

## Fundamentos para el Submódulo Evaluador en Sistemas Tutores Inteligentes: Diagnóstico, predicción y autoevaluación

**Zulma Cataldi,**

LIEMA - Laboratorio de Informática Educativa y Medios Audiovisuales. Facultad de Ingeniería. UBA.  
Facultad Regional Buenos Aires. Universidad Tecnológica Nacional  
liema@fi.uba.ar

**Fernando Salgueiro,**

LIEMA - Laboratorio de Informática Educativa y Medios Audiovisuales. Facultad de Ingeniería. UBA.  
fsalgueiro@fi.uba.ar

**Fernando Javier Lage<sup>1</sup>**

LIEMA - Laboratorio de Informática Educativa y Medios Audiovisuales. Facultad de Ingeniería. UBA.  
Facultad Regional Buenos Aires. Universidad Tecnológica Nacional.  
flage@fi.uba.ar

### Abstract

In this paper, we present the basic functions of the evaluator submodule and how it will carry out diagnosis and evaluation. Taking into consideration the mistakes committed in the evaluations and its analysis, a diagnosis is carried out. Based on data of the first evaluations, the prediction of future mistakes is useful to give suggestions, bearing in mind improving performance by making a corrective exercitation. In order to carry out this prediction of future errors, a neuronal network is used and based on this; a sequence of exercises and problems is suggested to the student in order to improve his production and therefore his grades.

Regarding the function of the evaluation, in order to help in the process of new conceptualization of errors, an application has been made so that students can carry out their own auto evaluation. Different series of questions based upon the errors were elaborated; each of them corresponds with a Didactic Unit already studied. This instrumentation intends to make the student realize his weaknesses and therefore, would be able to reinforce them before his partial or final evaluation.

**Key words:** *Intelligent Tutoring Systems, Neural networks, Evaluation, Prediction*

### Resumen

Se presentan las funciones básicas del submódulo evaluador y se describe cómo se llevará cabo el diagnóstico y la evaluación. Con base en los errores cometidos en sus evaluaciones y a partir del análisis de los mismos se busca efectuar un diagnóstico. En función de los datos de las primeras evaluaciones se podrán predecir los errores futuros para sugerirle ejercitación correctiva a fin de mejorar el rendimiento. Para efectuar la predicción de los errores siguientes se usa una red neuronal y sobre esta base se le sugiere al estudiante una secuencia de ejercicios y problemas a fin de mejorar su producción y por lo tanto sus calificaciones.

En cuanto a la función de evaluación a fin de ayudar en el proceso de reconceptualización de los errores, se ha realizado una aplicación para que los estudiantes puedan efectuar su autoevaluación. Se han elaborado diferentes series de preguntas con base en los errores que se corresponden a cada Unidad Didáctica trabajada. Con esta instrumentación se busca que el alumno pueda *darse cuenta* de sus debilidades y pueda reforzarlas antes de su evaluación parcial o final.

**Palabras Clave:** *Sistemas Tutores Inteligentes, Redes neuronales, Evaluación, Predicción.*

---

<sup>1</sup> Esta comunicación ha sido desarrollada en el marco del Convenio FI-UBA y UTN-FRBA. Proyecto *Modelado del tutor basado en redes neuronales para un Sistema Tutor Inteligente* del Programa de Incentivos 2007-2008.

## 1. INTRODUCCIÓN

En publicaciones previas [1] se ha señalado, el bajo rendimiento de los estudiantes de Programación Básica<sup>2</sup> en sus evaluaciones parciales y finales durante los últimos seis cuatrimestres [1]. Este trabajo de investigación surge buscando una solución a este problema a través de la predicción tomando como datos los resultados en las evaluaciones y su análisis para poder efectuar luego un diagnóstico, a fin de sugerir estrategias de refuerzo, ya que se piensa que un sistema predictor podría asistir a los estudiantes en su trayecto [2]. En la actualidad no existen sistemas expertos que resuelvan en forma eficiente este problema, ya que sólo se encuentran disponibles algunos programas para enseñanza de programación, pero de tipo tutorial [3,4].

La autoevaluación se fundamenta sobre bases epistemológicas provenientes de las corrientes que abogan por la autorregulación del alumno y se justifica a partir del problema del desgranamiento sostenido en los cursos iniciales de Programación Básica en la universidad. Se creó una herramienta a partir del listado de los errores que los alumnos comenten en las evaluaciones parciales y finales. Éstos fueron registrados y clasificados conformando una base de datos refinada para poder componer la autoevaluación diseñada en base a ejercicios para reforzar los aspectos críticos.

## 2. DESARROLLO

El sistema planteado resolvería el problema de la predicción del comportamiento de los alumnos en el marco universitario, dado que a partir de los primeros exámenes se podrían predecir posibles errores futuros a través del diagnóstico y la predicción, a fin de sugerir la ejercitación correctiva a para mejorar su rendimiento y encauzar su aprendizaje hacia conceptualizaciones incorporadas de forma más significativa y permanente [5]. Los errores cometidos por cada alumno en los exámenes son los datos de partida, se cargan en una base de datos y permiten efectuar una primera categorización en grandes grupos de temas problemáticos. A partir de estos datos el sistema que se propone deberá ser capaz de predecir las próximas fallas de un alumno y como consecuencia de ello, deberá determinar la siguiente secuencia de problemas y/o ejercicios que el estudiante deberá realizar para mejorar su rendimiento y por ende sus calificaciones [6].

