavaScript, PHP y una versión más sencilla de JavaScript llamada JQuery.

Para hacer un programa, se efectúa un análisis del problema que se quiere resolver, así como de los requisitos para su solución. Luego se hace un diseño, para lo cual se pueden utilizar diagramas de flujo que muestren la estructura del programa; después se escribe el código fuente y, finalmente, se realizan pruebas para verificar que se han alcanzado los objetivos deseados. Cuando los programas son muy complejos se dividen en funciones que realizan determinadas tareas.

10.2 HTML

Para hacer los programas que se presentarán, puede emplear programas como NetBeans y XAMPP, los cuales son *software* de utilización libre.

Un programa HTML (figura 46), en este caso versión 5, se identifica con `<!DOCTYPE html>`. La expresión DOCTYPE, indica la versión de HTML con la cual se está trabajando. Luego viene `<html>`, que al final debe cerrarse con `</html>`. La sílaba *lang* corresponde al lenguaje, que para nuestro caso está en idioma español. También en esta figura (46) puede verse `<head></head>`, `<title></title>` y `<body></body>`, que corresponde a una hoja de Internet. A partir de aquí se va a continuar con programación en jQuery, el cual se puede programar en el *head*, el *body* o en un programa aparte; los JQuery se programan dentro los rútilos `<script></script>`. Las páginas de Internet están divididas por medio de `<div></div>`. A las divisiones y demás elementos se les puede dar estilo con CSS (anexo A). CSS es un lenguaje empleado para darle estilo a las páginas realizadas con HTML, dado que contiene elementos necesarios para darles color y animación. Los div se identifican en CSS con las expresiones *id* o *class seguidas de un nombre*. Ejemplo: `<div id = 'entrada'></div>`. Luego, en el `<head>`, se pone una etiqueta `<style></style>` y se programa el estilo.

FIGURA 46. Aspectos básicos en HTML

```

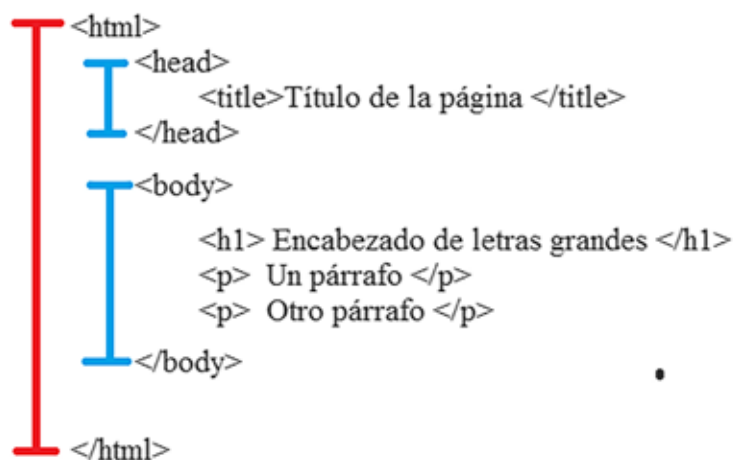
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <title>Lo básico de una página en HTML.</title>
  </head>
  <body>
  </body>
</html>

```

10.2.1 Las páginas de HTML y sus elementos

En la figura 47 se observa un diagrama de la estructura de una página de HTML

FIGURA 47. Estructura de una página de HTML



Para mostrar información se utiliza:

<p> </p> para párrafos

`<h1> </h1>` para títulos

A medida que el número que acompaña a la letra *h* aumenta, el título se hace más pequeño.

Para enfatizar textos:

``, ``, `` e `<i>`

Para hacer listas:

`` lista no ordenada

`` lista ordenada

`<dl>` lista descriptiva

Para los elementos de las listas se utiliza ``, `<dt>` y `<dd>`. En la figura 48 se indica la programación de los títulos y elementos resaltados, mientras que en la figura 49 se muestra el resultado.

FIGURA 48. *Títulos y elementos resaltados*

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Demostración HTML</title>
  </head>
  <body>
    <h1>Título principal</h1>
    <h2>Otro título</h2>
    <h3>Otro Título más</h3>
    <h6>Título pequeño</h6>
    <p>Este es un párrafo. Mire como se ve.</p>
    <p>Otro párrafo, luce interesante.</p>
    <p>Otro párrafo, luce interesante.<strong>Esta parte está resaltada con strong.</strong> </p>
    <p>Ahora em:<em> Mire la diferencia </em><b>Este es b </b><i> y este es i.</i></p>
  </body>
</html>
```


FIGURA 49. Resultados de títulos y párrafos resaltados



En la figura 50 aparece el resultado del programa para mostrar las listas ordenadas y no ordenadas, mientras que en la figura 48 se puede observar la programación.

FIGURA 50. Resultado de listas ordenadas y no ordenadas

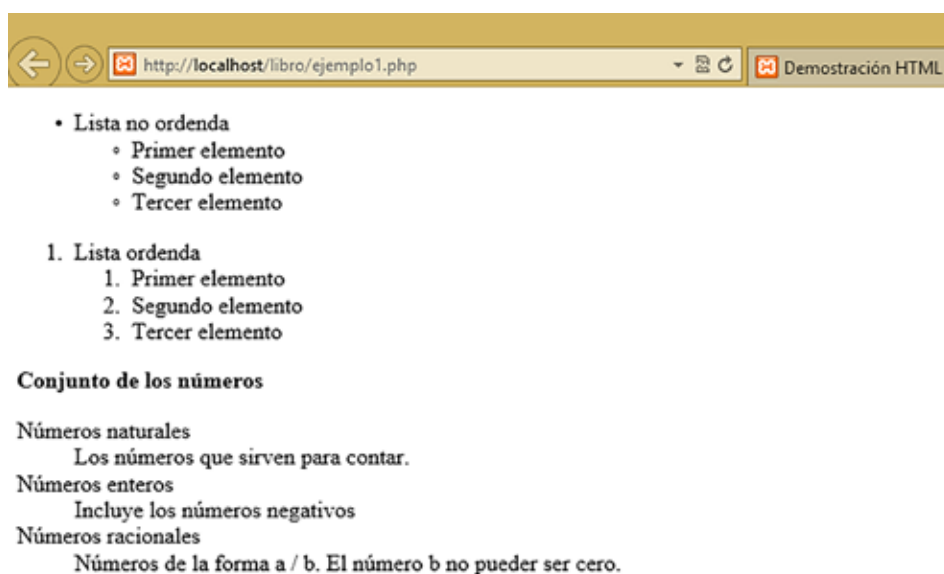


FIGURA 51. Programación de listas ordenadas y no ordenadas

```

<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <ul>
      <li>Lista no ordenda
        <ul>
          <li>Primer elemento</li>
          <li>Segundo elemento</li>
          <li>Tercer elemento</li>
        </ul>
      </li>
    </ul>

    <ol>
      <li>Lista ordenda
        <ol>
          <li>Primer elemento</li>
          <li>Segundo elemento</li>
          <li>Tercer elemento</li>
        </ol>
      </li>
    </ol>

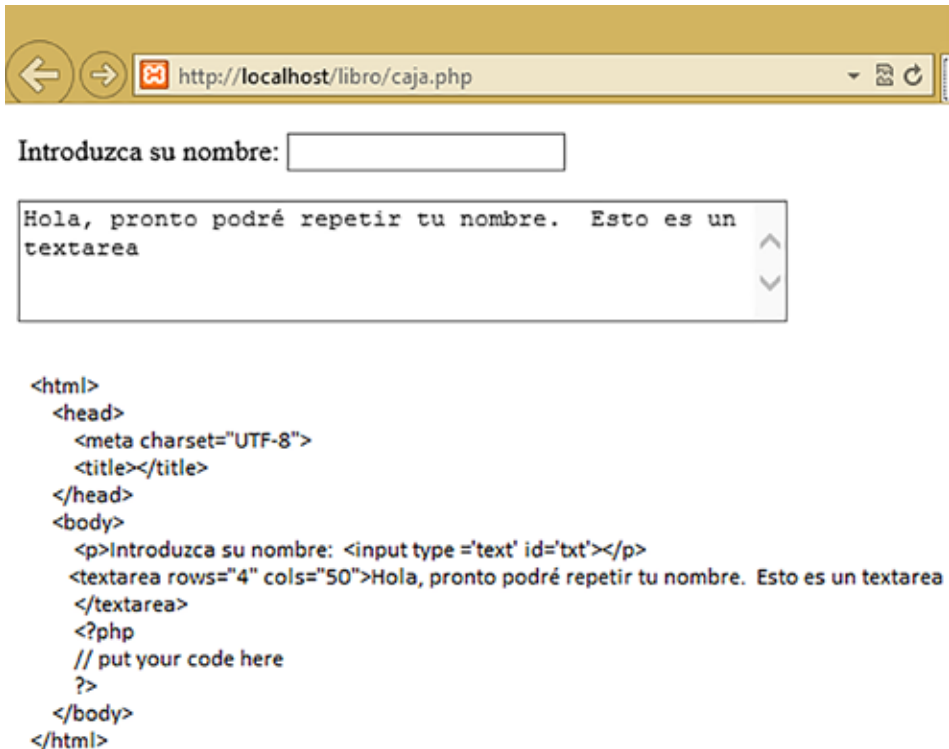
    <p><strong>Conjunto de los números</strong></p>
  <dl>
    <dt>Números naturales</dt>
    <dd>Los números que sirven para contar.</dd>
    <dt>Números enteros</dt>
    <dd>Incluye los números negativos</dd>
    <dt>Números racionales</dt>
    <dd>Números de la forma a / b. El número b no pueder ser cero.</dd>
  </dl>
  <?php
    // put your code here
  ?>
</body>
</html>

```

Para introducir información se utiliza *input* o *textarea*. La información se puede utilizar con programas escritos en PHP o JavaScript o, su forma

más sencilla, jQuery. La forma para programarla la información es: `<input type='text' id='txt'>`. Se le debe dar un *id* para poder recuperar el valor de la variable. En la figura 52 se presenta un ejemplo del resultado y la programación de los elementos explicados.

FIGURA 52. Capturando información



10.2.2 Tipos de datos en JavaScript

Como se mencionó, cuando los programas necesitan interactividad, además de emplear HTML, se requiere de otro programa de programación. En esta sección se explica el JavaScript que es uno de tales lenguajes. Comenzaremos explicando el concepto de variable ya que este es muy importante para tarea que se pretende efectuar.

Las variables son símbolos que permiten identificar un elemento no especificado en una situación. Declarar una variable en programación equivale a

reservar un espacio de memoria en el computador. En JavaScript una variable se declara con la partícula *var*. Ejemplo:

```
var a = 'Carro nuevo';
```

Cuando se desee mostrar el contenido de la variable, hay que indicarle al computador que imprima *a*; en lugar de *a*, de este modo se mostrará «Carro nuevo».

En la tabla 5 se presentan los tipos de datos en JavaScript. El tipo de dato «Clase» (véase anexo B) es creado por el usuario; en JavaScript, las clases se crean como funciones, las cuales contienen variables y otras funciones. Para ilustrar el concepto de «Clase», a continuación se escribe un pequeño programa que pregunta el nombre y luego devuelve el nombre con un saludo.

TABLA 5. Tipo de datos en JavaScript

Tipo	Ejemplo
Number	var = 5;
Cadena	var cad = «Hola , esta es una cadena»;
Booleano	var apagado = true;
Clase	

En la figura 53 se puede observar una función que actúa como una clase en JavaScript. Se destaca que la función aparece entre los *heads* (<head> </head>). Está el script jquery-2.1.1.min.js, que son las clases que permiten programar en jQuery. La función aparece entre <script> </script>, que es lo que se utiliza cuando se programa en JavaScript en una página de Internet. Estas funciones también se pueden hacer aparte, en un archivo con extensión .js.

Al escribir una función siempre se debe anteponer la palabra *function* y luego el nombre de la función. Como se puede observar que en el ejemplo de la figura 53, la función que tiene un argumento: nombre, argumento que recibe la información del nombre enviado desde otra función. Luego se declara una variable tipo cadena, *tuNombre*, a la cual se le asigna el nombre recibido y, a través de la variable *display*, se muestra un saludo con el nombre asignado

a través de la expresión `document.getElementById («saludos»).innerHTML`. La palabra ‘saludos’ que se ve entre paréntesis representa el nombre del *div* que muestra la información.

FIGURA 53. Función saludos

```
<head>
  <meta charset="UTF-8">
  <script src = 'jquery-2.1.1.min.js'></script>
  <title></title>
  <script>
    function Saludos(nombre)
    {
      var tuNombre = nombre;
      this.display = function()
      {
        document.getElementById("saludos").innerHTML += "<br> Gusto en
conocerle "+tuNombre;
      };
    }
  </script>
</head>
```

En la figura 54, en el `<body></body>`, aparece el programa principal, `hola.php`, que llama a la función «Saludos», donde se puede observar el *div* llamado «saludos»; luego, un *script* en el que aparece el símbolo pesos (\$), característico de la programación en JavaScript. El comando *append* sirve para agregar información a un *script* y para crear elementos de HTML. En este caso se está agregando una caja de texto donde se escribirá el nombre de la persona que está corriendo el programa. La expresión `$("#txtNombre")` pasa el nombre que está en la caja de texto a la variable *nom*. El *keypress* es un evento de la caja de texto *txtNombre*; al pulsar ENTER se corren las instrucciones que están en esta función. El número 13 representa la tecla ENTER. Se declara la variable «salud», que representa un objeto que se forma con la clase «Saludos». Observe que el argumento de «Saludos» es la variable «nom» que contiene el nombre. `Salud.display` es un miembro de «Saludos» y muestra el saludo con el nombre.

FIGURA 54. Programa hola.php

```

<body>
  <div id ='saludos'></div>;
  <script>
    $("#saludos").append("Cuál es tu nombre: <input type ='text' id='txtNombre'>");

    $("#txtNombre").keypress(function (e){

      if(e.keyCode===13)
      {
        var nom = $("#txtNombre").val();

        var salud = new Saludos(nom);
        salud.display();
      }
    });
  </script>
  <?php
  // put your code here
  ?>
</body>

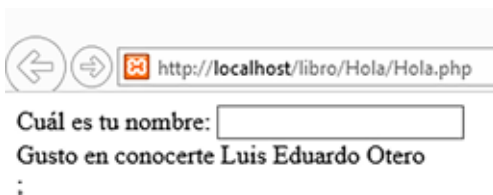
```

En la figura 55 se muestra el programa cuando solicita el nombre y, posteriormente, en la figura 56 aparece un saludo con el nombre que se digitó.

FIGURA 55. Programa Preguntando un nombre



FIGURA 56. Respuesta del programa Saludos



A continuación se muestra otro programa en jQuery para ilustrar lo que se ha visto; cuando se utilicen nuevos elementos se explicará su sintaxis. Se construirá un programa que realice la suma, la multiplicación y la división de dos números, para lo cual se crearán una clase y un menú que conducirán a la operación deseada. Una clase se asemeja a lo que es un concepto, ya que posee unos atributos que la definen y unos métodos que realizan acciones. Con la clase se pueden declarar objetos.

En JavaScript no se utiliza la palabra *class*. La declaración de una función es suficiente para declarar la clase (en el anexo 1 aparecen ejemplos de cómo se construye una clase, tanto en PHP como en JavaScript). En la figura 57 aparece parte de la programación HMTL del programa que se va a realizar. En la parte del *head*, aparece la expresión: `<script src="js/jquery-2.1.1.min.js"></script>`. El rótulo `<script>` se utiliza para programar en JavaScript. El archivo `jquery-2.1.1.min.js` es un archivo de jQuery que permite convertir la programación de JavaScript en una tarea más fácil.

La etiqueta *style* (anexo A, figura 129) se utiliza para dar formato a la página. La expresión «`ul li`» significa que se le va dar estilo a los *li* que aparecen debajo de los *ul*. La palabra *float* permite que el elemento flote, ya sea hacia la izquierda o hacia la derecha. El *width* establece el ancho y *border* el tipo de borde del elemento. La expresión «`#menu`» significa que se le está dando estilo al *div* que tiene como *id* `menu`. La barra doble (`//`) significa que la expresión aparece como comentario. En este caso, *border* no está programado porque está antecedido por la barra doble. El *height* indica la altura. La palabra clave *color* indica el color de la fuente. En el *body* aparecen programadas unas cajas de textos (`input type = 'text'`) y unos botones (`input type = 'button'`).

FIGURA 57. Programa de operaciones aritméticas 1

```

<html>
  <head>
    <meta charset="UTF-8">
    <script src="js/jquery-2.1.1.min.js"></script>
    <style>
      ul li{
        float: left;
        width: 240px;
        border:none;
      }
      #menu{
background-color: #cccccc;
// border: #0099cc solid;
width: 200px;
height: 300px;
// position: absolute;
// top:60px;
// left:0px;
float: left;

      }
      #resu{
color:#ff6600;
      }
    </style>
    <title>Operaciones</title>
  </head>
  <body>
    <H1>Operaciones aritméticas</H1>
    <div id='menu'>

      <p> Introduzca el primer número: <input type='text' id='txtNum1'></p>
      <p> Introduzca el segundo número: <input type='text' id='txtNum2'></p>
      <p>Presione sobre la operación deseada:</p>
      <input type='button' id='bot1' value='+'>
      <input type='button' id='bot2' value='*'>
      <input type='button' id='bot3' value='/'>

    </div>

    <div id='resu'>

  </div>

```

En la figura 58 aparecen programados los botones con la función *click*; al hacer clic sobre el botón se ejecutan las sentencias que estén en la función. Aparecen declaradas una variable y una clase que se explica en la figura 57.

Las expresiones `$('#txtNum1')` y `$('#txtNum2')` recuperan los valores de las cajas de texto y se las asignan a las variables `cad1` y `cad2`. La expresión `var opera = new operaciones(cad1,cad2,o)` declara el objeto que hace las operaciones aritméticas. La variable `o` indica la operación que se va a hacer con los números `cad1` y `cad2`; `opera.numas()` es una función que realiza la operación. El comando `document.getElementById` recibe la cadena resultado de la operación y lo muestra en pantalla en el *div* llamado *resu*.

FIGURA 58. Programa operaciones aritméticas 2

```
<script>
    $("#bot1").click(function(){
        var o=$('#bot1').val();
        var cad1 = $('#txtNum1').val();
        var cad2 = $('#txtNum2').val();
        var opera = new operaciones(cad1,cad2,o);
        opera.numas();
        var so=opera.display();
        document.getElementById('resu').innerHTML="Resultado de la suma "+so;
    });

    $("#bot2").click(function(){
        var o=$('#bot2').val();
        var cad1 = $('#txtNum1').val();
        var cad2 = $('#txtNum2').val();
        var opera = new operaciones(cad1,cad2,o);

        opera.numas();
        var so=opera.display();
        document.getElementById('resu').innerHTML="Resultado de la multiplicación es "+so;
    });

    $("#bot3").click(function(){
        var o=$('#bot3').val();
        var cad1 = $('#txtNum1').val();
        var cad2 = $('#txtNum2').val();
        var opera = new operaciones(cad1,cad2,o);

        opera.numas();
        var so=opera.display();
        document.getElementById('resu').innerHTML="Resultado de la división es "+so;
    });
</script>
```

La función que aparece en la figura 59 evalúa el contenido de la variable *operador* con la estructura de control *if* (en el anexo C se pueden observar otras estructuras de control); si contiene un más (+) se ejecuta la función *suma*; los paréntesis vacíos indican que la función no recibe ningún parámetro; además, si contiene un signo de multiplicación (*) se ejecuta la función *multi*, y si contiene un signo de división (/) se realiza la función *divi*.

FIGURA 59. Función con *this*

