



MODELANDO OBJETOS DE APRENDIZAJE a partir de la psicología del aprendizaje

Luis Eduardo Otero Sotomayor

 CORPORACIÓN UNIVERSITARIA

UNITEC



Modelando objetos de aprendizaje a partir de la psicología del aprendizaje



**Modelando objetos de
aprendizaje a partir de la
PSICOLOGÍA DEL
APRENDIZAJE**

Luis Eduardo Otero Sotomayor



371.307

O73m

Otero Sotomayor, Luis Eduardo

Modelando objetos de aprendizaje : a partir de la psicología del aprendizaje / Luis Eduardo otero Sotomayor ; Bogotá, Corporación Universitaria Unitec, 2018.

166 p.

Incluye bibliografía

ISBN 978-958-9020-10-4

1. EDUCACIÓN -- METODOLOGIA 2. OBJETOS DE APRENDIZAJE 3. CAPACITACIÓN DOCENTE -- METODOS EXPERIMENTALES 4. INNOVACIONES EDUCATIVAS.

© Corporación Universitaria Unitec®
2018. 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): 978-958-9020-10-4

ISBN (obra digital): 978-958-9020-09-8

Primera edición 2018

Consejo Editorial Corporación Universitaria Unitec

Carlos Alfonso Aparicio Gómez

Rector

Camilo Rizo Parra

Vicerrector Académico

Martín Edilberto Quevedo Bermúdez

Director Centro de Investigaciones

David Arturo Acosta Silva

Jefe del Departamento de Publicaciones

Edición

Centro de Publicaciones Académicas

Departamento de Publicaciones

Corporación Universitaria Unitec

Calle 76 #12-58, segundo piso

Bogotá, D. C. Colombia

Correo electrónico: david.acosta@unitec.edu.co

Producción editorial

David Arturo Acosta Silva

Corrección de estilo

Alejandra Muñoz Suárez

Fotografía cubierta

Réseau de neurones humains

Por Else If Then, 2016.

Licencia Creative Commons BY 4.0, vía Wikimedia Commons.

Diseño y diagramación

Taller de Edición • Rocca® S. A.S.

Carrera 4A N.º 26A-91, of. 203

Tel./fax: 243 2862 - 243 8561

Preprensa e impresión

Imagen Editorial S. A. S.

Calle 35 sur N.º 72 L - 63

Tel.: 311 239 6094

Contenido

Introducción	9
Capítulo 1.	
La teoría del aprendizaje en el diseño de un objeto de aprendizaje	11
1.1 Análisis de requisitos	11
1.2 Teorías de aprendizaje	13
1.3 Sistema aprendizaje de tutores inteligentes (ITS)	17
1.4 Análisis de la tarea cognitiva y análisis del error	18
1.5 Modelado del análisis	19
1.6 Estructura general del objeto de aprendizaje	28
1.7 Arquitectura	30
1.8 Diagramas de actividad para cada módulo	30
1.9 Interfaz del <i>software</i>	41
1.10 Proceso de codificación	42
Capítulo 2.	
Variables del diseño instruccional en la adquisición y aplicación de conceptos	95
2.1 Planteamiento del problema	96
2.2 Justificación	98
2.3 Hipótesis	99
2.4 Marco teórico	99
2.4.1 Aprendizaje de conceptos	99
2.4.2 Componentes de la arquitectura cognitiva humana	104
2.4.3 Interactividad	106
2.4.4 Retroalimentación	110

2.5	Marco metodológico	112
2.5.1	Diseño	112
2.5.2	Participantes	113
2.5.3	Instrumentos	113
2.5.4	Variables de la investigación	119
2.5.5	Procedimiento	119
2.6	Procesamiento y análisis de la información	121
2.6.1	Población, muestra y procedimientos experimentales	121
2.6.2	Datos y resultados de la experimentación	121
2.6.4	Análisis del resultado de las evaluaciones	127
2.6.5	Efecto de la adquisición de concepto sobre la retención	131
2.7	Resultados	132
Referencias		135
Anexos		139
ANEXO 1.	Ideas básicas sobre HTML	139
ANEXO 2.	CSS	144
ANEXO 3.	Diseño de la base de datos	146
	Diseño conceptual	146
	Diseño lógico	147
ANEXO 4.	Clases en PHP	150
ANEXO 5.	Recuperación de datos con MySQL	151
ANEXO 6.	Conceptos de JavaScript	156
ANEXO 7.	Función recorte	163
ANEXO 8.	Función de ordenación	164

Introducción

La presente obra comprende dos capítulos, ambos resultados de investigación y relacionados entre sí. En el primero se efectúa el análisis, diseño e implementación de un objeto virtual de aprendizaje, entendido como un recurso digital reutilizado en diferentes contextos educativos, desarrollado a partir de la teoría de formación de conceptos. Además, se emplea el análisis del error, los tipos de retroalimentación, la interactividad y la carga cognitiva para incrementar el éxito de un *software* para el aprendizaje de conceptos, que se estructura a partir de estas teorías.

Así, el objetivo de este capítulo es poner a disposición de profesores de cualquier asignatura e investigadores en el área de conocimiento de la psicología de aprendizaje un *software* para el aprendizaje en cualquier dominio de conocimiento, que se puede reutilizar con tan solo realizar algunos cambios. Se presenta el *software* completo, esto es, desde la construcción de los menús, hasta el desarrollo de contenidos, pasando por actividades de aprendizaje y evaluaciones, estas últimas con su respectiva retroalimentación. En el análisis del *software* se detalla cómo, a partir de los objetivos, se construyen las unidades de conocimiento, las actividades de aprendizaje y la retroalimentación que se facilita a los estudiantes por cada entrada que realizan frente a las preguntas que se les hacen.

Por su parte, en el segundo capítulo se presenta el desarrollo y los resultados de la investigación titulada *Influencia del uso de las variables de diseño instruccional en la adquisición y aplicación de conceptos*, la cual se originó en la Corporación Universitaria Unitec y pretendió medir el efecto que tiene un objeto de aprendizaje modelado (a partir de la teoría de conceptos) en el aprendizaje de conceptos y su utilización en la construcción de otros nuevos.

En el estudio se utilizó un diseño estadístico para verificar si hubo diferencia en el desempeño de dos grupos, uno que utilizó un objeto virtual de aprendizaje modelado a partir de la teoría de formación de conceptos y otro grupo que siguió la clase tradicional. En el primer capítulo se hace una descripción del *software* utilizado y se describen las teorías que orientaron la investigación.

Además, se explica el procedimiento estadístico para que quienes quieran trabajar en este campo, puedan comprobar que el objeto virtual de aprendizaje iguala o es superior a la estrategia metodológica que va a reemplazar. Un mensaje para los lectores que son profesores es que no teman ensayar nuevas estrategias metodológicas, tanto en el salón de clases como en el nivel de formación virtual. No tengan miedo de probar y comparar resultados de nuevas estrategias, más allá de las que usan actualmente. De la misma manera, traten de perderle el recelo a las pruebas estadísticas, ya que estas fueron diseñadas para asegurar si realmente las estrategias que se están implementando producen los cambios deseados.



Capítulo 1

La teoría del aprendizaje en el diseño de un objeto de aprendizaje

Este capítulo se basa en la investigación *Análisis, diseño e implementación de un objeto de aprendizaje sobre estadística para investigadores*, efectuada en la Corporación Universitaria Unitec, que tuvo como producto un objeto virtual de aprendizaje, al cual llamamos *Investa*. El capítulo trata de ofrecer a docentes, diseñadores de objetos de aprendizaje e investigadores un procedimiento que les facilite el análisis, diseño e implementación de objetos de aprendizaje. Se presentan ideas prácticas para personas que no tengan conocimientos de ingeniería de *software* o de diseño de bases de datos.

1.1 Análisis de requisitos

En el análisis de requisitos se describe lo que el *software* debe hacer y los actores que intervienen en la solución que se ofrece. En este caso, se trata de hacer objetos de aprendizaje que abordan temas que el profesor u otros interesados deseen diseñar para utilizarlo, ya sea como material de clase o para uso autónomo por parte de los estudiantes. Los actores serán los profesores, estudiantes y el sistema. El término «profesor» comprende a todos aquellos que deseen desarrollar las aplicaciones para uso de aprendizaje.

El modelo se hará tomando como base los principios de la psicología del aprendizaje, los cuales fueron sintetizados a través de un proceso previo de investigación. Lo anterior es fundamental, ya que la opinión o gustos personales (sin haber sido objeto de un estudio investigativo) podrían no generar los resultados esperados. En el presente libro se presentan algunas teorías que han sido desarrolladas durante investigaciones de la línea *Tecnología educativa*, del grupo de investigación *Educación, tecnología y sociedad* de la Corporación Universitaria Unitec. Ahora, en la aplicación puntual de lo que presentaremos a continuación, los lectores y profesores pueden seguir las teorías de aprendizaje de su predilección.

El proceso que seguiremos en la construcción del objeto de aprendizaje será el siguiente:

1. Selección de una o varias teorías de aprendizaje para realizar el diseño del objeto de aprendizaje.
2. El análisis del tema o temas que se van a tratar siguiendo los pasos descritos por Narciss (2001) para el diseño de tareas de aprendizaje complejas llamado «diseño de retroalimentación informativa» tomando como base la tarea cognitiva y el análisis del error.
3. A partir de este análisis se hace el diseño del objeto de aprendizaje, construyendo los componentes señalados en los tutores inteligentes (Groner & Kersten, 2001).
4. Se utiliza un lenguaje de programación para la implementación. Aquí se utiliza JavaScript, por ser un lenguaje de fácil manejo.
5. Se realizan las pruebas.

Para el diseño completo se tienen en cuenta los principios para construir tutores inteligentes, de la manera como lo describen Groner y Kersten (2001).

1.2 Teorías de aprendizaje

Para el aprendizaje de conceptos tomaremos como base la teoría de la formación de conceptos, la cual, aunque se expondrá en ambos capítulos, se presenta con mayor detalle en la sección que desarrolla el marco teórico de la investigación en el capítulo dos.

Tennyson y Cocchiarella (1993) enfocan la teoría de diseño instruccional en tres aspectos: los procesos cognitivos, la estructura de contenido y las variables de diseño instruccional. Los procesos cognitivos que se tienen en cuenta son la formación de conocimiento conceptual y el desarrollo de conocimiento procedimental. El objetivo de la teoría es la formación del conocimiento conceptual y el desarrollo del conocimiento procedimental, a través del uso de las variables del diseño instruccional; pero, para tal fin, se debe tener en cuenta la estructura del contenido de los conceptos.

El conocimiento conceptual según estos autores —Tennyson y Cocchiarella (1993)— se refiere al entendimiento del concepto por parte del estudiante, e implica la formación de atributos y la conexión apropiada con otros conceptos. Por su parte, el desarrollo del conocimiento procedimental se refiere al desarrollo de habilidades de clasificación, generalización y discriminación entre instancias nuevas que se encuentran en conceptos asociados.

Las variables de diseño instruccional: definición, instancias expositoras, instancias interrogativas y elaboración de atributos, ayudan a reducir tres tipos de errores: sobregeneralización, subgeneralización y concepciones erróneas.

Una definición es una regla o descripción que expone la estructura de los atributos críticos. Las instancias expositoras, que pueden ser ejemplos y contraejemplos, organizan y presentan el contenido en un formato declarativo. Las instancias interrogativas, en forma de ejemplos y contraejemplos, muestran el contenido en forma de preguntas. La elaboración de atributos ofrece un análisis crítico en las instancias expositivas y la retroalimentación sobre los atributos críticos en las instancias interrogativas.

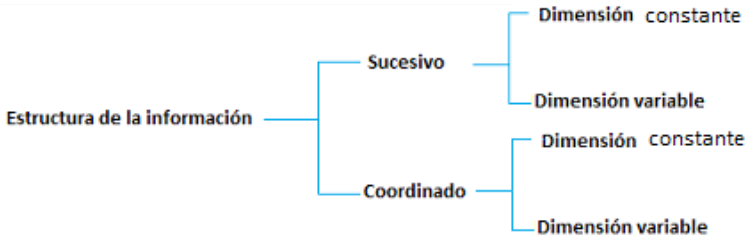
La estructura del contenido posee dos tipos de relaciones básicas: las *sucesivas* y las *coordinadas*; estas relaciones se definen con referencia al proceso cognitivo de clasificación. Así, una relación sucesiva es una relación vertical entre los conceptos. Un concepto básico tiene conceptos superordinados y otros subordinados. Con las relaciones sucesivas el aprendizaje se limita al desarrollo de la generalización dentro de la clase de un concepto. Por su parte, con las relaciones coordinadas el aprendizaje incluye el desarrollo de la generalización y la discriminación entre conceptos; con ellas se desarrolla el conocimiento procedimental.

Un ejemplo es el concepto de escala de medición; este concepto tiene como conceptos sucesivos las escalas nominales, ordinales, de intervalo y de razón. Las escalas mencionadas a su vez son conceptos coordinados. Para saber si cada una de las escalas es una de medición, se acude a un proceso de generalización: si la escala categoriza o cuantifica una variable, entonces es una escala de medición. Pero para distinguir entre las escalas se requiere de un proceso de discriminación; es decir, se deben tener en cuenta los atributos que marcan la diferencia. Por ejemplo, para categorizar a una persona como perteneciente al género masculino o femenino se necesita discriminar entre las escalas para determinar cuál se utiliza.

Las características de los conceptos pueden ser estables o variables. Los que tienen atributos relevantes estables a través de los contextos se denominan *conceptos de dimensión constante*; mientras que cuando varían a través de los contextos los llamamos *conceptos de dimensión variable*.

Tennyson y Cocchiarella (1993) afirman que la estructura de la información es la base para el diseño de la estrategia instruccional. En la figura 1 se puede observar las posibles combinaciones que se pueden obtener de las estructuras. Para hacer el diseño instruccional se deben en cuenta estas combinaciones.

FIGURA 1. Combinaciones de las estructuras de información



Los conceptos sucesivos de dimensión constante requieren de una mínima instrucción; para aprenderlos, se puede dar suficiente práctica al aprendiz según sea su experiencia. El aprendizaje se centra en el desarrollo de la habilidad de generalización; así, se pueden utilizar instancias expositivas (consistentes en definiciones organizadas y claras) y ejemplos trabajados. No se necesitan instancias interrogativas y se pueden usar contraejemplos.

Los conceptos coordinados constantes requieren de su presentación en secuencias simultáneas; como se debe discriminar entre conceptos, es conveniente implementar instancias interrogativas con retroalimentación sobre los atributos.

Para el caso de los conceptos sucesivos variables, estos se aprenden mejor cuando se presentan en contexto. Para ayudar al estudiante a entender la variabilidad de los atributos críticos, se puede utilizar información estratégica en los ejemplos expositivos e interrogativos dados, de manera que se pueda incrementar la generalización a través de los contextos.

Por su parte, los conceptos coordinados variables requieren de mucha instrucción para mejorar el aprendizaje, ya que son los conceptos más demandantes. Hay que tener en cuenta que se trata de discriminar entre conceptos en contextos variables.

Otra de las teorías que tendremos en cuenta en esta exposición es la de la *carga cognitiva* sobre la memoria de trabajo. Esta se entiende como la cantidad de actividad mental impuesta sobre la memoria de trabajo en un instante de tiempo. La carga cognitiva total está compuesta de la carga cognitiva intrínseca y extrínseca. La intrínseca es la impuesta por el tema que se va a estudiar, mientras que

la carga cognitiva extrínseca es la que se agrega por el material instruccional. Para que el aprendizaje pueda ocurrir, la suma de estas dos no debe sobrepasar los recursos mentales del estudiante.

Por otra parte, para informar al estudiante sobre los aciertos en sus respuestas utilizaremos la *retroalimentación* que, de acuerdo con Mason y Bruning (2001), presenta los siguientes niveles:

No retroalimentación: se le presenta la calificación al estudiante sin hacer referencia a los elementos individuales. Este tipo de retroalimentación no contiene ni verificación, ni elaboración. La verificación comprueba si una respuesta es correcta o incorrecta; mientras que la elaboración es el componente informacional que provee indicaciones que guían al estudiante hacia la respuesta correcta.

Conocimiento de resultado: este tipo de retroalimentación le informa al estudiante si su respuesta fue correcta o incorrecta. Este tipo de respuesta es esencial para la verificación, pero no provee información sobre los errores posibles.

Múltiples respuestas: el estudiante responde hasta que se genera la respuesta correcta. Este tipo de retroalimentación provee verificación, pero no elaboración.

Conocimiento de la respuesta correcta: facilita la verificación sobre cada uno de los puntos y le da conocer al estudiante la respuesta correcta. No brinda elaboración más allá de la respuesta correcta.

Contingente al tópico: provee verificación e información elaborada general sobre el tema. Cuando se produce la respuesta incorrecta, se dirige al estudiante al material en donde se encuentra la respuesta correcta; depende del estudiante encontrar la respuesta en el material instruccional.

Contingente a la respuesta: provee tanto verificación como elaboración. Además de proveer el conocimiento de la respuesta correcta, facilita la retroalimentación específica del porqué una respuesta incorrecta está equivocada y por qué la respuesta correcta lo es.

Relacionada con el error: brinda verificación y maneja errores específicos. Se apoya en librerías o reglas para identificar los errores más comunes de los

estudiantes. No facilita la respuesta correcta, pero ayuda al estudiante a detectar errores de procedimiento, para que él mismo haga la corrección.

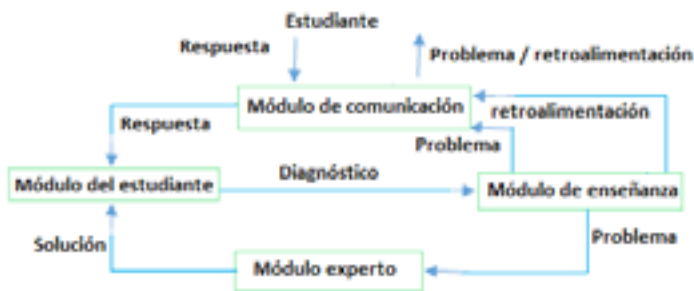
Aislamiento de atributos: provee verificación de elementos y enfatiza los atributos centrales del concepto. Se centra en los componentes claves para mejorar el entendimiento general del fenómeno.

A continuación, describimos el sistema de aprendizaje de tutores inteligentes, explicados por Groner y Kersten (2001). Este sistema permite estructurar el objeto virtual de aprendizaje, cuyo diseño se expone en este capítulo.

1.3 Sistema aprendizaje de tutores inteligentes (ITS)

El diseño del objeto virtual consta de los componentes utilizados para hacer tutores inteligentes. Posee 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; d) el módulo de comunicación. En la figura 2 se ilustra la interacción de estos componentes.

FIGURA 2. Secuencia de un ITS ideal



El módulo de enseñanza selecciona una tarea, se la comunica al módulo experto y, a través del módulo de comunicación, al estudiante. El módulo de enseñanza determina el contenido de la interacción y, el módulo de comunicación, la forma. El estudiante realiza la tarea y determina la solución o, de acuerdo con la situación, los pasos intermedios. 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. La siguiente unidad de enseñanza depende del estado de conocimiento del estudiante; se pueden generar tareas que concuerden con el conocimiento del estudiante o presentar ejemplos que sirvan para su comprensión del tema.

1.4 Análisis de la tarea cognitiva y análisis del error

Teniendo en cuenta las teorías expuestas y el sistema de aprendizaje que se utiliza para estructurar el objeto virtual de aprendizaje, a continuación se presenta el procedimiento del análisis del contenido disciplinar de las tareas cognitivas y el análisis del error siguiendo los pasos propuestos por Narciss (2001). Para identificar las operaciones cognitivas puede consultar la taxonomía de Bloom (1956).

- Seleccione y especifique objetivos para el dominio dado de conocimiento en términos de eventos de aprendizajes concretos.
- Identifique y clasifique los elementos específicos de contenido y las unidades de conocimiento relacionadas con el elemento del contenido (por ejemplo, hechos, conceptos, eventos, reglas, principios, modelos, teorías).
- Seleccione y especifique las operaciones cognitivas que se deben asociar con la unidad de conocimiento (por ejemplo, recordar, transformar, clasificar, argüir, inferir).
- Construya una matriz; un eje que contenga las unidades de conocimientos y otro que contenga las operaciones cognitivas.
- Seleccione y especifique una tarea de aprendizaje para cada celda de la matriz (por ejemplo, recordar hechos, identificar o definir conceptos, formular reglas, clasificar eventos, argüir con base en los modelos, inferir de las teorías).

- Identifique y clasifique factores posibles, concepciones erróneas o fuentes que pueden causar un error o estrategias ineficientes para cada tipo de tarea (por ejemplo, ignorar instancias negativas en la formación de conceptos).
- Seleccione y especifique 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, sugerencias sobre estrategias de éxitos).

1.5 Modelado del análisis

El análisis del objeto de aprendizaje que se tomó como ejemplo, se realizará con base en las ideas expuestas por Narciss (2001) sobre el diseño de retroalimentación elaborada en referencia a la tarea cognitiva y el análisis del error, así como en los principios de la formación de conceptos expuestos por Tennyson y Cocchiarella (1993). Los objetivos de aprendizaje los constituirán los conceptos que se desea que se aprendan.

Los asuntos objeto de diseño son temas tratados en la investigación que dio lugar a un objeto virtual de aprendizaje diseñado para enseñar estadística a investigadores (Otero, 2018). Expondremos el análisis que se efectuó con las escalas. Primero se presenta su definición de acuerdo con la teoría de formación de conceptos, luego ejemplos expositivos e interrogativos cuando sean necesarios.

Las escalas son conceptos coordinados constantes dentro del ámbito de la estadística; por ello se exponen actividades de aprendizaje en donde se presentan simultáneamente los conceptos para que el estudiante pueda discriminar entre los atributos y poder así diferenciar aquellos relevantes para cada concepto. Las actividades de aprendizaje constituyen ejemplos interrogativos de los conceptos. A continuación, expondremos parte de los contenidos de conocimiento, las operaciones cognitivas, las actividades de aprendizaje y la

retroalimentación que se usan en un objeto virtual de aprendizaje, el cual fue escogido como ejemplo para mostrar el proceso de diseño:

Objetivos virtuales de aprendizaje

Medición: implica asignar números a objetos o eventos según ciertas leyes, para lo cual se utilizan las escalas de medición.

Escala nominal: satisface solamente la propiedad de identidad. Clasifica a los objetos o individuos en categorías, tomando como base una o más características o atributos. Para la categorización se pueden utilizar nombres o números, pero estos últimos no se pueden utilizar en operaciones matemáticas o hacer comparaciones de orden; son ejemplo de ello los habitantes de una ciudad organizados por estrato.

Atributos relevantes: a) satisface la propiedad de identidad; b) clasifica a los objetos o individuos en categorías.

Mejor ejemplo: clasificación de las personas por género.

Ejemplos expositivos:

Ejemplo	Razones
Clasificación de habitantes de una ciudad por estrato.	Categoriza las personas según el estrato; no hace comparación.
Clasificación de los estudiantes según la carrera elegida.	Hace una categorización sin establecer una relación de orden.

Escala ordinal: satisface las propiedades de identidad y magnitud. Determina la posición de los objetos o individuos con relación a cierto atributo, pero sin indicar la distancia que existe entre los individuos. En las comparaciones se puede establecer cuál es mayor o menor, pero no se pueden realizar operaciones matemáticas.

Atributos relevantes: a) satisface las propiedades de identidad y magnitud; b) determina la posición de los objetos o individuos con relación a cierto atributo.

Mejor ejemplo: clasificación de los ciclistas en una carrera como primero, segundo y tercero, según el orden de llegada.

Ejemplos expositivos:

Ejemplo	Razones
Ordenar los objetos en pequeños, medianos y grandes.	Permitir comparar y establecer un orden.
Ordenar a los estudiantes en mejores, regulares y peores según el desempeño en un examen.	Permitir comparar y establecer un orden.

Escala de intervalos: proporciona intervalos iguales procedentes de un origen arbitrario. Además de ordenar objetos o eventos, establece intervalos iguales entre las unidades de medida.

Atributos relevantes: a) proporciona intervalos iguales procedentes de un origen arbitrario; b) establece intervalos iguales entre las unidades de medida.

Mejor ejemplo: la escala en centígrados para medir la temperatura. Está compuesta de unidades iguales, de tal manera que la diferencia entre 10 y 20 grados centígrados es la misma entre 30 y 40 grados centígrados.

Ejemplos expositivos:

Ejemplo	Razones
La nota en una prueba académica.	La diferencia entre el que obtuvo dos y tres es la misma diferencia de los que obtuvieron cuatro y cinco, es decir, un punto.
Medida del nivel del mar.	El cero se establece por definición. No es un cero verdadero.

Escala de razón: proporciona un verdadero punto cero y establece intervalos iguales entre las unidades.

Atributos relevantes: a) proporciona un verdadero punto cero; b) establece intervalos iguales entre las unidades.

Mejor ejemplo: un metro contiene el cero que equivale a una magnitud que no existe.

En tanto que la operación cognitiva que se asocia a estos conceptos es la *discriminación*, es por eso se presentan ejemplos expositivos, interrogativos y el mejor ejemplo.

Ejemplos expositivos:

Ejemplo	Razones
Cantidad de agua en un recipiente.	Existe el verdadero punto cero, cuando no hay agua.
Velocidad de un automóvil.	Cuando el automóvil está quieto es un verdadero punto cero.

Ejemplos interrogativos: se toman como actividades de aprendizaje y el proceso cognitivo consiste en discriminar entre los conceptos que se están estudiando.

Actividad de aprendizaje 1

Escriba en las cajas de textos ubicadas en la derecha, el tipo de escala que corresponde en la descripción de la izquierda.

1. Determinar la temperatura en grados centígrados.
2. Clasificar a las personas en bajas, medianas y altas.
3. Clasificar a las personas sanas en el grupo 1 y a las enfermas en el grupo 2.
4. Determinar la cantidad de metros cúbicos en un recipiente.

Factores de error

Errores de sobregeneralización: tomar instancia que no pertenece al concepto como ejemplo del concepto.

Retroalimentación

La retroalimentación se proporciona con relación a los atributos relevantes de las escalas. Se dan tres tipos de retroalimentación: de conocimiento de resultados, retroalimentación contingente al tópico y múltiples respuestas.

En cada una de las cajas de texto existen cuatro posibilidades de respuesta. Si no se produce la respuesta correcta existe un error de sobregeneralización o de concepción incorrecta que se produce cuando el estudiante tiende a sobregeneralizar.

Retroalimentación para la primera opción

Si el estudiante escribe *escala nominal* recibiría la siguiente retroalimentación:

«Su respuesta es errónea porque la temperatura en grados centígrados necesita una escala que proporcione intervalos iguales. La escala nominal no tiene esta característica; sirve para clasificar los objetos en categorías. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos*».

Si el estudiante escribe *escala ordinal* recibiría la siguiente retroalimentación:

«Su respuesta es errónea porque la temperatura en grados centígrados necesita una escala que proporcione intervalos iguales. La ordinal no tiene esta característica; sirve para ordenar los objetos según el valor de alguno de sus atributos. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos*».

Si escribe *escala de intervalos* recibiría la siguiente retroalimentación: «*Felicitaciones*. La escala de intervalos sirve en ese caso porque la temperatura en grados centígrados necesita una escala que proporcione intervalos iguales».

Si escribe *escala de razón* recibiría la siguiente retroalimentación:

«Su respuesta es errónea porque la temperatura en grados centígrados necesita una escala que proporcione intervalos iguales. La escala de razón sirve para objetos cuyos atributos se pueden expresar con un cero verdadero. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos*».

Retroalimentación para la segunda opción

Si el estudiante escribe *escala nominal* recibiría la siguiente retroalimentación:

«Su respuesta es errónea porque clasificar a las personas en bajas, medianas y altas necesita de una escala que proporcione ordenación de los objetos según el valor de alguno de sus atributos. La escala nominal no tiene esta característica; sirve para clasificar los objetos en

categorías. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos».*

Si escribe *escala ordinal*: «Muy bien. La clasificación de las personas en bajas, medianas y alto requiere establecer una comparación entre las estaturas; unas son más altas o menos altas que otras».

Si escribe *escala de intervalos*:

«Su respuesta es errónea porque clasificar a las personas en bajas, medianas y altas necesita de una escala que proporcione ordenación de los objetos según el valor de alguno de sus atributos. La escala de intervalos proporciona intervalos iguales procedentes de un origen arbitrario. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos».*

Si escribe *escala de razón*:

«Su respuesta es errónea porque clasificar a las personas en bajas, medianas y altas necesita de una escala que proporcione ordenación de los objetos según el valor de alguno de sus atributos. La escala de razón sirve para objetos cuyos atributos se pueden expresar con un cero verdadero. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos».*

Retroalimentación para la tercera opción

Si escribe *escala nominal*: «Muy bien. La clasificación de las personas sanas en el grupo 1 y a las enfermas en el grupo 2 constituye una escala nominal porque está clasificando las personas en categorías».