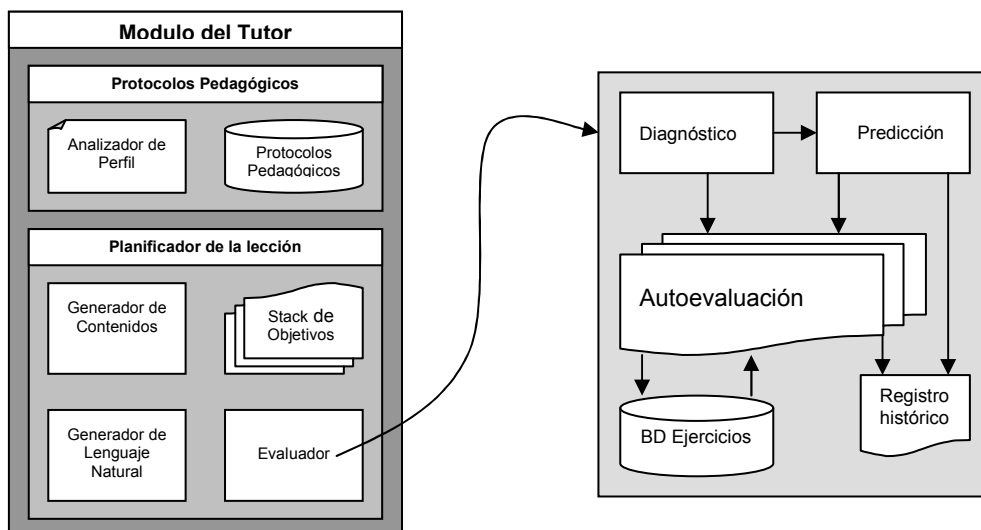


Figura 1: Módulo del tutor y submódulo Evaluador

En la Figura 1 se observa el esquema del módulo tutor y el submódulo evaluador con sus funciones básicas.

<sup>2</sup> En Facultad de Ingeniería se corresponde a las asignaturas: Computación que se dicta para todas las carreras excepto Informática y Electrónica y Algoritmos y Programación I para Ingeniería Informática y Licenciatura en Análisis de Sistemas.

El problema del diagnóstico y la predicción incluye:

- una primera etapa de diagnóstico, que está determinada por las respuestas dadas por los alumnos en sus exámenes parciales y finales,
- una segunda parte basada en la predicción de los errores futuros y en la determinación de la ejercitación que deberá resolver un alumno a fin de mejorar su rendimiento.

Este problema está relacionado directamente con los contenidos de la asignatura y además debe considerar, desde el punto de vista didáctico, las acciones del tutor y del estudiante que conforman el triángulo didáctico, en tanto es un proceso comunicacional [4]. Por lo tanto debería plantearse una solución con base en los sistemas inteligentes [7,8,9] buscando predecir comportamientos futuros sugiriendo líneas de acción, tomando alguna decisión pedagógica [9,10].

En un STI (Sistema Tutor Inteligente) existe un *modelo de dominio* donde se encuentra el conocimiento sobre el dominio en particular, de tipo declarativo (los primeros principios, la comprensión del dominio) y procedural (el conocimiento que es utilizado para realizar una tarea) [11]. Este conocimiento del dominio consiste en los hechos y en las relaciones entre ellos que, generalmente debe ser fortalecido en general por uno o más especialistas [12,13]. Este módulo de dominio, tiene algunas funciones básicas, ya que sirve como fuente del conocimiento a ser presentado al estudiante, lo que incluye la generación del material, generación de preguntas y de respuestas que conforma un patrón que permitirá evaluar el conocimiento del estudiante. Para ello, el sistema debe ser capaz de generar soluciones a los problemas en el contexto del estudiante, para que sus respuestas puedan ser evaluadas. La base del conocimiento del dominio es un componente clave en el *sistema predictor*, ya que es ahí donde está representado el material de enseñanza [14,15]. En los casos en que el dominio sea de naturaleza descriptiva y teórica (como en geografía ó física), la representación utilizada es la declarativa a través de redes semánticas ó “frames”. En los casos en que el dominio esté orientado a una tarea, tal el caso de la programación, la representación tiende a ser procedural, ya que son originalmente reglas de producción [16]. El modelo del dominio es un tema que ha sido estudiado [15, 16], así como el modo en que el sistema lo puede usar para razonar. Algunas de las representaciones posibles son las redes semánticas, reglas de producción y “constraints” [15, 16]. Su elección depende de cómo se lo usará ya que es común a todos los usuarios del sistema. A fin de dar solución a la problemática planteada de predicción y diagnóstico, se indagará sobre la aplicación de redes neuronales que han dado buenos resultados en diversas áreas [17].

También se hará un acercamiento a la problemática desde los aportes que puede hacer *la autoevaluación* a la formación del estudiante, en la adquisición de experticia en el dominio de la resolución de problemas en ingeniería desde los contenidos de la Programación Básica. En este sentido:

- Se propondrá una herramienta para que el estudiante pueda descubrir e interpretar cómo puede hacer para resolver correctamente sus problemas trabajando desde la reconceptualización de sus errores.

### 3.1. DIAGNÓSTICO Y PREDICCIÓN

#### 3.1.1. Las redes neuronales

Las redes neuronales (RN) son conjunto de elementos más simples que se interconectan en paralelo en forma jerárquica y que interactúan como los sistemas neuronales psicológicos [18]. A fin de poder utilizarlas para representar sistemas de mayor complejidad pueden tener retroalimentación. Una de sus características diferenciales es que pueden aprender de la experiencia a través de la generalización de casos [19]. Una red neuronal se caracteriza por cuatro elementos básicos: *su topología, el mecanismo de aprendizaje, tipo de asociación realizada entre la información de entrada y salida y la forma de representación de estas informaciones.*

Las neuronas se distribuyen en la red formando capas de un número determinado de elementos básico. Es decir, existe una capa de: *entrada* que recibe directamente la información proveniente de las fuentes externas de la red, capas *ocultas* que son internas a la red y no tienen contacto directo con el exterior (desde cero niveles hasta un número elevado), pudiendo estar interconectadas de distintas maneras, lo que determina junto a su número, las distintas topologías y una capa de *salida* que transfiere la información de la red hacia el exterior. La topología de las redes neuronales es la forma de organización de las neuronas en la red formando capas o agrupaciones de neuronas más ó menos alejadas de la entrada y la salida de la red. Por lo tanto, los parámetros fundamentales de la red serán: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones ente neuronas.