```

this. numas = function()
{
  if(operador=="+")
  {
    suma();
  }
  else if(operador=="*"){
    multi();
  }
  else
  {
    divi();
  }
};

```

En la figura 60 aparece la definición de una función, la cual recibe tres parámetros (*cad1*, *cad2* y *o*) que se asignan a unas variables internas; observe que los parámetros van separados por comas. La partícula *this* se utiliza para que las funciones que hacen las operaciones sean visibles fuera de la clase. Esta figura muestra que los parámetros enviados se reciben con las variables *n1* y *n2*, con las cuales se van a hacer las operaciones, y la operación que se va a realizar se indica con la variable *operador*. La expresión *parseInt()* se usa para pasar una variable tipo cadena a entero. El tipo de dato cadena tiene muchos métodos; lo mejor es consultar Internet para conocerlos mejor. La función *display()*, muestra el resultado de la operación que se hizo y con *return* regresa el valor resultado de un proceso que se quiere mostrar en otra función.

FIGURA 60. Programa operaciones aritméticas 3

```

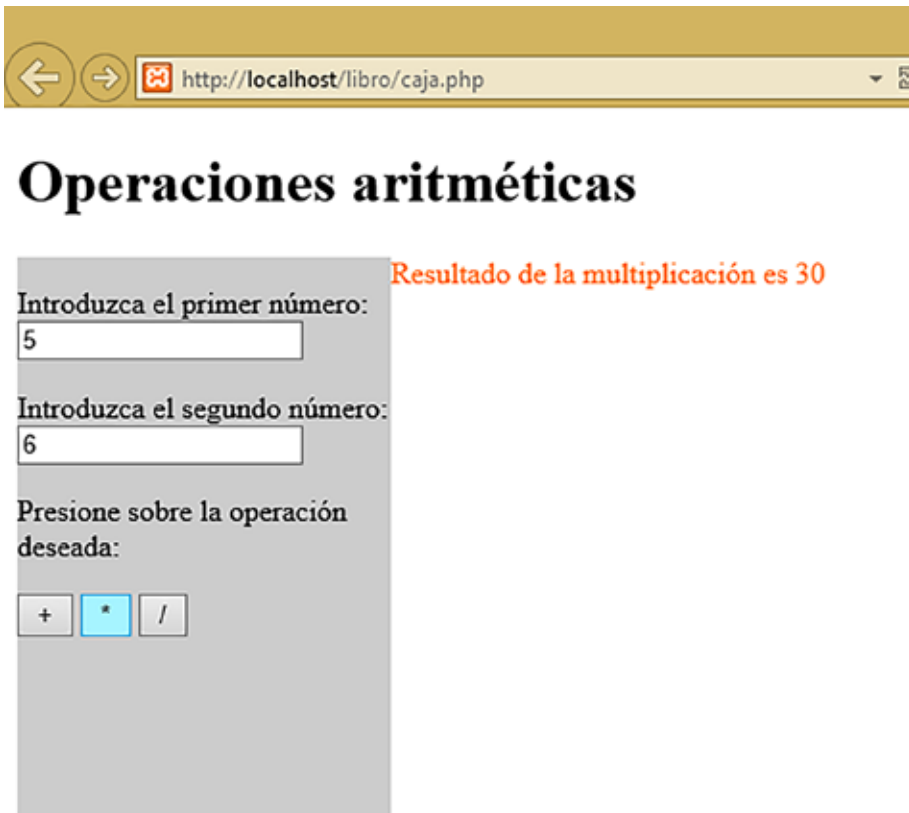
function operaciones(n1,n2,operador)
{
    var n=parseInt(n1);;
    var m=parseInt(n2);;
    var res;
    //alert("En clas");
    this.numas = function()
    {
        if(operador=="+")
        {
            suma();
        }
        else if(operador=="*"){
            multi();
        }
        else
        {
            divi();
        }
    };

    function suma()
    {
        res= n+m;
    };
    function multi()
    {
        res= n*m;
    };
    function divi()
    {
        res= n/m;
    };
    this.display = function()
    {
        return res;
    };
}
</script>
</body>
</html>

```

En la figura 61 se puede observar al programa corriendo. Al hacer clic sobre alguna de las operaciones se ejecuta la operación y muestra el resultado en rojo, a la derecha.

FIGURA 61. Resultado de operaciones aritméticas



Para hacer un test de escogencias múltiples se utiliza `<input type = 'radio'>`. Observe el siguiente ejemplo y su resultado. Este un problema en el que se le pide a los estudiantes que identifiquen las variables relevantes de un problema y, a medida que el estudiante pulsa una opción, se le da una retroalimentación acorde con su escogencia.

El problema es el siguiente: un carpintero produce en su casa veinte sillas en una hora para una empresa ubicada en Bogotá y registrada en la Cámara

de Comercio de Usaquén. Se desea saber cuántas sillas se pueden producir en un número determinado de horas. Determine cuál grupo de palabras constituyen las variables que definen el problema. La figura 62 muestra las opciones de respuestas.

FIGURA 62. Resultado de operaciones aritméticas

- Carpintero, casa.
- Sillas, horas.
- Empresa, Bogotá.
- Cámara de Comercio, Usaquén.

Para hacer el test de selección múltiple se utiliza `<input type="radio">`, cuyo uso se ilustra en las figuras 63. Entre los rótulos `<style> <style>` aparecen programados dos *divs*: contenido y opciones, que tienen los selectores *border* y *width*, el borde es gris y el ancho es de 600 píxeles; *solid* señala el tipo de *border*, que además de *solid*, puede ser: *dotted*, *dashed*, *double*, *groove*, *ridge*, *inset* o *outset*. Se están construyendo cuatro botones de radio que tienen un nombre (*name*) y un valor (*value*) con los cuales se pueden identificar en la programación. Adicionalmente, se encuentra el botón *answer2*, que al hacer clic sobre él llama a una función con el mismo nombre. Esta función se muestra en la figura 64, que de acuerdo con la opción seleccionada muestra mensajes de retroalimentación de tipo conocimiento de resultado, elaborada y de múltiples intentos.

FIGURA 63. Ejemplos de botones de radio

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <script src="js/jquery-2.1.1.min.js"></script>
    <title></title>
    <style>
      #contenido{
        border:grey solid;
        width:600px;

      }
      #opciones{
        border:grey solid;
        width:600px;

      }
    </style>
  </head>
  <body>
    <div id ='contenido'>

```

En el siguiente problema: Un carpintero produce en su casa, 20 sillas en una hora, para una empresa ubicada en Bogotá y registrada en la cámara de comercio de Usaquén. Se desea saber cuántas sillas se pueden producir en un número determinado de horas. Determine cuál grupo de palabras constituyen las variables que definen el problema:

```
</div>
```

```
<div id='opciones'>
```

```

<p> <input type="radio" value='1' name='opt'>Carpintero, casa.</p>
<p> <input type="radio" value='2' name='opt'>Sillas, horas.</p>
<p> <input type="radio" value='3' name='opt' >Empresa, Bogotá.</p>
<p> <input type="radio" value='4' name='opt'>Cámara de comercio, Usaquén</p>
<input type="button" id="answer2" value="Pulse opción deseada">

```

```
</div>
```

FIGURA 64. Determinación de la opción de los botones de radio.

```

<script type="text/javascript">
    $("#answer2").click(function(){
        var t= $("input[name=opt]:checked").val();
        if (t==='1')
        {
            alert("Lea con atención la pregunta del problema. Hay están las variables que busca.\n\
Ni carpintero ni casa están en la pregunta del problema.");
        }
        else if (t==='2')
        {
            alert("Perfecto esas es la respuesta correcta. Fijese que en la pregunta aparecen las variables que escogió.");
        }
        else if (t==='3')
        {
            alert("No es la opción correcta ¿Qué se desea en el problema. Ni Empresa o Bogotá están en la pregunta.");
        }
        else
        {
            alert("Incorrecto. Lea la pregunta con atención. Ni Cámara de comercio o Usaquéen están en ésta.");
        }
    });
</script>
<?php
// put your code here
?>
</body>
</html>

```

En la figura 65 aparece el resultado del programa. Después de marcar una opción se mostrará el mensaje de retroalimentación correspondiente.

FIGURA 65. Resultado de botones de radio

The screenshot shows a web browser window with the address bar displaying `http://localhost/libro/radios.php`. The page content is as follows:

En el siguiente problema: Un carpintero produce en su casa, 20 sillas en una hora, para una empresa ubicada en Bogotá y registrada en la cámara de comercio de Usaquéen. Se desea saber cuántas sillas se pueden producir en un número determinado de horas. Determine cuál grupo de palabras constituyen las variables que definen el problema:

- Carpintero, casa.
- Sillas, horas.
- Empresa, Bogotá.
- Cámara de comercio, Usaquéen

Below the options is a button labeled "Pulse opción deseada".

A continuación realizaremos un programa que maneje las funciones de arrastrar (*drag*) y soltar (*drop*) en un ejercicio de identificación de las funciones de las variables, dadas las unidades de la constante de proporcionalidad.

El ejercicio sería: si la constante de proporcionalidad está definida como la cantidad de variables dependientes por cada unidad de variable independiente, arrastre los elementos de la derecha hacia los que le correspondan a la izquierda.

Las funciones básicas aparecen en la figura 66. La variable *data* es la que se maneja para hacer el control de la información, lo demás se mantiene. En este caso, el programa que facilita el arrastre está localizado en un archivo de JavaScript aparte, el cual se declara en la sección *head* de la página web para que sea reconocido en el programa.

FIGURA 66. *Funciones para arrastrar y soltar*

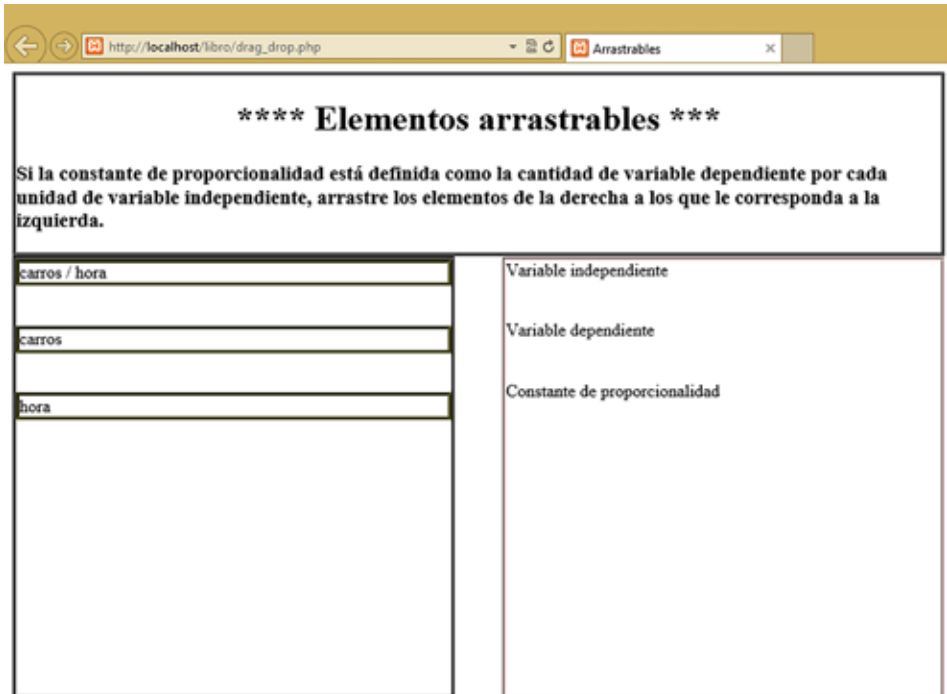
```
function allowDrop(ev) {
    ev.preventDefault();
}

function drag(ev) {
    ev.dataTransfer.setData("text", ev.target.id);
}

function drop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
}
```

En la figura 67 aparece el resultado del programa. El ejercicio consiste en pasar los elementos de la derecha a la caja correspondiente ubicada a la izquierda. Cuando el elemento no coincide, el rótulo escogido vuelve a su posición original.

FIGURA 67. Elementos arrastrables



La figura 68 muestra la programación en HTML de los elementos que se arrastran y de los que se sueltan. Están los *divs* que se utilizan para el arrastre y también los que reciben los rótulos que se arrastran. Se llama a la función *drag()* y se establece el tipo de puntero del ratón, llamado *move*. En la figura la función *drag()* aparece con comentarios, es decir, que no realiza ninguna acción. El programador pensó en otra solución y la anuló.

FIGURA 68. Programación de elementos arrastrables

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <script src="opp.js"></script>
    <script src="js/jquery-2.1.1.min.js"></script>
    <script src="js/jquery-ui.min.js"></script>

    <style>
      #general{
        width:850px;
        border:groove;
      }
      #variables{
        border:groove;
        height: 400px;
        width:400px;
      }
      #var1{
        border: #666600 inset;
      }
      #var2{
        border: #666600 inset;
      }
      #var3{
        border: #666600 inset;
      }
      #rol{
        position: relative;
        border:#ffcccc groove;
        height: 400px;
        width:400px;
        left:450px;
        top:-405px;
      }
    </style>
    <title>Arrastrables</title>
  </head>

  <body>
    <div id='general'>
      <h1>**** Elementos arrastrables ****</h1>
      <h3>Si la constante de proporcionalidad está definida como
        la cantidad de variable dependiente por cada unidad de
        variable independiente, arrastre los elementos de la
        derecha a los que le corresponda a la izquierda. </h3>
    </div>
    <div id='variables'>
      <div id='var1'>carros / hora</div><br><br>
      <div id='var2'>carros</div><br><br>
      <div id='var3'>hora</div><br>
    </div>
    <div id='rol'>
      <div id='rol1' >Variable independiente</div><br><br>
      <div id='rol2'>Variable dependiente</div><br><br>
      <div id='rol3'>Constante de proporcionalidad</div>
    </div>
    <script>draga();
    /* $("#rol1").css('cursor','move');
    $("#rol2").css('cursor','move');
    $("#rol3").css('cursor','move');*/
    </script>
    <?php
    // put your code here
    ?>
  </body>
</html>

```

La figura 69 muestra la función que permite que se arrastren los elementos. La palabra *draggable* permite que el elemento adquiera la propiedad de arrastre. El comando *revert: true* habilita al rótulo para volver a su posición original cuando la escogencia no corresponde a la respuesta correcta.

FIGURA 69. Función arrastrable

```

//*****opp.js
function draga()
{

//*****dragables
$("#rol1").draggable(
  {cursor:'move',
    revert:true});

  $("#rol2").draggable(
    {cursor:'move',
    revert:true});

  $("#rol3").draggable(
    {cursor:'move',
    revert:true});

//*****droppables

```

En la figura 70 se muestra la función «soltar» con el estado *hover*, que significa que se activa cuando el puntero del ratón se ubica sobre el elemento. *Droppable* indica que tiene la propiedad de aceptar los elementos que se arrastran. Observe el uso de la opción *revert*, cuando está *false*, indica que *var1* acepta el elemento que se está indicado en *accept*, cuando este elemento se suelte.

FIGURA 70. Función soltar

```
//*****droppables
$(function(){
  $("#var1").droppable({
    accept:"#rol3",
    activeClass:"ui-state-hover",
    hoverClass:"ui-state-active",
    drop:function(event,ui){

      alert("perfecto ");
      ui.draggable.draggable('option','revert','false');
    }
  });

  $("#var2").droppable({
    accept:"#rol2",
    activeClass:"ui-state-hover",
    hoverClass:"ui-state-active",
    drop:function(event,ui){

      alert("perfecto");

      ui.draggable.draggable('option','revert','false');

    }
  });

  $("#var3").droppable({
    accept:"#rol1",
    activeClass:"ui-state-hover",
    hoverClass:"ui-state-active",
    drop:function(event,ui){

      alert("perfecto");

      ui.draggable.draggable('option','revert','false');

    }
  });

}
```

10.3 PHP

Para trabajar con PHP se utilizan las etiquetas `<?php` cuerpo del programa `?>`. Se inicializa con `<?php`, se escribe el programa y al final se cierra con `?>`. Para mostrar la información se utiliza el comando *echo*. En la figura 71 se muestra un ejemplo sencillo de su utilización.

FIGURA 71. *Mostrando información en PHP*

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <?php
      echo 'Hola, ejemplo de mostrar datos PHP.';
    ?>
  </body>
</html>
```

Para declarar una variable se antepone el signo \$, por ejemplo: «*\$nombre*». Si una variable es global se conoce solo en el ámbito global; si es local, en el ámbito local. Para conocer una variable global dentro del ámbito local se antepone *global* a la variable (figura 72). En el ejemplo, el nombre «Antonio Lizarazo» está asignado a una variable global «*&nombre*»; para conocerla en la función *nombre()*, se le debe anteponer la palabra *global*: «*global \$nombre*».