Si escribe *escala ordinal*:

«Su respuesta es errónea porque clasificar a las personas sanas en el grupo 1 y a las enfermas en el grupo 2 necesita de una escala que proporcione categorización de los objetos. La escala ordinal facilita la ordenación de los objetos según el valor de alguno de sus atributos. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos».*

Si escribe *escala de intervalos*:

«Su respuesta es errónea porque clasificar a las personas sanas en el grupo 1 y a las enfermas en el grupo 2 necesita de una escala que proporcione categorización de los objetos. La escala de intervalos proporciona intervalos iguales procedentes de un origen arbitrario. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos*».

Si escribe *escala de razón*:

«Su respuesta es errónea porque clasificar a las personas sanas en el grupo 1 y a las enfermas en el grupo 2 necesita de una escala que proporcione categorización de los objetos. La escala de razón sirve para objetos cuyos atributos se pueden expresar con un cero verdadero. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos*».

Retroalimentación para la cuarta opción

Si escribe *escala nominal*:

«Su respuesta es errónea porque determinar la cantidad de metros cúbicos en un recipiente necesita una escala que proporcione cero verdadero. La escala nominal no tiene esta característica, esta sirve para clasificar los objetos en categorías. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos*».

Si escribe *escala ordinal*:

«Su respuesta es errónea porque determinar la cantidad de metros cúbicos en un recipiente necesita de una escala que proporcione cero verdadero. La escala de ordinal sirve para objetos cuyos atributos se pueden ordenar según el valor de alguno de ellos. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos*».

Si escribe *escala de intervalos*:

«Su respuesta es errónea porque determinar la cantidad de metros cúbicos en un recipiente necesita una escala que proporcione cero

verdadero. La escala de intervalos proporciona intervalos iguales procedentes de un origen arbitrario. *Repase los conceptos para que escoja la escala que reúne los atributos requeridos*».

Si escribe *escala de razón*: «Muy bien; determinar la cantidad de metros cúbicos en un recipiente necesita de una escala con cero verdadero».

En general, para la construcción de diferentes actividades sobre el tema se siguió el siguiente formato:

Señale a qué tipo de escala corresponde x ($x = \{\text{número del seguro social de una persona, fechas de calendario, ingresos mensuales de una persona, nivel educativo de una persona.}\}$)

- a. Nominal.
- b. Ordinal.
- c. De intervalo.
- d. De razón.

En este ejemplo se trata de varias actividades porque cuando se haga el programa la x va a ser reemplazada por alguno de los casos señalados. Lo que se encuentra entre paréntesis no va a aparecer en la actividad.

El error que se puede cometer es una sobregeneralización. Se va a señalar la retroalimentación en forma general.

Si en el análisis de la respuesta x_i la solución no corresponde, se dará una retroalimentación de resultado y seguida de otra contingente al tópico: «su respuesta es incorrecta; x_i no es la escala, porque la escala es aquella... Para el caso x_i , se necesita una escala que...».

De esta manera se hace el análisis de los demás temas que se van a incluir en el objeto de aprendizaje. El interés que se tiene es descubrir un patrón que ayude a diseñar un programa que realice los temas que se quieran cubrir en el momento o en trabajos futuros.

Evaluaciones

Las evaluaciones tienen la misma forma que las actividades de aprendizaje, pero en estas se da la retroalimentación al final señalando los errores y aciertos acaecidos. Se dan dos tipos de evaluación: cuantitativa y cualitativa. La finalidad de la evaluación cualitativa es señalar las debilidades en el conocimiento del estudiante, de manera que este (en forma autónoma) estudie nuevamente dichos temas. Miremos un ejemplo:

Evaluación N.º 1

Punto 1. Haga clic sobre la opción que mejor explique el concepto.

El género de una persona se mide con una escala nominal porque:

- a. Permite la ordenación del género, facilita establecer cuál es mayor y cuál es menor.
- b. Es una escala que contiene un valor cero arbitrario; en este caso, la ausencia de sexo de una persona.
- c. Categoriza las personas según su género. Por ejemplo, *m* para masculino y *f* para femenino.
- d. Tiene un cero que indica la ausencia de género

Punto 2. Escriba en la caja de texto de la izquierda el número que corresponda de la parte de la derecha.

- | | | |
|--|----------------------|-----------------|
| a. Número de hijos | <input type="text"/> | 1. Ordinal |
| b. Puntaje de un equipo en un campeonato | <input type="text"/> | 2. De razón |
| c. Grado de escolaridad | <input type="text"/> | 3. De intervalo |
| d. Credo religioso | <input type="text"/> | 4. Nominal |

En cada una de las opciones se dan cuatro probabilidades de retroalimentación que se muestran en la tabla 1. Se va a colocar la respuesta posible con dos puntos y luego el mensaje de retroalimentación. Emplearemos los símbolos ## como una marca para separar cada mensaje de retroalimentación.

TABLA 1. Mensajes de retroalimentación para una evaluación

Punto	Mensaje
Punto1	1: Confunde una escala ordinal con una escala de razón. ##2: Reconoce una escala de razón. ##3: Confunde una escala de intervalos con una escala de razón. ##4: Confunde una escala nominal con una escala de razón.
Punto 2	1: Confunde una escala ordinal con una escala de intervalos. ##2: Confunde una escala de razón con una escala de intervalos. ##3: Reconoce una escala de intervalos. ##4: Confunde una escala nominal con una escala de intervalos.
Punto 3	1: Reconoce una escala ordinal. ##2: Confunde una escala ordinal con una escala de razón. ##3: Confunde una escala ordinal con una escala de razón. ##4: Confunde una escala nominal con una escala ordinal.
Punto 4	1: Confunde una escala ordinal con una nominal. ##2: Confunde una escala de razón con una nominal ##3: Confunde una escala de intervalos con una nominal. ##4: Reconoce una escala nominal.

1.4 Estructura general del objeto de aprendizaje

En la figura 3 podemos observar la estructura general del programa que se realizó en la investigación.

FIGURA 3. Estructura general del programa



Por su parte, en la figura 4 aparece el diagrama de despliegue del concepto de conceptualización:

FIGURA 4. Despliegue de la conceptualización



En la figura 5 se observa el despliegue general para cualquiera tema.

FIGURA 5. Diagrama de despliegue de los temas



1.5 Arquitectura

La arquitectura que se sigue es la de programa principal/subprograma que separa la función en una jerarquía de control donde el programa estadístico para investigadores (*Investa*), descrito al comienzo del capítulo, el cual invoca a otros programas; el caso se ilustra en la figura 6.

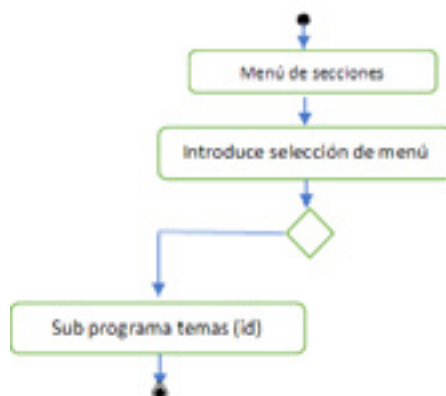
FIGURA 6. Arquitectura de programa principal/subprograma



1.8 Diagramas de actividad para cada módulo

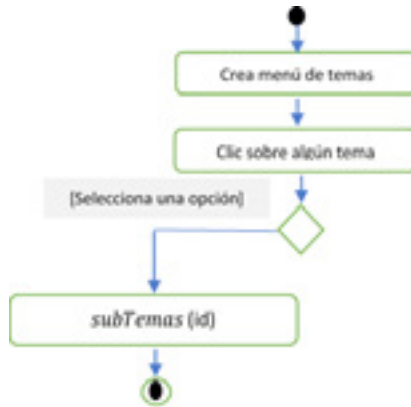
En la figura 7 se puede observar el diagrama de actividad para el programa *Investa*. Al hacer clic sobre alguna de las opciones del menú, llama al menú de temas de la sección correspondiente, pasando la identificación de dicho tema.

FIGURA 7. Programa estadístico



La figura 8 muestra el subprograma tema. Al hacer clic sobre algún elemento del menú, se invoca al programa *subTemas(id)* y se le pasa el parámetro *id* (identificación) del elemento sobre el cual se hizo clic.

FIGURA 8. Función temas



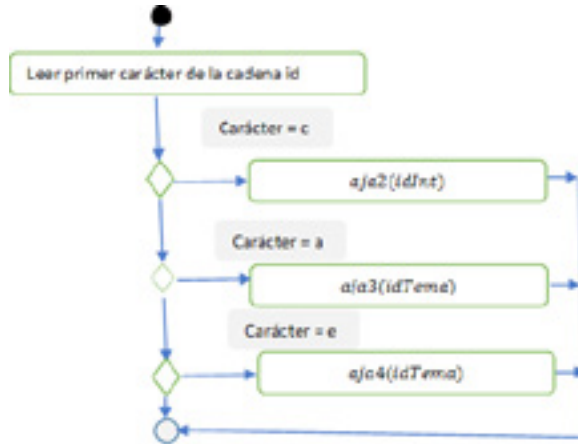
En la figura 9 aparece la función *subTemas (id)*, que llama a la función *ajax (idInt)* que recibe el *idInt* del tema que se va a tratar. Crea el menú de subtemas y, al hacer clic sobre cualquiera de sus elementos, invoca a la función *desaSubtemas(id, idTema)*, a la cual le pasa el *id* del subtema sobre el cual se hizo clic. En la variable *idTema* se pasa el *id* del tema.

FIGURA 9. Subprograma subTemas



La función *desaSubtemas(id)* (como lo muestra la figura 10) lee el primer carácter del *id* y, dependiendo de este valor, llama a las funciones *aja2(idInt)*, *aja3(idInt)* o *aja4(idInt)*. El argumento (*idInt*) identifica el número del subtema, de la actividad o de la evaluación.

FIGURA 10. Función *desaSubtemas(id)*



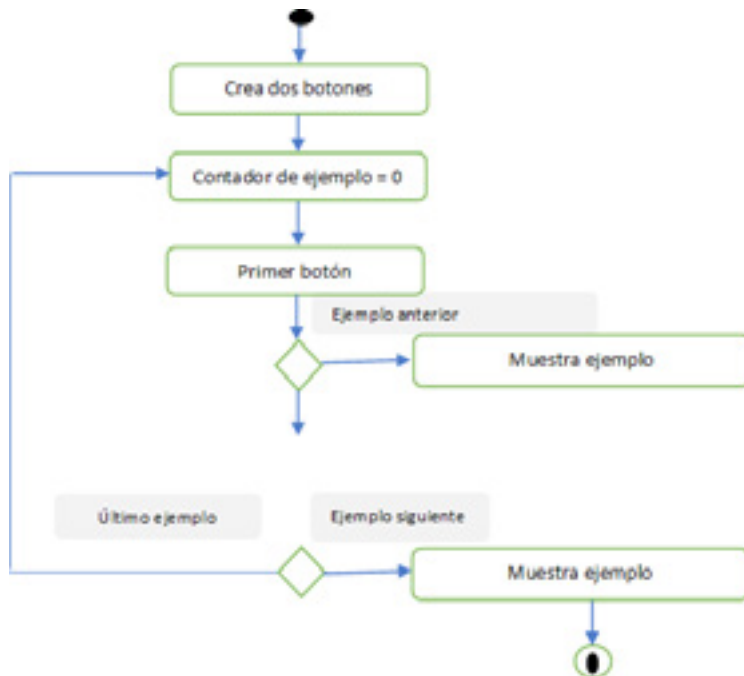
Como podemos observar en la figura 11, la función *aja2* muestra los títulos y los subtemas. Primero se llama a una función *subtema.php* que lee datos de una base de datos, limpia la división en donde se va a escribir la información, agrega los títulos y los contenidos de los subtemas, crea arreglo de ejemplos y llama a la función *bot()*, que es una función miembro de *aja2*.

FIGURA 11. Función que muestra subtemas



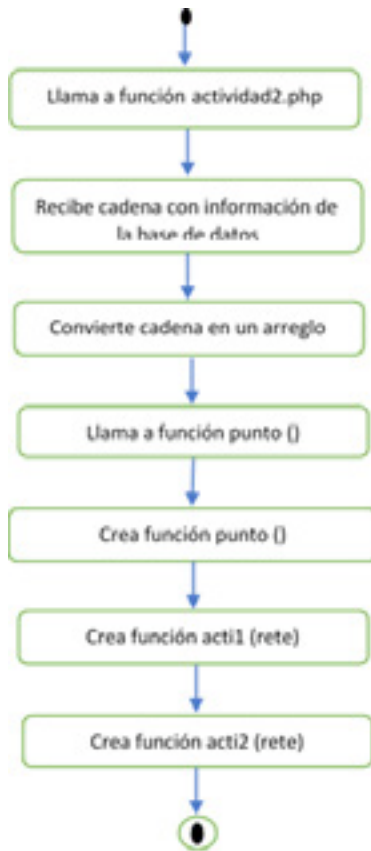
La figura 12 muestra la función *aja2.bot*. En esta se crean dos botones para recorrer los diferentes ejemplos. Cuando se hace clic sobre ellos, uno avanza y el otro retrocede. Si se llega al último ejemplo, el siguiente que se presenta es el primero.

FIGURA 12. Función *aja2.bot*



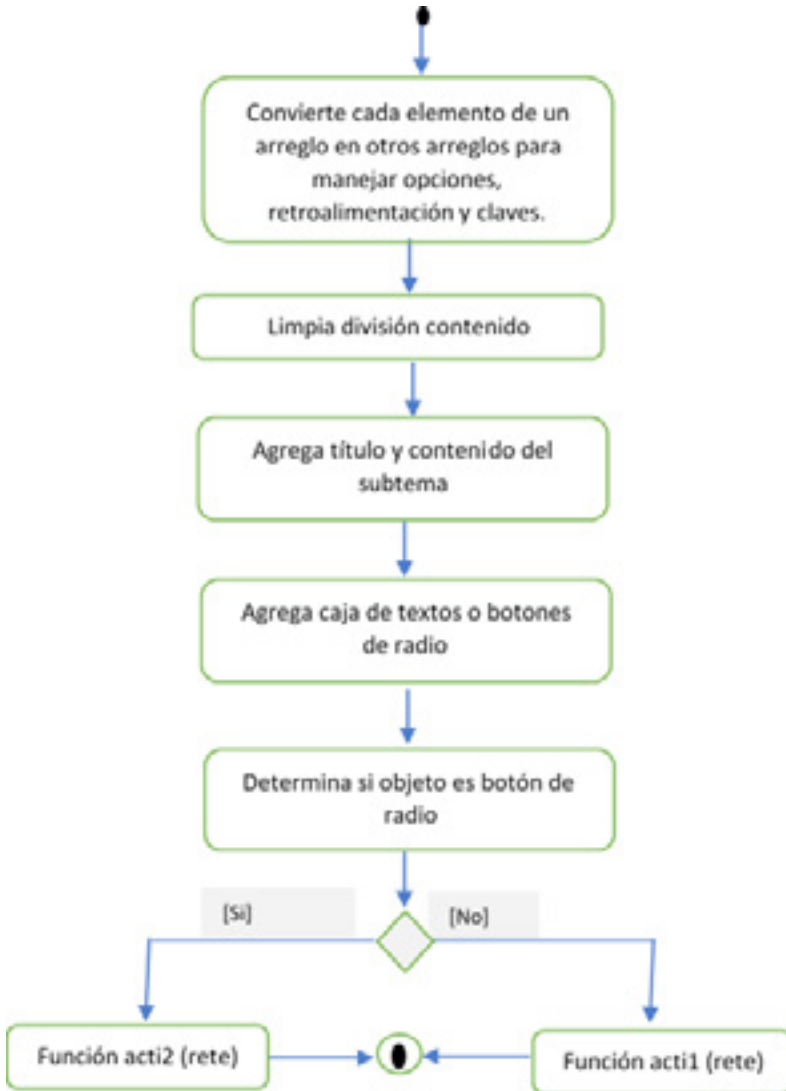
La figura 13 exhibe la función *aja3*. El objetivo de esta función es la presentación de las actividades de aprendizaje; tiene tres partes: el contenido de la actividad, las opciones correspondientes a las preguntas y la retroalimentación sobre cada opción. Primero, se hace un llamado a la función *actividad2.php*, que recupera la información de la base de datos. Los datos vienen en una cadena que, posteriormente, se convierte en un arreglo.

FIGURA 13. Función aja3



La figura 14 muestra la función *punto ()* que convierte el arreglo mencionado arriba en otros arreglos, los cuales contienen la información de las actividades de aprendizaje, de las opciones, las cajas de textos, los botones de radio y la retroalimentación. Esta función verifica si la actividad utiliza cajas de textos o botones y, de acuerdo con esta información, llama a las funciones *acti1* o *acti2*. Ambas funciones reciben un arreglo que contiene la retroalimentación.

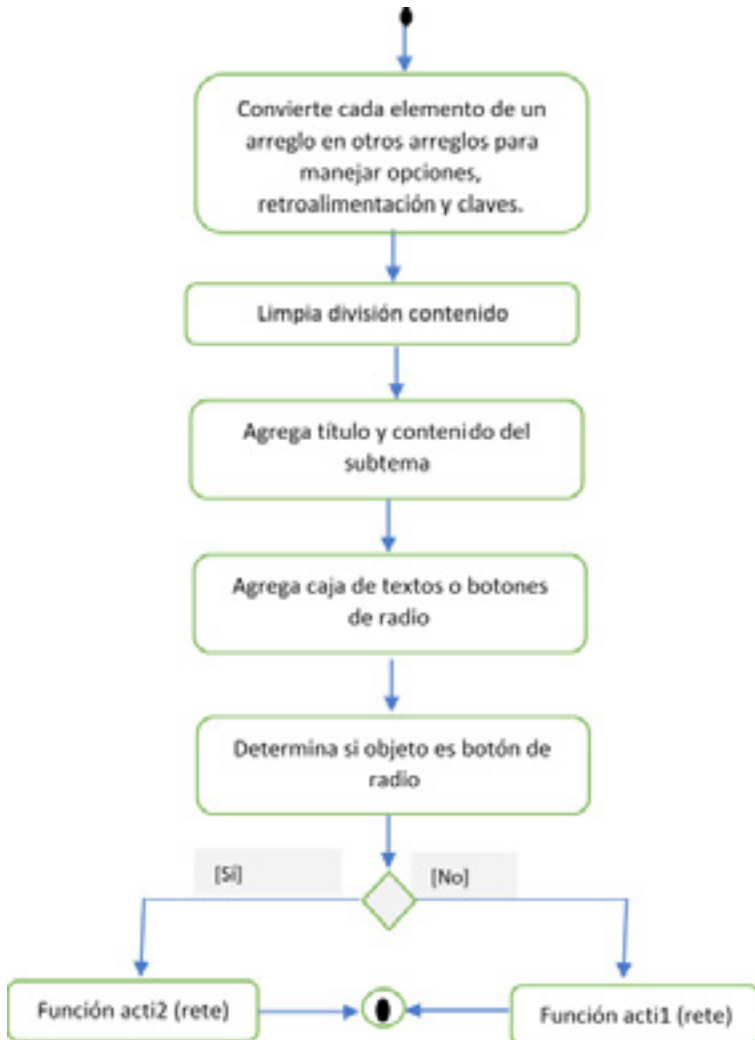
FIGURA 14. Función punto ()



Por su parte, la figura 15 muestra la función *acti1* que se activa cuando la interactividad se realiza a través de cajas de textos. Al presionar la tecla enter, se recupera la identificación (*id*) de la caja de texto con la cual se está trabajando y la información digitada. El *id* permite también obtener la información de retroalimentación contenida en un arreglo, que fue enviada a través de la variable *rete*. Cada elemento del arreglo contiene dos partes separadas por el carácter dos puntos (:); la primera parte contiene información sobre el error posible, mientras

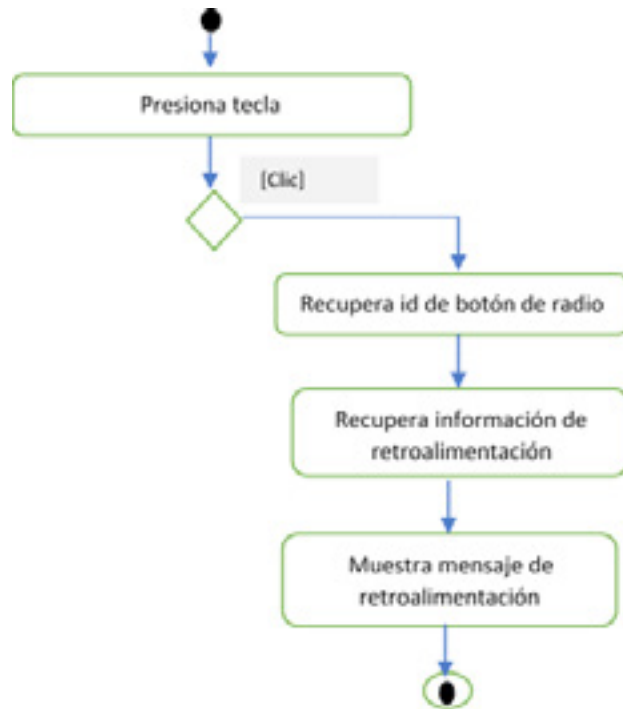
que la segunda la retroalimentación sobre el error. La función analiza el error y, acuerdo con este, emite el mensaje correspondiente. Fíjese que el diagrama de la actividad contiene procesos generales; en la sección de codificación se explica cada proceso en detalle.

FIGURA 15. Función acti1 (rete)



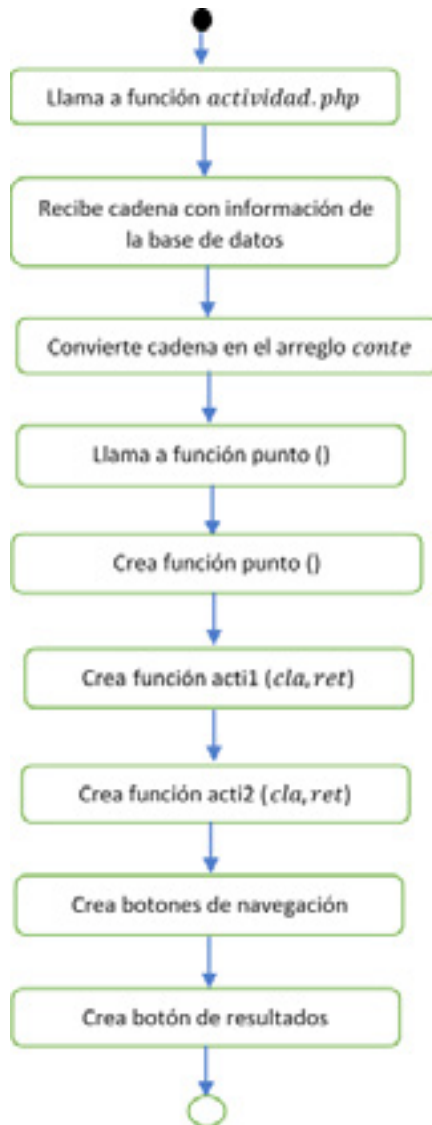
La función *acti2* (figura 16) utiliza botones de radio; con el *id* del botón de radio recupera la información de la retroalimentación y luego muestra el mensaje.

FIGURA 16. Función acti2 (rete)



La figura 17 presenta la función *aja4* que se encarga de las evaluaciones. Es muy parecida a la función *aja3*, pero la retroalimentación se da al final de cada evaluación con su respectiva nota numérica. En la *aja3* se muestran mensajes de retroalimentación que dan a conocer si la respuesta está bien o mal, más otros mensajes con retroalimentación relacionada con el tópico; el programa no controla si la respuesta del estudiante es correcta o no. En cambio, en *aja4* el programa controla si la respuesta es correcta, a través de las claves. Si es correcta coloca una nota. Esta retroalimentación tiene la intención de que el estudiante refuerce los temas que no se contestaron bien. En este programa el refuerzo se deja como una decisión autónoma.

FIGURA 17. Función *aja4*



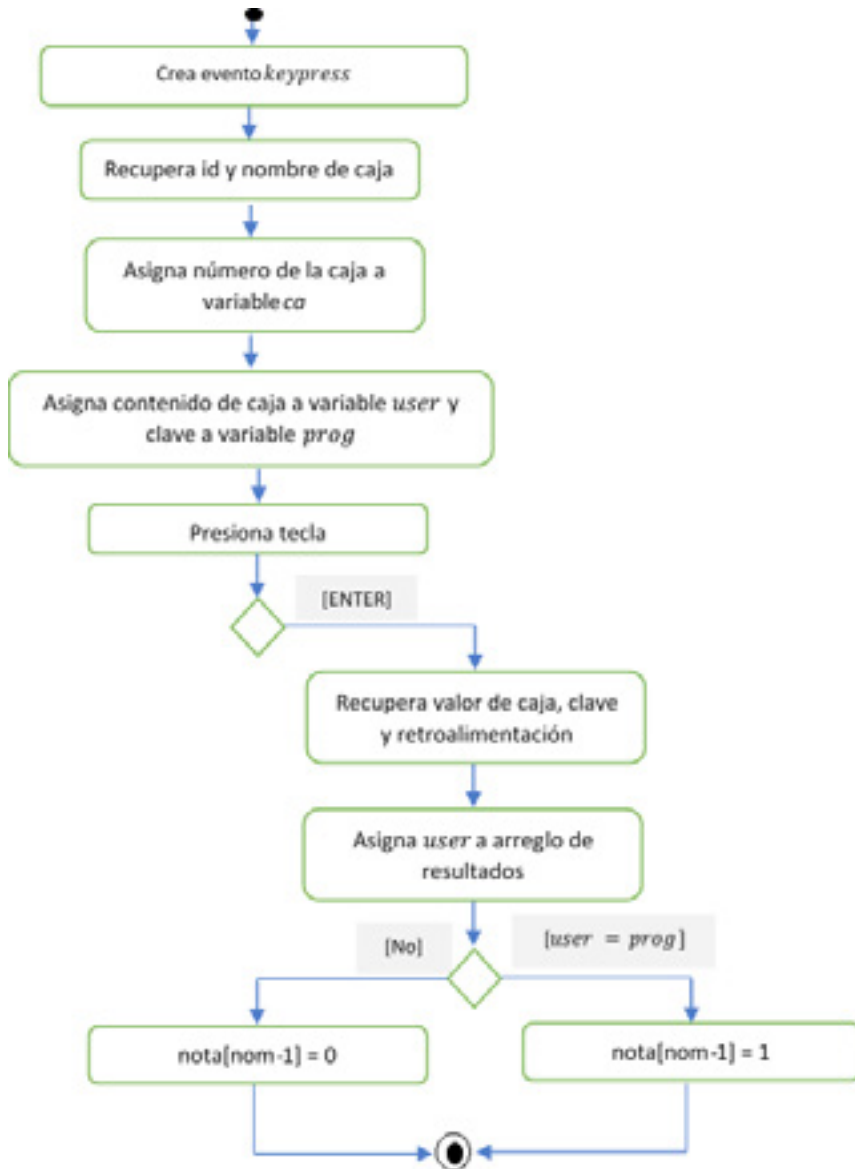
En la figura 18 aparece la función *punto ()*, miembro de la función *aja4*. El objetivo de la primera es crear arreglos con los puntos de las evaluaciones, los cuales contienen títulos, opciones, claves y retroalimentaciones. De acuerdo con el tipo de objeto proveniente de la base de datos, se llama a *acti1* o a *acti2*.

FIGURA 18. Diagrama de actividad función punto ()



La figura 19 muestra la función *acti1* miembro de *aja4*, por medio de la cual se crea un evento *keypress*, el cual se activa cuando se presiona la tecla *enter*. Se recupera el contenido de la caja en la variable *ca*, la cual se compara con la variable *prog* que tiene la clave de la pregunta; si la respuesta es correcta, se coloca una nota de 1. Al arreglo *resultados* se le asigna la evaluación cualitativa.