### 3.1.2. El algoritmo *backpropagation*

Rumelhart, Hinton y Williams [20] desarrollaron un método de aprendizaje automático que permitió que una red neuronal basada en el *perceptron* [21] aprendiera la asociación existente entre los patrones de entrada y las salidas correspondientes. El *Aprendizaje Automático (Machine Learning)* se puede definir como un conjunto de programas computacionales que mejoran con la experiencia y son sistemas que deben ser capaces de aprender a reconocer patrones a partir de la resolución de problemas o mediante ejemplos provistos por un instructor ó supervisor. Luego, se modificó el *perceptron* de Rosenblatt [21] agregándole capas ocultas, con conexión hacia adelante y sin conexiones recurrentes [20]. Pero, no fue suficiente con introducir algunas modificaciones topológicas a la red, sino que se requerían modificaciones en el algoritmo de aprendizaje; por lo tanto fue desarrollado el método de aprendizaje supervisado basado en la regla *delta generalizada* [20], logrando así, una ampliación del rango de aplicación de las redes neuronales.

El funcionamiento general de una red neuronal de este tipo, se puede dividir en dos partes: *una etapa de entrenamiento y una etapa de puesta en marcha*. La primera consiste en el aprendizaje de un conjunto predefinido de observaciones de entrada–salida dados como ejemplo (utilizando  $n$  atributos de entrada y un único atributo ó clase, de salida), empleando un ciclo propagación–adaptación de dos fases [17,22], donde en la *primera fase* se aplican los atributos de entrada a la capa de entrada de datos a la red y los valores generados se propagan desde esta capa hacia las superiores hasta generar una salida, en la capa de salida de la red. Para realizar el entrenamiento, se compara el resultado obtenido en cada neurona de salida con el valor deseado para cada neurona en particular y obteniéndose un error para cada una de las unidades de salida. En la *segunda fase*, los errores de las unidades de salida se transmiten hacia atrás, pasando por todas las neuronas de las capas intermedias que contribuyan directamente a la salida, recibiendo el porcentaje de error aproximado a la participación de las neuronas intermedias en la salida original. Este proceso se repite capa por capa hasta llegar a la capa de entrada y hasta que cada neurona haya recibido un error que describa su aporte al error total. Debido a ello, el algoritmo se denomina también de *retro-propagación* o propagación hacia atrás. Los errores se calculan con respecto a los aportes de las neuronas desde la capa de salida hasta la capa de entrada y se reajustan los pesos de las conexiones entre cada par de neuronas en la red, de manera tal que el error total cometido para ese patrón disminuya con respecto al valor del error recibido.

Dado que la fase de funcionamiento es similar a otras redes neuronales artificiales se debe realizar un análisis más profundo del método de aprendizaje. El método de *backpropagation* utiliza una función o superficie de error asociada a la red, buscando un estado de mínimo error estable a través del camino descendente sobre la superficie de error [20]. Es por esto que se debe realizar la retroalimentación para realizar las modificaciones en los pesos iniciales en un valor proporcional al gradiente decreciente de dicha función de error. En la Figura 21 se puede ver un esquema de este tipo de redes neuronales artificiales.

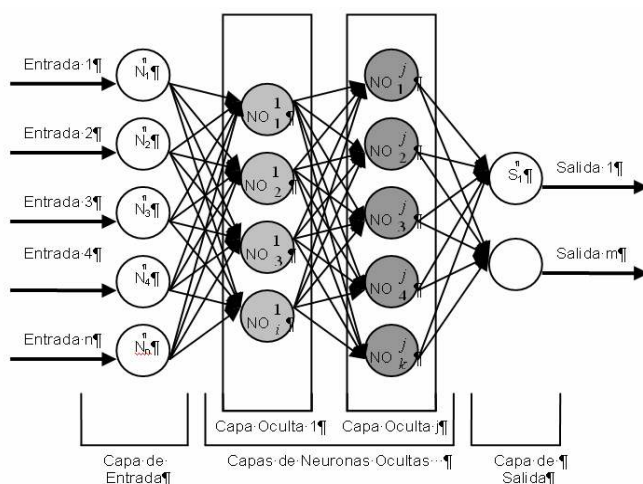


Figura 2: Modelo de red neuronal backpropagation [20].

La función de activación utilizada en cada una de las neuronas debe ser derivable de primer orden. Para obtener una aproximación funcional óptima se deben elegir cuidadosamente las variables a emplear, es decir se trata de incluir en el modelo las variables que realmente predigan la variable dependiente o de salida, pero que no covaríen entre sí [23], debido a que podrían provocar un *sobreajuste* (*overfitting*) innecesario. Esto sucede cuando el número de parámetros o de pesos de la red resulta excesivo en relación al problema a tratar y al número de patrones de entrenamiento disponibles. El *sobreajuste* disminuye la capacidad de la red de proporcionar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento. Se entiende por generalización de la red a la capacidad de dar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento [17].

### 3.2. La autoevaluación

En las evaluaciones parciales y finales *es donde aparecen las fallas documentadas a través de los errores que son registrados. Si bien su valor como indicadores de aprendizajes es relativo pues hay que considerar que la misma, es una situación de alto estrés donde el alumno esta más propenso a cometer fallas, es importante su recurrencia.*

Se construyó un registro de los errores cometidos en las evaluaciones parciales y finales que se plasmó en un listado, donde se observan los diferentes tipos de fallas cometidas en forma reiterada por los estudiantes. Ante sus fallas en las etapas iniciales del curso, el estudiante pierde más fácilmente la motivación y desiste de seguir cursando. Los errores citados son: a) *No leen detenidamente el problema a resolver, a veces no saben leer e interpretar el enunciado e interpretan lo que ellos quieren interpretar.* b) *Al principio confunden Read con Write* c) *No entienden para qué definir tipos y los definen mal* d) *Confunden los signos > y <* e) *Confunden variables y constantes* f) *No definen constantes* g) *Confunden filas y columnas con lo que se les dificulta el trabajo con vectores y matrices.* h) *Manejan mal los subíndices.* i) *No interpretan las estructuras cíclicas ni sus diferencias* j) *Usan if anidados en lugar de and* k) *No inicializan variables y contadores.* l) *No entienden cómo establecer sangrías por bloques y lo hacen mal* m) *No diferencian división real y entera* n) *No discriminan variables dependientes e independientes.* ñ) *No validan los datos de entrada a los programas* o) *No documentan los programas* p) *No verifican la división por cero, de haber un cociente* q) *Construyen ciclos infinitos*