FIGURA 72. Alcance de las variables

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <?php
      $nombre = "Antonio Lizarazo.";
      nombre();
      function nombre()
      {
        global $nombre;
        echo "El nombre es: $nombre";
      }

    ?>
  </body>
</html>

```

10.3.1 Tipos de datos

En la tabla 6 aparecen los principales tipos de datos que maneja el PHP con sus respectivos ejemplos (en Internet puede encontrar otros tipos).

TABLA 6. Tipos de datos

Tipo	Ejemplo
Entero	\$x = 5;
Flotante	\$x = 12,4;
Booleano	\$apagado = true;
Cadena	\$cad = "Hola, esta es una cadena";.

Además están los *arrays* y los objetos. En la figura 73 aparece la declaración de un *array*. A la variable «Contenido_pro» se le asigna el valor de otra variable denominada «\$contenido», que toma su valor en otro lugar. A las variables «nomVar_pro», «DescripcionVar_pro y Codigo_niv» se les asignan los valores que aparecen en la figura.

FIGURA 73. Declaración de un array en PHP

```

$problema = Array (
    'Contenido_pro'=>$contenido,
    'nomVar_pro'=>"Mango",
    'DescripcionVar_pro'=>"Independiente",
    'Codigo_niv'=>3
)

```

En la figura 74 aparece un *array* multidimensional; es decir, varios *arrays* dentro de un *array*. El *array* problema contiene los *arrays* 0 y 1. A los elementos de los *array* se les está dando un valor vacío, para ser reemplazados en la ejecución del programa por otros valores.

FIGURA 74. Array multidimensional

```

<?php
    $problema = Array(
        0 =>Array(
            'Contenido_pro'=>"",
            'nomVar_pro'=>"",
            'DescripcionVar_pro'=>"",
            'Codigo_niv'=>"",
            'Contenido_teo'=>""
        ),
        1 =>Array(
            'Contenido_pro'=>"",
            'nomVar_pro'=>"",
            'DescripcionVar_pro'=>"",
            'Codigo_niv'=>"",
            'Contenido_teo'=>""
        )
    );
?>

```

Para obtener un objeto se construye una clase (anexo B). En la figura 75 aparece la declaración de una clase que tiene una variable y un método. Las variables pueden ser públicas o privadas: las primeras se pueden ver desde afuera de la clase, las segundas, solo se conocen en la clase en donde se

declaran. Por ejemplo, a partir de la clase fruta que tiene la variable *\$fruta* y el método *mostrarFruta()*, que se utiliza para mostrar el tipo de fruta, en este caso manzana, se declara el objeto *\$fruti*; seguidamente, se llama a la función *\$fruti->mostrarFruta()* para conocer el nombre de la fruta.

FIGURA 75. Una clase en PHP

```
<?php
    $fruti = new frutas(); // Declaración del objeto

    $fruti->mostrarFruta(); // Llamado función mostrar fruta

    //Construcción de la clase.

    class frutas
    {
    // Declaración de una variable
    public $fruta = 'Manzana';    {

    // Declaración método
    public function mostrarFruta()
    echo $this->fruta;
    }
    }
    ?>
```

10.3.2 Un programa en PHP

Para ilustrar el uso de PHP realizaremos un programa que presenta un problema, del cual se deberán definir las variables y construir su modelo matemático. Luego mostraremos el resultado cualitativo de la evaluación.

En la tabla 6 se muestran las variables que maneja el problema, con su descripción y valores correspondientes.

TABLA 7. Variables del problema

Nombre variable	Descripción	Valor
\$conte	Contenido del problema	Un agricultor desea sembrar mangos y naranjos en su finca. Para producir 1 árbol de mango necesita 5 m ² de terreno y para un naranjo 3 m ² . Proponga un modelo matemático que facilite conocer la cantidad de terreno que se necesita para sembrar cierta cantidad de estas dos clases de árboles.
\$funVar	Funciones de las variables	Independiente1_ independent2_ Pendiente_ constante1 constante2
\$valor	Valor de las variables	0_0_0_5_3
\$descri	Descripción de las variables	Cantidad de mangos_ cantidad de naranjos_ cantidad de terreno_ terreno/mango_ terreno naranjo
\$problema = array (Contenido_pro nomVar_pro DescripcionVar_ pro Codigo_niv Contenido_teo ValorVar_pro)	Datos del problema	El valor inicial es vacío, excepto Codigo_niv = 2

En la figura 76 se muestra una lista desplegable que va a contener la descripción de las variables del problema. El estudiante debe definir la variable en la parte izquierda y, seguidamente, escoger una descripción en la parte derecha.

FIGURA 76. Lista desplegable del problema

Defina una variable con una letra y escoja una descripción de la lista desplegable.

variable

variable

variable

Luego, debe escribir el modelo matemático en la caja de texto que aparece en la figura 77.

FIGURA 77. Caja de texto para modelo matemático

Escriba la ecuación:

Ecuación	Enviar consulta
----------	-----------------

Posteriormente, el modelo experto compara la solución del estudiante con su solución y, a partir de ese punto, le comunica al módulo de enseñanza los aciertos y errores del estudiante durante el proceso. De esta manera el módulo de enseñanza se encargará de brindar la retroalimentación respectiva.

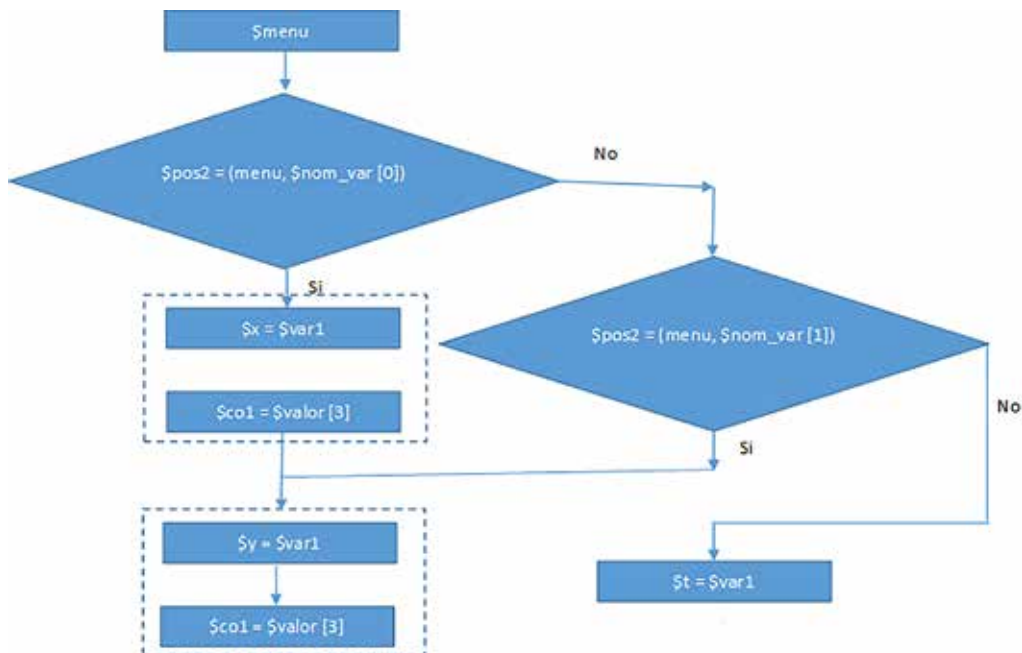
En la tabla 7 se representan los valores contenidos del *array \$problema* y en la figura 78 el diagrama de flujo del módulo experto, el cual posee el conocimiento de cómo se resuelve el problema. Para recibir la información introducida por el usuario existen tres pares de objetos, cada uno consistente en una caja de texto y una lista desplegable (figura 73): en la caja de texto se introduce una variable que consta de una letra y de la lista desplegable se escoge una descripción que determina la función que va a prestar la variable dentro del modelo matemático. La función puede ser de variable independiente o de dependiente. La tarea del módulo experto consiste en construir el modelo matemático a partir de las variables introducidas por el usuario; luego se compara con el modelo matemático de este último y el resultado de este proceso será enviado al módulo de enseñanza al cual se encarga de facilitar la respectiva retroalimentación.

TABLA 7. Valores del problema

\$funVar	\$valor	\$nomVar	\$descri	Índice
independiente1	0	mango	Cantidad de mangos	0
independiente2	0	naranja	Cantidad de naranjos	1
dependiente	0	terreno	Cantidad de terreno	2
constante1	5	terreno/mango	terreno/mango	3
constante2	3	terreno/naranja	terreno/naranja	4

Para el primer par de objetos, el módulo experto (figura 78) hace el siguiente análisis: la variable «menu» trae la información seleccionada de una lista desplegable (*descri*, columna 4, tabla 7) proveída por el usuario; por ejemplo, se escogió «cantidad de mangos». El array *\$nom_var* contiene la información mostrada en la columna 3 de la tabla. En el caso del *if* que se muestra en la figura 74 se trata de *\$nom_var[0]*, que corresponde a «mango». Por otra parte, si al buscar *\$nom_var[0]* (palabra mango) en *\$menu* (en el índice en la columna 5 de la tabla 7) se encuentra coincidencia, entonces la variable escrita en la caja de texto «\$var1» es la variable independiente del primer término (*\$funVar*, columna 1 de la tabla 7); si son diferentes (parte derecha de la figura 74), se comprueba si es «\$nom_var[1]» que corresponde a la variable independiente del segundo término (independiente2); si no son iguales, lo escrito en la caja de texto «\$var1» es la variable dependiente, representada en el diagrama con *t*; en caso de ser iguales, la variable escrita corresponde a la variable independiente del segundo término, y para los otros pares de objetos se utiliza el mismo diagrama, hasta tener todas las variables relevantes ubicadas, con lo que se construye el modelo matemático.

FIGURA 78. Diagrama de flujo del experto



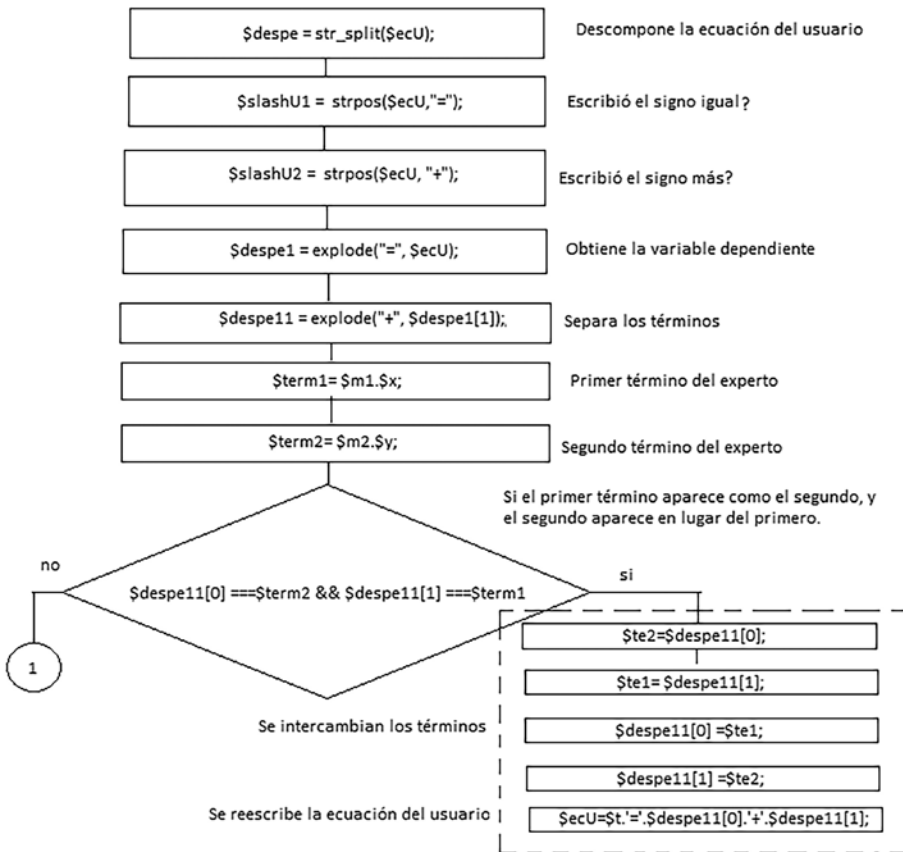
Para construir el módulo de enseñanza es necesario hacer un análisis de los errores que se puedan cometer en el proceso. A continuación se da una lista de los errores posibles:

- No se escribió el signo igual.
- No se escribió el signo de la operación.
- La variable dependiente la confunde con alguna de las variables independientes.
- La variable dependiente no fue definida.
- No establece diferencia entre una variable y una constante.
- Constante de proporcionalidad no coincide.
 - ~ Constante uno en el término dos y viceversa.
 - ~ La constante del primer término no aparece en el problema.

- Error en la primera variable independiente.
 - ~ Mezcló información del segundo término con el primero.
 - ~ Variable del primer término no fue definida.
- Error en el segundo término.
 - ~ Mezcló información del segundo término con el primero.
 - ~ Variable del segundo término no fue definida.

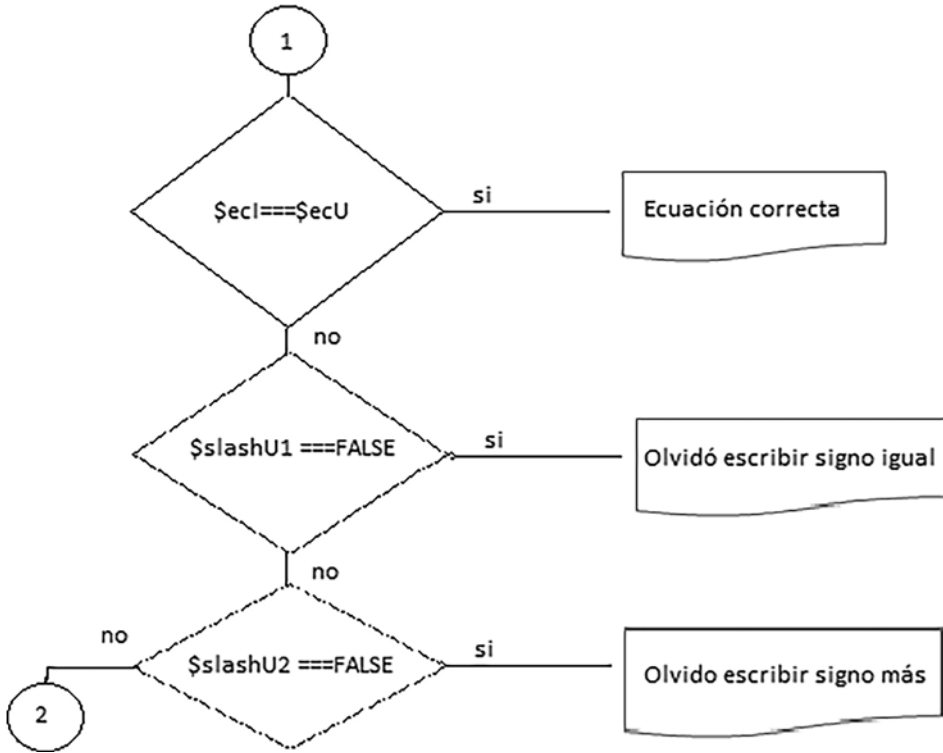
En la figura 79 se muestra parte del diagrama de flujo del módulo de enseñanza. Se desglosa la ecuación del usuario en sus partes componentes; para esto se convierte en *array* la cadena del usuario *\$ecu*, con la función *str_split*. Dentro de este *array* se busca que se haya escrito el signo igual y el signo más. Se convierte en *array* nuevamente la cadena del usuario, utilizando como separador el signo igual, de lo que se obtienen dos cadenas: *\$despe1[0]* y *\$despe1[1]*. Este último se convierte en *array*, utilizando como separador el signo más; luego se reconstruye el primer término del experto, que tiene la forma m_1x y, de igual manera, se construye el segundo término, que tiene la forma m_2y . También se puede observar que, si el usuario pasa la cadena en la forma $t = m_2y + m_1x$, el módulo de enseñanza la reescribe en la forma $t = m_1x + m_2y$, para poder hacer la comparación.

FIGURA 79. Módulo de enseñanza (primera parte)



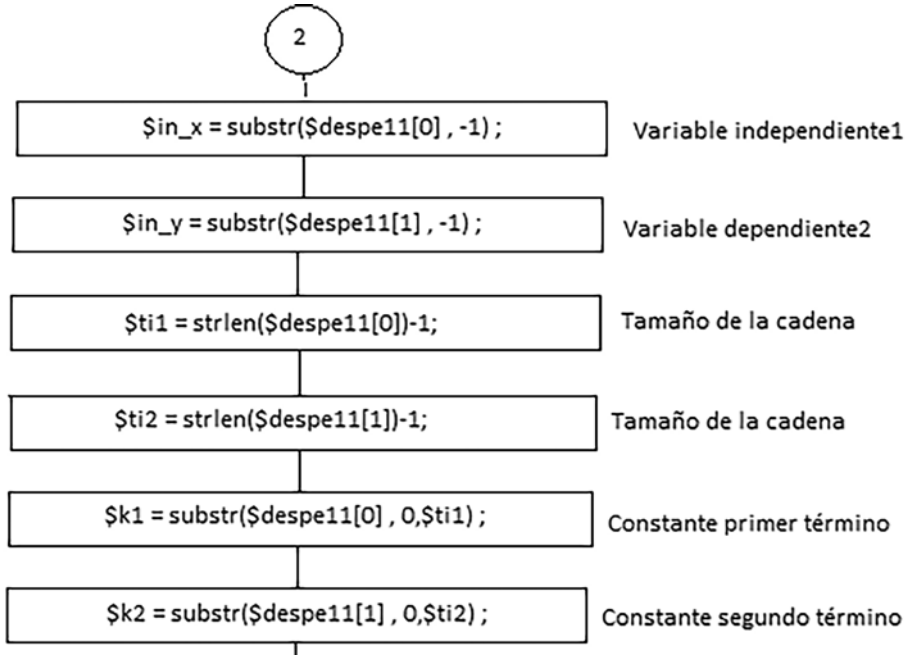
En la figura 80 se valida que el modelo escrito por el usuario esté correcto. Si no lo está, el error se debe tal vez a que olvidó escribir el signo igual o el signo más. El *if* verifica si la cadena del experto es igual a la del usuario: si son iguales, la ecuación se escribió correctamente; de lo contrario, se comprueba que se haya escrito el signo igual; si no es el error, tal vez no escribió el signo más.

FIGURA 80. Validando ecuación del usuario



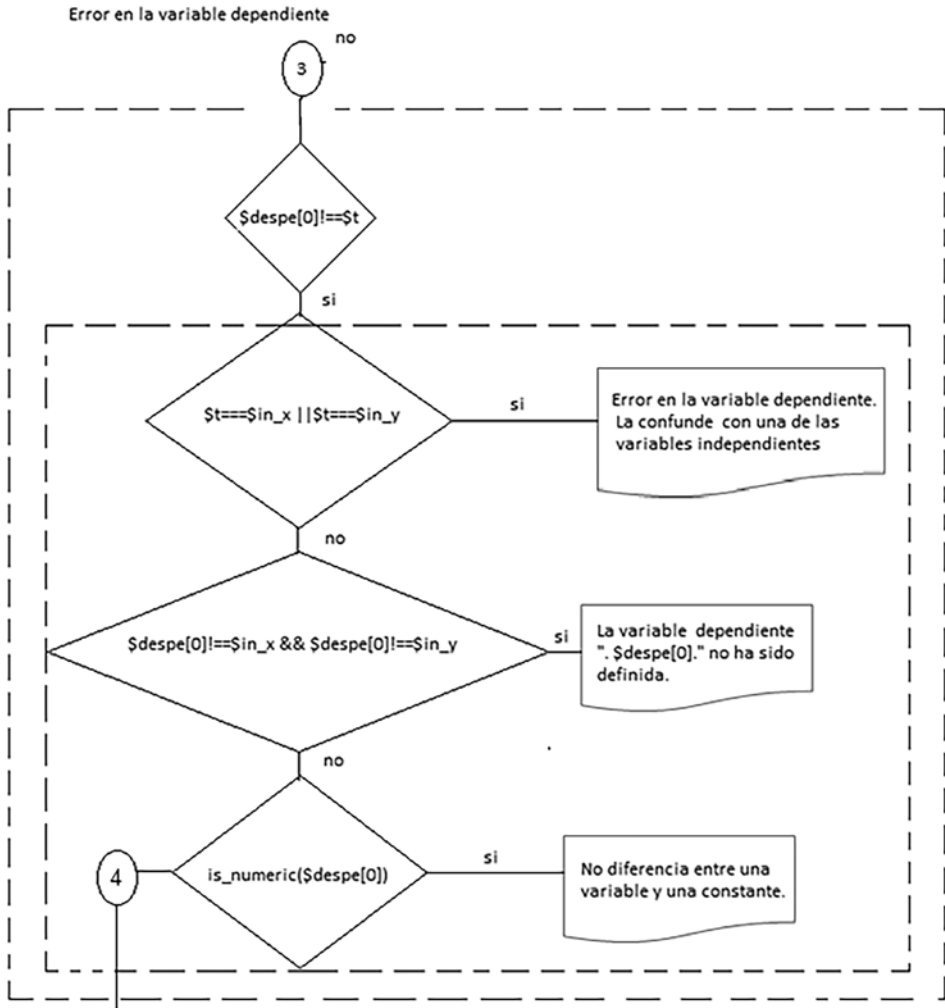
Si ninguna de las causas señaladas originó el error, la ecuación del usuario se descompone totalmente, para comparar cada una de sus partes (figura 81).

FIGURA 81. Descomposición de la ecuación del usuario



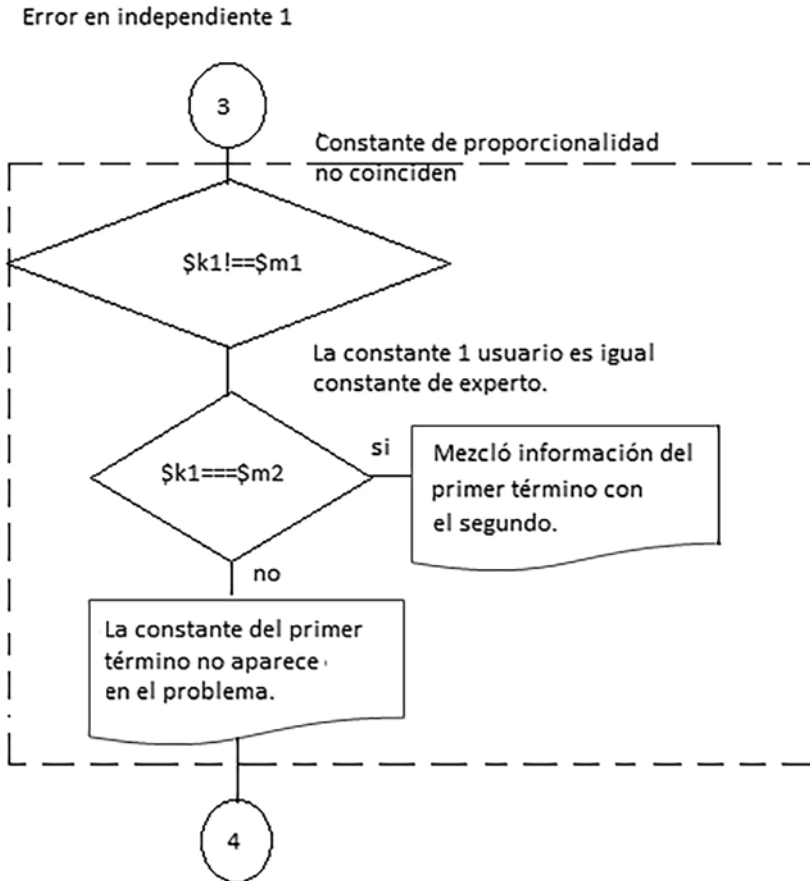
En la figura 82 se valida si el error está en la variable dependiente. Para ello se comparan las dos variables dependientes escritas. En la figura aparecen los comentarios de lo que puede originar el error: los signos (| |) y (&&) significan *o* e *y* respectivamente. Puede suceder que el error esté en la constante de proporcionalidad, por lo cual se contemplan dos posibilidades: que se escriba la constante de proporcionalidad del segundo término en el primero, o que se escriba una constante de proporcionalidad que no aparezca en el problema o que no sea relevante para el contexto.

FIGURA 82. Error variable dependiente



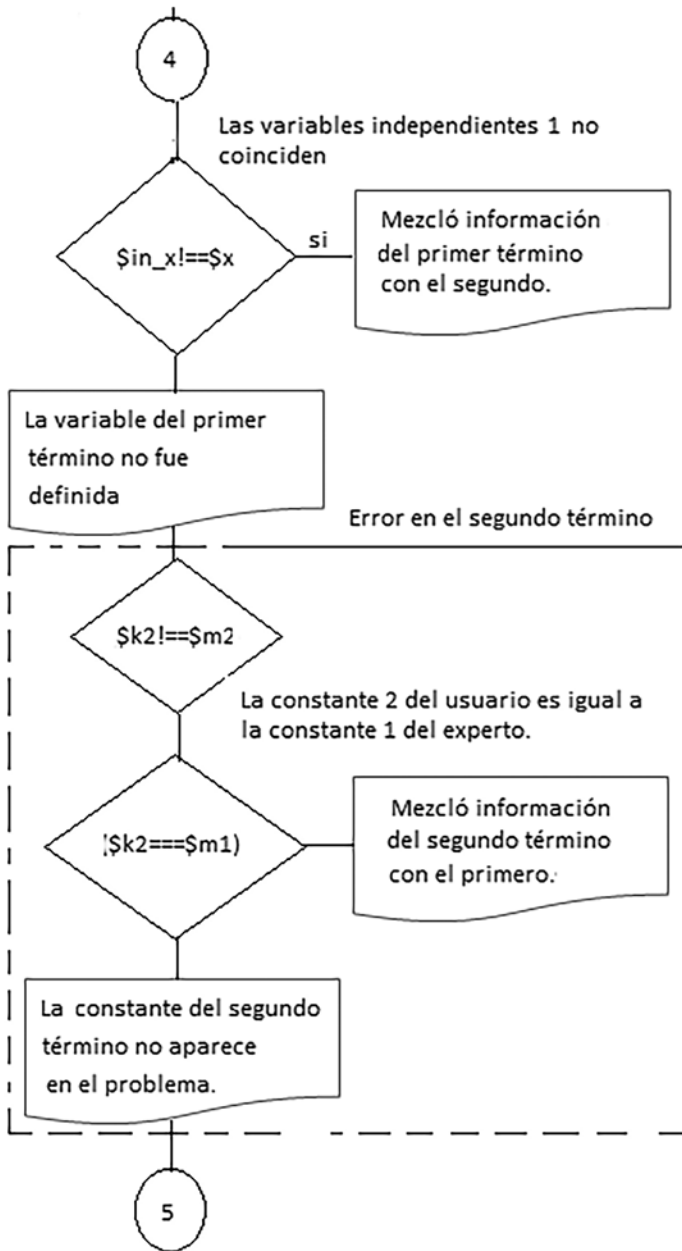
En la figura 83 se contempla la posibilidad de que el error esté en la constante de proporcionalidad del primer término. Puede suceder que se escriba la constante del segundo término en el primero, o que no se haya escrito la variable que se definió en la caja de texto. Luego, se hace el análisis para el segundo término.

FIGURA 83. Error en la primera constante



En la figura 84 se valida la variable independiente del primer término y si están intercambiadas las constantes de proporcionalidad. También puede suceder que no haya sido escrita la segunda constante de proporcionalidad.

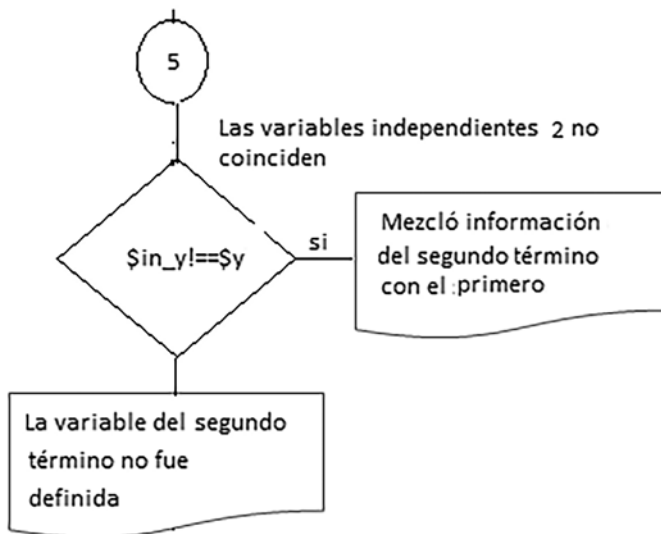
FIGURA 84. Error en la variable del segundo término



En la figura 85 se muestra el procedimiento en caso de error en la variable del segundo término, donde las posibilidades son las mismas que en el

primero. La información de los errores o aciertos se guarda en el módulo del estudiante para que la próxima vez que este ingrese al sistema, el módulo de enseñanza le asigne tareas para subsanar las deficiencias encontradas.

FIGURA 85. Error en la variable dependiente 2



Ahora, los diagramas de flujo se pasan a código fuente para que sean interpretados por un navegador, iniciando con la captura de los datos. Como ya se señaló, la definición de variables se captura en una caja de texto y su definición se escoge de una lista desplegable. Para hacer esta operación se utiliza un *form*, que luego envía los datos a otras funciones asociadas a los diagramas de flujo explicados.

En la figura 86 se muestra el *head* del programa. Hay una etiqueta *style* con el procedimiento de CSS que le da ancho, altura y fondo al *div* con *id proo*. También aparece el *script* que permite trabajar con jQuery.

FIGURA 86. Head del programa Problema

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <style>

      #proo{
        width: 600px;
        height: 260px;
        background-color: #cccccc;
      }
    </style>
    <script src="/js/jquery-2.1.1.min.js"></script>
  </head>