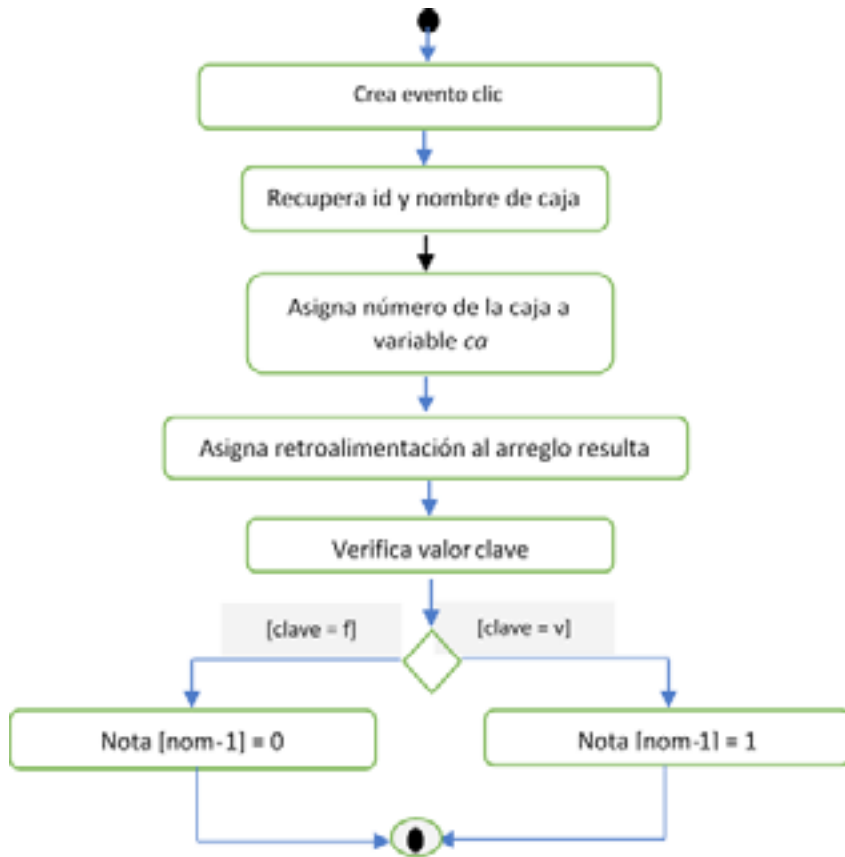
FIGURA 19. Diagrama de actividad función acti1



En la figura 20 se evalúa si el radio marcado contiene la respuesta correcta. De la misma manera que en el caso anterior, se guarda la evaluación cualitativa en *resultados* y se coloca una nota de 1 o de 0, dependiendo del resultado.

Aquí no se muestran las funciones que leen la base de datos, pero en el proceso de codificación se explica su estructura.

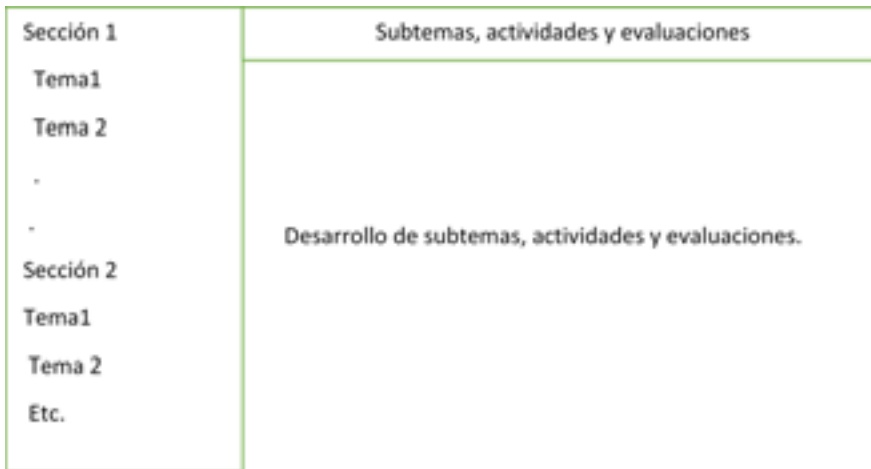
FIGURA 20. Diagrama de actividad función acti2



1.9 Interfaz del *software*

En la figura 21 se puede observar la distribución de la interfaz. Está dividida en tres partes; en la parte izquierda, se ubican la secciones y los temas; en la parte derecha, los subtemas, actividades y evaluaciones y debajo de esta última se lleva a cabo el desarrollo de los contenidos. Al dar clic sobre cada una de las secciones aparecen los temas, y dar clic sobre los temas aparecen los subtemas correspondientes.

FIGURA 21. Interfaz del software



En la figura 22 se puede observar una vista de la interfaz. En esta se muestra el submenú de escalas de medición.

FIGURA 22. Vista de la interfaz

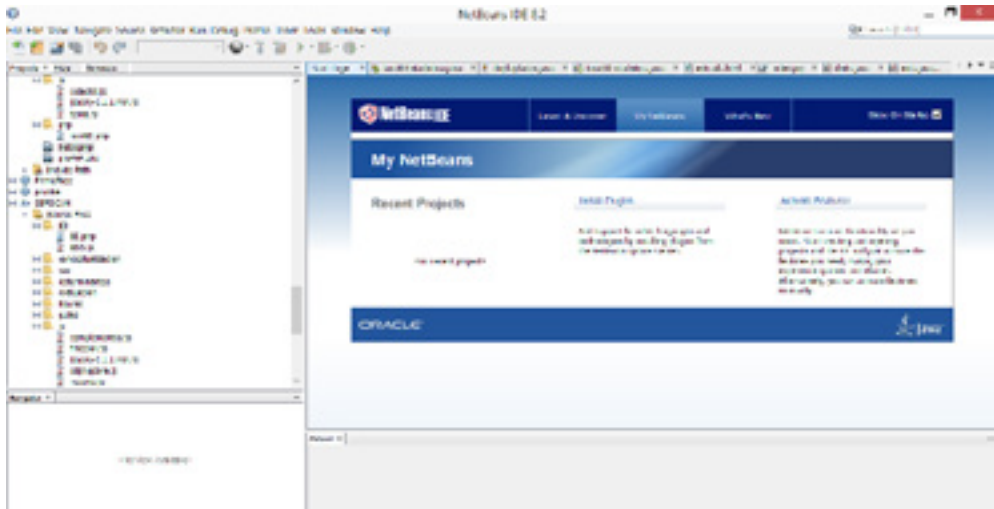


1.10 Proceso de codificación

Para codificar el programa se utilizará el editor Netbeans 8.2 (Oracle Corporation). La idea es utilizar un lenguaje de programación sencillo —como lo es JavaScript—; para el manejo de la base de datos se utilizará MySQL. Por ahora, explicaremos Netbeans. Este es un ambiente de desarrollo integrado libre que se puede bajar de internet, pero cuidando el escoger la versión más

reciente. Después de instalado, su apariencia durante la ejecución es como aparece en la figura 23.

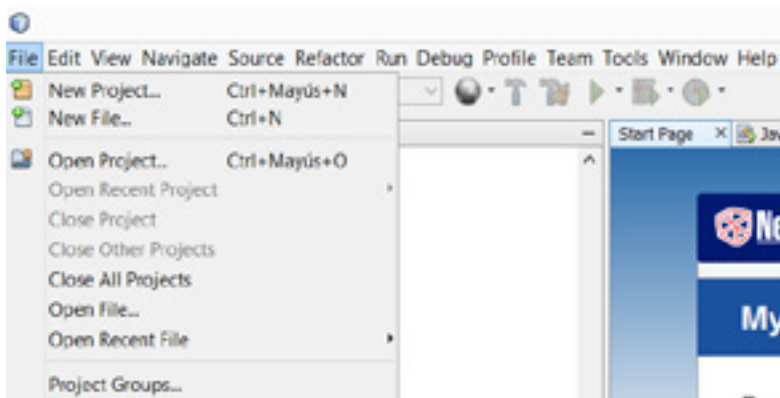
FIGURA 23. Apariencia de Netbeans



Para crear su primer programa en JavaScript se tienen varias opciones. Si no va a manejar base de datos, puede hacerlo en HTML5, el cual se podría correr sin necesidad de un servidor. Si se va a manejar base de datos, se puede escoger PHP o JSP. Aquí se utilizará PHP.

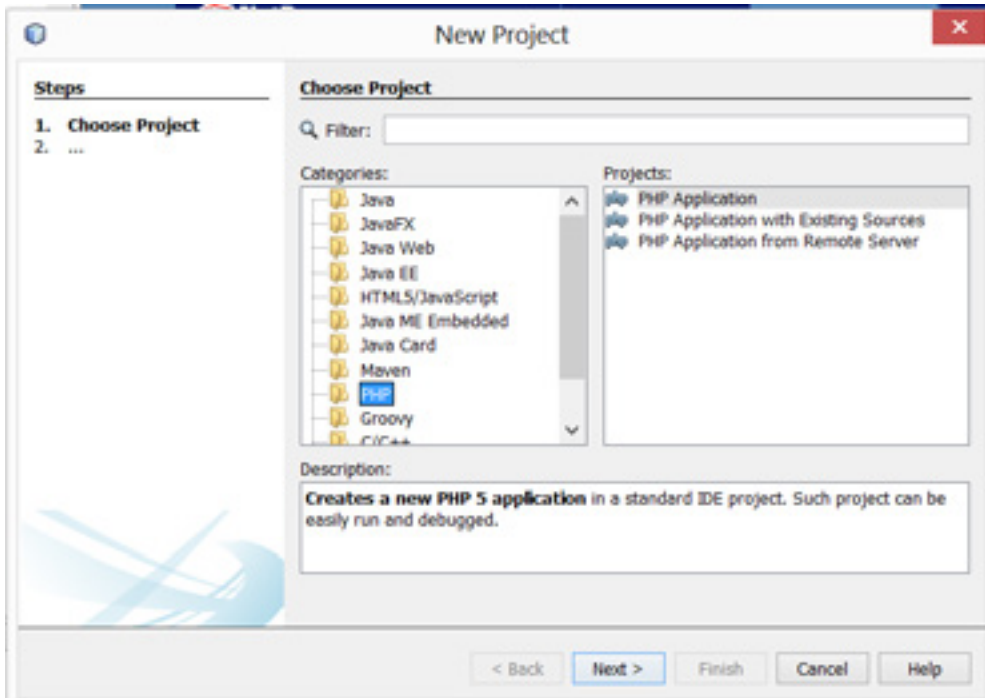
La figura 24 muestra que para hacer un programa se escoge la opción *file* y luego *New Project*.

FIGURA 24. Haciendo un programa en Netbeans



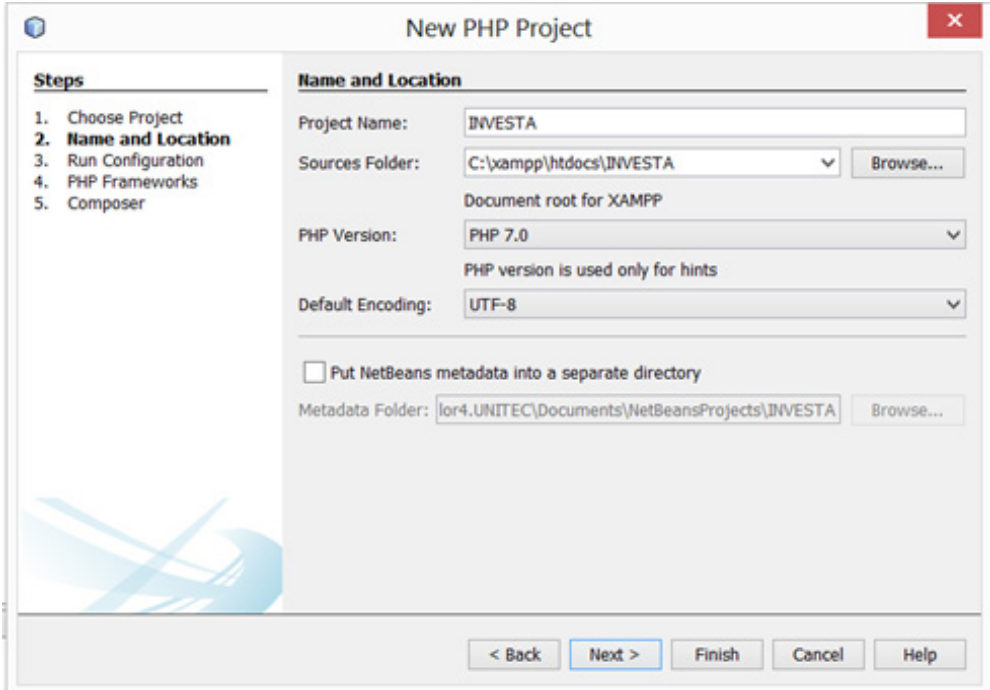
Aquí se muestra cómo se construyó el programa *Investa*. Se da clic sobre *New Project* y aparece la imagen de la figura 25. Obsérvese que está marcado PHP; si no lo está, lo marca y hace clic sobre *next*.

FIGURA 25. Creando el proyecto



En la figura 26, en la casilla *Project Name*, escribe *Investa* (el nombre del proyecto); luego haga clic sobre *finish*.

FIGURA 26. Nombre del proyecto



Netbeans le crea la plantilla que se puede observar en la figura 27. Este es un documento HTML, que contiene lo mínimo para hacer una aplicación. Lo primero que aparece es `<!DOCTYPE html>`, característico de las aplicaciones de HTML5. El color azul muestra el comienzo y el final del documento HTML. El color gris indica donde se abre y cierra el encabezado, mientras que el color verde en donde comienza y termina el cuerpo del documento.

FIGURA 27. Documento HTML

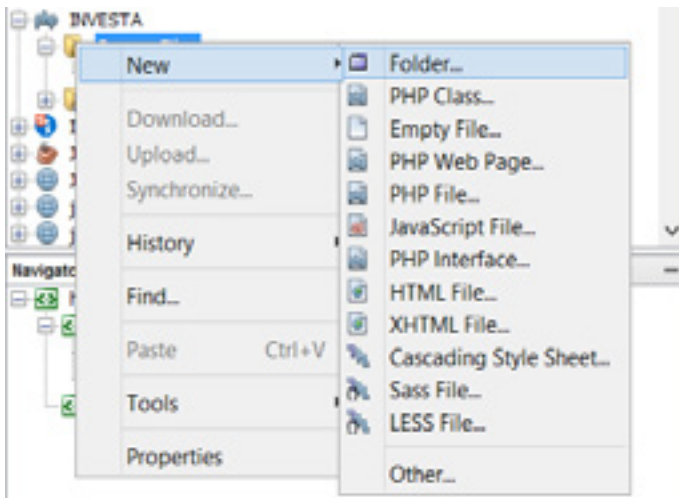
```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <?php
    // put your code here
    ?>
  </body>
</html>

```

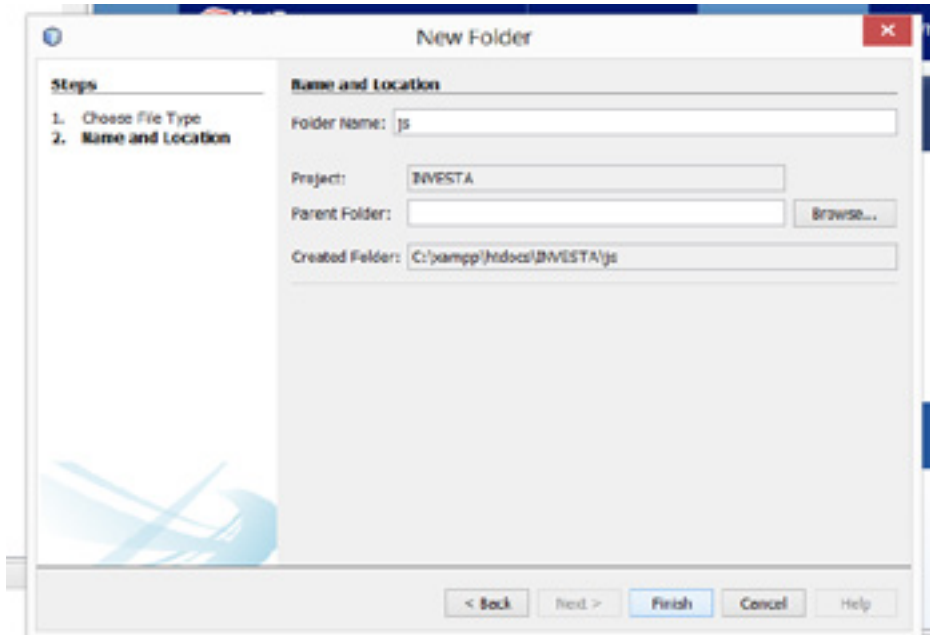
También se puede trabajar con un programa llamado jQuery (una versión sencilla de JavaScript) se puede descargar de internet; de nuevo, lo preferible es elegir la versión más reciente. Se coloca en la parte izquierda como se ve en la figura 28. Para esto se crea un fólder aparte.

FIGURA 28. Abriendo un fólder



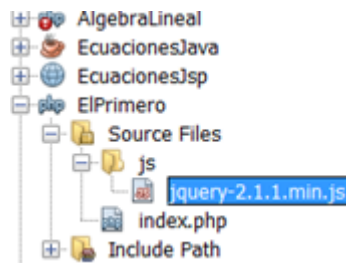
Se coloca el cursor sobre la opción *Source file* y se hace clic sobre el botón derecho del ratón. En *New*, se hace clic sobre *Folder*. Allí escribe *js* (figura 29); luego se hace clic en *finish*.

FIGURA 29. Nombre del f6lder



En la figura 30 se observa c6mo el archivo de jQuery qued6 contenido en la carpeta *js*.

FIGURA 30. jQuery



El jQuery se debe incluir en la cabecera del documento HTML de la siguiente forma:

```
<script src = "js/jquery - 2.1.1.min.s"
```

Cualquier programa que se desee utilizar con el documento HTML se debe incluir de esta manera. En la figura 31, se muestra su apariencia.

FIGURA 31. Inclusión de jQuery en el documento HTML

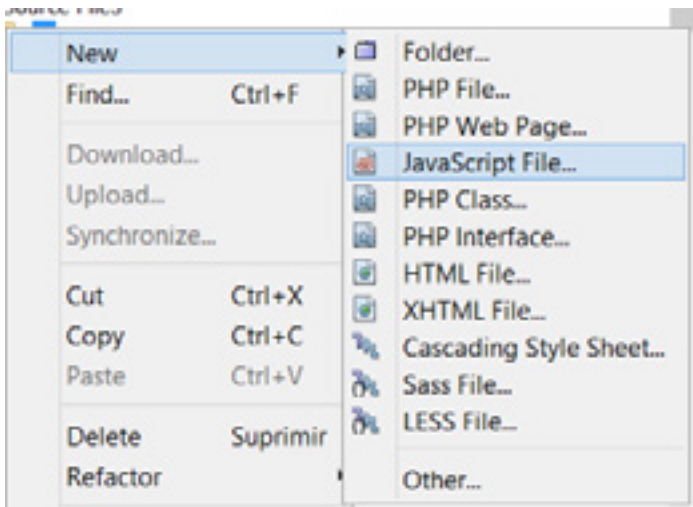
```

html>
<head>
  <meta charset="UTF-8">
  <script src="js/jquery-2.1.1.min.js"></script>
  <title></title>
</head>
<body>
  <?php
  // put your code here
  ?>
</body>

```

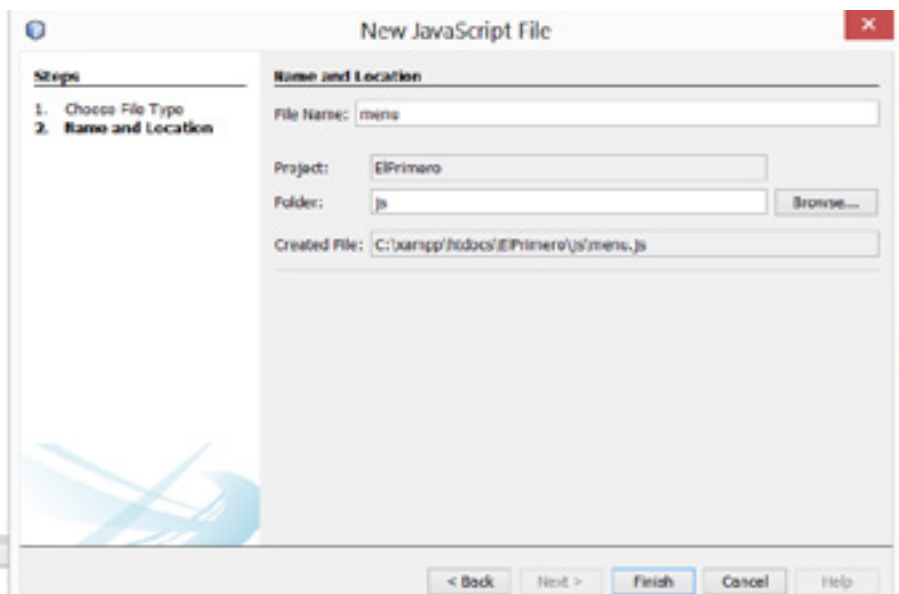
Dentro del fólдер *js* se pondrá otro archivo de JavaScript. En la figura 32 se muestran los pasos.

FIGURA 32. Un programa de JavaScript



Se le coloca el nombre *menú* y damos clic en *finish* (figura 33).

FIGURA 33. El archivo menú



En el archivo menú, se escribirá la función *menú*, que mostrará el menú principal; para lo cual se dividirá el documento HTML en cinco partes: *menPrin*, *menTem*, *menDesa*, *contenido* y *navegación* (figura 34). Para agregar información al documento HTML se utiliza *append* (véase el anexo 6 para su utilización). El `$(“body”)` indica que se está agregando el menú al cuerpo del documento, lo cual se realiza con el programa jQuery que se identifica con el signo `$`. El menú *menPrin* contendrá las secciones o diferentes niveles de un tema. El menú *menTem* contendrá el menú de los temas y *menDesa* contendrá el menú de los subtemas y el desarrollo de estos. Para dividir una página de internet se utiliza *div* (anexo 1). Observe que tiene un *id* que sirve para identificar el *div* y para programar eventos.

FIGURA 34. *Construyendo el menú principal*

```
function menu()
{
    $("body").append("<div id = 'menPrin' >");
    $("body").append("<div id = 'menTem' >");
    $("body").append("<div id = 'menDesa' >");
    $("body").append("<div id = 'contenido' >");
    $("body" ).append("<div id = 'navegacion' >");
}
```

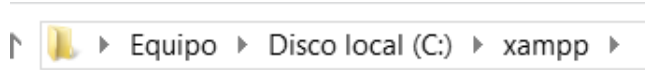
Pero este archivo se debe incluir en el documento para que HTML lo reconozca (figura 35).

FIGURA 35. *Inclusión del archivo menú en el documento HTML*

```
<html>
<head>
    <meta charset="UTF-8">
    <script src="js/jquery-2.1.1.min.js"></script>
    <script src="js/menu.js"></script>
    <title></title>
</head>
<body>
    <?php
    // put your code here
    ?>
</body>
```

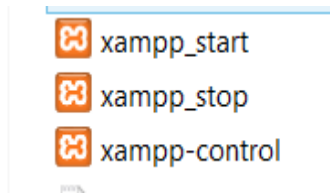
Para correr el programa, si se hace en un *software* diferente a HTML5 o si se utiliza base de datos, se debe hacer lo que se muestra en la figura 36. Después de instalado, el archivo *xampp* generalmente se encuentra en la carpeta C, a menos que lo haya instalado en otro lugar. En dicho caso, entre a la carpeta correspondiente.

FIGURA 36. Activando el servidor



Entre a la carpeta *xampp* y busque los programas *xampp_start* y *xampp_control*; los activa haciendo clic sobre ellos como aparece en la figura 37.

FIGURA 37. *Xampp_start* y *xampp_control*



Para crear los menús en las divisiones se utiliza el *li*; su forma es (`<li id = 'p0' class = 'prin' >Título del elemento `). El *id* se emplea para identificar el elemento del menú. La palabra clave *class*, para comportamientos comunes y *li* para crear una lista que corresponde a los elementos del menú.

En la figura 38 se puede ver cómo se crea el menú principal. Consulte el anexo 1 para el uso de ``. La sentencia `$(body).append(<div id = 'menPrin' >)` significa que se está agregando el menú principal al cuerpo del documento. Para dar al menú la apariencia de la figura 38, se da estilo a la hoja; esto se hará con el método *css* (anexo 2) de la siguiente manera: `objeto.css ({“atributo1”:” valor1”, “atributo2”:” valor2”, etc.})`. Además, se quiere escribir el título «Menú principal» con fondo gris, para lo cual se puede utilizar `h1`, `h2`... `h6`, según el tamaño del título. El título `h1` es el más grande. La sentencia sería: `< h3 > Menu principal < /h3 >`. En la figura 38 se puede observar cómo se coloca el título y se le otorga estilo a la hoja. Los códigos de los colores se pueden encontrar en internet. Algo muy importante es `< li id = 'p0' >` porque es la clave de cómo se identifica la información en la base de datos. En esta, el campo clave es `Id_tema = 0`, que coincide con el segundo carácter del *id* del elemento del menú. De esta manera, el programa sabe que se trata del tema conceptualización. A la clase *prin* se le agrega un evento clic que llama a la función *tema ()*, la cual tiene un parámetro

que representa el *id* del tema que se va a mostrar. Las clases se identifican con un punto. Observe el uso de *attr* para capturar el atributo del objeto. La palabra *this* se relaciona con *prin*.

FIGURA 38. Menú de entrada

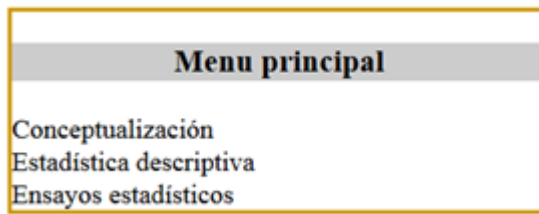
```

function menu()
{
  var id_Tem;
  $("body").append("<div id = 'menPrin' >");
  $("body").append("<div id = 'menTem' >");
  $("body").append("<div id = 'menDesa' >");
  $("body").append("<div id = 'contenido' >");
  $("body").append("<div id = 'navegacion' >");
  $("#menPrin").append("<h3 id = 'tit' >Menu principal</h3 >");
  $("#menPrin").append("<li id = 'p0' class = 'prin' >Conceptualización</li >");
  $("#menPrin").append("<li id = 'p1' class = 'prin' >Estadística descriptiva</li >");
  $("#menPrin").append("<li id = 'p2' class = 'prin' >Ensayos estadísticos</li >");
  $("#menPrin").css({"border": "#cc9900",
    "border-style": "solid",
    "width": "25%",
    "list-style-type": "none"});
  $("#tit").css({"background": "#cccccc", "text-align": "center"});
  $(".prin").click(function(){
    var id = $(this).attr("id");
    temas(id);
  });
}
//menu

```

Al correr el programa se obtiene la imagen que aparece en la figura 39.

FIGURA 39. Imagen del menú principal



Pero para correr la función *menu* (), se llama desde la hoja HTML como se muestra en la figura 40. Observe que aparece entre `<script>` `</script>`, que se utiliza para correr archivos JavaScript desde HTML.