A su vez, estos errores se pueden agrupar más globalmente de acuerdo a la clasificación siguiente: a) *Errores debidos a la mala interpretación del problema a resolver.* b) *Errores diversos en el uso del lenguaje* c) *Errores debido a procesos de decisión mal elaborados.*

La dificultad en la comprensión de los procesos de enseñanza y de aprendizaje se debe en parte a la multiplicidad de variables involucradas en ambos. Dentro de estos procesos, la evaluación da cuenta

En la etapa de aprendizaje, se busca minimizar el error entre la salida obtenida por la red y la salida deseada luego del entrenamiento con el conjunto de datos patrones. Es por ello, que en estas redes el aprendizaje es de tipo supervisado, ya que es el usuario o supervisor quien determina cuál es la salida deseada ante la presentación de un patrón de entrada dado

Para modificar los pesos se sigue la fundamentación matemática del algoritmo basado la técnica del *gradiente decreciente* [20].

de los errores cometidos en forma reiterada y representa uno de los aspectos de más trascendencia en el acto didáctico. Es frecuente, observar los diferentes tipos de errores cometidos por los estudiantes en las evaluaciones, que van desde omisiones leves debidas a la falta de atención o debidos al nivel estrés del proceso evaluativo, hasta errores conceptuales graves que dan lugar a la reprobación. La literatura existente determina el proceso de evaluación desde los aspectos cuantitativos de los errores, y aún en aquellos casos en que se realiza una consideración cualitativa, esta se lleva a cabo con un criterio globalizador [26].

Uno de los núcleos conceptuales que conforman la columna vertebral de la didáctica es la evaluación. Su importancia y necesidad ha sido puesta de manifiesto por numerosos autores e investigadores de la educación como Hernández Rojas [27] y House [28]. Angulo y Blanco [29] dicen al respecto: “*la evaluación es el proceso por el cual conocemos y valoramos la calidad del servicio y el papel de los distintos componentes en el mismo*”; significa entonces, que evaluar implica comprender qué le sucede al alumno y a todos los integrantes implicados en el proceso de enseñanza y aprendizaje considerado.

Como el *estudiante es el principal protagonista* y destinatario del proceso de enseñanza y aprendizaje, de algún modo se lo puede hacer partícipe de su evaluación a través de la *autoevaluación*. Blanco Prieto [30] dice: “*la autoevaluación por parte del alumno supone una concepción democrática y formativa del proceso educativo*” y Gimeno Sacristán [31], al respecto, enumera, entre otras, una serie de razones que justifican la necesidad de la *autoevaluación*: a) una persona formada adecuadamente es aquella capaz de dirigir su propio aprendizaje; por lo tanto habrá que preparar al estudiante para que participe en su evaluación, b) cuando un estudiante es capaz de corregirse en un determinado contenido, se puede decir que domina ese contenido, c) con la autoevaluación los alumnos se sentirán tratados como adultos responsables y d) la importancia de este tipo de proceso evaluativo radica en que es una actividad que mejora los aprendizajes de los alumnos, que en definitiva, es el fin que se persigue.

Retomando la idea de que el estudiante debe ser partícipe de su proceso de aprendizaje, y adscribiendo a la justificación de Gimeno Sacristán [31] es posible fundamentar la autoevaluación en el *aprendizaje autorregulado*. Puede decirse que existe aprendizaje autorregulado, *cuando el alumno es capaz de ser un participante activo de su proceso desde el punto de vista metacognitivo, conductual y motivacional*. [32,33]. Esto implica que él debe lograr una *toma de conciencia* de sus propias dificultades para establecer estrategias de acción (aspecto *metacognitivo*) y por otra parte debe ser capaz de controlar su conducta para alcanzar sus metas (aspecto *conductual* y *motivacional*).

#### 4. OBJETIVOS DEL TRABAJO

El objetivo general se desglosó en tres objetivos específicos:

- Determinar el tipo de red neuronal a usar para determinar en forma más eficiente la asesoría luego del diagnóstico realizado.
- Aplicar redes neuronales para predecir errores en futuros exámenes y a partir de éstos determinar que clase de problemas o ejercicios debe resolver el alumno para mejorar su rendimiento, en un dado dominio.
- Evaluar el sistema utilizando datos estadísticos almacenados y con un grupo de docentes y de estudiantes disponibles.
- Proponer y poner a prueba una herramienta que permite llevar a cabo la autoevaluación.

#### 5. METODOLOGÍA

- 1) Se determinó la herramienta inteligente más apropiada para resolver la cuestión de la predicción en el comportamiento del alumno y la determinación de la actividad futura que deberá realizar, entre las disponibles en un dado dominio.

- 2) Se creó una base de datos con todos los ejercicios tomados en las evaluaciones parciales y finales de una Programación Básica, indicando su tipo (ya sea de tipo práctico o teórico), su dificultad, su nivel de obligatoriedad en una evaluación, su puntaje y su objetivo de aprendizaje. El tamaño de la muestra fue de 450 alumnos, usándose 2/3 para entrenamiento y 1/3 para pruebas.
- 3) Se elaboraron las bases para la creación de un sistema predictor que cumpla con los objetivos propuestos. Para este punto se determinarán las herramientas metodológicas más apropiadas que provee la ingeniería de software y los sistemas inteligentes en cuanto a métodos, técnicas y herramientas disponibles.
- 4) Se realizó una prueba piloto (evaluación interna y externa del software) con un grupo de estudiantes y docentes de la cátedra disponibles.
- 5) Se diseñó el sistema evaluador desde la propia autoevaluación y se probó con alumnos disponibles de cursos de Programación Básica

## 6. PARTE EXPERIMENTAL APLICANDO LA RED NEURONAL

### 6.1. Descripción

Las etapas seguidas en la parte experimental fueron las que se describen en la Tabla 1. Se utilizó la herramienta *NNclass* [25]. La misma posee un instructivo que permite efectuar el ingreso de los datos. El valor máximo es de 50 variables de entrada y 40 de ellas pueden de tipo categóricas.