```

En las figuras 87 aparece la clase que maneja el problema. Se puede observar las declaraciones de variables tipo *private*

FIGURA 87. Clase problemas

```

<?php
class problemas {
  // Declaración de variables

  private $conte = "Un agricultor desea sembrar mangos y naranjos en su finca.
  Para producir 1 árbol de mango necesita 5 m2 de terreno y
  para un naranjo 3 m2. Proponga un modelo matemático que
  facilite conocer la cantidad de terreno que se necesita para
  sembrar cierta cantidad de estos dos clases de árboles.";

  private $funVar='independiente1_independiente2_pendiente_constante1_constante2';
  private $valor='0_0_0_5_3';
  private $nomVar = 'mango_naranjo_terreno_terreno/mango_terreno/naranjo';
  private $descri='Cantidad de mangos_Cantidad de naranjos_Cantidad de terreno_terreno/mango_terreno/naranjo';

  private $problema = Array(
    'Contenido_pro'=>"",
    'nomVar_pro'=>"",
    'DescripcionVar_pro'=>"",
    'Codigo_niv'=>"3",
    'Contenido_teo'=>"",
    'ValorVar_pro'=>""
  );

```

En la figura 88 se muestra la declaración de los métodos. Aquí se asignan el valor de las variables y existe un método para mostrar la información que se procesa.

FIGURA 88. Clase problemas (segunda parte)

```

// Declaración método
public function mostrarProble() {

    $this->problema['Contenido_pro']=$this->conte;
    return $this->problema['Contenido_pro'];
}
public function mostrarFunVar()
{
    $this->problema['nomVar_pro']=$this->funVar;
    return $this->problema['nomVar_pro'];
}
public function mostrarDescriVar()
{
    $this->problema['DescripcionVar_pro']=$this->descri;
    return $this->problema['DescripcionVar_pro'];
}

public function mostrarValorVar()
{
    $this->problema['ValorVar_pro']=$this->valor;
    return $this->problema['ValorVar_pro'];
}

public function mostrarNomVar()
{
    $this->problema['nomVar_pro']=$this->nomVar;
    return $this->problema['nomVar_pro'];
}
}

```

En la figura 89 aparece la forma. Comienza con un *include*, que es una clase que contiene los datos del problema, los cuales se pasan a las variables previamente explicadas. Hay unos datos que vienen enlazados con el subrayado (), que se utiliza como separador para los arreglos usados. También se asignan valor a las variables que se van a utilizar en el problema.

FIGURA 89. Captura de datos con una forma

```

<body>
  <?php
    //eje2.php
    include_once 'problemas.php';
    $pro = new problemas();
    $conte = $pro ->mostrarProble();
    $nomVar=$pro->mostrarNomVar();
    $descriVar = $pro->mostrarDescriVar();
    $valo = $pro->mostrarValorVar();

    $des = explode("_", $descriVar);
    $nom_var = explode("_", $nomVar);
    $valor= explode("_", $valo);

    echo '<div id = proo>';
    echo '<form action="eje2.php" method = "post" >';

    echo $conte;

    echo '<br><br>';
    echo 'Defina una variable con una letra y escoja una descripción de la
lista desplegable.<br>';
    echo '<input type ="text" name ="var1" placeholder="variable">';

    echo '<select name="menu1">';
    echo '<option>'. ""';
    for($x =0;$x<count($des);$x++)
    {

        echo '<option value = '.$x.'>'.$des[$x].</option>';

    }
    echo '</select>';

```

La forma (*form*) de la figura 89 define un archivo donde se van a enviar los datos (eje2.php) y la manera en que se va a hacer (método *post*). Se crea una caja de texto (var1) y, con *select*, una lista desplegable (menu1). En la lista desplegable se incluyen las definiciones de las variables que están contenidas en el arreglo $$des[x]$. Los datos recibidos por eje2.php se deben filtrar y asignar a variables locales.

La expresión `$pro->mostrarProble` muestra el contenido del problema y se asigna a `$conte`; de la misma manera sucede con las tres expresiones que siguen. El método `explode` convierte una cadena en un *array* que viene separado con «_». Seguido a esto viene un *div* con la forma `<form></form>`, que permite capturar los datos y enviarlos al programa, `eje2.php`, en este caso es el mismo programa que contiene la forma; es decir, se está llamando a sí mismo. La expresión `
` deja una línea en blanco.

Además de la forma y el `<select></select>` (que se usa para hacer una lista desplegable) hay un comando *for*, que pone la información en la lista desplegable. En la figura 90 se crean dos listas desplegables más porque se necesitan tres variables con sus descripciones. También aparece un botón identificado con *submit*, que sirve para enviar información a otra página web o a la misma página con la que se está trabajando.

FIGURA 90. Segundo y tercer select

```
// *****Segunda variable*****
echo '<br><br>';
echo '<input type = "text" name = "var2" placeholder = "variable">';
echo '<select name = "menu2">';
echo '<option>.''';
for($x = 0; $x < count($des); $x++)
{
    echo '<option value = '.$x.'>'.$des[$x].'\</option>';
}
echo '</select>';

// *****Tercera variable*****
echo '<br><br>';
echo '<input type = "text" name = "var3" placeholder = "variable">';
echo '<select name = "menu3">';
echo '<option>.''';
for($x = 0; $x < count($des); $x++)
{
    echo '<option value = '.$x.'>'.$des[$x].'\</option>';
}
echo '</select>';
echo '<input type = "submit">';
echo '</form >';
echo '</div>';
```


La función *filter_input* (INPUT_POST, Variable) (figura 91) verifica que la variable tenga algún contenido. Luego se asignan las variables a variables PHP. La función «*strpos*» comprueba que exista el *slash* dentro de la cadena; si esta última lo contiene significa que se escogió la descripción de una constante de proporcionalidad, lo cual no es permitido y se le avisa al usuario de tal error. El *if* que aparece en el programa es para evitar que lleguen datos vacíos.

FIGURA 91. Validación de contenido variable

```

if ((filter_input(INPUT_POST, 'var1') !==NULL) &&
    (filter_input(INPUT_POST, 'var2') !==NULL)
    && (filter_input(INPUT_POST, 'var3' ) !==NULL) &&
    (filter_input(INPUT_POST, 'menu1') !==NULL)
    && (filter_input(INPUT_POST, 'menu2') !==NULL) &&
    (filter_input(INPUT_POST, 'menu3') !==NULL))
{

    $var1= filter_input(INPUT_POST, 'var1');
    $menu1= filter_input(INPUT_POST, 'menu1');

    $var2= filter_input(INPUT_POST, 'var2');
    $menu2= filter_input(INPUT_POST, 'menu2');

    $var3= filter_input(INPUT_POST, 'var3');
    $menu3= filter_input(INPUT_POST, 'menu3');

    $pos1 = strpos($des[$menu1], "/");
    $pos2 = strpos($des[$menu2], "/");
    $pos3 = strpos($des[$menu3], "/");
    echo '<br>';
}

```

La figura 92 muestra la parte del programa que evita que el usuario repita una selección más de una vez.

FIGURA 92. *Función que evita repetición de variables.*

```

if(($var1===$var2) || ($var1===$var3) || ($var2===$var3))
{
    echo 'Repetió una variable con dos descripciones diferentes: ';
    echo '<br>';
    echo $var1. ". ". $des[$menu1];
    echo '<br>';
    echo $var2. ". ". $des[$menu2];
    echo '<br>';
    echo $var3. ". ". $des[$menu3];
    echo '<br>Introduzca nuevamente las variables.';
}
elseif (($menu1===$menu2) || ($menu1===$menu3) || ($menu2===$menu3)) {
    echo 'Tiene dos descripciones iguales: ';
    echo '<br>';
    echo $var1. ". ". $des[$menu1];
    echo '<br>';
    echo $var2. ". ". $des[$menu2];
    echo '<br>';
    echo $var3. ". ". $des[$menu3];
    echo '<br>Introduzca nuevamente las descripciones de las variables.';
}
}

```

En la figura 93 se valida que no falten definiciones y descripciones a la hora de enviar la información.

FIGURA 93. *Captación de falta descripciones o definiciones*

```

elseif (empty( $menu1) && empty($menu2)
        && empty( $menu3)) {
    echo "Faltaron una o más descripciones. $menu $menu1 $menu2";
}
elseif (empty(filter_input(INPUT_POST, 'var1'))|| empty(filter_input(INPUT_POST, 'var2'))
        || empty (filter_input(INPUT_POST, 'var3'))){
    echo 'Faltaron una o más definiciones de variables.';
}
elseif ($pos1!==FALSE || $pos2!==FALSE || $pos3!==FALSE) {
    echo 'En este problema no se necesita definir las constantes.';
    echo '<br>';
    echo $var1. ". ". $des[$menu1];
    echo '<br>';
    echo $var2. ". ". $des[$menu2];
    echo '<br>';
    echo $var3. ". ". $des[$menu3];
}
}

```

En la figura 94 aparece otra forma (*form*) que envía la ecuación aportada por el usuario a una página llamada *ecuacion.php*, donde se verifica si la ecuación digitada corresponde al problema propuesto. Allí se muestra que cuando la información proveída está completa y correcta se envía al módulo experto, que está en el archivo *ecuacion.php*, para que construya la ecuación.

FIGURA 94. Verificación si la información está correcta y completa

```

        else {

            echo $var1. ": ". $des[$menu1];
            echo '<br>';
            echo $var2. ": ". $des[$menu2];
            echo '<br>';
            echo $var3. ": ". $des[$menu3];

            echo '<br><br>';

            echo 'Escriba la ecuación:';
            echo '<br>';
            echo '<br>';
            echo '<form action="ecuacion.php" method = "post" >';
            echo '<input type ="text" name ="ecua" placeholder="Ecuación">';

            echo '<input type ="hidden" name ="var1" value=','$var1.'>';
            echo '<input type ="hidden" name ="var2" value=','$var2.'>';
            echo '<input type ="hidden" name ="var3" value=','$var3.'>';
            echo '<input type ="hidden" name ="menu1" value=','$menu1.'>';
            echo '<input type ="hidden" name ="menu2" value=','$menu2.'>';
            echo '<input type ="hidden" name ="menu3" value=','$menu3.'>';
            echo '<input type ="submit">';
            echo '</form>';
        }
    }
    ?>
</body>
</html>

```

En la figura 95 se puede observar la construcción de la clase del módulo experto. Se declaran las variables y luego se escriben las funciones miembros; posteriormente aparece la función *set*, que asigna las variables externas a la clase.

FIGURA 95. Construcción de la clase módulo experto

```

class experto {
    private $_var1;
    private $_var2;
    private $_var3;

    private $men1;
    private $men2;
    var $_nom_var;

    public function set ($cad1,$cad2,$var1,$var2,$var3,$nomVar)
    {
        $this->men1=$cad1;
        $this->men2=$cad2;

        $this->_var1=$var1;
        $this->_var2=$var2;
        $this->_var3=$var3;

        $this->_nom_var= explode("_", $nomVar);
        //var_dump( _nom_var);
        // echo "En set: ".$this->_nom_var[0];
    }
}

```

En la figura 96 aparece la función *indep1()*, que identifica ya sea la variable independiente 1, dependiente 2 o la variable dependiente, según la opción escogida en la lista desplegable. La función busca el contenido de *\$nomVar[0]* en *\$men1*; si la encuentra es la variable dependiente 1, de lo contrario verifica que sea la variable dependiente 2 y, si no es ninguna de las dos, es la variable dependiente. Lo mismo sucede con las otras dos funciones, *indep2* e *indep3*: *indep1* corresponde a la primera caja de texto, *indep2* a la segunda e *indep3* a la tercera. Existen dos variables independientes y una variable dependiente. La ecuación que se va a obtener tiene la forma: $t = m_1x + m_2y$.