FIGURA 40. Llamando a la función *menu* ()

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <script src="js/jquery-2.1.1.min.js"></script>
    <script src="js/menu.js"></script>
    <title>INVESTA</title>
  </head>
  <body>
    <script>
      menu();
    </script>
  </body>

```

La función *tema* (figura 41) sustrae el número de la sección que comienza en el carácter uno. El carácter de la posición cero es una letra. Se crea un arreglo con los temas de cada sección. Se construye un *div*, identificado con la palabra *div* y el número de identificación de la sección, y a este *div* se le agregan los temas correspondientes. Observe que la sección se identifica con la variable *idInt*. Cuando se lee *maMenu [1]*, significa que se le está agregando a la sección 1 (estadística descriptiva) los temas correspondientes. El evento clic de *subMen* recupera el

atributo del tema sobre el cual se hace clic mediante la sentencia `var id = $(this).attr("id");` luego, se llama a la función `subTemas`, pasando el parámetro `id` que identifican el tema.

FIGURA 41. Función temas ()

```

function temas(id)
{
    var idF =id.substr(1) ;
    var idTema= idF;
    var idInt = parseInt(idF);
    var maMenu =[];
    maMenu[0] = "<ul><li id='t0' class='subMen'>Conceptualización</li></ul>";
    maMenu[1] = "<ul>"+
        "<li id='t1' class='subMen'>Escalas</li>"+
        "<li id='t2' class='subMen'>Distribución de Frecuencias</li>"+
        "<li id='t3' class='subMen'>Medidas de Tendencia Central</li>"+
        "<li id='t4' class='subMen'>Mediciones de Variabilidad</li>"+
        "</ul>";
    maMenu[2] = "<ul>"+
        "<li id='t5' class='subMen'>Correlación</li>"+
        "<li id='t6' class='subMen'>Prueba t </li>"+
        "<li id='t7' class='subMen'>Análisis Entre Varianza</li>"+
        "<li id='t8' class='Otras'></li>"+
        "</ul>";
    $('#div'+idInt).empty();
    $("#" +id).append("<div id ='div'+idInt+'>");
    $('#div'+idInt).append( maMenu[idInt]);
    $(".subMen").click(function(){
        var id = $(this).attr("id");
        id_Tem= id.substr(1) ;
        subTemas(id);
    });
}

```

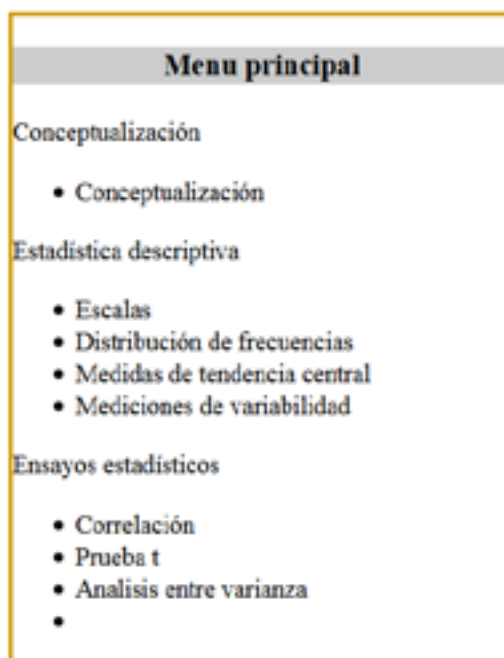
En la tabla 2 se detallan las sentencias de las líneas de la figura 41 con su respectiva explicación.

TABLA 2. Explicación de los comandos de temas

Sentencia	Explicación
<code>temas[id]</code>	Recibe identificación de la sección sobre el cual se hizo clic.
<code>var iff = id.substr(1);</code>	Asigna desde el segundo carácter de identificación a la variable <code>idInt</code> .
<code>var maMenu = [];</code>	Crea una matriz vacía
<code>maMenu[i]</code>	Se le está asignando valor al elemento <code>i</code> de la matriz. El valor son los elementos de otro menú. Se usa <code></code> para darle una sangría a los submenús.
<code>\$('#div' + idInt).empty();</code>	Se está limpiando la división correspondiente. Observe que el valor <code>idInt</code> depende del <code>id</code> que se pasa.
<code>\$('#' + id).append("<div id = 'div' + idInt + '>");</code>	Se está creando una división para los elementos del menú.
<code>\$('#div' + idInt).append(maMenu[idInt]);</code>	Se agregando el valor de la matriz al menú.

En la figura 42 se puede observar el menú principal con los temas de cada sección.

FIGURA 42. El menú principal con sus submenús



En las figuras 43, 44 y 45 se muestra cómo se construye el submenú de los subtemas; observe que se efectúa de la misma manera que se creó `menu()`, puede seguir este patrón para hacer sus menús. La función comienza recuperando el número del tema al cual se le van a agregar los subtemas. Se llama a la función `aja(idInt)`, se crea un arreglo con los subtemas de cada tema y se le agregan a la división `menDesa`. Se crea un evento clic para la clase `subi`; al hacer clic, se envía el `id`, que corresponde al subtema, a la función `desaSubtemas`; `idTema` es el número del tema.

FIGURA 43. Función `subtemas` (parte 1 de 3)

```

Function subTemas(id)
{
    var idF=id.substr(1) ;
    var idInt = parseInt(idF);
    aja(idInt);
    var subtemas = [];
    subtemas[0]="<ul>"+
        "<li id='c0' class='subi' >Introducción</li>"+
        "<li id='c1' class='subi'>Objetivos</li>"+
        "<li id='c2' class='subi'>Metodología </li>"+
        "<li id='c3' class='subi'>Prerrequisitos</li>"+
        "</ul>";
    subtemas[1]="<ul><li id='c4' class='subi' >Medición</li>"+
        "<li id='c5' class='subi' >Nominal</li>"+
        "<li id='c6' class='subi'>Ordinal</li>"+
        "<li id='c7' class='subi'>De intervalo </li>"+
        "<li id='c8' class='subi'>De razón</li>"+
        "<li id='a10' class='subi'>Actividades</li>"+
        "<li id='e11' class='subi'>Evaluación</li></ul>";
    subtemas[2]="<ul><li id='c11' class='subi' >Ordenación</li>"+
        "<li id='c11 class='subi' >Rango</li>"+
        "<li id='c12' class='subi'>Distribución</li>"+

```



```

"<li id='c13' class='subi'> </li>" +
"<li id='c14' class='subi'></li>" +
"<li id='a15' class='subi'>Actividades</li>" +
"<li id='e16' class='subi'>Evaluación</li></ul>";

```

FIGURA 44. Función subtemas (parte 2 de 3)

```

subtemas[3]="<ul><li id='c15' class='subi' >Media</li>" +
"<li id=c16' class='subi' >Mediana</li>" +
"<li id='c17' class='subi'>Moda</li>" +
"<li id='c18' class='subi'> Geométrica</li>" +
"<li id='c19' class='subi'>Armónica</li>" +
"<li id='a20' class='subi'>Actividades</li>" +
"<li id='e21' class='subi'>Evaluación</li></ul>";

subtemas[4]="<ul><li id='c20' class='subi' >Cuartil</li>" +
"<li id='c21' class='subi' >Percentil</li>" +
"<li id='c22' class='subi'>Típica</li>" +
"<li id='c23' class='subi'> Varianza</li>" +
"<li id='c24' class='subi'>Actividades</li>" +
"<li id='c25' class='subi'>Evaluación</li></ul>";

$("#menDesa").empty();
$("#menDesa" ).append( subtemas[idInt]);
$("#menDesa").append("<h3 id='subtit'>Subtemas</h3>");
$("#menDesa").css({"border":"#cc9900",
"border-style":"solid",
"width":"60%",
"list-style-type":"none",
"position":"absolute",
"top":16,
"left":"20.5em"});
$("#subtit").css({"background":"#cccccc","text-align":"center"});
$("#li.subi").css({"float":"left"});

```

FIGURA 45. Función subtemas (parte 3 de 3)

```

.....
$(“li”).css({“width”:100});
    $(“#subtit”).css({“background”：“#cccccc”,“text-align”：“center”});
    $(“li.subi”).css({“float”：“left”});
    $(“li”).css({“width”:100});
    $(“#contenido”).
css({“position”：“absolute”,“top”：“7em”,“left”：“21em”,“width”：“48em”});
    $(“.subi”).click(function(){
        var id = $(this).attr(“id”);
        desaSubtemas(id);
    });
} //subtemas
.....

```

Los subtemas se van a situar en la parte derecha del menú principal. Se van a ubicar con un comando de css, *position*, con el siguiente orden: *menDesa.css*({*position* : *absolute*, *top* : *altura*, *left* : *distancia hacia la derecha*}). Con *float* se construye un menú horizontal. En la figura 46 se observan los subtemas del tema escalas.

FIGURA 46. Menú con sus subtemas

Menu principal	Subtemas
Conceptualización Estadística descriptiva <ul style="list-style-type: none"> • Escalas • Distribución de frecuencias • Medidas de tendencia central • Mediciones de variabilidad Ensayos estadísticos	Medición Nominal Ordinal De intervalo De razón Actividades Evaluación

Creada la interfaz, se procede al desarrollo de los subtemas. La información correspondiente se puede guardar en una base de datos. En el anexo 3 se explica cómo se construyó, tratando de hacer las consultas lo más sencillo posible, para que el lector que no tenga conocimiento del tema lo pueda comprender fácilmente.

Este objeto de aprendizaje utiliza funciones genéricas. Es decir, todos los temas y subtemas son manejados por las mismas funciones. Los diseñadores de material instruccional pueden optar por escribir una función por temas, aunque se repita el código; pero lo más profesional es escribir funciones genéricas. Para explicar el programa se utilizará el tema escalas de la estadística descriptiva. Para mostrar los demás temas solo se necesita que estén en la base de datos.

En la figura 47 está la función *aja* que recibe la identificación del elemento sobre el cual se hizo clic. Esta función hace uso de una tecnología llamada Ajax. La primera parte hace un llamado a la función *tema.php* y le pasa el número del tema que se va a tratar. Los resultados son recibidos por una función anónima que captura los datos y se los asigna a la variable *responseText*, cuyo valor pasa a la variable *d* que es mostrada en la división *contenido*. Con la sentencia condicional se controla algún error al leer la base de datos.

FIGURA 47. Función *aja* ()

```
function aja(str)
{
    var response;
    $.ajax({
        url: 'php/tema.php',
        type: 'POST',
        timeout: 12000,
        dataType: 'text',
        data:{cod:str}
    }).done(function( responseText ){
        var d = String(responseText);
        if(d !== undefined) {
            $("#contenido").append(d);
        }
    }
    else {
        alert("No se encontró el código,inténtelo nuevamente.");
    }
    }).fail(function() {
```

```

        alert('El código '+str+" no se encuentra en la base de datos.");
    });
} //bd

```

En la figura 48 se muestra la función que maneja la base de datos. Tiene una clase y dos funciones miembros (temas que se explican en el anexo 4). La información que aparece en *header* es para que se muestren las tildes y otros elementos diacríticos. En el mismo anexo se explica el arreglo.

FIGURA 48. Función que maneja base de datos (1 de 3)

```

header('Content-Type: text/html; charset=ISO-8859-1');
class tema {
    public $tema = array(
        "id"=>"",
        "tit"=>"",
        "conte"=>""
    );
}

```

En la figura 49 se muestra la función *bdDatos()*, que es miembro de la clase *tema*. Las primeras cuatro líneas corresponden a la información para acceder a los datos. Con esta información se establece la conexión:

```
$conn = new mysqli($servername, $username, $password, $dbname)
```

Luego se asigna el número del tema enviado por la función *tema* a la variable *\$str*. Viene la cadena de consulta en MySQL, la cual se explica en el anexo 5. Se envía la cadena para hacer la consulta y los datos se leen mediante un *while* y se asignan estos valores al arreglo declarado.

FIGURA 49. Función que maneja base de datos (2 de 3)

```

function bdDatos() {
    $servername = "localhost";
    $username = "root";
}

```

```

$password = " "; //Escribir aquí la contraseña que posee para entrar a la base de datos.
$dbname = "estadistica";
$resu = "";
// Create connection
$conn = new mysqli($servername,$username,$password,$dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$str =filter_input(INPUT_POST,"cod");
$sql = "SELECT * FROM tema WHERE Id_tema = ".$str;
$result = $conn->query($sql);
$cct=0;
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $this->tema["id"]=$row["Id_tema"];
        $this->tema["tit"]=$row["Titulo_tema"];
        $this->tema["conte"]=$row["Contenido_tema"];
    }
} else {
    echo "0 results";
}
$conn->close();
}

```

En la figura 50 se muestra la función miembro que muestra los datos. Se utiliza la sentencia *echo* que se explica en el anexo 5. La clase termina y luego se declara un objeto con base en la clase para mostrar los datos en pantalla.

FIGURA 50. Función que muestra los datos (3 de 3)

```

function displayTit()
{
    echo "<b>".$this->tema["tit"]."</b><br><br>";
    echo $this->tema["conte"];
}

```

```
//clase
$tem = new tema;
$tem->bdDatos();
$tem->displayTit();
```

En figura 51 aparece la función *desaSubtemas* que recibe un parámetro: la identificación del tema. Hay una sentencia condicional que evalúa el valor del primer carácter de la cadena *id*. Si el valor es *c* se invoca a la función *aja2* () con la identificación del subtema; si es *a* se llama a *aja3* () y pasa la identificación del subtema y del tema. Lo mismo sucede con la función *aja4* (). En estos últimos la identificación del tema es importante porque es la clave para leer la información de la base de datos.

FIGURA 51. Función que llama a los subtemas, a las actividades o a la evaluación

```
function desaSubtemas(id)
{
    var idStr = id.substr(1);
    var idInt = parseInt(idStr);
    var idTema=id_Tem;
    if(id[0]==="c")
    {
        aja2(idInt);
    }
    else if(id[0]==="a"){
        aja3(idTema);
    }
    else if(id[0]==="e"){
        aja4(idTema);
    }
    $("#contenido").
css({"position":"absolute","top":"7em","left":"21em","width":"48em"});
}
//desaSubtemas(id)
```

En la figura 52 se observa la función *aja2(str)*, que llama al *subtema.php* que lee la base datos para recuperar los subtemas. Estos últimos contienen varios ejemplos que se muestran al hacer clic sobre un botón. Observe que los datos recibidos en forma de una cadena son asignados a un arreglo llamado *conte*. Los asteriscos dobles sirven como marca para separar la cadena mediante una función *split* de JavaScript (véase anexo 6). Luego el elemento dos de *conte* es, a su vez, separado en otro arreglo que contiene los ejemplos. La variable *str* contiene la identificación del subtema. Los subtemas comienzan a aparecer cuando *str* es mayor de 4; si es menor, el programa muestra el contenido de la conceptualización.

FIGURA 52. Función que maneja los subtemas (parte 1 de 2)

```
function aja2(str)
{
$.post("php/subtema.php",
{
cod: str
},
function(data,status){
var d= data;
var conte =d.split("**");
$("#contenido").empty();
var con = conte[1].replace("-|","Σ");
$("#contenido").append(conte[0]+"<br><br>"+con);
$("#contenido").
css({"position":"absolute","top":"7em","left":"21em","width":"48em"});

//0: Títulos subtemas.1: Contenido subtemas.2: Ejemplos subtemas.3: Razones
subtemas.

var ejem =conte[2].split("//");
var ef=0;

if (str >4) //Menor de 4 son los contenidos de la conceptualización.
```

```
{
  bot();
}
```

La figura 53 muestra la parte final de la función *aja2()*. Se trata de la función *bot*. En esta se crean dos botones para navegar entre los ejemplos: el botón 1 retrocede y el botón 2 adelanta. La expresión *ejem.length* contiene el número de ejemplos. La sentencia `
` agrega una nueva línea.

FIGURA 53. Función que maneja los subtemas (parte 2 de 2)

```
function bot ()
{
  $("#navegacion").empty();
  $("#navegacion").append("<br><br>Ejemplos<input type='button' id='btn1'
value='Ejemplo'>");
  $("#navegacion").append("<input type='button' id='btn2' value='Siguiente'>");
  $("#contenido").append("<div id='ejemplos'>");
  $("#btn1").click(function(){
    ef--;
    if (ef < 0)
    {ef=0;
    }
    $("#ejemplos").empty();
    $("#ejemplos").append("<br><br>"+ejem[ef]);
    $("#btn1").val("Anterior");
  });
  $("#btn2").click(function(){
    ef++;
    if (ef >=ejem.length-1)
    {ef=0;
    }
    $("#ejemplos").empty();
    $("#ejemplos").append("<br><br>"+ejem[ef]);
  });
}
```



```
});
} //aja2
```

La figura 54 llama a la función que trae los subtemas de la base de datos. Contiene un *header* que permite que aparezcan las tildes y las eñes en el documento y una clase (véase anexo 4) que contendrá los datos provenientes de la base de datos.

FIGURA 54. *Función que trae los subtemas de la base de datos (parte 1 de 3)*

```
<?php
header('Content-Type: text/html; charset=ISO-8859-1');
class subtema {
    var $subtema = array(
        "id_subtem" => "",
        "cont_subtema" => "",
        "tit_subtema" => ""
    );
```

En la figura 55 se continúan declarando otros arreglos y se comienza a leer la base de datos.

FIGURA 55. *Función que trae los subtemas de la base de datos (parte 2 de 3)*

```
var $ejemplo = array(
    array(
        "id_ejem" => "",
        "cont_ejem" => ""
    );
var $razon = array(
    array(
        "id_ejem" => "",
        "razon_ejem" => ""
    );
function bdDatos() {
```

```

$servername = "localhost";
$username = "root";
$password = " "; //Escribir aquí la contraseña que posee para entrar a la base de
datos.
$dbname = "estadistica";
$resu = "";
// Create connection
$conn = new mysqli($servername,$username,$password,$dbname);
// Check connection
$str =filter_input(INPUT_POST,"cod");
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT * FROM `ejemplo`,subtema WHERE subtema.Id_subtema = ".$str."
AND subtema.Id_subtema = ejemplo.Id_subtema";
$result = $conn->query($sql);
$ct=0;
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {

```

En la figura 56 termina la función. Nótese los asteriscos en la expresión: “**” `.$this->subtema[“cont_subtema”].**` y las barras inclinadas (`//`). Los dobles asteriscos sirven para separar títulos, subtemas y ejemplos, mientras que las barras inclinadas se utilizan para separar los ejemplos; el proceso de separación se hace mediante el comando *split* (véase el anexo 6). La lectura de la base de datos es parecida a la anterior.

FIGURA 56. Función que trae los subtemas de la base de datos (parte 3 de 3)

```

    $this->subtema[“id_subtem”]=$row[“Id_subtema”];
    $this->subtema[“cont_subtema”]=$row[“Contenido”];
    $this->razon[$ct][“razon_ejem”]=$row[“Razon_ejemplo”];
    $this->subtema[“tit_subtema”]=$row[“Titulo”];
    $this->ejemplo[$ct][“cont_ejem”]=$row[“Contenido_ejemplo”];
}

```

```

} else {
    echo "0 results";
}
$conn->close();
$resu = $this->subtema["tit_subtema"]."***".$this->subtema["cont_subtema"]."***";
$qty = count($this->ejemplo)-2;
for($i=1;$i<=$qty;$i++)
{
    $resu = $resu.$this->ejemplo[$i]["cont_ejem"]."//";
}
$resu = $resu."***";
for($i=1;$i<=$qty;$i++)
{
    $resu = $resu.$this->razon[$i]["razon_ejem"]."//";
}
echo $resu;
}
}

```

La figura 57 muestra la función *aja3* que tiene como objetivo presentar las actividades con sus opciones y su respectiva retroalimentación. La estructura es la que utiliza la tecnología Ajax. Primero se observa la función que llama a la función PHP, *actividad2.php* que es la que permite la consulta de la base de datos. Se pasa el número del tema con la variable *cod* y en seguida se observa la función anónima que recibe los datos. En esta última función se crean tres funciones: *punto ()*, *acti1* y *acti2*.

FIGURA 57. Función de actividades

```

function aja3(cade)
{
    $.post("php/actividad2.php",
    {
        cod: cade
    },

```

```
function(data,status){
    var d= data;
    var conte =d.split("**");
    /*
    * 0 Título de la actividad. 1. Contenido de la actividad. 2.Opciones 3.
    Retroalimentación
    * 4. Domes    5. Claves
    */
    var contRetro;
    var te;
    var nAct=0; //Número del punto.
    var mAct=0;
    var ca=0;
    var nota =[];
    var cantPT;
    var tx=0;
    var nn=0;//Número de la nota
    puntos();
}
```

En la figura 58 se observa la función *punto ()*. Cada elemento del arreglo *conte* se separa con la función *Split* de JavaScript. El nombre de las variables es sugestivo del contenido: *tit* se refiere al título y *cont* al contenido. Aquí se crea el arreglo de las opciones de las actividades de selección múltiple. La expresión *contDomes* se refiere a los objetos como cajas de textos, botones o botones de radio. La declaración *tipOp* es una cadena que le sobra una coma y por eso se le suprime y se crea la cadena *tip*; la cual es convertida en un arreglo mediante la sentencia *JSON.parse()*.

FIGURA 58. Creación de las partes de una actividad (1 de 12)

```
function puntos ()
{
    var titActividad =conte[0].split("//");
    var contActividad = conte[1].split("//");
    var contOpciones=conte[2].split("//");
}
```

```

        contRetro=conte[3].split("//");
var contDomes =conte[4].split("//");
var contClaves =conte[5].split("//");
var tipOp =conte[6];
var co = tipOp.lastIndexOf(",");
var tip =tipOp.substr(0,co);
tip= tip+"]";
var tipi = JSON.parse(tip);
cantPT= contOpciones.length-1; //Cantidad de opciones totales
var cantEn =titActividad.length-1;// Cantidad de enunciados
mAct= cantEn;
var cantOp = cantPT/cantEn; // cantidad de opciones por punto
var maxFil = parseInt(cantEn);
var maxCol = parseInt(cantOp);
var op = new Array (maxFil);//op: Opciones. maxFil: Cantidad de puntos

```

La figura 59 corresponde a la misma función y se está haciendo el mismo proceso, creación de las partes de las actividades. La variable *ob* denota a los objetos que se manejan en la actividad. Mediante esta variable se capturan los radios de botones o las cajas que vienen creadas desde la base de datos y *op*, denota las opciones de las actividades.

FIGURA 59. Creación de las partes de una actividad (2 de 12)

```

for(var i=0; i<maxFil;i++)
{
    op[i]=new Array (maxCol);//maxCol: cantidad de opciones
}
var j=0;
var k=0;
for(var i=0; i<cantPT;i++)
{
    op[k][j]= contOpciones[i] ;
    j++;
    if(j>{maxCol-1})
    {

```

```

        j=0;
        k++;
    }
}
//*****Objetos *****
    var ob = new Array (maxFil);//Cantidad de puntos
    for(var i=0; i<maxCol;i++)
    {
ob[i]=new Array (maxCol);//ob: Objetos dom (cajas de textos y botones de radio)
cantidad de opciones
    }

```

En la figura 60 se está asignando un valor al arreglo *ob*, el cual puede tener una o dos dimensiones.

FIGURA 60. Creación de las partes de una actividad (3 de 12)

```

j=0;
k=0;
if ((maxCol > 1)
{
for (var i=0; i < cantPT;i++)
{
    ob[k][j]= contDomes[i] ;
    j++;
    if(j>(maxCol-1))
    {
        j=0;
        k++;
    }
}
}
}
else
{
for(var i=0; i<(contDomes.length-1);i++)

```

```

{
  ob[0][i]= contDomes[i];
}
}

```

En la figura 61 se le están asignando los valores de las claves de las respuestas y los objetos (como las cajas de textos y los botones de radio).

FIGURA 61. Creación de las partes de una actividad (4 de 12)

```

//*****Claves*****
if (maxCol ===1)
{
  ob [0][0]=contDomes [0];
}
var cla= new Array (maxFil);//Cantidad de claves
for (var i=0; i < maxFil; i++)
{
  cla[i]=new Array (maxCol);//cantidad de opciones
}
j=0;
k=0;
for (var i=0; i<cantPT;i++)
{
  cla[k][j]= contClaves[i] ;
  j++;
  if(j>(maxCol-1))//
  {
    j=0;
    k++;
  }
}
}

```

En la figura 62 se asigna la retroalimentación y se escriben los títulos en la división; con la función de JavaScript *indexOf()* se determina si la información

de la base de datos se refiere a una función que crea el programa o a una información de uso inmediato.

FIGURA 62. Creación de las partes de una actividad (5 de 12)

```

-----
// *****Retroalimentación *****
    var ret = new Array (maxFil);//Cantidad de claves
    for (var i=0; i< maxFil; i++)
    {
        ret[i]=new Array (maxCol);//cantidad de opciones
    }
    j=0;
    k=0;
    for (var i=0; i<cantPT;i++)
    {
        ret[k][j]= contRetro[i];
        j++;
        if(j>(maxCol-1))
        {
            j=0;
            k++;
        }
    }
    /* Matriz de puntos por opciones *****
    $("#contenido").empty();
    //***** Título y contenido de la actividad
    *****
    $("#contenido").append("<b></b><br><br>" +titActividad[nAct]+ "<br><br>" +contActividad[nAct]+ "<br><br>");
    //***** Subtema ordenación*****
    $("#contenido").append("<table>");
    //***** Opciones y objetos *****
    var fun = cla[0][0].toString()
    var inCla =fun.indexOf("funcion");
    var ind2;
-----

```


En la primera línea de la figura 63 se determina si la actividad la crea el programa o ya está resuelta en la base de datos. Un valor diferente a 2 significa que es una actividad guardada en la base de datos. Luego se determina qué tipo de objetos maneja la actividad; si el valor de la variable *ind2* es diferente de -1, se trata de botones de radio. En seguida se crea una tabla en la división contenido donde se muestran los objetos que van recibir la entrada del usuario; el valor de la variable -1 indica que no se trata de botones de radio. En este caso se llama a *acti1* y se pasan las variables que contienen las claves y los objetos. Si se trata de cajas de textos, se llama a *acti2* y se envía la retroalimentación. Si el valor de la variable *tipi[nAct]* es 2, el programa debe crear los datos de la actividad y para eso llama a la función *dos*.

FIGURA 63. Creación de las partes de una actividad (6 de 12)

```

if (tipi[nAct] !==2) {
    ind2 = ob[nAct][0].indexOf("rad");
    $("#contenido").append("<table>");
    for (var i=0; i< cantOp; i++) {
        $("#contenido").append("<tr><td>" +op[nAct][i]+"</td><td>" + ob [nAct]
[i]+"</td></tr>");
    }//for
    $("#contenido").append("</table>");
    if(ind2 ===-1)
    {
        acti1(cla,ob);
    }//if
    else
    {
        acti2(ret);
    }
}
else//***** El programa resuelva la actividad
{
    ind2 = ob[0][nAct].indexOf("txt");
    $("#contenido").append("<table>");
}

```

```

for (var i=0; i<cantOp; i++)
{
    $("#contenido").append("<tr><td>" + op[nAct][i] + "</td><td></td></tr>");
    $("#contenido").append("<tr><td>" + ob[0][nAct] + "</td><td></td></tr>");
    $("#contenido").append("</table>");
    dos(ind2,ob,cla);
} //for

```

En la figura 64 se crean dos botones de navegación para avanzar o retroceder en las actividades. Luego aparece la función *dos* que recibe tres parámetros. Dentro de esta función se verifica el valor de la variable *inCla*; si es -1, se asigna a la variable *maOrd* el valor de la cadena *ob[0][0]*. En seguida se llama a la función ordenación (véase anexo 8) para que organice el arreglo en forma ascendente. Si el valor de la variable no es -1, se crea un evento *keypress* que convierte la cadena *cla* en un arreglo y llama a la función *retros()*, que devuelve un valor que se le asigna a la variable *y*. Después, se convierte la variable *re[ca]* en variable numérica.

FIGURA 64. Creación de las partes de una actividad (7 de 12)

```

$("#contenido").append("</table>");
$(".texto").css({"width":20});
$("#contenido").
css({"position": "absolute", "top": "7em", "left": "21em", "width": "48em"});
} //puntos
$("#navegación").empty();
$("#navegacion").append("<h3>Actividades<h3><input type='button' id='btn1'
value='Anterior'>" +
    "<input type='button' id='btn2' value='Siguiete'>");
function dos (inCla, ob, cla)
{
    var maOrd;

    if (inCla === -1)
    {
        maOrd = JSON.parse(ob [0][0]);

```

```

ordenacion(maOrd);
}
else
{
    $(".txt"). keypress(function(e){
        if(e.keyCode ===13)
        {
            var atr = $(this).attr("id");
            var tx=$("#"+atr). val ();
            var ca =atr [3];
            var re =JSON.parse(cla[nAct] [0]) ;
            var y= retres ();
            var sis = parseInt(re[ca]);

```

En la figura 65 se convierte la variable *tx*, que contiene el valor digitado por el usuario (en valor numérico). Luego, se compara el valor obtenido por el sistema con el digitado por el usuario. Si son iguales se emite un mensaje contenido en el primer elemento del arreglo *y*. Si son diferentes se muestra el valor del segundo elemento del mismo arreglo. A continuación aparece parte de la función *retres()*. El valor de la cadena *conRetro[nAct]* se convierte en un arreglo. Si el tipo de dato del primer elemento de este arreglo es de tipo *string* (cadena), se le asigna a *rete[0][k]*.

FIGURA 65. Creación de las partes de una actividad (8 de 12)

```

var user = parseInt(tx);
if(sis === user)
{
    alert(y[0][ca]);
}
else
{
    alert(y[1][ca]);
}
} //13

```

```

        });
    }//else
}
function retres ()
{
    var obj = JSON.parse(contRetro[nAct]);
    var tipo=typeof(obj[0]);
    var rete =[];
    if(tipo === "string")
    {
        for(var k=0;k<obj.length;k++)
        {
            rete[0][k] =obj[k];
        }
    }

}//if

```

La figura 66 muestra que si el dato no es de tipo *string* se crea un arreglo bi-dimensional, donde su dimensión *k* es del tamaño del arreglo *obj*. Luego se le asigna el valor de *ob* al arreglo *rete*, que contiene la información que la función *retres ()* devuelve cuando es llamada. En esta misma figura aparece la función *acti1()* con dos parámetros. Se crea un evento *keypress* que se activa al presionar la tecla enter. La identificación del objeto se pasa a la variable *atr* y, mediante ella, se recupera el valor de la caja de texto que se le asigna a la variable *tx*. El tercer elemento del valor de la cadena se le asigna a *ca*. Se llama a la función *retres()* que devuelve un valor que contiene una retroalimentación que se le asigna a *rete*.

FIGURA 66. Creación de las partes de una actividad (9 de 12)

```

else {
    for(var k=0;k<obj.length;k++)
    {
        rete[k] =new Array (obj [0]. length);
    }
}

```

```

        for(var j=0;j<obj[0].length;j++)
        {
            for(var k=0;k<obj.length;k++)
            {
                rete[k][j] =obj[k][j];
            }
        }
    }//else
    return rete;
}
function act1(cla,ob)
{
    $(".txt").keypress(function(e){
    if(e.keyCode ===13)
    {
        var atr = $(this).attr("id");
        var tx =$("#"+atr).val ();
        var ca =atr [3];
        var re;
        var rete = [];
        nAct=ca;
        rete = retres ();

```

En la figura 67, si la clave es igual a la cadena “*funcion2*”, se llama a la función *distri*, cuya codificación aparece en el anexo 8; esta función devuelve una retroalimentación. Si la clave no es igual a la cadena, la retroalimentación viene de la base de datos y está contenida en el arreglo *rete*.

FIGURA 67. Creación de las partes de una actividad (10 de 12)

```

if(cla[0][nAct]==="funcion2")
{
    distri(rete[nAct] [0][0], ca,tx);
}
else {
    var pos;

```