Please make sure that there are **no more than 50 neurons** in Input Layer.  
There should be **exactly 1 Output variable** - application will treat it as Categorical.  
There should be **no more than 40** Categorical Variables.

Omit	Cat	Omit	Omit	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat			
Errores	Sig	Carrera	Padron	Sezo	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17		
9	84585	M	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	
10	84722	M	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0
10	84653	M	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0
9	79600	M	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
10	84078	M	0	0	1	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	1	0
10	84134	M	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	84870	M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	84520	M	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	85032	F	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10	85413	M	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10	84962	F	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
9	84926	M	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	84734	M	1	1	1	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0
10	84091	M	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
10	82557	M	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0
10	84690	M	0	0	1	1	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	1
10	84087	M	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	84587	M	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
10	84371	M	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Figura 3: Pantalla ingreso de datos codificados.

#### 6.1.1. El ingreso de los datos.

Se observa que existen tres tipos de datos que se ingresaron en forma codificada:

- *Catagóricos*: son las palabras, que no son la salida esperada de la red.
- *Continuos*: son datos numéricos que representan números reales.
- *De Salida (Output)*: son los valores que se desea que la red prediga, los que a su vez pueden ser catagóricos ó continuos.

Cat	Cat	Cat	Cat	Cat	Cat	Predicted Out	Score (aprobado)	Score (desaprobado)
E16	E17	E18	E19	E20	E21			
0	1	0	0	10	0	aprobado	0,970997572	0,027049135
0	0	10	0	0	1	desaprobado	0,001156448	0,998752892
0	0	0	0	0	1	desaprobado	0,005972461	0,994004071
100	0	0	0	11	1	aprobado	0,979461491	0,019286392
0	0	10	0	101	101	aprobado	0,999360442	0,000711223
1	0	1	0	1	1	aprobado	0,944856524	0,058509205
1	0	0	0	1	0	aprobado	0,999345362	0,000726172
0	0	1	0	0	0	aprobado	0,544088364	0,455251902
1	11	1	0	1	0	aprobado	0,983933151	0,01474512
0	0	0	0	1	1	desaprobado	0,006141305	0,993789971
0	0	0	0	0	1	aprobado	0,680459857	0,331099778
0	0	0	0	0	0	aprobado	0,779129326	0,220500425
1	0	0	0	11	11	aprobado	0,999358594	0,000712385
1	0	0	0	1	1	aprobado	0,987725079	0,011409881
0	0	0	0	0	0	aprobado	0,994639397	0,005598276
0	0	0	0	1	1	aprobado	0,945939481	0,057353366
0	0	0	0	1	1	desaprobado	0,171479136	0,828849435
0	0	0	0	0	1	desaprobado	0,000927733	0,999024689
0	0	0	0	1	1	aprobado	0,954882205	0,047749929
0	0	0	0	10	1	aprobado	0,99622333	0,003323361
0	0	1	0	0	1	desaprobado	0,030887676	0,968400836

Figura 4: Pantalla predicción evaluación final



En los casos que se muestra en la Figuras 3 y 4, se muestra la salida que es de tipo categórica ya que será: *aprobado* ó *desaprobado*.

	Entrada	Acción	Salida
1	Datos de los errores de los alumnos de la materia en exámenes	Confección de base de datos de errores de cometidos	Errores codificados
2	Datos de entrenamiento	Entrenamiento de la red	Parámetros con error mínimo. Red entrenada
3	Datos de prueba	Aplicación de la red	Pronóstico
4	Errores codificados	Aplicación de la red	Diagnóstico

Tabla 1: Etapas y acciones seguidas

Predicted Output	Score (aprobado)	Score (desaprobado)
Aprobado	0,970997572	0,027049135
Desaprobado	0,001156448	0,998752892

Tabla 2: Ejemplo de Predicción Aprobado–Desaprobado

000	no se equivocó nunca
001	se equivocó en el parcial
010	se equivocó en el primer recuperatorio
100	se equivocó en el segundo recuperatorio

Tabla 3: Codificación de las Instancias de Aprobación

Por lo tanto, la red utilizará dos neuronas para ello; una cuya salida, a través de un número real, representa a los *aprobados* y otra, que también devuelve un número real, cuya salida está asociada a los *desaprobados*. En la primera fila del ejemplo de la Tabla 2 se observa que la para la salida de 0.970 la neurona está asociada a "*aprobado*" y sólo 0.027 la asociada a "*desaprobado*", por lo que la red da como resultado el primero de éstos. En la segunda fila se presenta el caso inverso. Los datos de las evaluaciones de los estudiantes (a través de 6 instancias de aprobación: un parcial con dos recuperatorios y tres oportunidades para el examen final) han sido codificados como se observa en la Tabla 3. El resto son combinaciones de las opciones presentadas. Con respecto a las columnas, E1 a E21 de la Figura 3, son los tipos de errores detectados en cada una de las evaluaciones agrupados luego del análisis realizado.

### 6.1.2. La selección de los parámetros de entrenamiento de la red.

En la Figura 5 se presentan los parámetros característicos de la red obtenidos. Se han explorado otras arquitecturas, con otras combinaciones de parámetros pero se obtuvieron errores mayores al 5%.