FIGURA 96. Modelo experto (segunda parte)

```

public function indep1()
{
    $pos1= strpos($this->men1, $this->_nom_var[0]);
    $pos2= strpos($this->men2, $this->_nom_var[0]);
    if($pos1 !==FALSE)
    {
        $x=$this->_var1;
    }

    elseif ($pos2 !==FALSE) {
        $x= $this->_var2;
    }
    else{
        $x= $this->_var3;
    }
    return $x;
}
//fin indep1

public function indep2()
{
    $pos1= strpos($this->men1, $this->_nom_var[1]);
    $pos2= strpos($this->men2, $this->_nom_var[1]);
    if($pos1 !==FALSE)
    {
        $y= $this->_var1;
    }

    elseif ($pos2 !==FALSE) {
        $y= $this->_var2;
    }
    else{
        $y= $this->_var3;
    }

    return $y;
}
//fin indep2

public function dep()
{
    $pos1= strpos($this->men1, $this->_nom_var[2]);
    $pos2= strpos($this->men2, $this->_nom_var[2]);
    if($pos1 !==FALSE)
    {
        $t= $this->_var1;
    }
    elseif ($pos2 !==FALSE) {
        $t= $this->_var2;
    }
    else{
        $t= $this->_var3;
    }
    return $t;
}
//fin dep
//fin class

```

En un archivo llamado *ecuacion.php* está la función que aparece en la figura 97 —el programa que construye la ecuación del experto—. Aquí se puede observar la asignación enviada por una forma a variables locales de la función, y aparece también la declaración del objeto *\$ecu* a partir de la clase *experto*; luego se envían a la función *set* para descubrir el papel de cada variable en la ecuación. Como se observa en la función, «*\$x*» es la variable independiente, «*\$y*» la variable independiente 2 y «*\$t*» es la variable dependiente.

FIGURA 97. La construcción de la ecuación por el experto

```

include 'eje2.php';
include_once 'problemas.php';

if (!empty(filter_input(INPUT_POST, 'ecua')))
{
    $menu1 = filter_input(INPUT_POST, 'menu1');
    $menu2 = filter_input(INPUT_POST, 'menu2');
    $var1 = filter_input(INPUT_POST, 'var1');
    $var2 = filter_input(INPUT_POST, 'var2');
    $var3 = filter_input(INPUT_POST, 'var3');

    $ecu = new experto();

    $ecu->set($des[$menu1], $des[$menu2], $var1, $var2, $var3, $nomVar);
    $x = $ecu->indep1();
    $y = $ecu->indep2();
    $t = $ecu->dep();

    $ecI=$t."=".$valor[3].$x."+".$valor[4].$y;//Ecuación del experto

    $ecua = filter_input(INPUT_POST, 'ecua');

    $newRec = new recor();
    $ecU = $newRec->recorte($ecua);//Ecuación del usuario.

    ensenanza( $ecI,$ecU,$valor[3],$valor[4], $var1, $var2, $var3);
}

```

En la figura 98 se presenta una clase (*recor*) que elimina los espacios en blanco en una cadena para poder facilitar la comparación. La función *str_split* (*str_split* —cadena, número de caracteres—) convierte una cadena en *array*. Luego, con el comando *for* (anexo 2), se lee cada carácter y se copia en otra cadena, evitando con el *if* (anexo 1) que en esta nueva cadena se copien los espacios en blanco. La función *count()* devuelve la cantidad de elementos en un *array*.

FIGURA 98. Función que recorta espacios en blanco

```

class recor{

    public function recorte($txt2)
    {
        $txt = str_split($txt2);
        //echo "txt".$txt[0];
        $txtR="";
        for ($j=0;$j < count($txt);$j++)
        {
            if($txt[$j] != " ")
            {
                $txtR.= $txt[$j];
            }
        }
        return $txtR;
    }
}

```

Posteriormente el módulo de enseñanza (figura 99) separa las dos ecuaciones en sus componentes para realizar la comparación y establecer los errores o aciertos obtenidos; se van guardando los mensajes y, al final, se da un informe cualitativo del desempeño. Después de declarar dos variables, la función busca que se haya escrito el signo igual y el signo más en la cadena *\$ecu* enviada por el usuario; luego convierte la cadena en un *array* llamado *\$despe* y, nuevamente, se convierte la cadena del usuario en dos *arrays*, utilizando como separador el signo igual. De aquí se obtienen dos cadenas: *\$despe1[0]* y *\$despe1[1]*; esta última se convierte en *array* utilizando como separador el signo más. Finalmente se reconstruye el primer término del experto, que tiene la forma m_1x y el segundo término que tiene la forma m_2y .

En la figura 99 se puede observar que si el usuario pasa la cadena en la forma $t = m_2y + m_1x$, el módulo de enseñanza la escribe en la forma $t = m_1x + m_2y$ para poder hacer la comparación. Para eso se usan dos variables temporales: «*\$te1*» y «*\$te2*».

FIGURA 99. Módulo de enseñanza (primera parte)

```

function ensenanza( $ec1,$ecU,$m1,$m2, $x, $y, $t)
{
    $ecuacion= "Su ecuación es $ecU";
    $retro="";
    $slashU1 = strpos($ecU,"=");
    $slashU2 = strpos($ecU, "+");
    $despe = str_split($ecU);
    $despe1 = explode("=", $ecU);//Separa variable dependiente del resto

    $despe11 = explode("+", $despe1[1]);//separa los términos

    $term1= $m1.$x;
    $term2= $m2.$y;

    if($despe11[0] ===$term2 && $despe11[1] ===$term1 )
    {
        $te2=$despe11[0];
        $te1= $despe11[1];

        $despe11[0] =$te1;
        $despe11[1] =$te2;

        $ecU=$t.'='.$despe11[0].'+'.$despe11[1];
    }
}

```

En la figura 100 está la sección del programa que elabora la retroalimentación cuando no se escribe el signo igual o el signo más que une los dos términos que conforman la ecuación. Si la ecuación está bien, se le aplica un color diferente al informe y se da a conocer el resultado.

FIGURA 100. Módulo de enseñanza (segunda parte)

```

$in_x = substr($despe11[0], -1);
$in_y = substr($despe11[1], -1);

$ti1 = strlen($despe11[0])-1;
$ti2 = strlen($despe11[1])-1;

$sk1 = substr($despe11[0], 0,$ti1); //constante 1
$sk2 = substr($despe11[1], 0,$ti2); //constante 2

if($ec1===$ecU)
{
    echo '<br>';
    echo '<br>';
    echo "Su ecuacion es: $ecU ";
    echo '<br>';
    echo '<br>';
    $retro= '<font face color = "orange">Muy bien, esa es la ecuación pedida. '
    . 'Ya sabe obtener un modelo matemático correcto.</font>';
} // $ec1===$ecU

elseif($slashU1 ===FALSE)
{

    echo "Olvidó escribir el signo igual: $ecU";
}
elseif ($slashU2 ===FALSE)
{
    echo "Faltó el signo más: $ecU";
}

```

Posteriormente, se valida la variable dependiente escrita (figura 101) y, de acuerdo con el error, se produce un mensaje que aparecerá en el informe final.

FIGURA 102. Módulo de enseñanza (cuarta parte)

```

if($in_x!=$x)
{
    if($in_x==$y)
    {
        $retro.= "Mezcló información del primer término con el segundo.<br>";
    }
    else
    {
        $retro.='La variable del primer término no fue definida <br>';
    }
}

if($k2!=$m2)
{
    if($k2==$m1)
    {
        $retro.= "Mezcló información del segundo término con el primero.<br>";
    }
    else
    {
        $retro.='La constante del segundo término no aparece en el problema. Incompresión de lectura.<br>';
    }
}

if($in_y!=$y)
{
    //echo 'Error en constante 2';
    if($in_y==$x)
    {
        $retro.= "Mezcló información del segundo término con el primero.<br>";
    }
    else
    {
        $retro.= 'La variable del segundo término no fue definida.<br>';
    }
}

} //Sec1==$SecU
echo "$Secuacion<br>";
echo 'El resultado de su evaluación es el siguiente: <br>'. $retro;
} //enseñanza

```

En la figura 103 aparece el resultado de la primera parte del programa, mientras que en la figura 104 se muestra el informe que aparece cuando se cometen algunos errores.

FIGURA 103. Problema propuesto

Un agricultor desea sembrar mangos y naranjos en su finca. Para producir 1 árbol de mango necesita 5 m² de terreno y para un naranjo 3 m². Proponga un modelo matemático que facilite conocer la cantidad de terreno que se necesita para sembrar cierta cantidad de estos dos clases de árboles.

Defina una variable con una letra y escoja una descripción de la lista desplegable.

variable		▼
variable		▼
variable		▼

Enviar consulta

FIGURA 104. Errores en la formulación del modelo matemático

- x: Cantidad de mangos
- y: Cantidad de naranjos
- t: Cantidad de terreno

Escriba la ecuación:

Ecuación	Enviar consulta
----------	-----------------

Su ecuación es $t=2x+4r$

El resultado de su evaluación es el siguiente:

- La constante del primer término no aparece en el problema. Incomprensión de lectura.
- La constante del segundo término no aparece en el problema. Incomprensión de lectura.
- La variable del segundo término no fue definida.

Finalmente, en la figura 105 se puede observar la formulación correcta del modelo matemático del problema propuesto.

FIGURA 105. Formulación correcta del modelo matemático

x: Cantidad de mangos
 y: Cantidad de naranjos
 t: Cantidad de terreno

Escriba la ecuación:

Ecuación	Enviar consulta
----------	-----------------

Su ecuacion es: $t=5x+3y$

Su ecuación es $t=5x+3y$

El resultado de su evaluación es el siguiente:

Muy bien, esa es la ecuación pedida. Ya sabe obtener un modelo matemático correcto.

10.3.3 Ejemplo de aplicación en aprendizaje de conceptos

Para ilustrar la utilización de estos programas en otro dominio disciplinar, presentaremos a continuación la aplicación de un objeto de aprendizaje cuyo objetivo es diferenciar los vertebrados. En este ejemplo están involucradas la generalización y discriminación de los atributos críticos de los conceptos a fin de poder establecer el grupo al que pertenece el vertebrado.

Comenzaremos utilizando el modelo de Narciss (2001).

1. Análisis del objeto:

1. Objetivos de aprendizaje: formación de conceptos sobre los vertebrados.
2. Elementos específicos del contenido: los vertebrados y su clasificación.
3. Operaciones cognitivas: generalizar y discriminar entre los conceptos.
4. Matriz de conocimiento contra operaciones cognitivas.

TABLA 8. Unidades de conocimiento y operaciones cognitivas para el problema.

Unidades de conocimiento / Operaciones cognitivas	DV	C
Comprender	x	
Generalizar	x	x
Discriminar		x

Nota: DV: Definición vertebrados. C: Clasificación

2. Tareas de aprendizaje:

Tarea de generalización. Arrastre el animal de la parte derecha (figuras de animales) que corresponda al grupo descrito en la parte izquierda.

- Respiran por los pulmones.
- Tienen sangre caliente.
- Son vivíparos (esto es, el embrión se desarrolla en el vientre materno).
- Amamantan a sus crías (característica que los diferencia de los demás).

Tarea de discriminación de atributos en un concepto. En la figura 106 se muestran una serie de preguntas para ayudar a formar el concepto de mamífero. A la persona se le pide que conteste las preguntas que se hacen a continuación. Si todas las respuestas son **sí**, el animal es un mamífero; por consiguiente, en la pregunta sobre si es mamífero conteste **sí**. Si alguna respuesta es **no**, el animal no es mamífero y la pregunta sobre si es mamífero se contesta con un **no**.

FIGURA 106. *Tareas de discriminación de atributos de un concepto.*

¿Respiran por los pulmones? -----	SÍ <input type="radio"/> / NO <input type="radio"/>
¿Tienen sangre caliente? -----	SÍ <input type="radio"/> / NO <input type="radio"/>
¿Amamantan a sus crías? -----	SÍ <input type="radio"/> / NO <input type="radio"/>
¿Son vivíparos (el embrión se desarrolla en el vientre materno)? -----	SÍ <input type="radio"/> / NO <input type="radio"/>
Mamífero -----	SÍ <input type="radio"/> / NO <input type="radio"/>

3. Factores posibles de concepciones erróneas

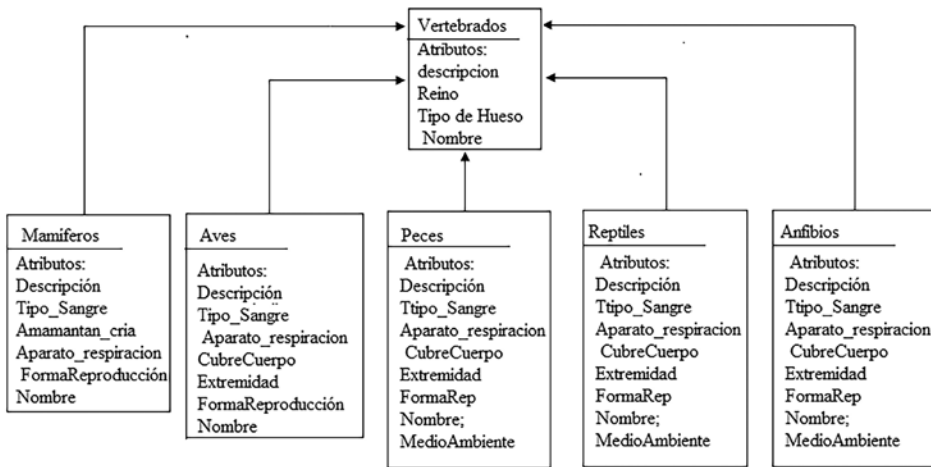
- Sobre generalización: tomar un contraejemplo como ejemplo del concepto.
- Generalización por debajo: tomar un ejemplo como contraejemplo.
- Concepción equivocada: los errores descritos al mismo tiempo.

4. Retroalimentación

Se comienza diseñando las clases³ necesarias para manejar la información del objeto de aprendizaje. Así como en la estructura de los conceptos existen conceptos superordinados, básicos y subordinados, las clases tienen una clase base, la cual es heredada por otras clases llamadas derivadas. Primero se suministra la estructura del concepto (figura 107) y luego, a partir de esta, se definen la clase base y sus clases derivadas.

3 Una clase es parecida a la definición de concepto.

FIGURA 107. Estructura del concepto vertebrado



A continuación se define la clase vertebrados (figura 108). Se puede observar la función `_construct`, la cual se utiliza para inicializar variables. Cada atributo tiene una función `display` que muestra sus valores, así como el comando `return`, que sirve para devolver un valor que ha procesado una función.

FIGURA 108. Clase vertebrado

```

class vertebrados{
    var $descrip;
    var $reino;
    var $esqueleto;
    var $tipo_Hueso;
    var $nombre;
    var $clas;

    function __construct()
    {
        $this->reino= "animal";
        $this->descrip="Son aquellos animales que tienen esqueleto interno con
columna vertebral y cráneo.";
    }
    public function display()
    {
        return $this->descrip;
    }
    public function displayClas()
    {
        $this->clas="Se agrupan en: <br>
        • Mamíferos.<br>
        • Aves.<br>
        • Peces.<br>
        • Reptiles.<br>
        • Anfibios.<br>";
        return $this->clas;
    }
}

```

En la figura 109 se puede notar la clase mamíferos, la cual tiene la palabra clave *extends* para denotar que se deriva de vertebrados; por tanto, hereda los atributos y las funciones de la clase principal.

FIGURA 109. Clase mamíferos (primera parte)

```

class mamiferos extends vertebrados{
    var $descrip2;
    var $tipo_Sangre;
    var $amamantan_cria;
    var $aparato_respiracion;
    var $formaRep;
    var $nombre;

    function __construct() {
        parent::__construct();
        //parent::display();
        $this->descrip2="
            • Respiran por los pulmones.<br>
            • Tienen sangre caliente.<br>
            • Son vivíparos, Estos es el embrión se desarrolla en el vientre materno.<b>
            • Amamantan a sus crias: característica que los diferencia de los demás
            vertebrados.<br>";
    }
    public function displayDescri()
    {
        parent::__construct();
        return $this->descrip2;
    }

    public function displayTipo_Sangre()
    {
        parent::__construct();
        $this->tipo_Sangre="Tienen sangre caliente";
        return $this->tipo_Sangre;
    }
}

```

En la figura 110 aparecen las funciones que permiten mostrar las información de los atributos de los mamíferos.

FIGURA 110. Clase mamíferos (segunda parte)

```

public function displayAmamantan_cria()
{
    parent::__construct();
    $this->amamantan_cria="Amamantan a sus crías";
    return $this->amamantan_cria;
}

public function displayAparato_respiracion()
{
    parent::__construct();
    $this->aparato_respiracion="Respiran por los pulmones";
    return $this->aparato_respiracion;
}

public function displayDesarrollo()
{
    parent::__construct();
    $this->formaRep="Son vivíparos, Estos es el embrión se desarrolla en el vientre materno";
    return $this->formaRep;
}

public function displayNombre()
{
    parent::__construct();
    $this->nombre="Mamífero";
    return $this->nombre;
}

}

```

Las otras clases de construyen de manera similar, por tanto, no se muestran aquí. La figura 111 presenta la entrada del programa, donde aparece la definición de vertebrado y cómo se estos agrupan. Al hacer clic sobre cualquier miembro del grupo se puede ver su definición. También se muestra el resultado de haber hecho clic sobre anfibios. Aquí aparecen la definición y las características para ese grupo.

FIGURA 111. Características de los anfibios

<h2>Ciencias Naturales</h2>						
<p>Los Vertebrados</p> <p>Los vertebrados: Son aquellos animales que tienen esqueleto interno con columna vertebral y cráneo.</p> <p>Se agrupan en (Haga clic en cada botón):</p> <table border="1"> <tr><td>Mamíferos</td></tr> <tr><td>Aves</td></tr> <tr><td>Peces</td></tr> <tr><td>Reptiles</td></tr> <tr><td>Anfibios</td></tr> </table>	Mamíferos	Aves	Peces	Reptiles	Anfibios	<p>Los anfibios:</p> <p>Anfibios: Animales con el cuerpo dividido en cabeza, tronco y cola, aunque esta última puede estar atrofiada como ocurre con los anuros o ranas adultas. Sus principales características son:</p> <ul style="list-style-type: none"> • Piel desnuda y provista de glándulas mucosas. • Esqueleto óseo. • Respiran cuando están pequeños por branquias; de adultos, por pulmones y por la piel. • Circulación doble e incompleta. • Temperatura variable. • Fecundación externa. Los huevos fecundados se desarrollan mediante metamorfosis. <p>Actividad de Aprendizaje</p>
Mamíferos						
Aves						
Peces						
Reptiles						
Anfibios						





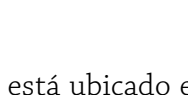
En la figura 112 se puede observar una actividad de aprendizaje que consiste en identificar a cuál de los animales ubicados en los recuadros de la derecha le corresponde los atributos que aparecen descritos en el recuadro de la izquierda.