```

        var pos2;
        var rec, rec2;
    var rec2min, recmin;
    //***** Retroalimentación *****
    for (var t =0;t<rete[0].length;t++)    {
        pos = rete[0][t]; //Las respuestas claves
        var rec3=recorte(pos);
        var rec4=recorte(tx);
        rec =rec3.toLowerCase();
        rec2 =rec4.toLowerCase();
        if(rec ===rec2)
        {
            alert(rete[1][t]); //La retroalimentación
        }
    }
} //keypress
}

```

En la figura 68 se muestra un evento clic que se activa al escoger un botón de radio. El *id* del botón de radio se captura con la variable *atr*, cuyo tercer carácter se le asigna a la variable *ca*. Observe los subíndices del arreglo *rete*: el primero corresponde al número de la actividad y el segundo coincide con la retroalimentación que viene de la base de datos.

FIGURA 68. Creación de las partes de una actividad (11 de 12)

```

function acti2(rete)
{
    $(".txt").click(function(){
        var atr = $(this).attr("id");
        var ca =atr [3];
        alert ("Retroalimentación: "+rete[nAct][ca] );
    });
} //acti1

```

En la figura 69 se muestra la parte final de la función *aja3* que está constituida por dos botones que permiten la navegación entre las actividades. El botón 1 devuelve y el botón 2 avanza. Si este último llega a la última actividad, muestra la primera de ellas. La función *puntos ()* carga las actividades.

FIGURA 69. Creación de las partes de una actividad (12 de 12)

```

$("#btn1").click(function(){
    nAct--;
    if(nAct <0)
    {
        nAct=0;
    }
    puntos();
});
$("#btn2").click(function(){
    nAct++;
    if (nAct>(mAct-1))
    {
        nAct=0;
    }
    puntos ();
});
} //data
} //aja3

```

Las figuras 70, 71, 72 y 73 muestran la función *actividad2.php* que lee la base de datos. Se crea la clase actividad que contiene la función miembro *bdDatos ()*. Se crean varios arreglos donde se aloja la información que viene de la base de datos. Cada arreglo corresponde a una parte de la actividad. Con cada una de estas partes se forma una cadena unida con el doble asterisco (**). Cuando se tienen varias actividades, estas se separan con barras dobles (//).

FIGURA 70. Función actividad2.php (1 de 4)

```

function bdDatos()
{
    $servername = "localhost";
    $username = "root";
    $password = ""; //Escribir para leer la base de datos
    $dbname = "estadistica";
    $resu = "";
    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    $str = filter_input(INPUT_POST, "cod");
    //$str=2;
    $maxct=0;
    $maxia=0;
    //$sql= filter_input(INPUT_POST, "cadena");
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "SELECT * FROM `actividad`,opcion WHERE actividad.Id_actividad = opcion.Id_
    actividad and actividad.Id_tema = ".$str;
    //$sql = "SELECT * FROM `actividad`,opcion WHERE actividad.Id_actividad = ".$str." and
    opcion.Id_actividad = ".$str." and actividad.Id_tema =1";
    $result = $conn->query($sql);

```

FIGURA 71. Función actividad2.php (2 de 4)

```

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $ct = $row["ld_opcion"];
        $ia = $row["tipo_act"]; //número de la actividad dentro del tema.
        if($ct > $maxct)
        {
            $maxct = $ct;

```



```

    }

    if($ia>$maxia)
    {
        $maxia =$ia;
    }
    $this->activy [$ia]=$row["Titulo"];
    $this->conte [$ia]=$row["Contenido_act"];
    $this->opciones[$ia][$ct]=$row["Contenido"];
    $this->claves[$ia][$ct]=$row["Clave"];
    $this->retro[$ia][$ct]=$row["Retro"];
    $this->domes[$ia][$ct]=$row["dom"];
    $this->tipoOp[$ia][$ct]=$row["Tipo_opcion"];
    /*1: Opción con respuesta y retroalimentación. 2. El programa realiza el
    ejercicio*/
    $ct ++; }
} else {
    echo "En construcción";
}
$conn->close();

```

FIGURA 72. Función actividad2.php (3 de 4)

```

for($i=1;$i<=$maxia;$i++)
{
    $resu = $resu.$this->activy[$i]."/"; //Título del punto
}
$resu = $resu."***";
for($i=1;$i<=$maxia;$i++)
{
    $resu = $resu.$this->conte[$i]."/"; // Contenido del punto
}
$resu = $resu."***";
for($j=1;$j<=$maxia;$j++) // $maxia
{
    for($i=1;$i<=$maxct;$i++)// $maxct

```

```

{
    $resu = $resu.$this->opciones[$j][$i]."/";
}
}
$resu = $resu."**";
for($j=1;$j<=$maxia;$j++)
{
    for($i=1;$i<= $maxct;$i++)
    {
        $resu = $resu. $this->retro[$j][$i]."/";
    }
}

```

FIGURA 73. Función actividad2.php (4 de 4)

```

$resu = $resu."**";
for($j=1;$j<=$maxia;$j++) {
    for($i=1;$i<=$maxct;$i++) {
        $resu = $resu.$this->domes[$j][$i]."/";
    }
}
$resu = $resu."**";
for($j=1;$j<=$maxia;$j++) {
    for($i=1;$i<=$maxct;$i++) {
        $resu = $resu.$this->claves[$j][$i]."/";
    }
}
$coma=",";
$resu = $resu."**[";
for($j=1;$j<=$maxia;$j++) {
    for($i=1;$i<=$maxct;$i++)
    {
        if($i==($maxct-1))
        {
            $coma="";
        }
    }
}

```

```

        $resu = $resu.$this->tipoOp[$j][$i].$coma;
    }
}
$resu = $resu."]";
    echo $resu;
}
} //clase
$tem = new actividad;
$tem->bdDatos();

```

Finalmente, las figuras 74 hasta la 78 contienen a la función *aja4* que tiene como objetivo manejar las evaluaciones. Es un programa parecido a *aja3*, pero se presenta para tener la parte de programación del *software* completa para aquellos lectores que quieran diseñar su propio objeto de aprendizaje. Lo único que deben hacer es cambiar el contenido de la base de datos y los títulos de los menús.

FIGURA 74. Función *aja4*, lectura de datos (1 de 5)

```

function aja4(str,cade)
{
    var resulta =new Array();
    var nText =0;
    $.post("php/actividad.php",
    {
        cod: cade
    },
    function(data,status){
        var d= data;
        var conte =d.split("***");
        /*
        * 0 Título de la actividad. 1. Tipo de actividad. 2. Título del punto de la actividad.3.
enunciado
        * de la actividad. 4. Contenido de la opción. 5. Retroalimentación. 6. domes.7.
Claves*/
        var nAct=0; //Número del punto.
        var mpun=0;
        var ca=0;

```

```

var nota =[];
var cantPT;
var tx=0;
var nn=0;//Número de la nota
puntos();
function puntos()
{ var titPunto =conte[2].split("//");
  var enunPunto = conte[3].split("//");
  var contPunto =conte[4].split("//");
  var contRetro=conte[5].split("//");
  var contDomes =conte[6].split("//");
  var contClaves =conte[7].split("//");

```

FIGURA 75. Función aja4 (2 de 5)

```

cantPT= contPunto.length-1; //Cantidad de opciones totales
var cantEn =enunPunto.length-1;// Cantidad de enunciados
mpun = cantEn;
var cantOp = cantPT/cantEn; // cantidad de opciones por punto
var maxFil = parseInt(cantEn);
var maxCol = parseInt(cantOp);
var op = new Array (maxFil);//op: Opciones. maxFil: Cantidad de puntos
for(var i=0; i<maxFil;i++)
{
  op[i]=new Array (maxCol);//maxCol: cantidad de opciones
}
var j=0;
var k=0;
for(var i=0; i<cantPT;i++)
{
  op[k][j]= contPunto[i] ;
  j++;
  if(j>3)
  {
    j=0;
    k++;

```

```

    }
  }
  /*******Objetos *****
  var ob = new Array (maxFil);//Cantidad de puntos
  for(var i=0; i<maxFil;i++)    {
    ob[i]=new Array (maxCol); //ob: Objetos dom (cajas de textos o botones de radio)
  }

```

De igual manera que *aja3*, *aja4* llama al programa *actividad.php* para traer los datos de la base de datos; recibe el código del tema y envía la información a la siguiente función, la cual es asignada a la variable *d*, que envía los valores al arreglo *conte*; se declaran algunas variables para posterior manejo de los datos y se llama a la función *punto ()*, que asigna estos datos a cada parte de la evaluación.

FIGURA 76. Función *aja4* (3 de 5)

```

j=0;
k=0;
for(var i=0; i<cantPT;i++)
{
  ob[k][j]= contDomes[i] ;
  j++;
  if(j>3)
  {
    j=0;
    k++;
  }
}
/*******Claves*****
var cla= new Array (maxFil);//Cantidad de claves
for(var i=0; i<maxFil;i++)
{
  cla[i]=new Array (maxCol);//cantidad de opciones
}

```

```

j=0;
k=0;
for(var i=0; i<cantPT;i++)
{
    cla[k][j]= contClaves[i] ;
    j++;
    if(j>3)
    {
        j=0;
        k++;
    }
}

```

Las figuras 77 y 78 continúan mostrando la función *punto ()*; se crean las opciones, los objetos que pueden ser cajas de textos o botones de radio y los mensajes de retroalimentación.

En la figura 78 se agregan los títulos, los enunciados y las opciones de la evaluación al *div* contenido. Se crea una tabla para un punto que contiene dos columnas.

FIGURA 77. Función *aja4* (4 de 5)

```

var ret = new Array (maxFil);//Cantidad de claves
for(var i=0; i<maxFil;i++) {
    ret[i]=new Array (maxCol);//cantidad de opciones
}
j=0;
k=0;
for(var i=0; i<cantPT;i++)
{
    ret[k][j]= contRetro[i] ;
    j++;
    if(j>3)
    {
        j=0;

```

```

        k++;
    }
}
$("#contenido").empty();        $("#contenido").append("<b>"+conte[0]+"</b><br><br>"+titPunto[nAct]+"<br><br>"+enunPunto[nAct]+"<br><br>");
var tind;
var der,izq;
$("#contenido").append("<table>");
for(var i=0; i<cantOp ;i++)      {
    tind = op[nAct][i].indexOf("lder");
    if(tind !==-1)              {
        izq = op[nAct][i].substring(0,tind-1);
        der = op[nAct][i].substring(tind+4);
    }
}

```

La figura 78 llama a *acti1* o *acti2* según el tipo de objeto que se maneje.

FIGURA 78. Función *aja4* (5 de 5)

```

        acti1(cla,ret);
    }
} //puntos
else
{
    izq = op[nAct][i];
    der = "";
}
$("#contenido").append("<tr><td>"+izq+ob[nAct][i]+"</td><td>"+der+"</td></tr>");
} //for
$("#contenido").append("</table>");
$(".texto").css({"width":20});
$("#contenido").
css({"position":"absolute","top":"7em","left":"21em","width":"48em"});
var ind2 = ob[nAct][0].indexOf("rad");
if(ind2 !==-1)
{

```

```

        acti2(cla,ret);
    }
    else
    {

```

La figura 79 muestra los botones de navegación a través de los puntos de la prueba. También aparece el botón resultado que contendrá la evaluación cuantitativa y cualitativa de la prueba.

FIGURA 79. Botones de navegación para aja4

```

$("#navegacion").empty();
    $("#navegacion").append("<br><br>Actividad<input type='button' id='btn1' value
='Anterior'>");
    $("#navegacion").append("<input type='button' id='btn2' value='siguiente'>");

    $("#navegacion").append("<input type='button' id='res' value='Resultados'>");

```

En la figura 80 aparecen los arreglos que guardan el desempeño del usuario, en el caso de que el objeto sea una caja de texto.

FIGURA 80. Función acti1 miembro de aja4

```

function acti1(cla,ret)
{
    $(".txt").keypress(function(e){
        var atr = $(this).attr("id");
        var nom = $(this).attr("name");
        ca =atr[3];
        var user;
        var prog;
        var nT;
        if(e.keyCode ===13)
        {
            tx=$("#"+atr).val();

```



```

        nota[nom-1]=0;
        alert("Retroalimentación: Error: "+resulta[nom-1]+ " Nota: "+nota[ nText]+
nText: "+nText} ;
    }
    nText++; //Índice de los resultados
    nn++;
    };//click
};//acti2

```

La figura 82 muestra la función que contiene los resultados de la prueba. Los resultados se dan en forma cualitativa y cuantitativa. Estos resultados se pueden guardar en la base de datos para que el usuario los consulte y pueda fortalecer más tarde de forma autónoma los conceptos que no fueron bien aprendidos.

FIGURA 82. Resultados de la prueba

```

        $("#btn1").click(function(){
nAct--;
if (nAct<0)
{
    nAct=0;
}
puntos();
});
$("#btn2").click(function(){
nAct++;
if (nAct>{mpun- 1})
{
    nAct=0;
}
puntos();
});
$("#res").click(function(){
for(var k=0;k<resulta.length;k++)
{

```

```

        $("#navegacion").append("<br><br>"+resulta[k]+" nota:" +nota[k]+"<br>+");
    }
    });
};//data
};//aja4
};//menu

```

En las figuras 83 a la 85 aparece la función *evaluacion.php* que trae los datos de la base de datos. Se crean dos arreglos y la función *bdDatos*. En la variable *\$str* se recibe la identificación del tema al cual pertenece la evaluación. Se declara el objeto *\$tem* para llamar a la clase; lo demás es similar al caso anterior.

FIGURA 83. Función *evaluacion.php* para la evaluación (1 de 4)

```

<?php
header('Content-Type: text/html; charset=ISO-8859-1');
class actividad {
    //put your code here
    var $activy = array(
        "id_act"=>"",
        "tit_act"=>"",
        "cont_act"=>""
    );
    var $puntos = array(
        "id_pun"=>"",
        "tit_pun"=>"",
        "cont_pun"=>""
    );
function bdDatos()
{
    $servername = "localhost";
    $username = "root";
    $password = ""; //Escribir aquí contraseña
    $dbname = "estadistica";
    $resu = "";
    // Create connection

```

```

$conn = new mysqli($servername,$username,$password,$dbname);
// Check connection
$str = filter_input(INPUT_POST,"cod");
$maxcp=0;
$maxct=0;

```

FIGURA 84. Función *evaluacion.php* para la evaluación (2 de 4)

```

    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "SELECT * FROM `opcion_eva`, punto_eva, evaluacion, tema WHERE tema.id_
    tema = ".$str." and punto_eva.id_puntoEva= opcion_eva.id_puntoEva";
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        // output data of each row
        while($row = $result->fetch_assoc()) {
            $cp = $row["Id_puntoEva"];
            if($cp > $maxcp)
            {
                $maxcp = $cp;
            }
            $ct = $row["Id_opcionEva"];
            if($ct > $maxct)
            {
                $maxct = $cp;
            }
            $maxct = $ct;
            $this->activy ["id_act"] = $row["Id_evaluacion"];
            $this->activy ["tit_act"] = $row["Titulo_evaluacion"];
            $this->activy ["cont_act"] = $row["tipo_actApre"];
            $this->puntos[$cp] ["tit_pun"] = $row["Titulo_punto"];
            $this->puntos[$cp] ["cont_pun"] = $row["Contenido_puntoEva"];
            $this->opciones[$cp][$ct] = $row["Contenido_opcionEva"];

```

FIGURA 85. *Actividad.php para la evaluación (3 de 4)*

```

$this->domes[$cp][$ct]=$row["dom"];
    $this->claves[$cp][$ct]=$row["Clave_opcionEva"];
    $this->retro[$cp][$ct]["cont_retro"]=$row["Retro_opcionEva"];

    $ct++;
}
} else {
    echo "0 results";
}
$conn->close();
$resu = $this->activy ["tit_act"]."***".$this->activy ["cont_act"]."***"; //Título de la
evaluación
for($i=1;$i<=$maxcp;$i++)
{
    $resu = $resu.$this->puntos[$i]["tit_pun"]."//"; //Título del punto
}
$resu = $resu."***";
for($i=1;$i<=$maxcp;$i++)
{
    $resu = $resu.$this->puntos[$i]["cont_pun"]."//"; // Contenido del punto
}
$resu = $resu."***";
for($j=1;$j<=$maxcp;$j++)
{
    for($i=1;$i<=$maxct;$i++)
    {
        $resu = $resu.$this->opciones[$j][$i]."//";
    }
}
}

```

FIGURA 86. *Actividad.php para la evaluación (4 de 4)*