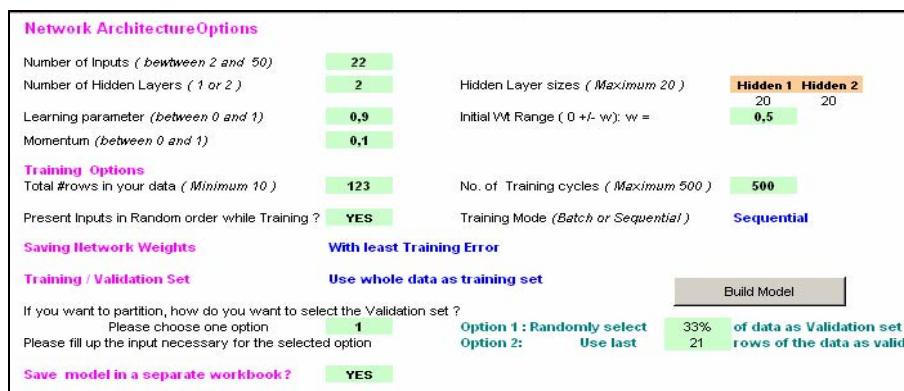


Figura 5: Características de la red

### 6.1.3. La predicción de la red

Cuando un estudiante se equivoca o no en alguno de los temas, aunque no haya completado aún todo el curso, es decir, cuando solo rindió la evaluación parcial y no accedió aún a la evaluación recuperatoria, por ejemplo, se puede predecir si aprobará o no en la instancia próxima. La idea es utilizar esta red como primer paso para el uso de una serie de dos redes. Cuando la primera red prediga que no aprobará, una segunda red del mismo tipo, basada en los errores cometidos, le puede indicar cuáles son los temas que debe estudiar, a fin de recomendarle los ejercicios por núcleos temáticos. Para este primer entrenamiento predictivo la red operó con un error del 4%, lo que es más que aceptable para el trabajo con grupos humanos [9].



### 6.1.4. Resultados de aplicación de la red

En la Figura 4 se observan las características de la red usada. Cuando el error es inaceptable, o sea mayor al 5%, se debe descartar la red y comenzar el proceso de entrenamiento nuevamente variando los distintos parámetros que la definen, entre los que se pueden citar el parámetro de aprendizaje  $\alpha$  (el valor inicial y el régimen de modificación a lo largo de los ciclos, el momentum  $\beta$ , los pesos aleatorios iniciales, el vecindario gaussiano, la cantidad de atributos que se utilizarán para el entrenamiento, la cantidad de las observaciones que se utilizarán y las que se descartarán, justificando por qué se descartarán. El tiempo de entrenamiento de una red neuronal para los valores utilizados es de aproximadamente 1.5 horas en promedio. Se entrenaron redes con más de 600 variaciones en los parámetros hasta encontrar los más adecuados y los valores mínimos correspondientes al error.

## 7. UNA APLICACIÓN PARA ASISTIR A LOS ESTUDIANTES EN SUS AUTOEVALUACIONES

### 7.2.1. Descripción

A fin de ayudar en el proceso de reconceptualización de los errores, se elaboró una aplicación basada en la *web* que está disponible en los servidores de la universidad para que los estudiantes puedan efectuar sus autoevaluaciones. Esta herramienta cuyas pantallas se pueden ver en las Figuras 6 a 8), les permite evidenciar sus fallas a fin de que puedan tomar conciencia de sus limitaciones [34]. La autoevaluación está dirigida a estudiantes del área de Programación Básica. Como se cuenta desde el año 2000 con una base de datos de los errores más comunes de los estudiantes en sus evaluaciones, se pudieron construir series de preguntas con base en ellos donde cada grupo de preguntas se corresponde a cada Unidad Didáctica trabajada. Con esta instrumentación se buscó que el alumno pudiera darse cuenta de sus debilidades a fin de reforzarlas antes de su evaluación parcial o final. (En el ANEXO se presentan los diferentes tipos de preguntas, con sus respuestas y con las observaciones de cada caso).

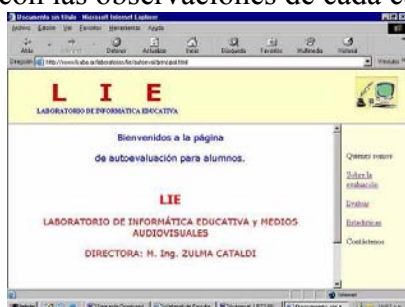


Figura 6: Pantalla Principal



Figura 7: Los contenidos.

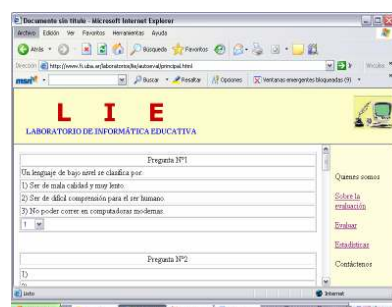


Figura 8: La autoevaluación

La aplicación *Autoeval*, es un programa muy simple que permite a los alumnos autoevaluar sus conocimientos mientras los profesores pueden ver los datos estadísticos en cualquier momento para tener un panorama general respecto del nivel del curso que está interactuando. El software está basado en formularios HTML, *scripts* PHP y una pequeña base de datos MySQL, utilizando de esta manera herramientas de muy bajo costo y sin problemas legales. Básicamente, se muestra a los alumnos un formulario HTML con  $n$  preguntas de tipo *múltiple choice*, dándoles 3 opciones a elegir, más la opción de no contestar. Las preguntas han sido formuladas considerando los datos estadísticos acerca de los errores más frecuentes de los alumnos en las evaluaciones parciales y finales. Al finalizar su autoevaluación, el alumno realiza un "submit", llamando así a un *script* que corrige el examen mostrándole al estudiante la cantidad de respuestas correctas, y las opciones correctas de los puntos que haya contestado erróneamente. El programa<sup>3</sup> consiste de las siguientes

<sup>3</sup> *El modelo de datos*: En las cátedras donde se utilizará el programa se han desarrollado los contenidos en 10 Unidades Didácticas (o Capítulos), por lo cual se podría haber optado por usar 10 tablas en la base de datos, donde cada una almacenara los ítems del capítulo que le correspondiera, así, sería un modelo muy intuitivo y ordenado, sin embargo, para independizar el programa de la estructura de datos se optó por tener una única tabla de preguntas con la estructura de la Tabla 1, donde sabiendo que *Chapter* contiene el número de capítulo deseado, en *#1* se genera la consulta que se

etapas: a) creación de registros (carga de las preguntas de examen), b) presentación de examen, c) corrección y d) presentación de estadísticas [34].