FIGURA 112. Actividad de aprendizaje de generalización de concepto

Actividad de Aprendizaje

Arrastre el animal de la parte derecha que corresponda al grupo descrito en la parte izquierda

- Respiran por los pulmones.
- Tienen sangre caliente.
- Son vivíparos. Estos es el embrión se desarrolla en el vientre materno.
- Amamantan a sus crías: característica que los diferencia de los demás vertebrados.

 Para arrastrar
 Para arrastrar
 Para arrastrar
 Para arrastrar
 Para arrastrar

En la figura 113 se muestra el CSS del programa, el cual está ubicado en el *head* y tiene varios *scripts* correspondientes a los *divs*. Nótese el *float left* de *conte* y *grupo*; esto facilita mostrar la información a modo de columnas. En el

input tipo *submit* se le quitó el fondo al botón con *background*. *Fra* es un *fieldset* que contiene un marco para la actividad de aprendizaje.

FIGURA 113. CSS del programa

```
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <style>
      #conte{
        width:300px;
        float:left;
      }
      #grupo{
        width:700px;//700
        float:left;
      }
      input[type="submit"]{
        background: none;
        width:130px;
      }
      #fra {
        width:220%;
        // height: 5%;
      }
    </style>
  </head>
```

En la figura 114 aparece una forma (*form*), la cual servirá como base para construir un menú. La expresión *include* se utiliza para incluir las clases en este programa. Se construyen los objetos mamíferos, aves, peces, reptiles y anfibios. Los botones tienen un nombre, *grup*, y cada uno tiene un valor.

FIGURA 114. Forma del menú agrupación

```

<body>
  <fieldset>
    <h1>Ciencias Naturales</h1>
  </fieldset>

  <?php
  include 'mamiferos.php';
  $mamifero =new mamiferos();
  $save=new aves();
  $speces=new peces();
  $reptiles=new reptiles();
  $anfibiobios=new anfibios();
  // $reptiles=new reptiles();

  echo '<form action="concepto.php" method="POST">';
  echo '<div id="conte">';
  echo '<fieldset>
    <h2>Los Vertebrados</h2>';
    echo "Los vertebrados: ".Save->display().<br><br>';

    echo 'Se agrupan en (Haga clic en cada botón):'
    . '<br><input type="submit" name="grup" value="Mamiferos">';
    echo '<br><input type="submit" name="grup" value="Aves">';
    echo '<br><input type="submit" name="grup" value="Peces">';
    echo '<br><input type="submit" name="grup" value="Reptiles">';
    echo '<br><input type="submit" name="grup" value="Anfibios">';
    // echo "Los vertebrados: ".Save->displayClas().<br>';
  echo '</fieldset>';
  echo '</div>';

  echo '</form>';

```

En la figura 115 se observa la parte del programa que muestra la característica de cada grupo. La información se llama desde las clases con las funciones *display*. La expresión

```
<a href="actividad2.php" target="myFrame">Actividad de Aprendizaje</a>
```

crea un enlace con el que se llama a la página *actividad2.php* y la introduce en un marco (*frame*) llamado *myFrame*. En seguida se le dan las dimensiones al marco.

FIGURA 115. Característica de cada grupo

```

//echo $grup .En mamiferos';
echo '<div id="grupo">';
if( $grup=="Mamiferos")
{
    echo "<h3>Los mamiferos</h3><br>".Smamifero-> displayDescri0.'<br>';

}
elseif( $grup=="Aves")
{
    echo "<h3>Las aves: </h3><br>".Save->display30.'<br>';
}

elseif( $grup=="Peces")
{
    echo "<h3>Los Peces: </h3><br>".Speces->display40.'<br>';
}
elseif( $grup=="Reptiles")
{
    echo "<h3>Los Reptiles: </h3><br>".Sreptiles->display50;
}
elseif( $grup=="Anfibios") {
    echo "<h3>Los anfibios: </h3><br>".Sanfibios->display60;

    echo '<a href="actividad2.php" target= "myFrame"> Actividad de Aprendizaje</a>';
//echo "<iframe id= 'myFrame">";
echo '<fieldset id="fra">';
    echo '<iframe name="myFrame" width="900" height="380" frameborder="0" ></if
    echo "</fieldset>";
}

echo '</div>';
?>
</body>

```

En la figura 116 se muestra el *head* de la actividad de aprendizaje. Se pueden observar dos *divs* con *float left*, donde se construyen dos columnas: en *obj* aparecen las características de un grupo perteneciente a los vertebrados, mientras que en *obj3* aparecen las fotos de los animales.

FIGURA 116. Head de la actividad de aprendizaje

```

<html>
  <head>
    <meta charset="UTF-8">
    <script src="jquery-2.1.1.min.js"></script>
    <script src="jquery-ui.min.js"></script>

    <style>

      #obj{
        float:left;

      }
      #obj3{
        float:left;
      }
    </style>
    <title></title>
  </head>

```

La figura 117 muestra la creación de un objeto llamado *\$mamDes* que recupera la características de la clase mamíferos. Luego aparece la expresión

```
var mamDes = ' <?php echo json_encode( $mamDes); ?> ';
```

que trae un valor asignado en PHP a JavaScript. Posteriormente, se utiliza la expresión

```
draggable = true
```

```
ondragstart = ' drag (event)
```

para convertir un objeto en arrastrable, y con la expresión `)` comienza el arrastre del objeto. Los *drags* son *divs* que contienen las imágenes.

FIGURA 117. Convertir un objeto en arrastrable

```

<body>
  <?php
  include 'mamiferos.php';
  $mami = new mamiferos();
  $mamDes= $mami->descrip2;

  ?>
  <script>
    var mamDes= '<?php echo json_encode( $mamDes); ?>';
    S("body").append("<div id='obj'>");

    S("body").append("<div id='obj2'>");
    S("body").append("<div id='obj3'>");
    var grupo = "Arrastre el animal de la parte derecha que corresponda al grupo
descrito en la parte izquierda<br><br>";
    S("#obj").append(grupo);
    S("#obj3").append("<p id = 'drag1' draggable='true' ondragstart='drag(event)'>
</p>");
    S("#obj3").append("<p id = 'drag2' draggable='true' ondragstart='drag(event)'>
</p>");
    S("#obj3").append("<p id = 'drag3' draggable='true' ondragstart='drag(event)'>
</p>");
    S("#obj3").append("<p id = 'drag4' draggable='true' ondragstart='drag(event)'>
</p>");
    S("#obj3").append("<p id = 'drag5' draggable='true'
ondragstart='drag(event)'></p>");

    S("#drag1").append("<br><br><img src= 'bagre.png' draggable='true'
ondragstart='drag(event)'><span>Para arrastrar</span>");
    S("#drag2").append("<img src= 'canario.png' draggable='true'
ondragstart='drag(event)'><span>Para arrastrar</span>");
    S("#drag3").append("<img src= 'elefente.png' draggable='true'
ondragstart='drag(event)'><span>Para arrastrar</span>");
    S("#drag4").append("<img src= 'culebra.png' draggable='true'
ondragstart='drag(event)'><span>Para arrastrar</span>");
    S("#drag5").append("<img src= 'rana.png' draggable='true'
ondragstart='drag(event)'><span>Para arrastrar</span>");

```

En la figura 118 se muestra cómo se le asigna el color de fondo a los *drags*.

FIGURA 118. CSS de los objetos arrastrables

```

S("#obj").append("<div id='canvis2'>");
S("#canvis2").append("<div id = 'myCanvas2' ondrop='drop(event)'
ondragover='allowDrop(event)'>"+mamDes+"<br><br></div>");
S("#drag1").css(
{
"background": "grey",
"width": 150

});

S("#drag2").css(
{
"background": "grey",
"width": 150});

S("#drag3").css(
{
"background": "grey",
"width": 150});

S("#drag4").css(
{
"background": "grey",
"width": 150});
S("#drag5").css(
{
"background": "grey",
"width": 150});

S("#myCanvas2").css(
{
"background": "grey",
"width": 200,
"color": "white",
"height": 300});

```

En la figura 119 se presentan las funciones que controlan los arrastres y cuándo se suelta el elemento. La expresión `var data=ev.dataTransfer.getData("Text");` recupera el *id* del elemento arrastrado y lo asigna a *data*, con lo que se controla la retroalimentación según el elemento arrastrado. En este caso existen dos mensajes de retroalimentación cuando se escoge la respuesta correcta u otra respuesta diferente, los cuales se pueden observar en el *alert*.

FIGURA 119. Funciones para controlar el arrastre de los objetos

```
function allowDrop(ev)
{
    ev.preventDefault();
}

function drag(ev)
{
    ev.dataTransfer.setData("Text",ev.target.id);
}

function drop(ev)
{
    ev.preventDefault();
    var data=ev.dataTransfer.getData("Text");

    if(data==='drag3')
    {
        ev.target.appendChild(document.getElementById(data));
    }
    else
    {
        alert("El animal seleccionado no tiene las características del grupo dado.");
    }
} </script>

</body>
</html>
```

10.4 Base de datos con MySQL

Para conectar una base con MySQL se utiliza la función que aparece en la figura 120.

FIGURA 120. *Conexión a una base de datos*

```
<?php
$servidor = "localhost";
$usuario = "usuario";
$password = "password";

// Crear la conexión
$conn = new mysqli($servidor, $usuario, $password);

// Verificar la conexión
if ($conn->connect_error) {
    die("Fallo la conexión: " . $conn->connect_error);
}
echo "La conexión tuvo éxito.";
?>
```

A continuación se muestra cómo se crea una base de datos (figura 121) y cómo las tablas (figura 122).

FIGURA 121 *Creación de una base de datos*

```
// Creando la base de datos
$sql = "CREATE DATABASE problema";
if ($conn->query($sql) === TRUE) {
    echo "La base de datos se creó exitosamente";
} else {
    echo "Se presentaron errores en la creación de la base de datos: " . $conn->error;
}

$conn->close();
?>
```

FIGURA 122. Creación de tablas

```

<?php

    // include 'creaBD.php';
    $servidor = "localhost";
    $usuario = "root";
    $password = "miclave";
    $BD = "problemas";

    // Create connection
    $conn = new mysqli($servidor, $usuario, $password, $BD);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // sql to create table
    $sql = "CREATE TABLE pro (
    Codigo_pro INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    Contenido_pro TEXT NOT NULL,
    nomVar_pro VARCHAR(300) NOT NULL,
    descrpcionVar_pro VARCHAR(300) NOT NULL,
    Codigo_niv INT (2)
    )";

    if ($conn->query($sql) === TRUE) {
        echo "Se creó la tabla pro exitosamente.";
    } else {
        echo "Error al tabl pro: " . $conn->error;
    }

    $conn->close();
?>

```

Para insertar datos en una base de datos debemos emplear la función que se muestra en la figura 123.

FIGURA 123. Inserción de datos

```

<?php
    $servidor = "localhost";
    $usuario = "root";
    $password = "miclave";
    $BD= "problemas";

    // Creando LA CONEXIÓN.
    $conn = new mysqli($servidor, $usuario, $password, $BD);
    // Validando conexión.
    if ($conn->connect_error) {
        die("Falló la conexión: " . $conn->connect_error);
    }

    $sql = "INSERT INTO pro (Codigo_pro, nomVar_pro, descrpcionVar_pro,funVar_pro,Codigo_niv)
    VALUES ('Un agricultor desea sembrar mangos y naranjos en su finca.
    Para producir 1 árbol de mango necesita 5 m2 de terreno y
    para un naranjo 3 m2. Proponga un modelo matemático que
    facilite conocer la cantidad de terreno que se necesita para
    sembrar cierta cantidad de estos dos clases de árboles.',
    'mango_naranja_terreno_terreno/mango_terreno/naranja',
    'Cantidad de mangos_Cantidad de naranjos_Cantidad de terreno_terreno/mango_terreno/naranja',
    'independiente1_independiente2_pendiente_constante1_constante2',
    '3')";

    if ($conn->query($sql) === TRUE) {
        echo "Se agregaron registros exitosamente.";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }

    $conn->close();
?>

```

Para leer una base de datos (figura 124) se utiliza un *array* que permite guardar los datos recuperados y enviarlos con *return* al programa que los requiere. Donde se lee «clave» debe escribirse la clave que se utiliza para entrar a la base de datos.

FIGURA 124. Lectura de base de datos

```

<?php

    $niv = 3;
    $problema = BD($niv);

    echo $problema[0]['Contenido_pro'];
    function BD ($_codigo_niv){

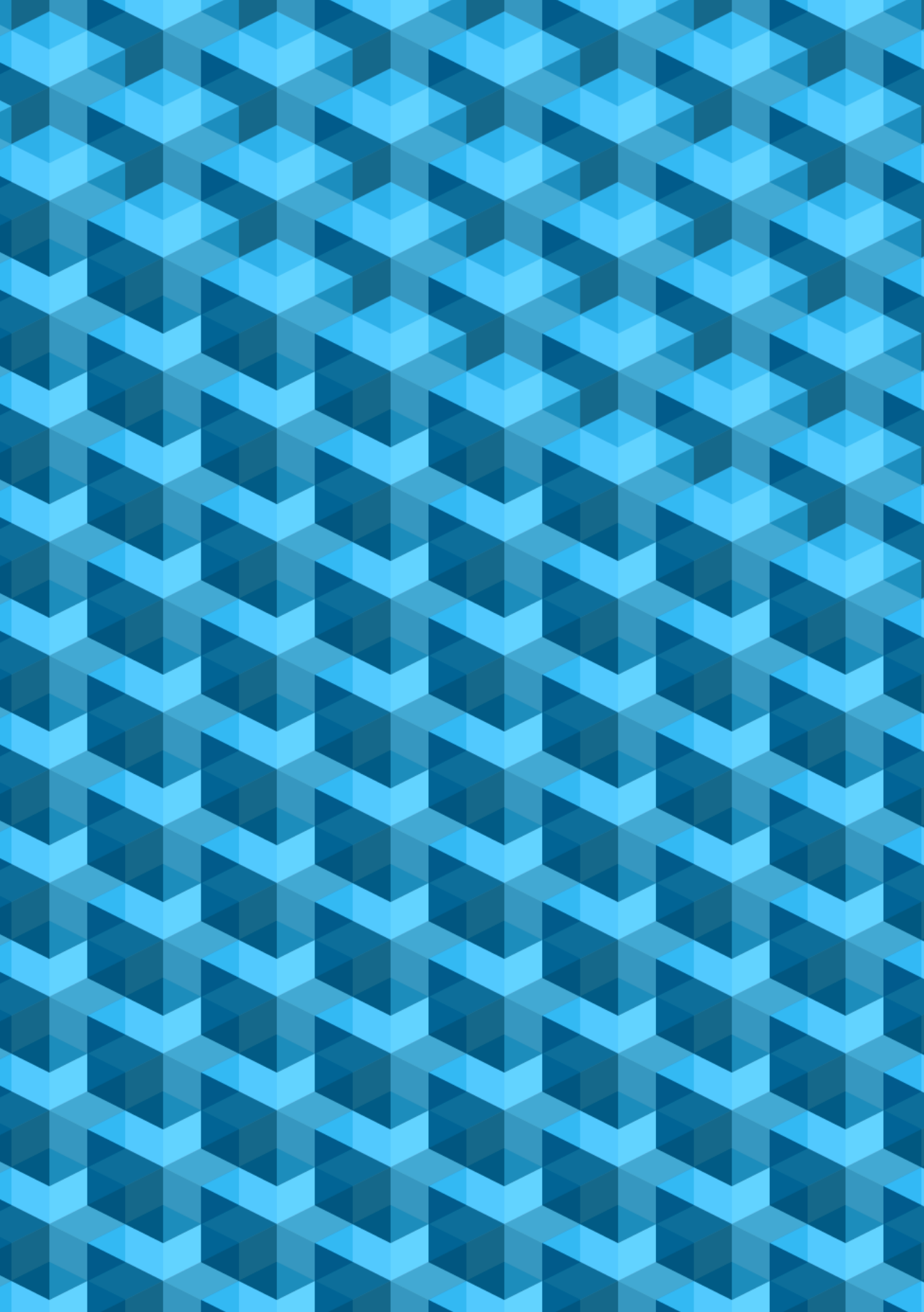
        $codigo_niv=$_codigo_niv;

        $cont =0;
        $problema = Array(
            0 =>Array(
                'Contenido_pro'=>"",
                'nomVar_pro'=>"",
                'DescripcionVar_pro'=>"",
                'funVar_pro'=>"",
                'Codigo_niv'=>""
            )
        );

        $con=mysqli_connect("localhost","root","clave ","problemas");
        $cade="select * from pro where Codigo_niv= ".$_codigo_niv;
        $result2 = \mysqli_query($con,$cade);
        while($row = mysqli_fetch_array($result2))
        {
            $problema[$cont]['Contenido_pro']=$row['Contenido_pro'];
            $problema[$cont]['nomVar_pro']=$row['nomVar_pro'];
            $problema[$cont]['desrcpcionVar_pro']=$row['desrcpcionVar_pro'];
            $problema[$cont]['funVar_pro']=$row['funVar_pro'];
            $problema[$cont]['Codigo_niv']=$row['Codigo_niv'];
            $cont++;
        }
        return $problema;
    }
?>

```

En conclusión, hemos pretendido mostrar al lector la posibilidad de poner en práctica las teorías del aprendizaje al servicio del diseño de material instruccional, para apoyar el proceso de enseñanza-aprendizaje. El lector que se quiera aventurar en esta tarea puede utilizar las teorías de sus preferencias, no necesariamente las que se dan aquí. Lo importante es que verifique a través de la investigación el efecto que esto puede tener en el aprendizaje de nuestros estudiantes.



Referencias

- Bangert-Drowns, R. L., Kulik, C.-L. C., Kulik, J. A., & Morgan, M. (1991). The instructional-effect of feedback in test-like events. *Review of Educational Research*, 61(2), 213-238.
- Barker, P. (1994). Designing interactive learning. En T. de Jong & L. Sarti (Eds.), *Design and production of multimedia and simulation-based learning material*. Dordrecht: Kluwer Academic Publishers.
- Bransford, J., & Stein, B. S. (1983). *The IDEAL problem solver: A guide for improving thinking, learning, and creativity*. Nueva York: W. H. Freeman.
- Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A study of thinking*. Nueva York: Wiley.
- Chan, M. E. (2001). *Objetos de aprendizaje: una herramienta para la innovación educativa*. Innova, U de G. Recuperado de www.udgvirtual.udg.mx/apertura/pdfs/.../RevApertura_Dic2002.pdf
- Chi, M. T. H., Feltovich, P. J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5(1), 121-152.
- Chi, M. T. H., & VanLehn A. K. (2012). Seeing deep structure from interactions of surface features. *Educational Psychologist*, 47(3), 177-188.