```

$resu = $resu."***";
for($j=1;$j<=$maxcp;$j++)

```

```

{
  for($i=1;$i<= $maxct;$i++)
  {
    $resu = $resu.$this->retro[$j][$i]["cont_retro"]."//";
  }
}
$resu = $resu."**";
for($j=1;$j<=$maxcp;$j++)
{
  for($i=1;$i<=$maxct;$i++)
  {
    $resu = $resu.$this->domes[$j][$i]."//";
  }
}
$resu = $resu."**";
for($j=1;$j<=$maxcp;$j++)
{
  for($i=1;$i<=$maxct;$i++)
  {
    $resu = $resu.$this->claves[$j][$i]."//";
  }
}
$resu = $resu."**";
  echo $resu;
}
} //clase
$tem = new actividad;
$tem->bdDatos();

```



Capítulo 2

Variables del diseño instruccional en la adquisición y aplicación de conceptos

Este capítulo tiene como finalidad mostrar el desarrollo y los resultados de la investigación titulada *Influencia del uso de las variables del diseño instruccional en la adquisición y aplicación de conceptos*. Expondremos todo su desarrollo metodológico de manera que este sirva de guía para aquellos que comienzan a incursionar en el tema.

El estudio en mención trató el problema del impacto de la teoría de diseño instruccional en el aprendizaje de conceptos de álgebra lineal, su aplicación en la resolución de problemas matriciales y la formación de nuevos conceptos, poniendo en práctica los conceptos aprendidos. Para evaluar el impacto de la teoría instruccional, se realizó una comparación entre dos estrategias instruccionales: una empleó un objeto de aprendizaje modelado a partir de la teoría de formación de conceptos y otra usó la clase tradicional. Para tal fin, se conformaron inicialmente dos grupos de estudiantes, cada uno en una estrategia instruccional diferente. A su vez, cada grupo se subdividió en dos nuevos grupos, según el rendimiento académico mostrado históricamente: un grupo de alto rendimiento y uno de bajo rendimiento.

El análisis se realiza a partir de los planteamientos de Gronen y Kersten (2001), quienes manifiestan que los sistemas tutoriales inteligentes buscan

emular las funciones que tiene un profesor. Los resultados mostraron que no hubo diferencia entre las estrategias instruccionales; es decir, el desempeño que mostraron los estudiantes con el objeto virtual de aprendizaje fue el mismo que tuvieron los que participaron en la clase tradicional. El uso del objeto de aprendizaje puede ser una alternativa para los desarrollos de los temas, sobre todo en los cursos virtuales, en donde no hay presencia de un docente. También se puede utilizar como una herramienta de ayuda durante las clases para recuperar clases que por cualquier razón no se pudieron llevar a cabo presencialmente y como medio para que el estudiante repase temas vistos. La diferencia de desempeño entre los estudiantes de alto y bajo rendimiento se mantuvo, así que el objeto de aprendizaje no logró el acercamiento en desempeño entre estos dos grupos. Las estrategias instruccionales probablemente ejercen un efecto significativo en los estudiantes de alto rendimiento en comparación con los de bajo rendimiento.

2.1 Planteamiento del problema

En la investigación se abordó el problema que enfrentan los estudiantes de no poder transferir conceptos trabajados en el aula de clase, es decir, aplicarlos a la solución de problemas que dependen de su dominio o al desarrollo de otros conceptos, incurriendo en lo que Perkins (2002) denomina *conocimiento frágil, pobre o inerte*. La fragilidad del conocimiento se manifiesta en el olvido del concepto o en la incapacidad del estudiante de aplicarlo para resolver una situación problemática en donde su uso es pertinente. En el conocimiento pobre los estudiantes no saben pensar valiéndose de lo que saben, mientras que en el conocimiento inerte no conectan los conocimientos que tienen con la situación que están enfrentando.

Como es una situación que atañe al aprendizaje y al uso de los conceptos, el fenómeno se estudió desde la teoría del *aprendizaje de conceptos*. Para esta, un concepto es una representación mental de una categoría que les facilita a las personas clasificar los objetos dentro de esa categoría, si cumplen con una

regla. Los conceptos se adquieren mediante la abstracción de información de los objetos y se reconocen mediante una actividad mental denominada *categorización* (Gulmans, 1998). De acuerdo con Morrison (2008), las investigaciones en el aprendizaje de conceptos han conducido al diseño de estrategias instruccionales que se centran en: a) definir y presentar los atributos de los conceptos; b) crear y presentar instancias de ejemplos y contraejemplos del concepto; y c) apoyar la práctica guiada de los estudiantes en el aislamiento de atributos, la discriminación de instancias, la generalización y el uso del concepto. La aplicación de estas estrategias podría influenciar el aprendizaje y el uso de los conceptos.

Para la adquisición de conceptos, Tennyson y Cocchiarella (1986) proponen una estrategia basada en cuatro variables de diseño instruccional:

1. *Definición*: una descripción que señala la estructura de los atributos críticos.
2. *Instancias expositoras*: ejemplos y contraejemplos que organizan y presentan sistemáticamente el contenido de la información en un formato declarativo.
3. *Instancias interrogativas*: ejemplos y contraejemplos que presentan el contenido de la información en un formato de preguntas.
4. *Elaboración de atributos*: análisis de los atributos críticos en instancias expositoras y retroalimentación sobre los atributos críticos en las instancias interrogativas.

Así las cosas, la investigación se centró en el efecto que tiene un objeto de aprendizaje, construido a partir de la teoría de la formación de conceptos, sobre estudiantes de bajo y alto rendimiento, en el aprendizaje de conceptos y su aplicación al desarrollo de nuevos. Por lo tanto, se buscó contestar a la siguiente pregunta: ¿qué efecto tiene un objeto de aprendizaje construido a partir de la estrategia de diseño instruccional en el aprendizaje del concepto de matriz y su aplicación para el desarrollo de nuevos conceptos, en estudiantes de bajo y alto rendimiento de primer semestre de ingeniería?

Para contestar dicha pregunta se recurrió a las siguientes subpreguntas:

¿Qué efecto tiene la implementación de un objeto virtual de aprendizaje basado en la estrategia de diseño instruccional en el aprendizaje de conceptos de álgebra lineal en un grupo de estudiantes de ingeniería de bajo y alto rendimiento?

¿Qué efecto tiene la implementación de un objeto virtual de aprendizaje basado en la estrategia de diseño instruccional en la transferencia o en el desarrollo de nuevos conceptos de álgebra lineal en un grupo de estudiantes de ingeniería de bajo y alto rendimiento?

¿Qué características de desempeño en el aprendizaje de conceptos y transferencia o desarrollo de nuevos conceptos de álgebra lineal presenta el grupo de estudiantes de ingeniería de bajo y alto rendimiento expuestos al objeto de aprendizaje de diseño instruccional, respecto al grupo expuesto a la metodología de enseñanza tradicional?

2.2 Justificación

Con el desarrollo de la universidad virtual se necesita garantizar que los medios utilizados para el proceso de enseñanza-aprendizaje al menos igualen la formación recibida en la formación tradicional. La utilización de simples textos en la red equivale al uso de libros, lo cual en la educación regular es acompañado por otras ayudas didácticas, entre ellas, la orientación de un profesor. Por otro lado, la universidad virtual es un medio masivo que busca ofrecer un ambiente de aprendizaje las 24 horas del día; no obstante, no existe un profesor como tal. Ahora, se puede recurrir a las tecnologías existentes, de manera que se emule el comportamiento del profesor, analizando errores, dando retroalimentación y generando tareas que concuerden con el estado de conocimientos del estudiante. Lo anterior, junto con las teorías del aprendizaje, puede generar el medio de enseñanza-aprendizaje adecuado para lograr tales fines.

2.3 Hipótesis

En la investigación se buscó verificar las siguientes hipótesis de investigación:

H1: los estudiantes sometidos a la influencia del objeto de aprendizaje (caracterizado por el manejo de la estrategia de variables de diseño instruccional) tienen un desempeño significativo en la formación de conceptos y aplicación al desarrollo de nuevos conceptos, frente a aquellos que están en la clase tradicional.

H2: los estudiantes que se encuentran en el grupo de alto rendimiento académico en ciencias básicas muestran un desempeño significativo en la formación de conceptos y la aplicación al desarrollo de nuevos conceptos frente a aquellos que están en el grupo de bajo rendimiento.

2.4 Marco teórico

Nuestra presentación girará principalmente alrededor de la teoría de la formación de conceptos, pero también incluiremos a las teorías de la retroalimentación y de la carga cognitiva, las cuales hacen parte de todos los objetos de aprendizaje que se emplearon como herramienta experimental en la investigación. La manera como se manejen estos dos últimos temas influirá en el aprendizaje a través de la formación de conceptos.

2.4.1 Aprendizaje de conceptos

Markle y Tiemann (1970, citados por Morrison, 2008) definen *concepto* como una categoría que agrupa objetos, eventos, símbolos o relaciones que comparten características o propiedades (estas últimas llamadas atributos). Los atributos describen las dimensiones que diferencian los objetos y los eventos. La membresía a una categoría se basa en los atributos físicos percibidos o por la definición.

Por otro lado, Klausmeier y Feldman (1975) consideran que un concepto se ha aprendido cuando el estudiante puede discriminar entre los atributos del concepto y evaluar nuevos ejemplos basados en la membresía de su categoría.

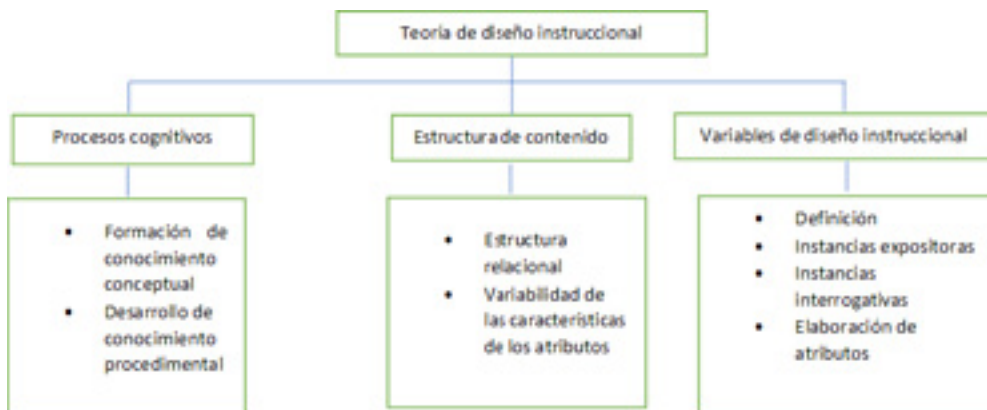
Por su parte, para Gulmans (1998) un concepto se define por uno o más atributos relacionados con una regla. La estructura mental mediante la cual un sujeto representa una categoría es a lo que este autor llama *concepto*. De esta manera, este último es una representación mental de una categoría, la cual 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 reglas de combinación, su aprendizaje requiere del descubrimiento de los atributos y de la regla.

Gulmans (1998) describe dos teorías sobre cómo se forman los conceptos: la teoría de la *formación de hipótesis* y la teoría del *prototipo*. En la teoría de formación de conceptos este autor cita a Trabasso y Bower (1968), quienes manifiestan que un concepto presuntivo se adopta y se mantiene hasta que se llega a la evidencia de otra alternativa. El descubrimiento de atributos relevantes se describe como un proceso de prueba de hipótesis. La presunción de la prueba de hipótesis es una parte fundamental del aprendizaje e implica que el aprendizaje de conceptos es un proceso constructivo, en el cual el sujeto genera hipótesis candidatas. Además, enfatiza el papel de la retroalimentación en el aprendizaje de los conceptos, al punto que considera que sin retroalimentación no es posible el aprendizaje.

Por su parte, la teoría del prototipo es planteada por Rosch y Mervis (1975, citados por Gulmans, 1998). Según estos autores, los conceptos se organizan alrededor del mejor ejemplo (el prototipo), el cual se concibe como un miembro promedio de una categoría. El prototipo contiene los atributos que ocurren con más frecuencia en los miembros de la categoría. De esta forma, un prototipo es la abstracción idealizada de los miembros de una categoría

Por su parte, Tennyson y Cocchiarella (1986) presentan su teoría del *diseño instruccional* para el aprendizaje de conceptos. Las tres partes de la teoría se presentan en la figura 87.

FIGURA 87. Partes de la teoría de diseño instruccional



El objetivo de la teoría es la formación de conocimiento conceptual y el desarrollo de conocimiento procedimental, a través del uso de las variables del diseño instruccional; pero, para tal fin, se debe tener en cuenta la estructura de contenido de los conceptos.

El conocimiento conceptual según estos autores (Tennyson y Cocchiarella, 1986) se refiere al entendimiento del concepto por parte del estudiante e implica la formación de atributos y la conexión apropiada con otros conceptos. El conocimiento conceptual se forma en la memoria por medio el almacenamiento integrado de dimensiones significativas (seleccionadas de ejemplos conocidos) y conectando estas entidades a un dominio dado de información.

El desarrollo del conocimiento procedimental se refiere al desarrollo de habilidades de clasificación, generalización y discriminación entre instancias nuevas que se encuentran en conceptos asociados. El conocimiento procedimental se desarrolla usando el conocimiento conceptual para resolver problemas de un dominio específico. Estos dos procesos interactúan de tal manera que el uso de conocimiento procedimental en la solución de problemas elabora el conocimiento conceptual.

La estructura de contenido tiene dos tipos de relaciones básicas: las sucesivas y las coordinadas; estas relaciones se definen con referencia al proceso cognitivo de clasificación. Una relación sucesiva es una relación vertical entre

los conceptos. Hay un concepto básico que tiene conceptos superordinados y otros subordinados. Por su parte, con las relaciones sucesivas, el aprendizaje se limita al desarrollo de la generalización dentro de la clase de un concepto. Con las relaciones coordinadas, el aprendizaje incluye el desarrollo de la generalización y la discriminación entre conceptos. Con las relaciones coordinadas se desarrolla el conocimiento procedimental.

La variabilidad de las características significa que estas pueden ser estables o variables. Los conceptos que tienen atributos relevantes estables a través de los contextos se llaman conceptos de dimensión constante, mientras que cuando varían a través de los contextos se las denomina conceptos de dimensión variable.

La interacción entre la estructura relacional y la variabilidad de las características de los atributos determina la selección de las variables del diseño instruccional para estrategias de enseñanza específicas.

En cuanto a las variables de diseño instruccional, la definición está asociada al proceso de hacer conexiones en la memoria, con conocimientos que son prerrequisitos. Así, una definición se entiende porque se conocen los conceptos que la definen. En las instancias expositivas se direcciona al estudiante para que compare y contraste los ejemplos expositivos; de esta manera, se codifica un conocimiento conceptual más completo y elaborado. Las instancias interrogativas fortalecen el desarrollo de conocimiento procedimental; los estudiantes se direccionan hacia el uso del mejor ejemplo cuando se intenta resolver problemas de clasificación.

Tennyson y Cocchiarella (1986) afirman que la estructura de la información es la base para el diseño de la estrategia instruccional. En la figura 88 se observan las posibles combinaciones que se pueden obtener de las estructuras.

FIGURA 88. Estructura de información de un concepto



Los conceptos sucesivos constantes requieren de mínima instrucción; se puede dar suficiente práctica según la experiencia del aprendiz. El aprendizaje se centra en el desarrollo de la habilidad de generalización, de manera que se pueden utilizar instancias expositivas (consistentes en definiciones organizadas y claras) y ejemplos trabajados. No se necesitan instancias interrogativas y se pueden usar contraejemplos.

Los conceptos coordinados constantes requieren de su presentación en secuencias simultáneas. Como se debe discriminar entre conceptos, es conveniente implementar instancias interrogativas con retroalimentación sobre los atributos.

Los conceptos sucesivos variables se aprenden mejor cuando se presentan en contexto. Para ayudar al estudiante a entender la variabilidad de los atributos críticos se puede utilizar información estratégica en los ejemplos expositivos e interrogativos dados, de forma tal que se incremente la generalización a través de los contextos.

Los conceptos coordinados variables requieren de mucha instrucción para mejorar el aprendizaje; por tanto, son los conceptos más demandantes. Se debe tener en cuenta que se trata de discriminar entre conceptos en contextos variables.

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

Gulmans cita a Rosch (1978), quien afirma que para la categorización de objetos naturales se distingue el nivel básico, el superordinado y el subordinado. Un concepto subordinado tiene siempre los atributos de los conceptos superordinados y también otros atributos. De tal manera, hay una herencia de atributos de conceptos de niveles más altos por parte de los niveles más bajos. Los conceptos que están al mismo nivel se llaman *coordinados*.

De acuerdo con Tennyson y Cocchiarella (1986) los conceptos existen en la memoria como parte de estructuras de conocimientos más grandes y complejas. De acuerdo con Millward (1980), las estructuras de conocimiento cambian con las experiencias de los individuos.

2.4.2 *Componentes de la arquitectura cognitiva humana*

De acuerdo con Sweller (1994), la arquitectura cognitiva está compuesta por la memoria sensorial, la memoria de trabajo y la memoria a largo plazo. Cooper (1998) define la memoria sensorial como aquella que trata los estímulos de los sentidos. Estos son la vista, el oído, el olfato, el gusto y el tacto. Cada sentido tiene una partición de la memoria sensorial por separado. La memoria sensorial se extingue rápidamente.

La memoria a largo plazo está definida por Cooper (1998) como el inmenso cuerpo de conocimientos y de habilidades que se guardan en formas de esquemas, de manera permanente y que se pueden recordar y aplicar en un tiempo posterior cuando se necesite. Se cree que la memoria de trabajo es la parte de la mente que provee la conciencia. Es el vehículo que permite pensar, resolver problemas y ser expresivos. Su limitación más grande es la capacidad de tratar solamente cerca de siete elementos a la vez. Se puede expandir un poco mezclando los sentidos que se utilizan para presentar la información.

Existen diferencias individuales en la naturaleza y detalles de los esquemas poseídos. Las personas especialistas en un dominio particular poseen esquemas más extensos y detallados que otras personas. De hecho, la diferencia

entre expertos y novatos se da por la expansión de los esquemas y el alto nivel de automatización.

El aprendizaje de algún concepto o de alguna información depende de la interactividad de los elementos. Cuando la interactividad entre los elementos es baja se hace fácil la comprensión. Los elementos aislados se comprenden fácilmente. Pero cuando existe una alta interacción entre los elementos que se aprenden se dificulta más la comprensión porque hay que atender a los elementos y a la relación entre estos. La interactividad entre elementos se define como el grado en el cual la información se puede o no entender aisladamente (Cooper, 1998).

La comprensión de los elementos tiene que ver con la *carga cognitiva*, la cual se define como la cantidad de actividad mental impuesta sobre la memoria de trabajo en un instante de tiempo. Si la información no es atendida y procesada por la memoria de trabajo, no se produce el aprendizaje.

Como presentamos previamente, la diferencia entre un experto y un novato se basa en la extensión de los esquemas. Lo que puede ser un esquema para un experto, puede constituir varios esquemas para un novato. Cooper (1998) da como ejemplo algo parecido a las siguientes frases:

El perro persigue al gato.

Pl oqppesq laue orro go.

En la primera frase cada palabra puede ser un esquema, esto es, se maneja como un elemento. En la segunda frase cada letra es un elemento que debe ser atendido por la memoria de trabajo, lo que hace más difícil su comprensión. Además, las personas tienen esquemas formados sobre el perro y el gato. Para un niño que esté aprendiendo a leer, la primera frase se complica porque no son expertos en la lectura. Para un experto en lectura, toda la frase puede ser un esquema, por lo que le queda suficiente capacidad a la memoria de manejar otros elementos.

A la hora de hacer un diseño instruccional, se deben tener en cuenta los recursos mentales y la carga cognitiva que impone el material que se va a aprender (llamada *carga cognitiva intrínseca*) y la carga cognitiva que se le agrega al tema debido al material instruccional que se le presenta a los estudiantes (llamada *carga cognitiva extrínseca*).

La carga cognitiva intrínseca no se puede reducir, mientras que la carga cognitiva extrínseca se puede manejar, según se presente al estudiante. Lo importante es que, sumando los dos tipos de carga, no se superen los recursos mentales; en este caso, no se produciría el aprendizaje.

2.4.3 Interactividad

En términos generales, *interactividad* significa actuar mutuamente; esto es, para ejecutar actos recíprocos se requiere cooperación y coordinación por parte de los sujetos. A lo anterior debemos sumarle el poder y la influencia que ejerce uno sobre otro, así como la presencia de algún grado de negociación sobre lo que el otro hará, cuándo y cómo (Kirsch, 1997).

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 un rango de habilidades y entendimiento por parte del aprendiz, la importancia de un diseño instruccional riguroso y la aplicación de interfaces gráficas apropiadas.

Por otro lado, la multimedia —como tecnología educativa— se refiere al conjunto de medios (texto, audio, video), tecnologías (computador) y productos como la información, los juegos y programas educativos. Al integrar estos elementos podemos decir que existe interactividad.

La interactividad se puede enfocar 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

(1982, citado por Sims, 1997), las características que hacen interactivo a un *software* instruccional incluyen una serie de opciones: observar, encontrar, hacer, usar, construir y crear. Por su parte, también se puede analizar el planteamiento de Jonassen (1988 citado por Sims, 1997), según el cual la interactividad es descrita como una actividad entre dos organismos con una aplicación basada en el computador, comprometiendo al aprendiz en un diálogo verdadero. De tal manera, la calidad de la interacción dependerá de la naturaleza de la respuesta del aprendiz y de la retroalimentación por parte del computador. Si la respuesta es consistente con las necesidades de procesamiento de información del aprendiz, entonces la interactividad es significativa. Por ende, se la puede comprender como la función de entrada requerida por el aprendiz para responder al computador, el análisis de las respuestas y la naturaleza de la acción por parte del computador.

Schwier y Misanchuk (1993, citados por Sims, 1997) identificaron diferentes niveles de interactividad basados, a su vez, 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 aprendiz 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 como los usuarios pueden acceder, manipular y navegar a través del contenido material. A partir de estos se construyen los 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; estos son la *interactividad*:

- 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 activados 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 en donde el usuario se puede mover (hacia delante o hacia atrás) mediante una secuencia lineal predeterminada de material instruccional. Esta clase de interacción no facilita la retroalimentación a la acción del aprendiz; simplemente provee acceso a la siguiente o a la anterior pantalla en la secuencia.

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

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

La interactividad *de construcción* (elaboración proactiva) es una extensión para actualizar la interactividad y requiere de la creación de un ambiente instruccional en donde 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 en forma textual.

La interactividad *de simulación* extiende el papel del aprendiz a controlador u operador, en la cual las selecciones individuales determinan la secuencia del entrenamiento. Las interactividades de simulación y de construcción están muy relacionadas y pueden necesitar que el aprendiz complete una secuencia específica de tareas antes de que se produzca una actualización adecuada.

En la interactividad *de hipervínculo* (navegación proactiva) el aprendiz tiene acceso a una riqueza de información, y puede viajar por 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 estudiante puede trabajar en un contexto relacionado de trabajo significativo. A través de una serie de secuencia 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 aprendiz 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 en tres dimensiones: la primera se refiere a la interactividad de navegación en la que el usuario se mueve de un lado a otro, o instruccional, que 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.

2.5.4 Retroalimentación

De acuerdo con Mason y Bruning (2001) la retroalimentación es cualquier mensaje que se genera en respuesta a la acción de un aprendiz. Una de sus funciones es la de ayudar a identificar errores y hacer tomar conciencia sobre las concepciones equivocadas que se tienen. Según Kulhavy y Stock (1989), la retroalimentación le facilita al estudiante dos clases de información: verificación y elaboración. La verificación informa si la respuesta es correcta o no, mientras que la elaboración es el componente que brinda guías relevantes para que el estudiante encuentre la respuesta correcta. Para que suceda una retroalimentación exitosa se deben dar estas dos formas de retroalimentación.

La retroalimentación elaborada típicamente puede ser informacional, específica al tópico o específica a la respuesta. La elaboración informacional no provee información específica, sino que facilita un marco de información relevante del cual se puede deducir la respuesta correcta. La elaboración específica al tópico, contrariamente, provee información más específica sobre el tópico o la meta.

La elaboración específica a la respuesta está dirigida tanto a las respuestas correctas como a las incorrectas; si el estudiante selecciona una respuesta incorrecta, este tipo de retroalimentación explica por qué es correcta o incorrecta, y si es el último caso proporciona la información sobre cuál es la respuesta correcta (Whyte, Karolick, Neilsen, Elder, & Hawley, 1995; citados por Mason & Bruning, 2001).

Los niveles de retroalimentación de acuerdo con Mason y Bruning (2001) pueden ser:

- *No retroalimentación*: se le da la nota al estudiante sin hacer referencia a los elementos individuales. Este tipo de retroalimentación no contiene ni verificación, ni elaboración.

- *Conocimiento de la respuesta*: este tipo de retroalimentación le informa al estudiante si su respuesta es correcta o incorrecta. Este tipo de respuesta es esencial para la verificación, pero no provee información sobre los errores posibles.
- *Múltiples respuestas*: el estudiante responde hasta que se genera la respuesta correcta. Este tipo de retroalimentación provee verificación, pero no elaboración.
- *Conocimiento de la respuesta correcta*: facilita verificación sobre cada uno de los puntos y le da conocer al estudiante la respuesta correcta. No brinda elaboración más allá de la respuesta correcta.
- *Contingente al tópico*: provee verificación e información elaborada general sobre el tema. Cuando se produce la respuesta incorrecta, se redirige al estudiante al material en donde se encuentra la respuesta correcta; adicionalmente, se le da información sobre cómo encontrar la respuesta. Así, depende del estudiante encontrar la respuesta en el material instruccional.
- *Contingente a la respuesta*: provee tanto verificación como elaboración. Además de proveer el conocimiento de la respuesta correcta, facilita retroalimentación específica del porqué una respuesta incorrecta está equivocada y por qué la respuesta correcta lo es.
- *Relacionada con el error*: brinda verificación y maneja errores específicos. Se apoya en librerías o reglas para identificar los errores más comunes de los estudiantes. No facilita la respuesta correcta, pero ayuda al estudiante a detectar errores de procedimiento, para que él mismo haga la corrección.
- *Aislamiento de atributos*: provee verificación de elementos y enfatiza los atributos centrales del concepto. Se centra en los componentes claves para mejorar el entendimiento general del fenómeno.

2.6 Marco metodológico

La investigación expuesta en este capítulo fue de tipo explicativo cuasiexperimental, porque mediante el planteamiento de una relación causal entre las variables se pretendió determinar los posibles efectos de un objeto de aprendizaje sobre la formación de nuevos conceptos y en su aplicación en el desarrollo de otros. El procedimiento de aleatorización se empleó en gran parte del proceso investigativo, salvo en la selección de los participantes del estudio, por la limitación de asignar a los participantes en forma aleatoria al grupo experimental.

2.6.1 Diseño

Se aplicó un diseño factorial de 2 x 2 con dos tratamientos experimentales que constituyeron las variables activas y dos niveles de rendimiento que conformaron las variables atributivas. En la tabla 3 aparece el diseño factorial 2 x 2 para medir los efectos que las dos estrategias producen en la formación de conceptos y su aplicación al desarrollo de nuevos conceptos. Las dos estrategias son el *objeto de aprendizaje* y la *clase tradicional*; se manejó una variable atributiva con dos valores: rendimiento bajo y el rendimiento alto. El efecto de tratamiento experimental sobre la variable dependiente se evaluó sobre los dos niveles de la variable atributiva.

En este diseño algunos sujetos del nivel de bajo rendimiento recibieron el tratamiento del objeto de aprendizaje y otros el tratamiento de la clase tradicional. El rendimiento se midió con el desempeño general que mostraron los estudiantes en las asignaturas de las ciencias básicas. Con esta variable atributiva se pretendió controlar variables extrañas que pudieran alterar el estudio, tales como la cantidad de tiempo sin estudiar, la formación deficiente y otras no previstas, que se esperaba que se reflejaran en el rendimiento académico del área de ciencias básicas. Se consideraron estudiantes de bajo rendimiento aquellos que tuvieran en promedio las notas bajas en el área de ciencias básicas; y estudiantes de alto rendimiento a aquellos que tuvieran un promedio superior.

TABLA 3. Diseño factorial de 2 x 2

Variable atributiva	Estrategia de aprendizaje (variable experimental)		Media
	Objeto de aprendizaje	Clase tradicional	
Bajo rendimiento	X ₁	X ₂	
Alto rendimiento	X ₁	X ₂	
Media			

2.6.2 Participantes

Participaron de la investigación estudiantes de educación superior de primer semestre y que estaban matriculados en un curso básico de álgebra lineal (en el que verían el concepto de matriz). Estos estudiantes elegidos tenían un promedio de rendimiento menor o igual a 3.2 o mayor o igual a 3.8. Se tomaron dos cursos como grupos intactos, por lo que la investigación se clasifica dentro de los estudios cuasi-experimentales.

2.6.3 Instrumentos

Se emplearon un objeto virtual de aprendizaje, la clase tradicional y una prueba final (postest). El objeto virtual de aprendizaje fue modelado a partir de la teoría de conceptos. Contiene una conceptualización (el concepto de matriz), unas determinantes y la evaluación que aparece como aplicación. En cuanto a la clase tradicional, esta hace referencia al desarrollo de un curso a través de la exposición de un profesor, que ilustra los conceptos a través de ejemplos y ejercicios en clase. Después de haber aplicado las dos estrategias anteriores se realizó una prueba final para determinar, a través del desempeño en estas, si hubo diferencia entre los dos grupos.

A continuación realizaremos una descripción más detallada del objeto virtual de aprendizaje, comenzando por la contextualización en donde se introduce al estudiante en los objetivos, metodología y prerrequisitos del objeto.

El desarrollo del tema comienza con matrices, que se diseña siguiendo la teoría de formación de conceptos. El concepto de «matrices» contiene el concepto de «matriz» que se da en forma de definición y físicamente a través de un ejemplo (figura 89). Matriz es un concepto sucesivo del concepto de arreglo rectangular.

FIGURA 89. Concepto de matriz

Matriz

Definición.

Una matriz es un arreglo rectangular de símbolos en filas y columnas encerradas entre corchetes o paréntesis. En álgebra lineal, las matrices se representan con letras mayúsculas.

Ejemplo

$$A = \begin{pmatrix} -3 & 8 & -3 \\ -3 & -3 & -4 \\ 0 & -8 & 2 \end{pmatrix}$$

De acuerdo con Tennyson y Cocchiarella (1988), basta con mostrar instancias expositivas para entenderlo, pero el objeto de aprendizaje presenta también instancias interrogativas. En las instancias expositivas se muestran ejemplos y contraejemplos (figura 90) del concepto. La finalidad es que el estudiante logre discriminar entre conceptos que son coordinados.

FIGURA 90. Instancias expositivas

Ejemplos y contra ejemplos de matriz

Observe con atención las siguientes figuras, compárelas y lea atentamente porque unas son matrices y otras no