NUM	Int unsigned auto_increment <b>Primary key</b>
<b>Question</b>	Text
<b>Ans1</b>	Text
<b>Ans2</b>	Text
<b>Ans3</b>	Text
<b>Chapter</b>	Char(5)
<b>correct</b>	Char(1)

**Tabla 1:** Estructura de la base de Datos<sup>4</sup>

```
#1 $query = "select * from questions where
chapter like '%".$chapter."'";
#2 $result= mysql_query($query);
#3 $num_rows= mysql_num_rows($result);
#4 $stop= rand(1,$num_rows);
#5 for ($i=0; $i< $stop ; $i++)
{
    $row= mysql_fetch_array($result); }
#6 return $row;
```

**Tabla 2:** Selección de las preguntas

De esta manera el programa seleccionará para cada Unidad Didáctica una pregunta como se ve en la Tabla 2. En cuanto a las *estadísticas* el programa muestra la cantidad de exámenes corregidos, la cantidad de exámenes con más de 6 respuestas correctas, la cantidad de respuestas incorrectas por cada Unidad Didáctica y su respuesta correcta.

### 7.2.2. Resultados

La aplicación se probó con alumnos voluntarios desde 2005. A partir de esta primera exploración se ha visto que los alumnos fueron capaces de elaborar preguntas e inquietudes con mayor frecuencia y relevancia. Por otra parte, se convirtieron en agentes de apoyo a sus compañeros ya que pudieron ayudarlos en los inconvenientes de menor envergadura que ellos fueron capaces de reelaborar merced a su autoevaluación. Los docentes lograron modelizar algunos esquemas de los errores que se presentaron con mayor frecuencia. A partir de ahí, se observó que los alumnos trabajaron de manera más autónoma, la temática de las preguntas cambió desplazándose de los temas instrumentales como el uso del lenguaje hacia temas que tenían que ver con la manera de optimizar los algoritmos que ellos creaban. Por otra parte, el índice de aprobación ha mejorado sensiblemente.

## 8. CONCLUSIONES Y TRABAJOS FUTUROS

Se trabajó en el sistema de predicción donde a partir del rendimiento obtenido en las evaluaciones parciales se puede predecir cómo será la *“performance”* en futuras evaluaciones. Los datos obtenidos permitirán diseñar un sistema recomendador de modo que el alumno ante la situación de ser examinado pueda acceder al sistema quien le sugerirá una serie de ejercicios y problemas para que pueda internalizar los errores que fueron clave para la reprobación de las evaluaciones previas. Se busca brindar a los estudiantes una herramienta para que puedan tomar conciencia de sus propios errores y no cometer las mismas fallas en las evaluaciones finales. Este objetivo está cumplido, pero se ha visto que el impacto podría ser de mayor envergadura con el tiempo si se consideran los resultados que se presentaron. Quedó establecido que, indudablemente, la *autoevaluación* resulta un acercamiento, entre aquellos que son posibles, hacia la autonomía del alumno y la mejora del proceso de aprendizaje.

Como líneas futuras de investigación se prevé trabajar en el diseño del sistema recomendador de ejercicios y problemas de modo que el estudiante pueda adquirir una cierta autonomía en la preparación de sus exámenes finales. Como trabajos posteriores se propone: a) Ampliar los contenidos disponibles para la autoevaluación, b) Escalar el sistema informático para que se pueda realizar un seguimiento del alumno, de esta forma el docente puede tener una *clusterización* de las necesidades cognitivas de su clase. c) Escalar el sistema informático de tal forma que evolucione hacia bases de datos e interfaces capaces de interactuar con el alumno de manera autónoma y d)

quiere realizar en un string, en #2 se ejecuta la consulta, y se devuelve en \$result todos los registros donde el campo capítulo sea el buscado (\$chapter). En #3 se calcula la cantidad de registros obtenidos en #2 (\$num\_rows), para luego generar un número aleatorio entre 1 y \$num\_rows y en #5 acceder al registro que posee ese número y finalmente devolverlo al programa principal en #6.

<sup>4</sup> “num” es la clave única identificatoria del registro, “question” es la pregunta en sí, “ans1” “ans2” y “ans3” son las respuestas posibles a “question”, “chapter” es el número de capítulo (unidad didáctica) en cuestión y finalmente “correct” es el número de respuesta correcta (1, 2 ó 3).

Integrar los hallazgos en el módulo de evaluación para un Sistema Tutor Inteligente cuya arquitectura se está desarrollando.

## 9. AGRADECIMIENTOS

Esta comunicación forma parte de los proyectos de investigación: *Sistemas inteligentes aplicados a la predicción del comportamiento de los estudiantes y diagnóstico* 2005-2006 LIE-DC/04-07 del Laboratorio de Informática Educativa y Medios Audiovisuales (LIEMA) de la Facultad de Ingeniería, de la Universidad de Buenos Aires y *25/C099 Modelado del tutor basado en redes neuronales para un Sistema Tutor Inteligente*, de la Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional 2007-2008 convenio FI-UBA y UTN-FRBA. Los autores agradecen a los alumnos que participaron de la experiencia.