- Clariana, R. B., Wagner, D., & Murphy, L. C. (2000). Applying a connectionist description of feedback timing. *Educational Technology Research and Development*, 43(3), 5-21.
- Cooper, G. (2003). Cognitive load theory as an aid for instructional design. *Australian Journal of Educational Technology*, 6(2), 108-113.
- Corbalan, G., Kester, L., & Van Merriënboer, J. J. G. (2009). Dynamic task selection: Effects of feedback and learner control on efficiency and motivation. *Learning and Instruction*, 19(6), 455-465.
- Day, S. B., & Goldstone, R. L. (2012). The import of knowledge expert: Connecting findings and theories of transfer of learning. *Educational Psychologist*, 47(3), 153-176.
- Dempsey, J. V. (1988). The effects of four methods of immediate corrective feedback on retention, discrimination error, and feedback study time in computer-based instruction. (Tesis de grado doctoral), Florida State University, 1988). *Dissertation Abstracts International*, 49, 1434A.
- Dempsey, J. V. (1990). *Learning successive and coordinate concepts*. Mobile, AL: University of South Alabama, Mobile.
- Elley, W. B. (1966). The role of errors in learning with feedback. *British Journal of Educational Psychology*, 36(3), 296-300.
- Flowerday, T., & Schraw, G. (2000). Teachers' beliefs about instructional choice: A phenomenological study. *Journal of Educational Psychology*, 92(4), 634-645.
- Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology*, 15(1), 1-38.
- Goodman, J. S., & Wood, R. E. (2004). Feedback specificity, learning opportunities, and learning. *Journal of Applied Psychology*, 89(5), 809-821.
- Greeno, J. G. (1991a). Number sense as situated knowing in a conceptual domain. *Journal for Research in Mathematics Education*, 22(3), 170-218.
- Groner, R., & Kersten, B. (2001). *Entwurf eines intelligenten feedbacks in tutoriellen lernsystemen*. Seminararbeit am Institut für Psychologie der Universität Bern, Sommersemester. Recuperado de <http://visor.unibe.ch/SS01/evaluation/TypologieLernsysteme.pdf>

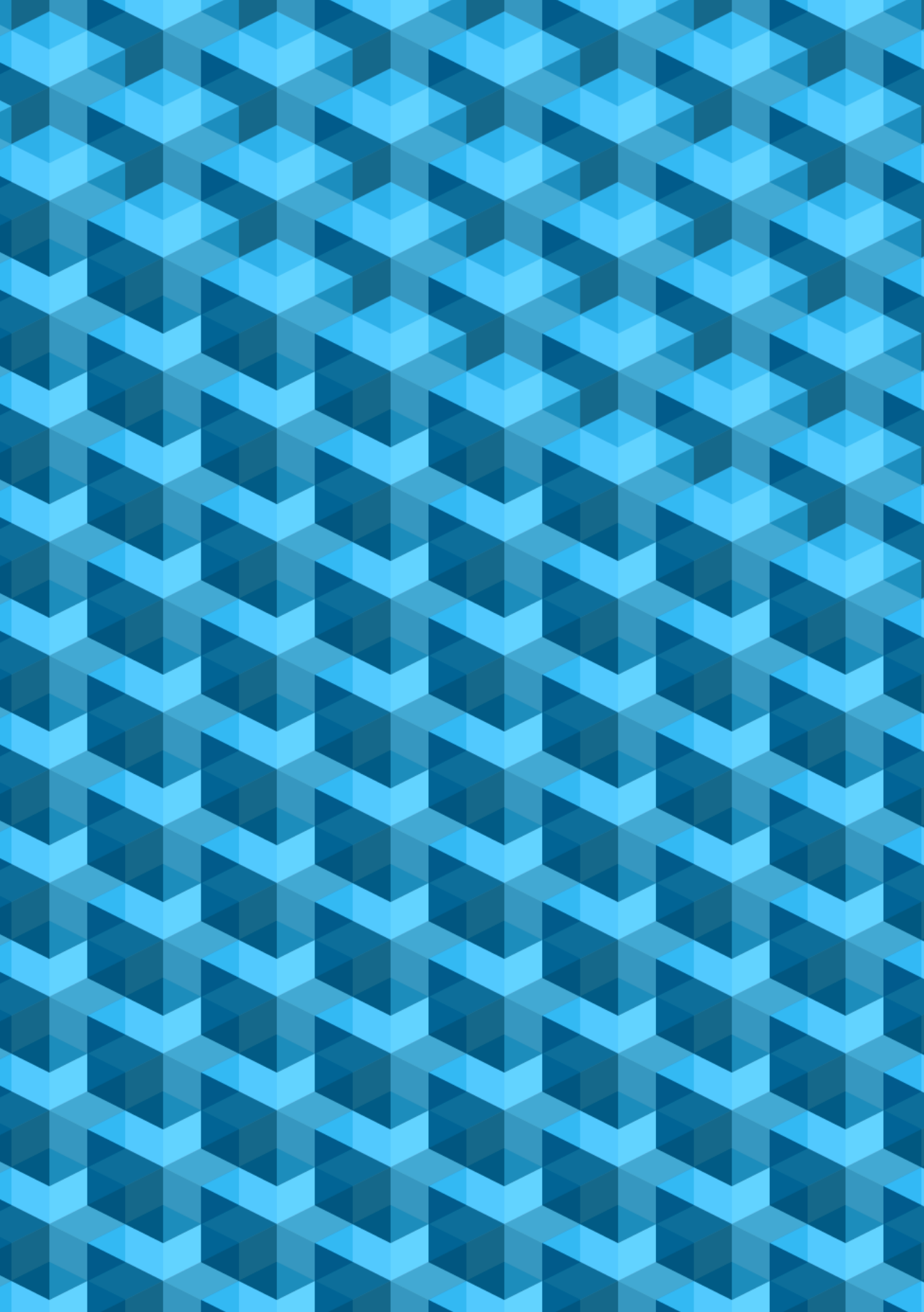
- Gulmans, J. (1998). *The acquisition of coordinate concepts in health education*. Enschede: Universiteit Twente.
- Hansen, J. B. (1974). Effects of feedback, learner control, and cognitive abilities on state anxiety and performance in a CAI task. *Journal of Educational Psychology*, 66(2), 247-254.
- Harrop, A. G. (2003). *A prototype computer based-learning environment for teaching calculator representations*. University of Leeds. Recuperado de www.ppi.org/papers/11th-harrop
- Janetzko, D. (1999). *Künstliche Dialoge-Dialog-und Interviews-Bots für das WWW*. Recuperado de www.psychol.uni-giesen.de/~Batinic/kapitel/janetzko.pdf
- Johnson-Laird, P. N. (1983). *Mental models*. Cambridge: Cambridge University Press.
- Jonassen, D. H. (2006). Toward a meta-theory of problem solving. Missouri, Columbia: University of Missouri. Recuperado de <http://coe.missouri.edu/~jonassen/problems.htm>
- Jonassen, D. H., Tessmer, M., & Hannum, W. H. (1999). *Task analysis methods for instructional design*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Jong, T. de, Ainsworth, S., Dobson, M., Hulst, A. van der, Levonen, J., Reimann, P., Sime, J. A., Someren, M. van, Spada, H., & Swaak, J. (1998). Acquiring knowledge in science and mathematics: The use of multiple representation in technology-based learning environments. En Maarten W. van Someren (Ed.), *Advances in learning and instructions series: Learning with multiple representation* (pp. 9-41). Oxford: Elsevier Science.
- Kinzie, M. B., Sullivan, H. J., & Berdel, R. L. (1988). Learner control and achievement in science computer-assisted instruction. *Journal of Educational Psychology*, 80(3), 299-303.
- Klausmeier, H. J., & Feldman, K. V. (1975). Effects of a definition and a varying number of examples and non-examples on concept attainment. *Journal of Educational Psychology*, 67(2), 174-178.
- Kulhavy, R. W. (1977). Feedback in written instruction. *Review of Educational Research*, 47(1), 211-232.

- Kulhavy, R. W., & Stock, W. A. (1989). Feedback in written instruction: The place of response certitude. *Educational Psychology Review*, 1(4), 279-308.
- Lee-Sammons, W. H., & Wollen, K. A. (1989). Computerized practice tests and effects on in-class exams. *Behavioral Research Methods Instruments & Computers*, 21(2), 189-194.
- Lemley, D. C. (2005). *Delayed versus immediate feedback in an independent study high school setting*. Utah: Brigham Young University.
- Lobato J. (2012). The actor-oriented transfer perspective and its contributions to educational research and practice. *Educational Psychologist*, 47(3), 232–247.
- Lusti, M. (1992). *Handbuch der informatik: Intelligente tutorielle systeme. Einführung in wissensbasierte Lernsysteme*. München, Wien: Oldenbourg.
- Markle, S. M., & Tiemann, P. W. (1970). *Really understanding II*. Champaign, IL: Stipes.
- Markle, S. M., & Tiemann, P. W. (1971). Conceptual learning and instructional design. En M. D. Merrill (Ed.), *Instructional design: Reading*. Englewood Cliffs, NJ: Prentice-Hall.
- Mason, B. J., & Bruning, R. (2001). *Providing feedback in computer-based instruction: What the research tells us*. Center for Instructional Innovation, University of Nebraska-Lincoln. Recuperado de <http://dwb.unl.edu/Edit/MB/Mason-Bruning.html>
- Merrill, M. D., & Tennyson, R. D. (1978). Concept classification and classification errors as a function of relationship between examples and non-examples. *Improving Human Performance*, 7(4), 351-364.
- Meyer, L. (1986). Strategies for correcting students' wrong responses. *Elementary School Journal*, 87(1), 227-241.
- Mory, E. H. (2004). *Feedback research revisited*. Wilmington: University of North Carolina.
- Narciss, S. (2001). *Informative feedback as a bridge from instruction to learning in computer-based trainings*. Dresden: Institut für Psychologie IV.

- Newkirk, R. L. (1973). A comparison of learner control and machine control strategies for computer-assisted instruction. *Programmed Learning and Educational Technology*, 10(2), 82-91.
- Otero, L. (2010a). Efecto de los niveles de especificidad de la retroalimentación sobre la competencia en resolución de problemas. En *Encuentro interdisciplinario de grupos de investigación* (pp. 47-60). Bogotá: Corporación Universitaria Unitec®.
- Otero, L. (2010b). *Análisis, diseño, desarrollo e implementación de un ambiente de un aprendizaje computacional basado en el manejo de errores*. Investigación no publicada. Corporación Universitaria Unitec, Sistema Institucional de Investigación, Bogotá, Colombia.
- Otero, L. (2012). Manejo de errores en un dominio de conocimiento y la competencia para la resolución de problemas. *Revista Científica*, (15), 67-77.
- Otero, L. (2013). *Análisis, diseño, desarrollo e implementación de un objeto aprendizaje basados en la teoría de la transferencia*. Investigación no publicada. Corporación Universitaria Unitec, Sistema Institucional de Investigación, Bogotá, Colombia.
- Otero, L. (2014a). El efecto del control de la retroalimentación sobre la competencia de resolución de problemas. *Revista Científica*, (18), 100-111.
- Otero, L. (2014b). Efectos de la transferencia de las estructuras profundas y del control de la retroalimentación sobre la competencia de la resolución de problemas. En *Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación*, Buenos Aires, Argentina.
- Pavio, A. (1990). *Mental representations: A dual coding approach*. Nueva York: Oxford University Press.
- Perales P. J., Álvarez S. P., Fernández G. M., García, G. J., & Rivarossa, A. (2000). *Resolución de problemas*. Madrid: Síntesis.
- Perkins, D. (2003). *La escuela de la inteligencia*. Barcelona: Gedisa.
- Perkins, D., & Salomon, G. (1992). *Transfer of learning*. Oxford: Pergamon Press.
- Perkins, D., & Salomon G. (2012). Knowledge to go: Motivational and dispositional view of transfer. *American Psychological Association*, 47(3), 248-258.

- Phye, G. D., & Andre, T. (1989). Delayed retention effect: Attention, perseveration, or both? *Contemporary Educational Psychology*, 14(2), 173-185.
- Piaget, J. (1972). *To understand is to invent*. Nueva York: The Viking Press.
- Pontificia Universidad Javeriana (2009). *Objetos de aprendizaje: prácticas y perspectivas educativas*. Cali: Pontificia Universidad Javeriana.
- Pridemore, D. R., & Klein, J. D. (1991). Control of feedback in computer-assisted instruction. *Educational Technology Research and Development*, 39(4), 27-32.
- Roper, W. J. (1977). Feedback in computer-assisted instruction. *Programmed Learning and Educational Technology*, 14(1), 43-49.
- Rosch, E. (1978). Principles of categorization. En E. Rosch & B. L. Lloyd (Eds.), *Cognition and categorisation* (pp. 27-48). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Ross, S. M., & Morrison, G. R. (1989). In search of a happy medium in instructional technology research: Issues concerning external validity, media replications, and learner control. *Educational Technology Research & Development*, 37(1), 19-33.
- Sassenrath, J. M., & Garverick, C. M. (1965). Effects of differential feedback from examinations on retention and transfer. *Journal of Educational Psychology*, 56(5), 259-263.
- Schloss, P. J., Wisniewski, L. A., & Cartwright, G. P. (1988). The differential effect of learner control and feedback in college student's performance on CAI modules. *Journal of Educational Computing Research*, 4(2), 141-150.
- Schulmeister, R. (1996). *Grundlagen hypermedialer lernsysteme: Theorie, didaktik, design*. Bonn: Addison-Wesley.
- Schwier, R. A., & Misanchuk, E. (1993). *Interactive multimedia instruction*. Englewood Cliffs, NJ: Educational Technology Publications.
- Sims, R. (1997, 27 de enero). *Interactivity: A forgotten art?* Recuperado de <http://intro.base.org/docs/interact>
- Skinner, B. F. (1958). Teaching machines. *Science*, 128(3330), 969-977.

- Smith, M. U. (1991). A view from biology. En M. U. Smith (Ed.), *Toward a unified theory of problem solving*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257-285.
- Tennyson, R. D., & Cocchiarella, M. J. (1986). An empirically based instructional design theory for teaching concepts. *Review of Educational Research*, 56(1), 40-71.
- Thompson, P. W. (2011). Quantitative reasoning and mathematical modeling. En L. L. Hatfield, S. Chamberlain & S. Belbase (Eds.), *New perspectives and directions for collaborative research in mathematics education*. Wisdom Monographs (vol. 1, pp. 33-57). Laramie: University of Wyoming Press.



Anexos

Anexo A. CSS

Para darle estilo a una hoja web se utiliza el CSS. Aquí se hará una breve descripción de algunos elementos. La sintaxis es:

Selector	Propiedad	Valor
H1:	font-size	30px;

Ejemplo:

```
#bot{
    font-size:10px;
    font-family:Verdana,Helvetica;
    font-weight:bold;
    color:white;
    background:#638cb5;
    border:0px;
    width:180px;
    height:19px;
}
```

Los colores se pueden dar en forma hexadecimal (#638cb5) o utilizando el nombre de los colores (*red*). Los tamaños se miden en porcentajes de unidades,

entre otras, px, em, pt y cm. La unidad px corresponde a pixeles; 1 em es igual a 16 px; pt corresponde 1/72 pulgadas; y cm representa a los centímetros.

Se puede programar con la etiqueta `<style></style>` al comienzo de la página o escribir un archivo aparte con la extensión .css (punto css), para darle formato a un elemento de la página, por ejemplo, h1:

```
h1 {  
  color: blue;  
  font-size = 20px;  
}
```

Si se trata de un elemento con *id*, por ejemplo que se tenga `<div id = "men"></div>`, se representa así:

```
#men {  
  color: blue;  
  font-size = 20px;  
}
```

Por otra parte, si se trata de una clase, por ejemplo, `<div class = "men"></div>`, se representa así:

```
.men {  
  color: blue;  
  font-size = 20px;  
}
```

Ejemplo:

```
.conce{  
  padding-left: 5px;  
  position: absolute;  
  top:65px;  
}
```

La clase se reconoce por comenzar con un punto (.).

Los archivos externos se referencian en el *head*:

```
<link href="estilos/css.css" rel="stylesheet"/>
```

Y se escriben:

```
#last{
  background-color: #cccccc;
  position: absolute;
  bottom:45px;
  width: 750px;
  height: 30px;
  opacity: 0.5;
}
```

Un ejemplo en una página web es:

```
<style>
  #destino{
    position: absolute;
    //float:right;
    top:40px;
    left: 320px;
  }
</style>
```

Existen 17 palabras clave para los colores:

**aqua, black, blue, fuchsia, gray, green, lime, maroon, navy,
olive, orange, purple, red, silver, teal, white, yellow.**

Los colores también se pueden representar en RGB: `p {color: rgb (71, 98, 176) ;}`. El RGB corresponde a rojo, verde y azul (red, green, blue) y cada parámetro define la intensidad del color entre 0 y 255. Por ejemplo, `rgb (255,0,0)` corresponde al color rojo de más alto valor, los otros valores están en cero.

En la figura A1 se pueden ver los componentes que se utilizan en un menú para navegar.

FIGURA A1. Estilos en navegación

```

<!DOCTYPE html>
<html>
<head>
<style>
a:link {
    text-decoration: none;
}

a:visited {
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}

a:active {
    text-decoration: underline;
}
</style>
</head>
<body>

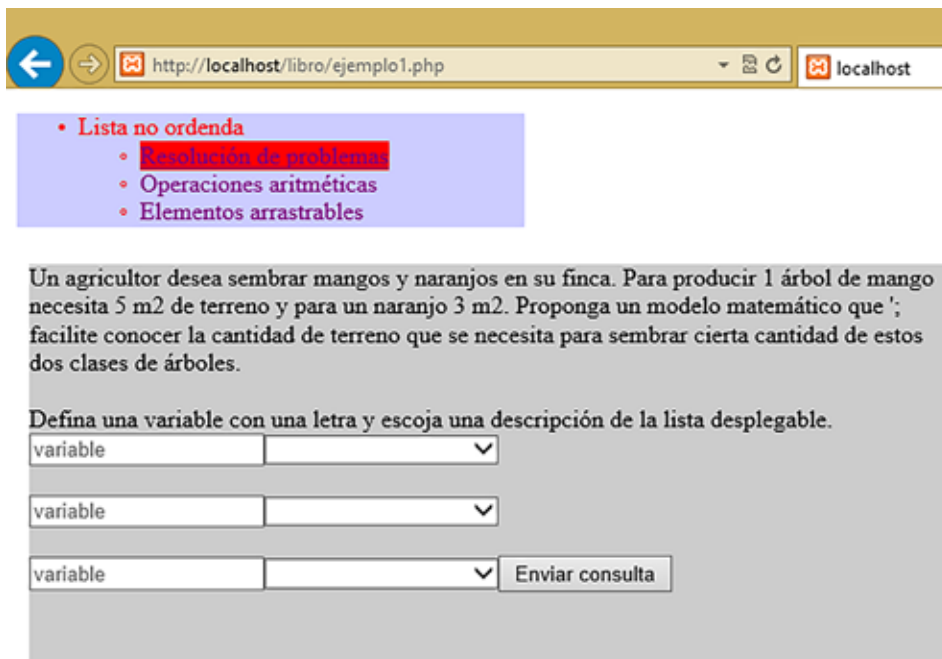
<p><b><a href="suma.php" target="_blank">Esto es un enlace</a></b></p>
<p><b>Advertencia:</b> un :hover se debe escribir después de un :link y un:visited
en la definición CSS para que se ejecute.</p>
<p><b>Advertencia:</b> un :active debe ir situado después un: hover en la
definición de CSS para que se ejecute.</p>

</body>
</html>

```

En la figura A2 aparece un menú realizado con CSS. Se puede observar que el *hover* permite siempre que se ubique el cursor sobre una de las opciones, haciendo que el fondo cambie a rojo. En la figura aparece sobre la primera opción.

FIGURA A2. Menú CSS con una opción seleccionada



Tenga en cuenta el orden de programación de los enlaces para que el programa pueda funcionar adecuadamente (figura A3): primero *link*, en seguida *visited*, después *hover* y, finalmente, *active*. En la parte derecha se encuentran elementos que ya se han explicado.