Esta figura que se observa no es una matriz porque tiene forma circular. Las matrices tienen forma rectangular.

$$\begin{array}{|c|c|c|} \hline 3 & 9 & 5 \\ \hline 2 & -1 & 8 \\ \hline 5 & -5 & 7 \\ \hline \end{array}$$

Esta figura tampoco representa una matriz porque los números están encerrados entre líneas verticales. Las matrices van encerradas entre paréntesis o corchetes.

$$\begin{pmatrix} 12 & -9 \\ -23 & 14 \\ 0 & 3 \end{pmatrix}$$

Esta figura es una matriz porque es un arreglo rectangular en filas y columnas y está encerrada entre paréntesis.

$$\begin{bmatrix} -45 & 7 & \pi \\ 12 & -6 & 8 \end{bmatrix}$$

Esta figura también es una matriz porque es un arreglo rectangular en filas y columnas está encerrada entre corchetes.

En la figura 91 se pueden observar instancias interrogativas del concepto. Como respuesta a la entrada del estudiante se presenta una retroalimentación elaborada, dándole a conocer si su respuesta es correcta o no, explicándole por qué está bien o mal.

FIGURA 91. *Instancias interrogativas*

Ejemplos interrogativos

Mire las siguientes figuras, la tarea consiste en determinar cuál de las figuras es una matriz. Siempre que vea un arreglo de símbolos, hágase las siguientes preguntas. Si todas las preguntas se contestan con sí, el objeto es una matriz.

	<p>¿Es un arreglo rectangular de filas y columnas? <input type="radio"/> Sí <input type="radio"/> No</p> <p>¿Está encerrado entre paréntesis o corchetes? <input type="radio"/> Sí <input type="radio"/> No</p> <p>¿Es una matriz? <input type="radio"/> Sí <input type="radio"/> No</p>
	<p>¿Es un arreglo rectangular de filas y columnas? <input type="radio"/> Sí <input type="radio"/> No</p> <p>¿Está encerrado entre paréntesis o corchetes? <input type="radio"/> Sí <input type="radio"/> No</p> <p>¿Es una matriz? <input type="radio"/> Sí <input type="radio"/> No</p>
	<p>¿Es un arreglo rectangular de filas y columnas? <input type="radio"/> Sí <input type="radio"/> No</p> <p>¿Está encerrado entre paréntesis o corchetes? <input type="radio"/> Sí <input type="radio"/> No</p> <p>¿Es una matriz? <input type="radio"/> Sí <input type="radio"/> No</p>
	<p>¿Es un arreglo rectangular de filas y columnas? <input type="radio"/> Sí <input type="radio"/> No</p> <p>¿Está encerrado entre paréntesis o corchetes? <input type="radio"/> Sí <input type="radio"/> No</p> <p>¿Es una matriz? <input type="radio"/> Sí <input type="radio"/> No</p>

El concepto de matriz involucra otros conceptos tales como fila y columna. Estos dos conceptos son coordinados; para su aprendizaje se muestran simultáneamente de manera que se logre la discriminación entre sus atributos. La figura 92 muestra que las filas son componentes horizontales y las columnas componentes verticales.

FIGURA 92. Concepto de fila y columna

Filas de una matriz

Las filas son los componentes horizontales de las matrices. Dé clic sobre el botón para que observe cómo se cuentan.

$$\begin{pmatrix} 5 & 3 & 8 \\ 8 & 4 & -6 \\ 0 & -6 & -2 \end{pmatrix}$$

Fila 2

Pulse para parar conteo

Columnas de una matriz

Las columnas son los componentes verticales de las matrices. Dé clic sobre el botón para que observe cómo se cuentan.

$$\begin{pmatrix} 7 & -8 & 0 & -5 \\ 0 & 0 & -8 & 5 \\ -7 & -2 & -7 & 0 \end{pmatrix}$$

Pulse para parar conteo

En el objeto de aprendizaje las filas y columnas se mueven para indicar cómo se cuentan. De acuerdo con Morrison (2008), un concepto se ha aprendido cuando el estudiante es capaz de discriminar entre los atributos de un concepto y evaluar nuevos ejemplos basados en la categoría del concepto. La figura 93 muestra una actividad de aprendizaje para evaluar precisamente si el estudiante, mediante nuevos ejemplos, puede identificar el concepto pertinente.

FIGURA 93. Identificación de filas y columnas

Actividad de aprendizaje 1.1 Identificación de filas y columnas

Marque la componente que está resaltada:

Fila
 Columna

$$\begin{pmatrix} -4 & 5 & -3 & -1 \\ 4 & 0 & -1 & 0 \\ -7 & 0 & 3 & 0 \end{pmatrix}$$

Otro ejercicio

Los conceptos de fila y columna se utilizan para construir el concepto de elemento. En la figura 94 se observa el concepto; la posición del elemento se indica mediante el movimiento de un cuadrado rojo. Debajo está la representación, una letra minúscula con subíndices que indican la fila y la columna de la ubicación y que van cambiando con la posición del elemento.

FIGURA 94. Concepto de elemento

Elementos de una matriz

Los elementos son cada uno de los componentes individuales de una matriz.

Se representan con letras minúsculas acompañadas de dos subíndices que indican su ubicación dentro de la matriz. El primero indica la fila y el segundo indica a columna.



En la figura 95 aparecen ejemplos y contraejemplos del concepto de elemento. La idea es que se pueda discriminar entre los conceptos para reconocer dicho concepto.

FIGURA 95. Ejemplos y contraejemplos de elementos

Observe con atención las siguientes figuras, compárelas y determine cuál de ellas son representaciones de los elementos.

A: Esta expresión no representa un elemento porque está en mayúscula y no tiene subíndices que indiquen su ubicación.

B $i j$: Esta expresión no representa un elemento porque está en mayúscula.

d_{ij} : Esta expresión es un elemento porque está en minúscula y está acompañada de dos subíndices.

b_{12} : Esta expresión representa un elemento porque está en minúscula y está acompañada de dos subíndices.

De la misma manera, aparecen instancias interrogativas del concepto con su respectiva retroalimentación elaborada. La figura 96 muestra un mensaje de retroalimentación como resultado de escoger una opción equivocada.

FIGURA 96. Ejemplos interrogativos de elemento

Mire las siguientes expresiones, la tarea consiste en determinar cuál de las expresiones corresponde a la representación de elemento. Siempre que vea una expresión del algebra lieneel hágase las siguientes preguntas. Si todas las preguntas se contestan todas con sí, la expresión representa un elemento.

a_{ijk}

¿El primer símbolo de la expresión está escrito en minúscula? Sí No
 ¿Está acompañado de subíndices? Sí No

B_{kj}

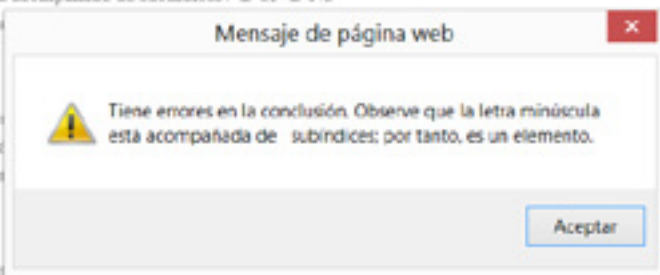
¿El p...
 ¿Est...
 ¿Repe...

d

¿El p...
 ¿Está acompañado de subíndices? Sí No
 ¿Representa un elemento? Sí No

f_{ik}

¿El primer símbolo de la expresión está escrito en minúscula? Sí No
 ¿Está acompañado de subíndices? Sí No
 ¿Representa un elemento? Sí No



Sobre este tema existen más ejercicios para que se forme el concepto. En la figura 97 aparece una actividad de aprendizaje sobre sumatoria. En esta actividad se está utilizando lo aprendido sobre el elemento. El concepto de elemento es clave para construir los conceptos siguientes.

FIGURA 97. Actividad de aprendizaje sobre sumatoria

Actividad de Aprendizaje 2.1

Haga clic sobre la opción apropiada. En la expresión:

$$\sum_{i=1}^n a_{ij} =$$

Se está sumando:

la fila La columna

En la parte inversa de una matriz se espera un dominio del lenguaje matemático que se trató de construir a lo largo del experimento. El programa termina con una evaluación que tiene dos partes: en la primera se evalúa la formación de conceptos y, en la otra, el desarrollo procedimental, en el que se espera que con los conocimientos adquiridos el estudiante pueda construir el concepto.

2.6.4 Variables de la investigación

La variable independiente la constituyeron las estrategias de aprendizaje con dos factores: el objeto de aprendizaje y la clase tradicional; estos se midieron teniendo en cuenta dos niveles de rendimiento: el nivel alto y el nivel bajo, que se operativizaron a través del desempeño mostrado en las pruebas realizadas. La variable dependiente la constituyó el desempeño mostrado en una prueba sobre la formación e interpretación de nuevos conceptos, contruidos a partir de las estrategias de aprendizajes dadas.

2.6.5 Procedimiento

Para comprobar la hipótesis se utilizó el análisis de varianza (ANOVA); esta es una razón (llamada F) que utiliza la varianza de las medias de los grupos como una medición de las diferencias observadas entre ellos.

El principio general de la ANOVA es que la varianza total de todos los sujetos en un experimento puede dividirse en dos fuentes: la varianza entre los grupos y la varianza dentro de los grupos. La varianza entre los grupos se incorpora al numerador en la razón F; por su parte, la varianza dentro de los grupos se incorpora al denominador. A medida que aumenta el primer tipo de varianza la razón aumenta. A medida que aumenta el segundo tipo de varianza la razón disminuye. Cuando el numerador y el denominador son iguales las diferencias entre las medias son debidas a la casualidad. Si el numerador es mayor que el denominador se consultará una tabla de valores de F para determinar si la razón es suficientemente grande para rechazar la hipótesis de nulidad al nivel establecido.

Para determinar la razón F se utilizaron las fórmulas presentadas en la tabla 4.

TABLA 4. Fórmulas de F

Nombre de la fórmula	Fórmulas
Suma total de los cuadrados	$\sum x_i^2 = \sum X^2 - (\sum X)^2 / N$
Suma de los cuadrados entre grupos	$\sum x_b^2 = (\sum x_1)^2 / n_1 + (\sum x_2)^2 / n_2 + \dots - (\sum X)^2 / N$
Suma de los cuadrados dentro de los grupos	$\sum x_w^2 = \sum x_1^2 - (\sum x_1)^2 / n_1 + \sum x_2^2 - (\sum x_2)^2 / n_2 + \dots$
Suma de los cuadrados entre columnas	$\sum x_{bc}^2 = (\sum X_{c1})^2 / n_{c1} + (\sum X_{c2})^2 / n_{c2} - (\sum X)^2 / N$
Suma de los cuadrados entre fila	$\sum x_{br}^2 = (\sum X_{r1})^2 / n_{r1} + (\sum X_{r2})^2 / n_{r2} + \dots - (\sum X)^2 / N$
Suma de la interacción de los cuadrados	$\sum x_{int}^2 = \sum X_b^2 - (\sum X_{bc}^2 + \sum X_{br}^2)$
Cuadrado de la media entre los grupos	$MS_b = \sum x_b^2 / df_b$
Cuadrado de la media dentro los grupos	$MS_w = \sum x_w^2 / df_w$
Razón F	MS_b / MS_w

La determinación de los grados de libertad aparece en la tabla 5.

TABLA 5. Determinación de grados de libertad

Nombre	Fórmula
<i>gdl para suma de los cuadrados entre columnas</i>	C-1
<i>gdl para suma de los cuadrados entre filas</i>	R-1
<i>gdl para la interacción</i>	(C-1) (R-1)

Nombre	Fórmula
<i>gdl para suma de los cuadrados entre grupos</i>	$(G-1)$
<i>gdl para suma de los cuadrados dentro de los grupos</i>	$\sum(n-1)$
<i>gdl para suma total de los grupos</i>	$N-1$

2.7 Procesamiento y análisis de la información

2.7.1 Población, muestra y procedimientos experimentales

La investigación se realizó con una muestra de 36 estudiantes de la Escuela de Ingeniería de la Corporación Universitaria Unitec, de la jornada nocturna. Esta muestra se dividió en dos grupos de acuerdo con el rendimiento académico. Los estudiantes con promedios de notas más bajos en las asignaturas de ciencias básicas se clasificaron en el grupo de rendimiento bajo, mientras que aquellos con los mayores promedios se clasificaron dentro de los estudiantes de rendimiento alto. Cada uno de estos grupos se subdividió a su vez aleatoriamente en dos grupos; el grupo sometido al objeto de aprendizaje caracterizado por estar diseñados con base en la teoría de la formación de conceptos y el grupo que siguió como estrategia metodológica la clase magistral tradicional.

2.7.2 Datos y resultados de la experimentación

En la tabla 6 aparecen las notas de acuerdo con la estrategia metodológica, con sus respectivos valores estadísticos.

TABLA 6. Clasificación según estrategia metodológica

0	1	2	3	4	5	6	7	8	9
1	Estrategia instruccional								
2			Objeto de aprendizaje				Clase tradicional		
3			1				2		
4			1.5				1		
5		Bajo	2				3		
6			2.5	Grupo 1			3	Grupo	3
7			2.8				2.2		
								$\sum X_{r1}$	43.4

	0	1	2	3	4	5	6	7	8	9	
1				Estrategia instruccional							
8					4.5			2.1			
9					3.5	$\bar{X} =$	2.91	1.3	$\bar{X} =$	1.91	$\bar{X}_{r1} =$ 2.4111
10					4.6			1.3			
11					3.8			1.3			
12											
13	Rendimiento			$\sum X$	26.2		$\sum X$	17.2			
14					5			4.3			
15					4.8			5.0			
16					5			4.4			
17					4.7	Grupo	2	5	Grupo	4	$\sum X_{r2} =$ 82.8
18			Alto		4.8			4.1			
19					3.9			4.8			
20					3.5	$\bar{X} =$	4.52	5	$\bar{X} =$	4.68	$\bar{X}_{r2} =$ 4.6
21					4.8			5			
22					4.2			4.5			
23											
24				$\sum X$	40.7		$\sum X$	42.1			
25				$\sum X_{c1} =$	66.9		$\sum X_{c1} =$	59.3		$\sum X_{total} =$	126.2
26				$\bar{X}_{c1} =$	3.717		$\sum X_{c2} =$	3.294		$\bar{X} =$	3.5056

El resumen de los resultados aparece en la tabla 7.

TABLA 7. Resumen de resultados para F

Fórmula	Resultado
$\sum x_t^2 = \sum X^2 - (\sum X)^2 / N$	$= 511.02 - 442.4011 = 68.6189$
$\sum x_b^2 = (\sum x_1)^2 / n_1 + (\sum x_2)^2 / n_2 + \dots - (\sum X)^2 / N$	$= 26.2^2 + 17.2^2 + 40.7^2 + 42.1^2 - 126.2^2 = 47.73$
$\sum x_w^2 = \sum x_t^2 - \sum x_b^2$	$68.6189 - 47.73 = 20.9$
$(\sum x_{bc}^2) / n_{c1} + (\sum x_{c2}^2) / n_{c2} - (\sum X)^2 / N$	$= 66.9^2 / 18 + 59.3^2 / 18 - 126.2^2 / 36 = 1.60444$
$\sum x_{br}^2 = (\sum x_{r1})^2 / n_{r1} + (\sum x_{r2})^2 / n_{r2} + \dots - (\sum X)^2 / N$	$= 43.4^2 / 18 + 82.8^2 / 18 - 126.2^2 / 36 = 43.1211$
$\sum x_{int}^2 = \sum x_b^2 - (\sum x_{bc}^2 + \sum x_{br}^2)$	$= 47.73 - (1.60444 + 43.1211) = 3.004$
$MS_b = \sum x_b^2 / df_b$	
$MS_w = \sum x_w^2 / df_w$	
Razón F	MS_b / MS_w

Tomando como base la tabla 5, en la tabla 8 se realiza el cálculo de los grados de libertad.

TABLA 8. Grados de libertad

Nombre	Fórmula
<i>gdl para suma de los cuadrados entre columnas</i>	$C-1 = 2-1 = 1$
<i>gdl para suma de los cuadrados entre filas</i>	$R-1 = 2-1 = 1$
<i>gdl para la interacción</i>	$(C-1)(R-1) = (1)(1) = 1$
<i>gdl para suma de los cuadrados entre grupos</i>	$(G-1) = 4-1 = 3$
<i>gdl para suma de los cuadrados dentro de los grupos</i>	$\sum(n-1) = 32$
<i>gdl para suma total de los grupos</i>	$N-1$

Los resultados de los cálculos basados en los datos de la tabla 5 aparecen en la tabla 7. Los valores de los cuadrados medios (MS) se encontraron dividiendo cada suma de los cuadrados entre sus correspondientes grados de libertad. Las razones F se hallaron dividiendo los cuadrados medios intergrupales entre el cuadrado medio dentro de los grupos. Para obtener la significación de cada uno de esos valores se consulta la tabla de valores de F. Para iniciar la lectura de la tabla de F se usa el número de grados de libertad asociados a cada razón F para el numerador y el número de grados de libertad asociados al cuadrado medio dentro de los grupos (el denominador). La razón F entre columnas es 2.4579; que con 1 y 32 cuadros libertad será necesario una razón F de 4.15 o más para lograr la significación al nivel 0.05; y 7.50 para un nivel de 0.01. La razón es menor; por tanto, el valor obtenido *no es significativo*. Es decir, no hubo diferencia significativa entre el desempeño entre los estudiantes que trabajaron con el objeto de aprendizaje y los que trabajaron en la clase tradicional.

Para que la razón entre filas (rendimiento) sea significativa (con 1 y 32 grados de libertad) se necesitaría alcanzar 4.15 al nivel 0.05 o 7.50 al nivel de 0.01. Como el valor obtenido de F es igual a 66.058 (que supera ambos valores), resulta significativo al 0.01. Es decir, los estudiantes de alto rendimiento tuvieron un mejor desempeño que los de bajo rendimiento.

Para que haya interacción entre columnas y filas, con 1 y 32 grados de libertad, se necesitan los mismos niveles de F: 4.15 y 7.50. Con un valor 5.178 se

puede concluir que el resultado es significativo al nivel de 0.05, pero no al 0.01; por tanto, se dice que el resultado es *probablemente significativo*.

TABLA 9. Resumen del análisis factorial

Fuente de varianza	SS	df	MS	F	Nivel de significación
Entre columnas (estrategia)	1.604	1	1.604	2.4579	-
Entre filas (rendimiento)	43.121	1	43.121	66.058	0.01
Columnas por filas (interacción)	3.004	1	3.004	4.6026	0.05
Entre grupos	47.730	3	15.910		
Dentro de grupos	20.889	32	0.6528		
Total	68.619	39			

La primera razón F (entre columnas) muestra que los resultados de las estrategias instruccionales no difieren significativamente. Este análisis es la comparación del rendimiento combinado de los grupos 1 y 2 con el rendimiento combinado de los grupos 3 y 4.

La segunda razón F (entre filas) que presentó significación al nivel 0.01 se basa en la comparación del rendimiento académico de los sujetos de los grupos 1 y 3 con el de los grupos 2 y 4. Tomando como base esta razón se puede inferir que la diferencia entre el rendimiento de los estudiantes que tenían bajo rendimiento y los que tenían alto rendimiento *no se debió a la casualidad*. Al examinar los datos de la tabla 5 se advierte que los grupos de bajo rendimiento consiguieron una media combinada de 2.4, comparada con la media de los grupos de alto rendimiento con una media de 4.6. Como la diferencia muestra una razón F significativa, se llega a la conclusión de que en condiciones similares al del experimento efectuado cabe esperar un desempeño más alto cuando el nivel de rendimiento del estudiante es alto.

La tercera razón F muestra el efecto de la interacción de las dos variables. En este caso la significación de la razón F denota que el efecto que la estrategia instruccional produce en el desempeño depende probablemente del nivel de rendimiento académico.

Es posible ver el fenómeno con mayor claridad si se comparan los resultados observados con aquellos que se conseguirían si no hubiera ninguna interacción entre las variables. En la tabla 10 se muestra el valor esperado si no hubiera habido interacción.

TABLA 10. *Eliminación de efectos de interacción*

	Media Total +	Diferencia de estrategia +	Diferencia de habilidad =	Valor esperado
Grupo 1	3.5056	0.2111	-1.0944	2.6222
Grupo 2	3.5056	0.2111	1.0944	4.8111
Grupo 3	3.5056	-0.2111	-1.0944	2.2000
Grupo 4	3.5056	-0.2111	1.0944	4.3889

La tabla 11 muestra las medias reales y esperadas de los grupos. Se puede apreciar que el grupo 1 lo hizo mejor de lo que se esperaba: la media esperaba fue de 2.62 y la media real fue de 2.91. El grupo 2 con rendimiento alto y objeto de aprendizaje no logró el rendimiento esperado, con medias de 4.52 y 4.81 respectivamente. El grupo 3 tampoco alcanzó el rendimiento esperado, con medias respectivas de 1.91 y 2.20. El grupo 4 alcanzó un rendimiento por encima de lo esperado con medias de 4.68 y 4.39.

TABLA 11. *Medias reales de los grupos*

Real		Esperado	
Objeto	Tradicional	Objeto	Tradicional
2.91	1.91	2.62	2.20
4.52	4.68	4.81	4.39

Ahora, comparemos el rendimiento de los estudiantes de baja habilidad matemática con las dos estrategias, mediante la prueba de diferencias medias con la distribución *t* de student (tabla 12).

TABLA 12. *Baja habilidad matemática ensayo de diferencia de medias con t de student.*

obj	Tra	dat- med	(dat- med) ²	dat- med	(dat- med) ²	
1	2	-1.9111	3.652345679	0.08889	0.007901235	
1.5	1	-1.4111	1.991234568	-0.9111	0.830123457	
2	3	-0.9111	0.830123457	1.08889	1.185679012	
2.5	3	-0.4111	0.169012346	1.08889	1.185679012	
2.8	2.2	-0.1111	0.012345679	0.28889	0.08345679	
4.5	2.1	1.59	2.524567901	0.18889	0.035679012	
3.5	1.3	0.59	0.346790123	-0.6111	0.37345679	
4.6	1.3	1.69	2.852345679	-0.6111	0.37345679	
3.8	1.3	0.89	0.790123457	-0.6111	0.37345679	
∑	26.2000	17.2000	0.0000	13.1689	0.0000	4.4489
\bar{X}	2.9111	1.9111		1.4632		0.4943
S				1.2096		0.7031

$$\sigma = \text{raíz} \left(\frac{(N_1 S_1^2 + N_2 S_2^2)}{(N_1 + N_2 - 2)} \right) = 1.0493$$

$$t = \frac{(\bar{X}_1 - \bar{X}_2)}{\sigma \cdot \text{raíz} (1/N_1 + 1/N_2)} = 2.0215$$

Valor 0,05 2.12

Valor 0,01 2.92

De acuerdo con el valor obtenido de *t*, no existe diferencia significativa entre las dos distribuciones; es decir, los estudiantes del grupo de baja habilidad matemática que trabajaron con el objeto de aprendizaje obtuvieron el mismo desempeño frente a los de baja habilidad matemática, que estuvieron en el grupo de la estrategia instruccional tradicional. A pesar de la diferencia entre los valores del valor medio, se da la no significatividad entre las dos distribuciones y esto es debido a la gran dispersión de los datos del grupo del objeto de aprendizaje. Para este grupo el rango es de 3.06 y la desviación es de 1.2096; mientras que, para el grupo tradicional, el rango es de 1.2 y la dispersión es de 0.7.

La tabla 13 muestra el resultado del ensayo estadístico de la diferencia de medias en los grupos de habilidad matemática alta. El valor de *t* de student (-0.13505) indica que no hay diferencia en las medias de las dos distribuciones; es decir, los estudiantes de la estrategia instruccional del objeto de aprendizaje tuvieron el mismo desempeño que quienes siguieron la estrategia tradicional.

TABLA 13. Alta habilidad matemática ensayo de diferencia de medias con t de student

	obj	Trad.	dat- med	(dat- med) ²	dat- med	(dat- med) ²
	5.0	4.3	2.089	4.363	2.389	5.707
	4.8	5.0	1.889	3.568	3.089	9.541
	5.0	4.4	2.089	4.363	2.489	6.195
	4.7	5.0	1.789	3.200	3.089	9.541
	4.8	4.1	1.889	3.568	2.189	4.791
	3.9	4.8	0.989	0.978	2.889	8.346
	3.5	5.0	0.589	0.347	3.089	9.541
	4.8	5.0	1.889	3.568	3.089	9.541
	4.2	4.5	1.289	1.661	2.589	6.702
Σ	40.7	42.1	14.500	25.617	24.900	69.906
\bar{X}	4.5	4.7		2.846		7.767
S				1.687		2.787

S 2.443

$\sigma = \text{raíz} [(N1S1^2+N2S2^2) / (N1+N2-2)] = 2.443$

$t = (\bar{X}1-\bar{X}1) / \sigma*\text{raíz} (1/N1+1/N2) = -0.13505.$

2.7.4 Análisis del resultado de las evaluaciones

Realicemos ahora un análisis del resultado de las evaluaciones punto por punto, para señalar cuál fue el efecto de la teoría del aprendizaje sobre el reconocimiento del concepto, su aplicación y su utilización en la formación de nuevos conceptos. En la tabla 14 se puede observar la cantidad de preguntas y su valor según la cantidad de elementos que esta poseía. En total fueron 5 preguntas y 60 elementos.

TABLA 14. Resultados de la evaluación por preguntas

	6	4	18	31	1	Total	Nota/5
Valor preguntas						60	5
pregunta	1	2	3	4	5		
1	6	4	18	30	1	59	4.9
2	6	4	14	29	1	54	4.5
3	5	2	16	31	1	55	4.6
4	3	0	14	20	0	37	3.1
5	4	0	13	25	0	42	3.5
6	6	4	14	31	0	55	4.6

						Total	Nota/5
7	6	4	15	31	1	57	4.8
8	6	4	14	31	1	56	4.7
9	3	2	11	21	0	37	3.1
10	4	2	17	11	1	35	2.9
11	6	4	17	31	1	59	4.9
12	6	3	18	27	1	55	4.6
13	6	4	12	0	1	23	1.9
14	6	4	18	22	1	51	4.3
15	6	2	18	22	1	49	4.1
16	6	3	11	29	0	49	4.1
17	4	3	14	31	1	53	4.4
18	6	3	17	14	1	41	3.4

En la tabla 15 se muestra la distribución de las respuestas del punto 1. Se puede observar que el 72.3% obtuvo una calificación de 5 o más sobre un puntaje total de 6. El 66.7% contestó bien los elementos que constituyen la pregunta. El mínimo puntaje fue 3, que corresponde al 50% de la nota del punto. El punto tuvo como finalidad reconocer diferentes conceptos relacionados con el tema de matrices. Los resultados muestran que sí pueden reconocer los conceptos tratados.

TABLA 15. *Distribución de las respuestas del punto 1*

Datos	f	f n			
3	2	6	2	11.1%	11.1%
4	3	12	5	16.7%	27.8%
5	1	5	6	5.6%	33.3%
6	12	72	18	66.7%	100%
Total	18	95			

En la tabla 16 aparece la distribución de las respuestas del punto 2. El total de la calificación del punto fue de 4. Este punto también tuvo como finalidad el identificar conceptos de los componentes de una matriz. Se puede observar que el 66.66% obtuvieron una nota superior a 3. El 44.4% contestaron todos los puntos bien. El 11.1% no fue capaz de contestar ningún punto bien.

TABLA 16. *Distribución de respuestas del punto 2*

Datos	f	F n	
0	2	0	11.1 %
2	4	8	22.2 %
3	4	12	22.2 %
4	8	32	44.4 %
Total	18	52	

En la tabla 17 aparece la distribución del punto 3, que a su vez contiene 18 preguntas. Su objetivo fue revisar la aplicación de los conceptos en el empleo de procedimientos para hacer operaciones matriciales. El mínimo de preguntas contestadas correctamente fue de 11 que corresponde al 61 % de las preguntas. El puntaje promedio fue de 15 preguntas correctas que corresponde al 83 % de la nota. En general, el grupo obtuvo un desempeño bueno en este punto.

TABLA 17. *Distribución de respuestas en el punto 3*

Datos	f	F n	%
11	2	22	11.1 %
12	1	12	5.6 %
13	1	13	5.6 %
14	5	70	27.8 %
15	1	15	5.6 %
16	1	16	5.6 %
17	3	51	16.7 %
18	4	72	22.2 %
Total	18	271	100 %
Media		15.0556	

En la tabla 18 aparece la distribución del punto 4. El promedio de preguntas correctas fue de 24 de un total de 31. Hubo un estudiante que no pudo contestar ninguna pregunta, pero el luego se observa que el número que le sigue es 11 de 31 preguntas que corresponde al 5.6 %. Es decir, el 11.2 % contestó menos o igual a 11 preguntas correctas. El 33.3 % contestó correctamente todas las preguntas y corresponde al porcentaje más alto de preguntas respondidas correctamente.

TABLA 18. *Distribución de respuestas punto 4*

Datos	f	F n	%		
0	1	0	5.6 %	1	5.6 %
11	1	11	5.6 %	2	11.1 %
14	1	14	5.6 %	3	16.7 %
20	1	20	5.6 %	4	22.2 %
21	1	21	5.6 %	5	27.8 %
22	2	44	11.1 %	7	38.9 %
25	1	25	5.6 %	8	44.4 %
27	1	27	5.6 %	9	50.0 %
29	2	58	11.1 %	11	61.1 %
30	1	30	5.6 %	12	66.7 %
31	6	186	33.3 %	18	100 %
Total	18	436	100 %		
Promedio		24.2222	78.1 %		

El punto 5 consistió en una sola pregunta y se enfocó en desarrollar un concepto a partir de lo aprendido. Debemos hacer notar que durante la evaluación los estudiantes no contestaban la pregunta hasta que se les dio una sugerencia; con eso relacionaron con lo aprendido y pudieron resolver la pregunta. Esto lleva a pensar que se necesita más que el conocimiento del concepto para poder avanzar a construir nuevos conceptos de los que se saben. Los estudiantes no hacen la conexión entre los conocimientos adquiridos y la situación donde los van a aplicar. La distribución se muestra en la tabla 19. El 72.2 % contestó este punto bien, después de haber recibido la sugerencia.

TABLA 19. *Distribución de respuestas punto 5*

Datos	f	F n	%
0	5	0	27.8 %
1	13	13	72.2 %
Total	18	13	100 %
Promedio		0.72222	

2.7.5 Efecto de la adquisición de concepto sobre la retención

A los 4 meses se evaluó la permanencia del aprendizaje de los conceptos. En la tabla 18 se muestra la comparación de los dos momentos. Aquí se evaluó el reconocimiento del concepto y su aplicación al desarrollo de operaciones matriciales.

TABLA 20. Diferencia estadística para medir retención

n	x ₁	x ₂	x ₁ - \bar{x}	x ₂ - \bar{x}	(x ₁ - \bar{x}) ²	(x ₂ - \bar{x}) ²
1	2.9	2.4	-1.1	-1.6	1.2874	2.6699
2	4.4	4.8	0.4	0.8	0.1527	0.6675
3	4.9	4.2	0.9	0.2	0.8086	0.0523
4	3.6	4.2	-0.5	0.2	0.2086	0.0523
5	4.6	4.1	0.6	0.1	0.3139	0.0171
6	4.2	4.1	0.2	0.1	0.049	0.0171
7	3.4	1.8	-0.6	-2.2	0.3921	4.9383
8	4.0	4.0	0.0	0.0	0.0011	0.0011
9	4.5	4.8	0.5	0.8	0.2261	0.6675
10	3.1	3.8	-0.9	-0.2	0.7751	0.0267
11	3.1	3.7	-0.9	-0.3	0.7751	0.0683
12	4.7	4.6	0.6	0.6	0.416	0.3855
13	4.2	3.6	0.1	-0.4	0.0186	0.1292
14	4.7	4.5	0.7	0.5	0.5325	0.2734
15	4.9	4.5	0.9	0.5	0.8086	0.2734
16	4.6	4.0	0.6	0.0	0.3139	0.0011
17	1.9	4.1	-2.2	0.1	4.6294	0.0171
18	4.7	4.4	0.6	0.4	0.416	0.1805
Σ	72.3	71.8	0.0	0.0	12.1	10.4
\bar{X}	4.0160075	3.9869			0.6736	0.5799
s					0.8207	0.7615

Con los datos de la tabla 14 se obtuvieron los siguientes estadígrafos:

$$\begin{aligned} \sigma &= 0.9152391 \\ t &= 0.0105908 \\ G &= 34 \\ -t_{.975} \text{ y } t_{.975} &= 2.03 \end{aligned}$$

Con un contraste bilateral al nivel de 0.05 para 34 grados de libertad se acepta la hipótesis de que no hay diferencia significativa entre los grupos; ello

significa que el tiempo transcurrido entre las evaluaciones no afectó el aprendizaje de conceptos.

2.8 Resultados

De acuerdo con la pregunta y el objetivo planteados, el propósito de la investigación fue conocer el efecto que tiene un objeto de aprendizaje construido a partir de las estrategias propuestas para el aprendizaje de conceptos, sobre la formación de conceptos en estudiantes de bajo y alto rendimiento. Para aprender los conceptos se utilizó un software, en forma de objeto de aprendizaje modelado a partir de la teoría de formación de conceptos. El proceso arrojó los siguientes resultados:

No hubo diferencia significativa en los desempeños de los estudiantes que constituyeron los grupos de las estrategias instruccionales, objeto de aprendizaje y clase magistral, lo que significa que los objetos de aprendizaje se pueden utilizar como una alternativa de instrucción.

Los estudiantes de alto rendimiento tuvieron mejor desempeño que los de bajo rendimiento en las dos estrategias instruccionales; el objeto de aprendizaje no logró acortar la diferencia de desempeño en los grupos del objeto de aprendizaje. Tal vez los estudiantes de bajo desempeño necesitan un tratamiento especial que permita descubrir y manejar sus necesidades específicas para que permitan superar las dificultades que les impiden beneficiarse de los objetos de aprendizaje.

Los estudiantes de alto rendimiento mostraron mejor desempeño en las dos estrategias instruccionales. En los estudiantes de bajo rendimiento no hubo diferencia significativa entre los grupos de la clase magistral y los del objeto de aprendizaje. Con el objeto de aprendizaje se garantiza como mínimo el mismo rendimiento de una clase tradicional.

En la utilización de los conceptos aprendidos para el desarrollo de nuevos conceptos, se necesitó ayuda exterior para aplicar los aprendidos. Los

estudiantes por sí solos tuvieron dificultades para conectar la generación de nuevos conocimientos con los que ya poseían. En general, el objeto virtual de aprendizaje facilitó el aprendizaje de conceptos a los estudiantes dentro del nivel respectivo de la variable atributiva. En conclusión, para que el estudiante construya nuevos conceptos a partir de los aprendidos es necesario ofrecer actividades de aprendizaje encaminadas al desarrollo de esta habilidad.





Referencias

- Barker, P. (1994). Designing interactive learning. En T. de Jong & L. Sarti (Eds.), *Design and production of multimedia and simulation-based learning material* (pp. 1-30). Dordrecht: Kluwer.
- Bloom, B. S. (Ed.). (1956). *Taxonomy of educational objectives. Vol. 1: Cognitive domain*. Nueva York; Toronto: Longmans, Green.
- Cooper, G. (2003). Cognitive load theory as an aid for instructional design. *Australian Journal of Educational Technology*, 6 (2), 108-113.
- Damarin, S. (1982). *Fitting the tool with the task: A problem with the instructional use of computers*. Paper presented at the Annual Meeting of the American Educational Research Association, Nueva York.
- Groner R., & Kersten, B. (2001). Entwurf eines intelligenten Feedbacks in tutoriellen Lernsystemen, Seminararbeit am Institut für Psychologie der Universität Bern, Sommersemester 2001, [citado junio 1, 2017]. Recuperado desde <http://visor.unibe.ch/SS01/evaluation/TypologieLernsysteme.pdf>
- Gulmans, J. (1998). *The acquisition of coordinate concepts in health education*. (Tesis no publicada). Universiteit Twente Enschede. Met samenvatting in het Nederlands.

- Kirsh, D. (1997). Interactivity and multimedia interfaces. *Instructional Science*, 25(2), 79-96.
- Klausmeier, H. J., & Feldman, K. V. (1975). Effects of a definition and varying number of examples and nonexamples on concept attainment. *Journal of Educational Psychology*, 67(2), 174-178. <http://dx.doi.org/10.1037/h0076998>
- Kulhavy, R. W., & Stock, W. A. (1989). Feedback in written instruction: The place of response certitude. *Educational Psychology Review*, 1(4), 279-308.
- Lusti, M. (1992). *Intelligente tutorielle Systeme*. München: Oldenbourg.
- Markle, S. M., & Tiemann, P. W. (1970). Problems of conceptual learning. *British Journal of Educational Technology*, 1(1), 52-62. <https://doi.org/10.1111/j.1467-8535.1970.tb00517.x>
- 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 desde <http://dwb.unl.edu/Edit/MB/MasonBruning.html>
- Narciss, S. (2001). *Informative feedback as a bridge from instruction to learning in computer based-trainings*. Dresden: Institut für Psychologie IV.
- Perkins, D. (2003). *La escuela de la inteligencia*. Barcelona: Gedisa.
- Rosch, E. (1978). Principles of categorization. En E. Rosch & B. B. Loyd (Eds.). *Cognition and categorization* (pp. 27-48). Hillsdale, NJ: Erlbaum.
- Rosch, E., & Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7(4), 573-605. [https://doi.org/10.1016/0010-0285\(75\)90024-9](https://doi.org/10.1016/0010-0285(75)90024-9)
- Schulmeister, R. (1996). *Grundlagen hypermedialer Lernsysteme: Theorie, Didaktik, Design*. Bonn, París: Addison-Wesley.
- Schwier, R. A., & Misanchuk, E. (1993). *Interactive multimedia instruction*. Englewood Cliffs, NJ: Educational Technology Publications.
- Sims, R. (1997) Interactivity: A forgotten art? *Computers in Human Behavior*, 3(2), 157-180.

- Stevens, S. S. (1946). On the theory of scales of measurement. *Science, New Series*, 103(2684), 677-680.
- Stevens, S. S. (1957). On the psychophysical law. *Psychological Review*, 64(3), 153-181. <http://dx.doi.org/10.1037/h0046162>
- Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4(4), 295-312. [https://doi.org/10.1016/0959-4752\(94\)90003-5](https://doi.org/10.1016/0959-4752(94)90003-5)
- Tennyson, R. D., & Cocchiarella, M. J. (1986). An empirically based instructional design theory for teaching concepts. *Review of Educational Research*, 56(1), 40-71. <https://doi.org/10.3102/00346543056001040>
- Trabasso, T., & Bower, G. H. (1968). *Attention in learning: Theory and research*. Nueva York: Wiley.





ANEXO 1.

Ideas básicas sobre HTML

Traducido al español, *HTML* denota un lenguaje de marcas de hipertexto que se utiliza para estructurar páginas web. La información se muestra con rótulos rodeados de corchetes. Por ejemplo:

```
<nombre del rótulo> Contenido </nombre del rótulo>.
```

Todos los documentos comienzan con una declaración del tipo de documento. Para HTML5 se utiliza `<!DOCTYPE html>`. Luego viene `<html>` y termina con `</html>`. La información se visualiza en `<body> Contenido </body>`.

Los encabezados se muestran con los rótulos que van desde `<h1>` para el más grande y `<h6>` para el más pequeño. En la figura 98 aparece un archivo grabado en el bloc de notas con la extensión *html* que muestra el tamaño de los diferentes encabezados. Para correrlo se debe pulsar el ratón con la mano derecha y escoger cualquier navegador.

FIGURA 98. Archivo *rótulo.html*

```
<h1>Este es el rótulo h1</h1>  
<h2>Este es el rótulo h2</h2><h3>Este es el rótulo h3</h3>  
<h4>Este es el rótulo h4</h4>  
<h5>Este es el rótulo h5</h5>  
<h6>Este es el rótulo h6</h6>
```

En la figura 99 se muestra el resultado de correr el programa *rótulo.html*.

FIGURA 99. Resultado de correr archivo rótulo.html

Este es el rótulo h1

Este es el rótulo h2

Este es el rótulo h3

Este es el rótulo h4

Este es el rótulo h5

Este es el rótulo h6

Para los párrafos se utiliza `<p> </p>`. La figura 100 muestra la programación de dos párrafos.

FIGURA 100. Programando párrafos

```
<p> Este es el primer párrafo </p>
```

```
<p> Y este es el segundo párrafo </p>
```

Para mostrar imágenes se utiliza el siguiente rótulo: ``. La figura 101 muestra el resultado.

FIGURA 101. Imágenes en html

Nota. © 2018 Luis Eduardo Otero.

Para hacer una lista se utiliza ``; luego vienen los `li` para una lista no ordenada, o `` para una lista ordenada. En la figura 102 se presenta el archivo html, con una lista no ordenada.

FIGURA 102. Archivo que construye una lista

```
<li>Conceptualización</li>
<li>Estadística descriptiva</li>
<li>Ensayos estadísticos</li>
```

La figura 103 muestra el resultado.

FIGURA 103. Resultado de lista.html

-
- Conceptualización
 - Estadística descriptiva
 - Ensayos estadísticos
-

Los menús se construyen utilizando las listas; basta agregarles un *id*, *class* o un evento *onclick* para correr el elemento correspondiente. En la figura 104 se agregó un evento *onclick* a conceptualización que llama a la función *conce* (). Fíjese que la función va entre los rótulos `<script>` `</script>`, que utiliza el lenguaje de programación JavaScript. El *alert* se utiliza para mostrar mensajes.

FIGURA 104. Construcción de un menú

```
<ul>
<li onclick = "conce()">Conceptualización</li>
<li>Estadística descriptiva</li>
<li>Ensayos estadísticos</li>
</ul>
<script>
function conce()
{
```

```

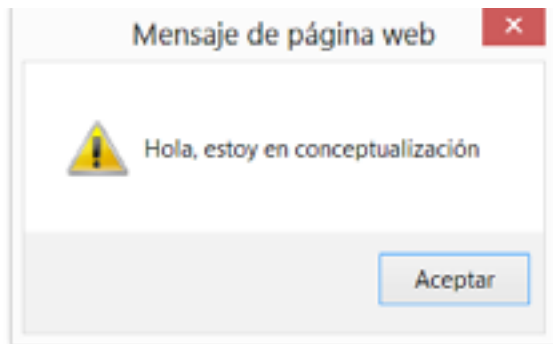
    alert("Hola, estoy en conceptualización");
}
</script>

```

Al hacer clic sobre conceptualización aparece como resultado lo que muestra la figura 105.

FIGURA 105. Resultados de la función conce

- Conceptualización
- Estadística descriptiva
- Ensayos estadísticos



Para la interactividad se utiliza `<input>` y `<select>`. Con este último se construye una lista. Se tienen diferentes `<input>`; aquí solo se muestra `radio` y `text`. La figura 106 muestra el archivo html y la 107 el resultado. Observe el rótulo `
`, se utiliza para agregar una línea en blanco.

FIGURA 106. Tipos de `<input>`

```

Escriba algo: <input type = "text" >
<br><br>
Marque la opción: <input type = "radio" >
<br><br>
<select>
  <option value= "pollo" >Pollo</option>
  <option value= "carne" >Carne</option>
  <option value= "pescado" >Pescado</option>
  <option value= "moluscos" >Molusco</option>
</select>

```

FIGURA 107. Resultado de *input.html*

Escriba algo:

Marque la opción:

Pollo ▾
 Pollo
 Carne
 Pescado
 Molusco

Para dividir una hoja HTML en el programa que se describe en este libro se utiliza *div*, que es rótulo derivado de división. En la figura 108 se divide la hoja en tres divisiones. Observe que las divisiones van acompañadas de la palabra *id* con el nombre que identifica la división.

FIGURA 108. Creando divisiones

```
<div id = "menu" ></menu>
<div id = "submenu" ></menu>
<div id = "desarrollo" ></menu>
```

En HTML se pueden realizar muchas más tareas, pero solo se quiere dar una introducción para que se entiendan los programas.



ANEXO 2.

CSS

El CSS se utiliza para dar estilo a una hoja. Ello se puede hacer de diferentes maneras:

1. `<h1 style = "color : blue;">` Esto es un encabezado azul `</h1>`
2. Colocando `style` en el `<head>` como muestra la figura 109.

FIGURA 109. *Estilo en el <head>*

```
<html>
  <head>
    <style>
      #colores {
        color: blue;
      }
    </style>
  </head>
</body>
</html></html>
```

3. Un archivo con extensión CSS aparte.

```
<head> <link rel = "stylesheet" type = "text/css" href =
"miestilo.css"> </head>
```

4. En este libro se hace con programación en jQuery, como se muestra en la figura 110.

FIGURA 110. Estilo con jQuery

```

Function stilo ()
{
$("#colores").css({"color":"blue"})
}

```

Existen una gran variedad atributos; en la figura 111 se utilizan *border* para establecer los bordes; *border - style* para el estilo del borde; *width* para su ancho; *list -style - type* para el tipo de lista; *position* para establecer la posición del elemento dentro de la hoja; *top* para la altura; *left* para la posición con respecto a la izquierda; *float* para que los menús o el texto que se presenten alineados en forma horizontal y *background* para darle color al fondo. La expresión `$("#menDesa")` significa que se le está dando estilo al *div* con identificación *menDesa*. Observe que para los *li* no se necesita escribir el carácter `#`. La unidad de medida que se utiliza es la *em*.

FIGURA 111. Ejemplo de estilos con jQuery

```

$("#menDesa").css({"border":"#cc9900",
"border-style":"solid",
"width":"60%",
"list-style-type":"none",
"position":"absolute",
"top":16em,
"left":"20.5em"});
$("#subtit").css({"background":"#cccccc","text-align":"center"});
$("li.subi").css({"float":"left"});
$("li").css({"width":100});
$("#contenido").
css({"position":"absolute","top":"7em","left":"21em","width":"48em"});

```



ANEXO 3.

Diseño de la base de datos

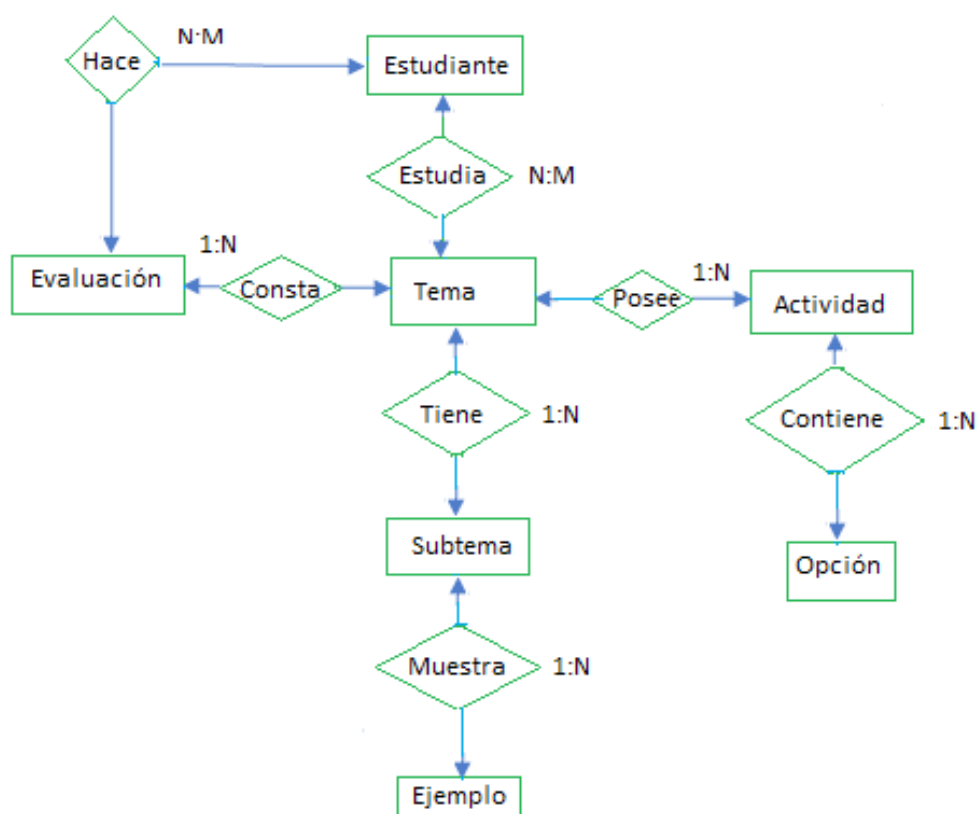
Diseño conceptual

Son las entidades y acciones que conforman el objeto de aprendizaje que salen de la consulta a los profesores que van a utilizar el software. En este caso son:

- Los temas (número del tema, título, contenido).
- Los subtemas (número del subtema, número del tema, título, contenido).
- Los ejemplos (número del tema, número del tema, contenido del ejemplo).
- Las actividades (número del tema, número del tema, contenido de la actividad, retroalimentación).
- Evaluación (número del tema, número del tema, número de la evaluación).

Del análisis anterior se obtienen las siguientes entidades: temas, subtemas, ejemplos, actividades y opciones. A partir de esta información se construye el diseño conceptual de la figura 112.

FIGURA 112. Esquema de diseño conceptual de una base de datos



Diseño lógico

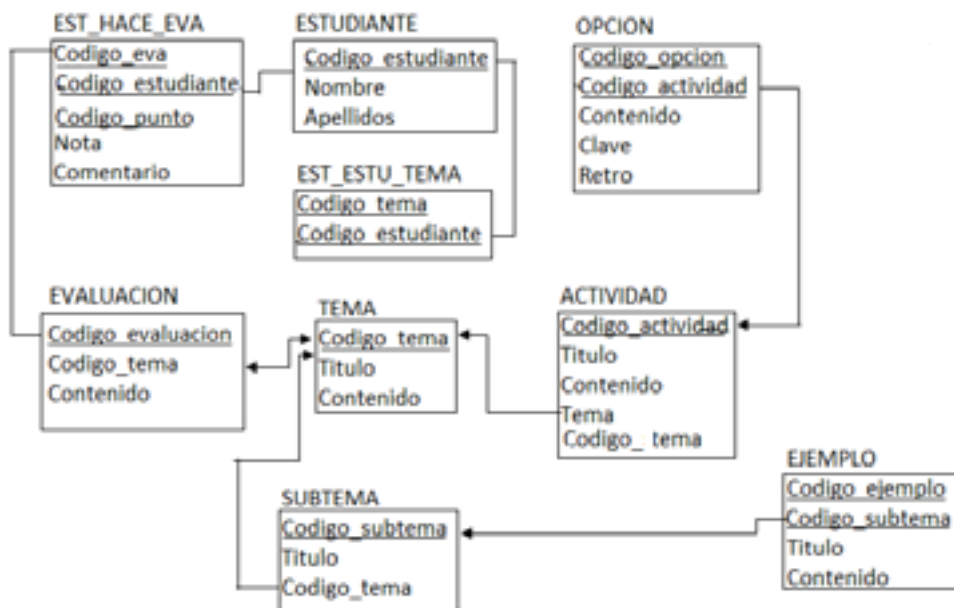
Para guardar la información se utilizará una base de datos que contiene las entidades presentadas en la figura 113. Estas tablas contienen los cambios que se fueron efectuando en el proceso.

FIGURA 113. Tablas de la base de datos



En la figura 114 se observan las identidades con sus relaciones.

FIGURA 114. Entidades con sus relaciones



ANEXO 4.

Clases en PHP

Una clase en PHP es un tipo de dato que contiene propiedades y métodos. En la figura 115 se observa la clase *subtema* que contiene un arreglo llamado también *subtema* y un método: *displayTit*. En la declaración se les asignó a los elementos del arreglo un valor vacío que, posteriormente, se llenará con la información obtenida de la base de datos. La función *displayTit* muestra la información con una sentencia *echo*. Al final del archivo se declara un tipo de dato *subtema* que corresponde a la clase; se llama a la función *bdDatos()* para leer la base de datos y finalmente a *bdDatos()* para mostrar la información.

FIGURA 115. Una clase en PHP

```
class subtema {
    var $subtema = array(
        "id_subtema"=>"",
        "cont_subtema"=>"",
        "tit_subtema"=>"");
}
function displayTit()
{
    echo "<b>".$this->tema["tit"]."</b><br><br>";
    echo $this->tema["conte"];
}
}
$tem = new tema;
$tem->bdDatos();
$tem->displayTit();
```



ANEXO 5.

Recuperación de datos con MySQL

MySQL utiliza tablas para guardar información; estas están compuestas por filas (*row*) y columnas (*column*). Las columnas corresponden a los atributos y las filas a los registros (figura 116).

FIGURA 116. Datos de una tabla de MySQL

Id_tema	Titulo_tema	Contenido_tema
0	Conceptualización	La conceptualización ofrece una introducción sobre...
1	Escalas	Se refiere a las maneras como se definen y se cate...
2	Distribuciones de frecuencia	Es una disposición tabular de datos por clases jun...

Si se quiere obtener información de una o más tablas se utiliza el comando *select*.

```
SELECT * FROM nombre de la tabla
```

El asterisco significa que se están recuperando todas las columnas de la tabla; la cláusula *from* indica que tabla se hace la recuperación. Para obtener los datos mostrados en la figura 116 se hace la siguiente consulta:

```
SELECT * FROM tema
```

Si se quiere mostrar el título y el contenido:

```
SELECT Titulo_tema, Contenido_tema FROM tema
```

Si se va a seleccionar información de varias tablas, la información aparecerá de la siguiente manera:

```
SELECT * FROM actividad, opcion WHERE actividad.Id_actividad=
opcion.Id_actividad and actividad.Id_tema = 1
```

En la sentencia mostrada, el asterisco indica que se están seleccionando todos los campos que tienen el mismo número de actividad en las tablas actividad y opción, pertenecientes al tema uno. Analice el uso de *WHERE* y *AND*; estos se utilizan para establecer los criterios de búsqueda. La base de datos se puede crear directamente en MySQL, lo mismo que la inclusión de los datos. Pero si quiere hacer por programación, la figura 117 muestra la manera de cómo hacerlo.

FIGURA 117. Creación de una base de datos

```
<?php
$servername = "localhost";
$username = "username";
$password = "contraseña";
// Creando la conexión.
$conn = new mysqli($servername,$username,$password);
//Comprobando la conexión.
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Creando la base de datos.
$sql = "CREATE DATABASE INVESTA";
if ($conn->query($sql) === TRUE) {
    echo "Se creó la base de datos exitosamente";
} else {
    echo "Se presentaron errores a crear la base de datos: " . $conn->error;
}
$conn->close();
?>
```

Para crear una tabla (por ejemplo, tema) proceda así:

```
CREATE TABLE tema (
    id INT(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    Titulo_tema VARCHAR(100) NOT NULL,
    Contenido_tema TEXT NOT NULL)
```


En la sentencia se está creando un campo tipo entero de 11 caracteres que se incrementa automáticamente, otro campo de tipo cadena de 100 caracteres y otro tipo texto. Para insertar datos puede utilizar el programa que aparece en la figura 118.

FIGURA 118. *Insertando datos en una base de datos*

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "INVESTA";
//Create connection
$conn = new mysqli($servername,$username,$password,$dbname);
//Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error );
}
$sql = "INSERT INTO tema (Id_tema,Titulo_tema,Contenido_tema)
VALUES ('1','Escalas ','Se refiere a las maneras como se definen y se categorizan
las variables. Según su grado de exactitud puede ser: nominales,ordinales,de
intervalo y de razón (Stevens,1946).')";
if ($conn->query($sql) === TRUE) {
    echo "\nSe agregaron registros satisfactoriamente.\n";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>

```

Se puede crear un formulario para capturar los datos como nos lo muestra la figura 119.

FIGURA 119. *Formulario para capturar datos*

```

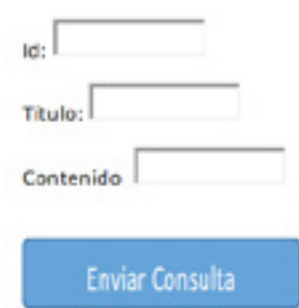
<html>
<body>

<form action="tema.php" method="post">
Id: <input type="text" name="id\"><br>
Titulo: <input type="text" name="tit\"><br>
Contenido: <input type="textarea" name="conte\"><br>
<input type="submit">
</form>
</body>
</html>

```

El resultado lo podemos observar en la figura 120.

FIGURA 120. *Resultado de usar un formulario*



El programa que recibe los datos aparece en la figura 121.

FIGURA 121. *Recibiendo los datos de un formulario*

```

<html>
<head>
<meta charset="UTF-8">
<title></title>
</head>
<body>

```

```
<?php
// put your code here
$id = filter_input(INPUT_POST,"id");
$tit = filter_input(INPUT_POST,"tit");
$conte = filter_input(INPUT_POST,"conte");
echo $id."<br><br>";//Solo para mostrar los datos recibidos, no se necesita
echo $tit."<br><br>";//Solo para mostrar los datos recibidos,no se necesita
echo $conte."<br><br>";//Solo para mostrar los datos recibidos,no se necesita
?>
</body>
</html>
```

En las sentencias de inserción se utiliza:

```
$sql = "INSERT INTO tema (Id_tema,Titulo_tema,Contenido_tema)
VALUES ($id,$tit,$conte);
```



ANEXO 6.

Conceptos de JavaScript

Para guardar los valores en JavaScript se utilizan las variables. Para declararlas se utiliza *var*; el programa determina el tipo de las variables. El siguiente es un ejemplo de declaración de variables:

```
var t = "Escalas nominales";
```

La variable *t* es de tipo cadena. Otro tipo de variable son los arreglos, los cuales se utilizan para guardar varios valores en una sola variable. Se pueden declarar arreglos vacíos como se observa en el siguiente ejemplo:

```
var maMenu = [ ];
```

Los arreglos utilizan un índice para identificar los valores:

```
maMenu[0] = " <ul><li id = 't0' class = 'subMen'> Conceptualización </li></ul>";
```

En este caso se está asignando el elemento de un menú en la primera posición del arreglo *maMenu*. Para agregarlo a una página HTML se emplea *append*:

```
$("#menu).append(maMenu[0])
```

El *#menu* se refiere a una división identificada con el *id* menu. Otra manera de crear arreglos es con la sentencia *Array*:

```
var maMenu = new Array (5);
```

Se está creando un arreglo con 5 elementos. Si se conocen los valores de los elementos, el arreglo se puede declarar así:

```
var arreglo = ["mariposa", "gusano", "sapo"];
```

El elemento arreglo [1], por ejemplo, es igual a gusano. Esa sentencia es lo mismo que escribir:

```
arreglo [0] = " mariposa";  
arreglo [1] = " gusano";  
arreglo [2] = " sapo";
```

Para manejar los programas se utilizan las instrucciones condicionales que son muy importantes para el manejo de errores de los estudiantes. Se ha utilizado con frecuencia la instrucción *if*, que sirve para escoger entre varias alternativas. En el listado de la figura 122 se determina la acción que hace cada animal de arreglo según el valor de una variable.

Si el animal es la mariposa, se muestra mediante la función *alert* el mensaje: «La mariposa vuela». Observe que *la* y *vuela* van entre comillas porque son cadenas. La expresión *arreglo [0]* es una variable a la cual se le ha asignado previamente el valor *mariposa*. El *else if* se utiliza para las alternativas; si el animal no es mariposa, puede ser el gusano o el sapo. En caso de que no sea ninguno de los contenidos en el arreglo se muestra el mensaje informando sobre la situación.

FIGURA 122. Instrucciones condicionales

```
.....  
If (animal === arreglo [0])  
{  
  alert("La " + arreglo [0] + "vuela");  
}  
else if(animal === arreglo [1])  
{  
  alert("El " + + arreglo [1] + "se arrastra");  
}  
else if(animal === arreglo [2])  
{  
  alert("El " + arreglo [2] + "le gusta el agua");  
}  
else  
{  
  alert("Ese animal no existe en el arreglo");  
}  
.....
```

Otra instrucción usada en los programas es el *for*. Su forma general es:

```
for (var i = 0; i < 20; i ++)  
{  
}
```

En el ejemplo tiene un valor inicial de cero; el valor máximo al cual puede llegar es 20 y finalmente se señala el incremento. La expresión *i++*, es semejante a expresar $i = i + 1$. Cuando llega a 20 se finaliza la instrucción. En la figura 123 aparece un ejemplo de su uso. En la figura se está asignando a un arreglo *ob*, el valor de otro arreglo *contDomes*, hasta llegar al valor contenido en *cantPT*. También, hay una sentencia *if*; si *j* supera el valor de 3, *j* toma el valor 0 y *k* aumenta en 1.

FIGURA 123. Instrucciones de repetición

```
for(var i=0; i<cantPT;i++)  
{  
    ob[k][j]= contDomes[i] ;  
    j++;  
    if(j>3)  
    {  
        j=0;  
        k++;  
    }  
}
```

La instrucción *while* es otra sentencia de repetición que significa *mientras que*. En la figura 124 se da un ejemplo. Si *x* es menor que 15 se hace la operación indicada por la fórmula y luego *x* se incrementa en 1.

FIGURA 124. Sentencia de repetición *while*

```
while (x <15)  
{  
    y=x+4;  
    x++;  
}
```

Ahora se tratan las funciones. Una función de JavaScript es un bloque de código diseñado para ejecutar una tarea particular. Las funciones se invocan (se llaman) desde otras funciones. En JavaScript las funciones pueden realizar el papel que en otros lenguajes realiza la clase. En la figura 125 aparece una función que recibe un nombre y luego lo muestra en un *div* llamado nombre.

FIGURA 125. *Función nombre*

```
function nombre(nom)
{
  $("#nombre").append(nom);
}
```

El lenguaje JavaScript tiene funciones incorporadas, algunas de las cuales aparecen en la tabla 21.

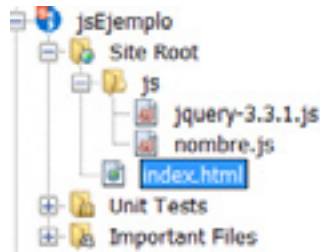
TABLA 21. *Algunas funciones de JavaScript*

Función	Uso
<i>objeto.attr</i>	Devuelve el atributo de un objeto
<i>objeto.click()</i>	Añade un evento clic al objeto.
<i>variable.indexOf()</i>	Devuelve la posición del primer carácter en el cual comienza una cadena que hace parte de otra cadena.
<i>variable.substr(inicio,fin)</i>	Extrae parte de una cadena desde inicio hasta fin.
<i>parseInt(variable)</i>	Convierte una cadena en un número.
<i>objeto.empty()</i>	Limpia el contenido de un objeto.
<i>cadena.split(separador)</i>	Convierte una cadena en un arreglo tomando como referencia el separador.
<i>objeto.keypress()</i>	Añade un evento <i>keypress</i> . Es decir, al presionar un carácter se activa la función.
<i>objeto.val()</i>	Devuelve el valor contenido en un objeto.

Durante todo el desarrollo se manejó las hojas de HTML y CSS con JavaScript. Finalmente, presentaremos un ejemplo completo para ilustrar el uso de JavaScript con NetBeans 8.2 y el uso de objetos de HTML. En la figura 126 se

observa que se creó un proyecto con el nombre *jsEjemplo*, en el cual se abrió una carpeta llamada *js* y dentro se colocó el jQuery y el programa que se implementando llamado *nombre.js*. Recuerde que el jQuery debe descargarlo de internet.

FIGURA 126. *Ejemplo completo del uso de JavaScript*



En la figura 127 se muestra la hoja de HTML. En la figura se puede observar la inclusión de los programas *jQuery - 3.3.1.js* y *nombre.js* en la hoja de HTML. En el *body* aparece la función *nombre ()* entre los rótulos *<script>*.

FIGURA 127. *Codificación ejemplo de JavaScript*

```

<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <script src = "js/jquery-3.3.1.js" > </script>
    <script src = "js/nombre.js"> </script>
  </head>
  <body>
    <script>
      nombre ();
    </script>
  </body>
</html>

```


En la figura 128 aparece la función *nombre ()*, con *append* se agregan las divisiones *caja* y *demo* al cuerpo del documento. Se declara la variable *tx* que contiene un párrafo *<p>* y una caja de texto con la identificación *txt*. Luego se agrega la variable *tx* a la división *caja*. A *caja* por medio de su identificación se le añade un evento *keypress*. Si se da enter lo escrito en la caja se le asigna a la variable *nom*; a continuación se llama a la función *Saludo ()* que regresa una cadena y se le asigna a la variable *sal*, cuyo contenido se muestra en la división *demo*.

FIGURA 128. Función que demuestra el uso de JavaScript

```
function nombre()
{
    $("body").append("<div id = 'caja' >");
    $("body").append("<div id = 'demo' >");
    var tx = "<p>Introduce tu nombre: </p><input type = 'text' id = 'txt' >";
    $("#caja").append(tx);
    $("#txt").keypress(function (e){
        if(e.keyCode === 13)
        {
            var nom = $("#txt").val();
            var sal = Saludo(nom);
            $("#demo").append(sal);
        }
    });
}
```

En la figura 129 aparece la función *Saludo ()*; aquí se declara la variable *deSal* que contiene «Te saludo con alegría» + *nom*; *nom* es reemplazado con el nombre enviado desde la función *nombre ()*. La sentencia *return* sirve para enviar respuestas desde una función. La función *nombre ()* invoca a la función *Saludo ()* y esta como respuesta devuelve el contenido de la variable *deSal*.

FIGURA 129. *Función que devuelve un saludo*

```
function Saludo(nom)
{
  var deSal = "Te saludo con alegría " + nom;
  return deSal;
}
```

Internet contiene material abundante sobre el tema. Una página recomendada es www.w3schools.com, que contiene un desarrollo bastante completo sobre diferentes temas.



ANEXO 7.

Función recorte

En la figura 130 la función *recorte* tiene como fin eliminar los espacios en blanco de una cadena. Observe el uso del condicional, si es diferente de blanco (“ ”) agregue el carácter a la variable *txtR*. Se utiliza para comparar cadenas porque no se sabe que espacios en blanco puede dejar usuario cuando escribe una información.

FIGURA 130. Cadena que elimina espacios en blanco

```
recorte(txt)
{
  var = "";
  var j;
  for (j=0;j < ;j++)
  {
    if(txt[j] != " ")
    {
      = txtR+txt[j];
    }
  }
  return ;
}
```



ANEXO 8.

Función de ordenación

FIGURA 131. Ordenando un arreglo

```
function burbuja (mat)
{
  var temp=0;
  for(var i=1;i < mat.length;i++){
    for (var j=0 ; j < mat.length- 1; j++){
      if (mat[j] > mat[j+1]){
        temp = mat[j];
        mat[j] = mat[j+1];
        mat[j+1] = temp;
      }
    }
  }
  return mat;
}
```

FIGURA 132. Convertir una cadena en arreglo de números

```
function ordenacion(maOrd){
  var ordenados=[];
  for (var i=0;i<maOrd.length;i++) {
    ordenados[i]=parseInt(maOrd[i]) ; }
  ordenados =burbuja(maOrd);
  for (var i=0;i<maOrd.length;i++) {
    $("#contenido").append("<input type = 'text' id = '"+i+"' class = 'orden'>"); }
  $(".orden").css({"width":20});
  $(".orden").keypress(function(e){
    if(e.keyCode ===13) {
      var atr=$(this).attr("id");
      var ca = $("#"+atr).val();
```

```
var caP =parseInt(atr);
var num = parseInt(ca);
if(ordенados[atr]===num)      {
    alert ("Muy bien");
    $("#"+{caP+1}).focus();
}
else {
    alert ("No coincide");
} }
});
}
```





Esta obra se terminó de imprimir en diciembre de 2018
con tipo Chaparral Pro con punto 11/15,2
sobre papel Bond de 90 gramos
en Imagen Editorial S. A. S.
Bogotá, D. C., Colombia

Esta obra abarca dos temáticas complementarias: en primer lugar, el diseño y la implementación de objetos virtuales de aprendizaje, los cuales son modelados a partir de la teoría de formación de conceptos, utilizando como herramienta de análisis a la teoría cognitiva. La idea es poner a disposición de profesores y diseñadores de objetos virtuales de aprendizaje recursos bibliográficos y prácticos que los orienten en la construcción de tales objetos. En segundo lugar, presentamos el desarrollo y resultados de una investigación que buscó medir el impacto de los objetos virtuales, modelados a partir de la teoría de psicología de aprendizaje, sobre el aprendizaje de los estudiantes. Estos temas se articulan por la presentación y discusión del concepto de retroalimentación, así como a través del desarrollo de ideas básicas sobre la teoría de elaboración de *software*.

ISBN: 978-958-9020-09-8



9 789589 020098