## 10. REFERENCIAS

- [1].Lage, F.; Cataldi, Z. y Denazis, J. M. (2000). *The Scripts of University Students and Experts in the Preparation of the Examinations: A Study in Process*. FIE 2000: 30<sup>th</sup> ASEE/IEEE Frontiers in Education Conference, Kansas City Missouri, 18-21 de octubre. Paper 1154. Proceedings en CD-ROM. ISBN 0-7803-6242/0/
- [2].Copello, G.; Cataldi, Z. y Lage, F. (1999). *La comprensión de los errores*. Proceedings del V Congreso Internacional de Ingeniería Informática. Páginas 210-217. Editado por Departamento de Publicaciones de la Facultad de Ingeniería.
- [3].Pozo, J. I. (1998). *Teorías cognitivas del aprendizaje*. Morata.
- [4].Pozo, J. I. (1999). *Aprendices y Maestros*. Alianza.
- [5].Ausubel, D.; Novak, J. y Hanessian, H. (1983) *Psicología educativa: un punto de vista cognitivo*. 2ª Ed. México: Trillas. 624p.
- [6].Ohlsson, S. (1996) *Learning from performance of errors*. Psychological Review 3 (2) p. 241-262.
- [7].Khuwaja, R.A. (1994) A Model of Tutoring: Facilitating Knowledge Integration Using Multiple Models of the Domain. *Ph.D., Illinois Institute of Technology*
- [8].Giraffa, L.M.M.; Nunes, M. A.; Viccari, R.M. (1997) *Multi-Ecological: an Learning Environment using Multi-Agent architecture*. MASTA'97: Multi-Agent System: Theory and Applications. Proceedings. Coimbra: DE-Universidade de Coimbra.
- [9].Cataldi, Z. 2005. *Sistemas tutores inteligentes: los estilos del estudiante para selección del tutorizado*. WICC 2005. 13 y 14 de mayo. Universidad Nacional de Río Cuarto. Córdoba. RED UNCI
- [10].Salgueiro, F. A, Costa, G., Cataldi, Z., García Martínez, R. y Lage, F. J. 2005. *Sistemas inteligentes para el modelado del tutor*. GCETE'2005, Global Congress on Eng. and Technology Education. marzo 13-15
- [11].Abbas, H. (1998) Designing a New Domain Knowledge Base for an Intelligent Tutoring System. *Ph.D., Illinois Institute of Technology*.
- [12].Brachman, R.J. (1988) *The basis of knowledge representation and reasoning*. AT&T. Technical Jurnal. 67, 1:15.
- [13].Brachman; R.J. (1985) On the epistemological status of semantic networks. En Brachman, R. y Levesque, H. (Eds.) *Readings in knowledge representation* (191-215). Los Altos. Morgan Kaufman Pub. Inc.
- [14].Viccari, R. M. (1993). *Inteligência Artificial e Educação: Indagações Básicas*. IV Simpósio Brasileiro de Informática e Educação.
- [15].Viccari, R.M. y Giraffa, L.M. (1996). *Sistemas Tutores Inteligentes: Abordagem Tradicional x Abordagem de Agentes*. XIII Simpósio Brasileiro de Inteligência Artificial, Curitiba.
- [16].Casas, M. (1999) *contribuições para a modelagem de um ambiente inteligente de educação baseado em realidade virtual*. Tesis Doctoral Univ. de Federal de Sta. Catarina. Pós-graduação em Eng. de Produção.
- [17].del Brío, B. M. y Sanz Molina, A. (2001) *Redes neuronales y sistemas difusos*. Paraninfo.
- [18].Kohonen, T. (1988) *Self-Organizing Maps* Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, NY(pp 236)
- [19].Hilera González; R. y Martínez Hernando, A. (2000) *Redes Neuronales Artificiales: Fundamentos, modelos y aplicaciones*. Ra-ma, Madrid.
- [20].Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. (1986). *Learning internal representations by back-propagating errors in Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Eds. Cambridge, MA: MIT Press, vol. 1, p. 318-362.

- [21].Rosenblatt, F. (1958), *The perceptron: A probabilistic model for information storage and organization in the brain*. Psychological Review, 65, 386-408.
- [22].Palmer, A., Montaña, J.J. y Jiménez, R. (2001) *Tutorial sobre Redes Neuronales Artificiales: El Perceptrón Multicapa*. Revista Electrónica de Psicología Vol. 5, No. 2, Julio ISSN 1137-8492.
- [23].Smith, M. (1993). *Neural networks for statistical modeling*. New York: Van Nostrand Reinhold.
- [24].Masters, T. (1993). *Practical neural networks recipes in C++*. London: Academic Press
- [25].Saha, A. (1998) *Application of Ridge Regression for Improved Estimation of Parameters in Compartmental Models*; Tesis Doctoral. Departamento de Estadística.
- [26].Ausubel D., Novak J. y Hanesian H.(1993, 1997). *Psicología educativa. Un punto de vista cognitivo*. Trillas. Décima impresión
- [27].Hernández Rojas, G. (1998) *Paradigmas en Psicología de la educación*. Paidós. México
- [28].House, E. R. (1997) *Evaluación, ética y poder*, Ed. Morata: Madrid
- [29].Angulo, J. F. y Blanco, L. (coordinadores) (1994). *Teoría y desarrollo del currículum*". Ediciones Aljibe. Málaga.
- [30].Blanco Prieto, F. (1994). *La evaluación en la Educación Secundaria*. Amarú. Salamanca.
- [31].Gimeno Sacristán, J. (1981). *Teoría de la enseñanza y desarrollo del currículo*. Anaya. Madrid.
- [32].Zimmerman, B. J. (1989) A social cognitive view of self-regulated academic learning. *Journal of Educational Psychology*, 81(3), 329-339.
- [33].Zimmerman, B. J. (1994) Dimensions of academic self-regulation: a conceptual framework for education. En D. H. Schunk y B. J. Zimmerman (Eds.), *Self-regulation of learning and performance. Issues and educational applications*. Hillsdale, NJ: Erlbaum.
- [34]. Cataldi, Z.; Méndez, P. y Lage, F: 2006. *La autoevaluación en Programación Básica en carreras no informáticas desde la perspectiva constructivista del aprendizaje autorregulado*. JENUI 2006. XII Jornadas de Enseñanza Universitaria de la Informática. Bilbao, 12 a 14 de julio. Thomson. 279-286.

#### ANEXO: Algunas preguntas del programa

La respuesta correcta es a.

**Unidad Didáctica 3:** *Estructuras secuenciales*. Indique cuál es la respuesta correcta

Se tiene el precio y nombre de un producto en una variable de tipo real (precio) y en una variable de tipo string (s\_nombre) respectivamente. Se desea mostrar en pantalla el nombre y el valor de dicho producto. Seleccione la manera óptima de hacerlo.

- Writeln('El precio de ', s\_nombre, ' es de \$', precio);
- Write('El precio de ');  
Write(s\_nombre);  
Write(' es de ');  
Writeln(s\_precio);
- Writeln('El precio de s\_nombre es de \$precio');

**Unidad Didáctica 6:** *Subprogramas*. Indique cuál es la respuesta correcta. Se tiene el siguiente procedimiento

```
procedure sumados(a: integer);
begin
a:= a+2;
end;
```

Si se lo llama desde el programa principal siendo en él a=8, cuál será el valor de a luego de ejecutarse el procedimiento?

- 8
- 10
- 0

**El resultado correcto es a.**

*Nota:* El estudiante debe darse cuenta que el parámetro es pasado por valor y no por referencia.