FIGURA A3. Programación del programa Menú

```

<html>
  <head>
    <meta charset="UTF-8">
    <script src='problema/eje2.php'></script>
    <style>
      #menu{
        background: #ccccff;
        width: 330px;      }

      a:link {
        text-decoration: none;
      }

      a:visited {
        text-decoration: none;
      }

      a:hover {
        text-decoration: underline;
        background: red;
      }

      a:active {
        text-decoration: underline;
      }

      li{
        color:red;
      }
    </style>
    <title></title>
  </head>

  <body>
    <div id = 'menu'>
      <ul>
        <li> Lista no ordenda
          <ul>
            <li><a href="problema/eje2.php"
              target="unMarco">Resolución de problemas</a>
            </li>

            <li><a href="caja.php"
              target="unMarco">Operaciones aritméticas</a>
            </li>

            <li><a href="drag_drop.php"
              target="unMarco">Elementos arrastrables</a>
            </li>
          </ul>
        </li>
      </ul>
    </div>

    <?php
      // put your code here
    ?>

    <iframe name="unMarco" width="1200"
      height="700" frameborder="0" ></iframe>
  </body>
</html>

```

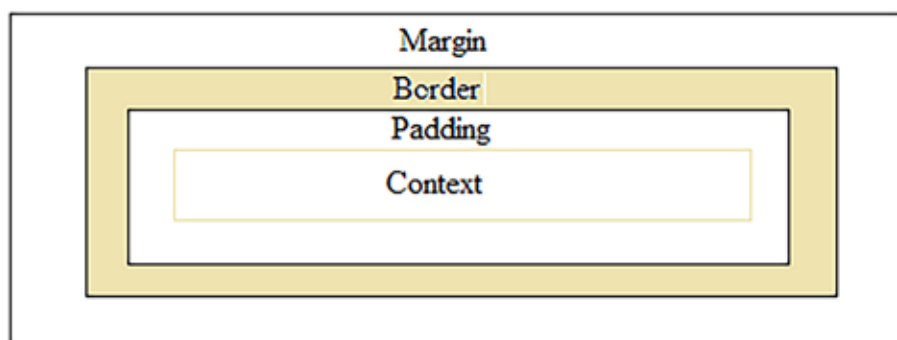
Se pueden especificar otros marcadores diferentes al círculo que acompaña a los enlaces con *list-style*. Pueden ser: *circle*, *square* y *lower-alpha*. Por ejemplo:

```

ul li{
  list-style-type: square;
}

```

Los elementos en CSS son cajas que contienen márgenes (*margin*), bordes (*border*), *padding* y contenido. En la figura A4 se muestra el orden en que se presentan.

FIGURA A4. *Modelo de caja CSS*

Hay muchos elementos que no fueron tratados en este anexo pero que puede obtener en Internet. Se recomienda especialmente consultar la página <http://www.w3schools.com>

Anexo B.

Una clase en PHP y en JavaScript

La clase en PHP se construye con la palabra clave `class`. En la figura B1 se muestra una clase en PHP con sus métodos, en los que no hay nada programado, así que solo se utiliza como ilustración. Para declarar un objeto se procede así:

```
$nissan = new vehiculo( );
```

Para llamar alguna de las funciones, por ejemplo, `parar`:

```
$nissan → parar( );
```

FIGURA B1. Una clase y sus métodos en PHP

```
Class vehiculo {  
  
    var $llanta;  
    var $volante;  
    var $motor;  
    var $asientos;  
  
    public function parar()  
    {  
  
    }  
  
    public function acelerar()  
    {  
  
    }  
  
    public function doblar()  
    {  
  
    }  
}
```

En la figura B2 se muestra la misma clase pero construida con JavaScript. Para definir un objeto:

Para invocarlo:

FIGURA B2. *Una clase en JavaScript*

```
function vehiculo()
{
  var llanta;
  var volante;
  var motor;
  var asientos;

  this.parar= function ()
  {
  };
  this.acelerar = function()
  {
  };
  this.doblar = function()
  {
  };
}
```

Anexo C.

Estructuras de control

Las estructuras de control sirven para generar reglas de decisión, entre las cuales está el *if* (figura C1). Si la condición es verdadera se ejecuta lo que está dentro del *if*; de lo contrario se realiza lo que está dentro del *elseif*; finalmente, si no es ninguna de las anteriores, se ejecuta el *else*. Se pueden construir tantos *elseif* como se necesite.

FIGURA C1. Estructura de control *if* en PHP

```
if (condicion = true)
{
}
elseif
{
}

else
{
}
```

En la figura C2 se muestra la estructura de control *switch*, que permite correr una serie de opciones de acuerdo con el valor de una variable.

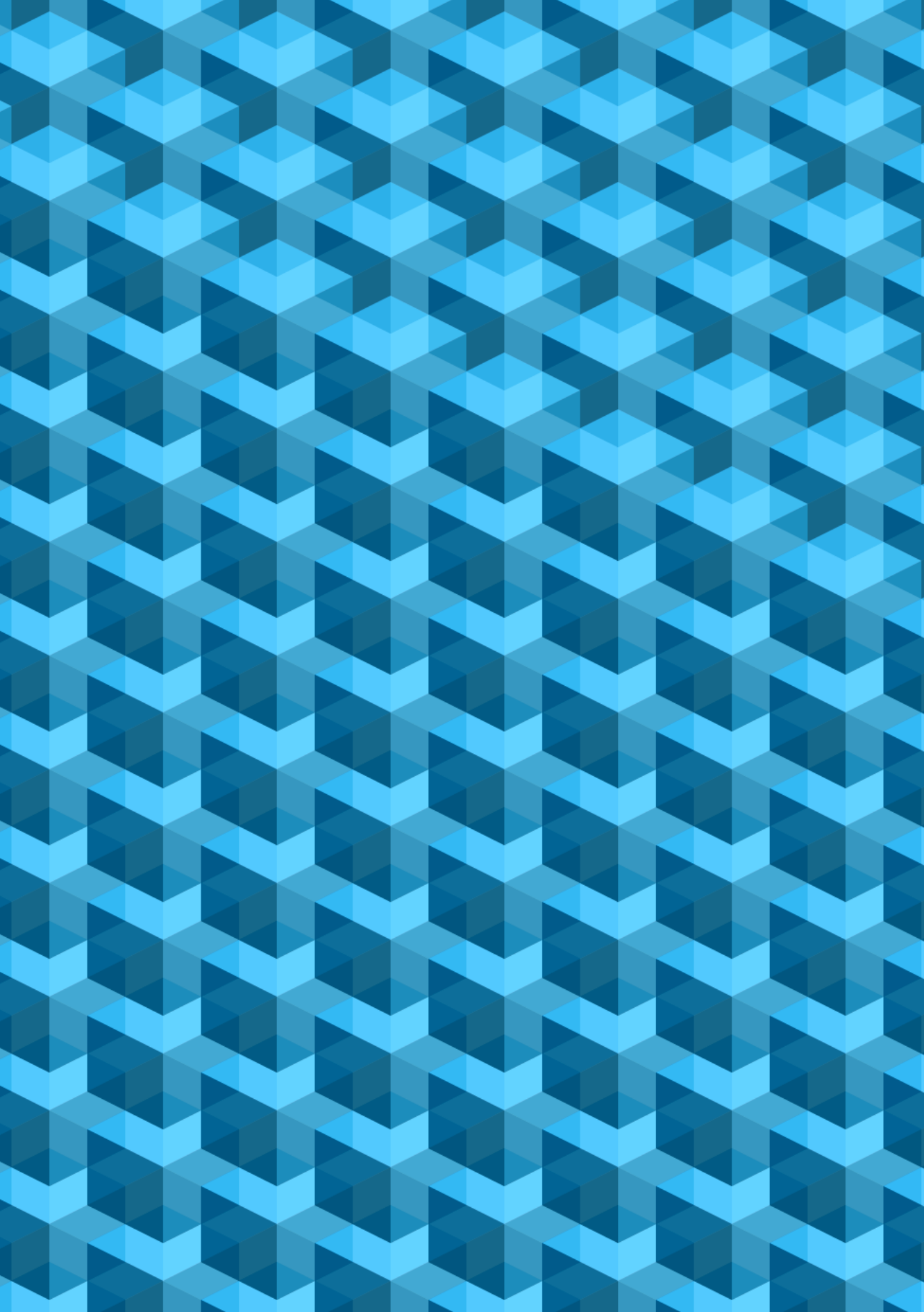
FIGURA C2. Estructura de control *switch*

```
switch ($valor) {
  case 1:
    Se ejecuta esta sentencia si $valor =1;
    break;
  case 2:
    Se ejecuta esta sentencia si $valor =2;
    break;
  case 3:
    Se ejecuta esta sentencia si $valor =3;
    break;
  ...
  default:
    Si ninguna de las anteriores se ejecuta, se ejecuta ésta.;
}
```

En la figura C3 aparecen otras estructuras de control. La función *foreach* solo se utiliza para *array*.

FIGURA C3. Otras estructuras de control

```
while (condición = true) {  
    Se ejecuta este código ;  
}  
  
do {  
    Código a ejecutarse;  
} while (condicion =true);  
  
for ($i=valor inicial; valor final; incremento ) {  
    Sentencias a ejecutarse;  
}  
  
foreach ($array as $valor) {  
    Sentencias a ejecutarse;  
}
```



Índice de Figuras y tablas

FIGURA 1. <i>Combinación de respuestas según los tipos de errores.</i>	21
FIGURA 2. <i>Análisis de la tarea.</i>	33
FIGURA 3. <i>Competencia en la resolución de problemas</i>	50
FIGURA 4. <i>Modelo de proceso de información de la arquitectura cognitiva humana</i>	60
FIGURA 5. <i>Relación entre los componentes de la carga cognitiva</i>	64
FIGURA 6. <i>Carga cognitiva intrínseca baja</i>	64
FIGURA 7. <i>Carga cognitiva intrínseca alta</i>	65
FIGURA 8. <i>Arquitectura de un sistema experto basado en conocimiento</i>	71
FIGURA 9. <i>Montaje de un programa de instrucciones asistidas por computador</i>	72
FIGURA 10. <i>Secuencia de un sistema de tutor inteligente ideal</i>	73
FIGURA 11. <i>Árbol de solución de un problema</i>	81
FIGURA 12. <i>Ejemplos de presentación de formas expositivas</i>	103
FIGURA 13. <i>Ejemplo de presentación de una forma interrogativa</i>	104
FIGURA 14. <i>Determinación de la pendiente</i>	109
FIGURA 15. <i>Tablas de la base de datos PLINKI.</i>	117
FIGURA 16. <i>Clases de un objeto de aprendizaje</i>	118
FIGURA 17. <i>Diagrama de la sección teórica</i>	119
FIGURA 18. <i>Menú de la parte teórica</i>	119
FIGURA 19. <i>Concepto de variable</i>	120
FIGURA 20. <i>Actividad de aprendizaje 1</i>	121
FIGURA 21. <i>Respuesta correcta actividad de aprendizaje 1</i>	122

FIGURA 22. <i>Relación entre variables</i>	123
FIGURA 23. <i>Actividad de aprendizaje 2</i>	124
FIGURA 24. <i>Respuesta correcta actividad de aprendizaje 2</i>	125
FIGURA 25. <i>Variable dependiente</i>	126
FIGURA 26. <i>Actividad de aprendizaje 3</i>	127
FIGURA 27. <i>Respuesta correcta actividad de aprendizaje 3</i>	128
FIGURA 28. <i>Variable independiente</i>	128
FIGURA 29. <i>La variable independiente en un contexto problemático</i>	129
FIGURA 30. <i>Identificación de la variable independiente</i>	130
FIGURA 31. <i>Constante de proporcionalidad</i>	131
FIGURA 32. <i>Constante de proporcionalidad</i>	133
FIGURA 33. <i>Diagrama de modelación matemática</i>	134
FIGURA 34. <i>Algoritmo para escoger nivel</i>	134
FIGURA 35. <i>Definición de variables</i>	135
FIGURA 36. <i>Actividad de identificación de variables</i>	136
FIGURA 37. <i>Variables ya definidas</i>	137
FIGURA 38. <i>Ejemplo modelo matemático nivel 1</i>	138
FIGURA 39. <i>Actividad para el nivel 1</i>	139
FIGURA 40. <i>Teoría de problemas nivel 2</i>	142
FIGURA 41. <i>Actividad nivel 2</i>	143
FIGURA 42. <i>Teoría de resolución de problemas nivel 3</i>	144
FIGURA 43. <i>Actividad nivel 3</i>	145
FIGURA 44. <i>Parte de los problemas nivel 4</i>	146
FIGURA 45. <i>Actividad para los problemas nivel 4</i>	147
FIGURA 46. <i>Aspectos básicos en HTML</i>	151
FIGURA 47. <i>Estructura de una página de HTML</i>	151
FIGURA 48. <i>Títulos y elementos resaltados</i>	152
FIGURA 49. <i>Resultados de títulos y párrafos resaltados</i>	153
FIGURA 50. <i>Resultado de listas ordenadas y no ordenadas</i>	153

FIGURA 51. <i>Programación de listas ordenadas y no ordenadas</i>	154
FIGURA 52. <i>Capturando información</i>	155
FIGURA 53. <i>Función saludos</i>	157
FIGURA 54. <i>Programa hola.php</i>	158
FIGURA 55. <i>Programa Preguntando un nombre</i>	158
FIGURA 56. <i>Respuesta del programa Saludos</i>	158
FIGURA 57. <i>Programa de operaciones aritméticas 1</i>	160
FIGURA 58. <i>Programa operaciones aritméticas 2</i>	161
FIGURA 59. <i>Función con this</i>	162
FIGURA 60. <i>Programa operaciones aritméticas 3</i>	163
FIGURA 61. <i>Resultado de operaciones aritméticas</i>	164
FIGURA 62. <i>Resultado de operaciones aritméticas</i>	165
FIGURA 63. <i>Ejemplos de botones de radio</i>	166
FIGURA 64. <i>Determinación de la opción de los botones de radio.</i>	167
FIGURA 65. <i>Resultado de botones de radio</i>	167
FIGURA 66. <i>Funciones para arrastrar y soltar</i>	168
FIGURA 67. <i>Elementos arrastrables</i>	169
FIGURA 68. <i>Programación de elementos arrastrables</i>	170
FIGURA 69. <i>Función arrastrable</i>	171
FIGURA 70. <i>Función soltar</i>	172
FIGURA 71. <i>Mostrando información en PHP</i>	173
FIGURA 72. <i>Alcance de las variables</i>	174
FIGURA 73. <i>Declaración de un array en PHP</i>	175
FIGURA 74. <i>Array multidimensional</i>	175
FIGURA 75. <i>Una clase en PHP</i>	176
FIGURA 76. <i>Lista desplegable del problema</i>	177
FIGURA 77. <i>Caja de texto para modelo matemático</i>	178
FIGURA 78. <i>Diagrama de flujo del experto</i>	180
FIGURA 79. <i>Módulo de enseñanza (primera parte)</i>	182

FIGURA 80. <i>Validando ecuación del usuario</i>	183
FIGURA 81. <i>Descomposición de la ecuación del usuario</i>	184
FIGURA 82. <i>Error variable dependiente</i>	185
FIGURA 83. <i>Error en la primera constante</i>	186
FIGURA 84. <i>Error en la variable del segundo término</i>	187
FIGURA 85. <i>Error en la variable dependiente 2</i>	188
FIGURA 86. <i>Head del programa Problema</i>	189
FIGURA 87. <i>Clase problemas</i>	189
FIGURA 88. <i>Clase problemas (segunda parte)</i>	190
FIGURA 89. <i>Captura de datos con un form</i>	191
FIGURA 90. <i>Segundo y tercer select</i>	192
FIGURA 91. <i>Validación de contenido variable</i>	193
FIGURA 92. <i>Función que evita repetición de variables.</i>	194
FIGURA 93. <i>Captación de falta descripciones o definiciones</i>	194
FIGURA 94. <i>Verificación si la información está correcta y completa</i>	195
FIGURA 95. <i>Construcción de la clase módulo experto</i>	196
FIGURA 96. <i>Modelo experto (segunda parte)</i>	197
FIGURA 97. <i>La construcción de la ecuación por el experto</i>	198
FIGURA 98. <i>Función que recorta espacios en blanco</i>	199
FIGURA 99. <i>Módulo de enseñanza (primera parte)</i>	200
FIGURA 100. <i>Módulo de enseñanza (segunda parte)</i>	201
FIGURA 101. <i>Módulo de enseñanza (tercera parte)</i>	202
FIGURA 102. <i>Módulo de enseñanza (cuarta parte)</i>	203
FIGURA 103. <i>Problema propuesto</i>	204
FIGURA 104. <i>Errores en la formulación del modelo matemático</i>	204
FIGURA 105. <i>Formulación correcta del modelo matemático</i>	205
FIGURA 106. <i>Tareas de discriminación de atributos de un concepto.</i>	207
FIGURA 107. <i>Estructura del concepto vertebrado</i>	208
FIGURA 108. <i>Clase vertebrado</i>	209

FIGURA 109. Clase mamíferos (primera parte)	210
FIGURA 110. Clase mamíferos (segunda parte)	211
FIGURA 111. Características de los anfibios	212
FIGURA 112. Actividad de aprendizaje de generalización de concepto	212
FIGURA 113. CSS del programa	213
FIGURA 114. Forma del menú agrupación	214
FIGURA 115. Característica de cada grupo	215
FIGURA 116. Head de la actividad de aprendizaje	216
FIGURA 117. Convertir un objeto en arrastrable	217
FIGURA 118. CSS de los objetos arrastrables	218
FIGURA 119. Funciones para controlar el arrastre de los objetos	219
FIGURA 120. Conexión a una base de datos	220
FIGURA 121. Creación de una base de datos	220
FIGURA 122. Creación de tablas	221
FIGURA 123. Inserción de datos	222
FIGURA 124. Lectura de base de datos	223
FIGURA A1. Estilos en navegación	236
FIGURA A2. Menú CSS con una opción seleccionada	237
FIGURA A3. Programación del programa Menú	238
FIGURA A4. Modelo de caja CSS	239
FIGURA B1. Una clase y sus métodos en PHP	240
FIGURA B2. Una clase en JavaScript	241
FIGURA C1. Estructura de control if en PHP	242
FIGURA C2. Estructura de control switch	242
FIGURA C3. Otras estructuras de control	243

TABLA 1. <i>Ejemplos tripleta objeto, atributo, valor</i>	88
TABLA 2. <i>Modelo de enseñanza de aprendizaje basados en los conceptos de Tennyson y Cochiarella.</i>	100
TABLA 3. <i>Matriz de unidades de conocimiento versus operaciones cognitivas.</i>	115
TABLA 4. <i>Matriz de unidades de conocimiento versus operaciones cognitivas para los problemas.</i>	132
TABLA 5. <i>Tipo de datos en JavaScript</i>	156
TABLA 6. <i>Tipos de datos</i>	174
TABLA 7. <i>Valores del problema</i>	179
TABLA 7. <i>Variables del problema</i>	177
TABLA 8. <i>Unidades de conocimiento y operaciones cognitivas para el problema.</i>	206



Esta obra se terminó de imprimir en junio de 2016
con tipo Myriad Pro con punto 11/15,2
sobre papel Bond de 75 gramos
en Molher Impresores Ltda.
Bogotá, D. C., Colombia

