



# Tecnologias da Informação em Educação

## Aplicación de la investigación social a la evaluación y su relación con la taxonomía de Bloom.

**Isidoro Hernán Losada**

Universidad Rey Juan Carlos, Spain

isidoro.hernan@urjc.es

**Jesús Ángel Velázquez Iturbide**

Universidad Rey Juan Carlos, Spain

angel.velazquez@urjc.es

### Resumen

En este artículo se esboza una técnica extraída de los métodos de investigación social que permite desgranar el concepto en estudio, de forma que podamos dividirlo en las distintas dimensiones, o lo que es lo mismo distinguir diferentes aspectos del mismo. Cada dimensión tendrá asociados una serie de indicadores para medirla. A su vez, las medidas de los indicadores se pueden agrupar adecuadamente para darnos un índice (lo que el alumno sabe). Este método es especialmente útil a la hora de fijar objetivos pedagógicos usando como marco de referencia la taxonomía de Bloom. Se ejemplifica su aplicación a un concepto concreto: la herencia simple en POO. De esta forma podemos dar solución a la pregunta: ¿Cómo podemos medir lo que un alumno ha aprendido o conoce sobre la herencia en POO?

**Palabras clave:** Evaluación; Programación; Bloom; Herencia.

### Resumo

Neste artigo esboça-se uma técnica retirada dos métodos de investigação social que permite isolar o conceito em estudo, de forma a que possamos dividi-lo nas suas diferentes dimensões, ou mesmo distinguir os seus diferentes aspetos. Cada dimensão terá associados uma série de indicadores para a medir. Por sua vez, as medidas dos indicadores podem-se agrupar adequadamente para obtermos um índice (o que o aluno sabe). Este método é especialmente útil no momento de se



fixarem objetivos pedagógicos utilizando como referência a taxonomia de Bloom. Exemplifica-se a sua aplicação a um conceito concreto: a herança simples em POO. Desta forma podemos dar solução à pergunta: Como podemos medir o que um aluno aprendeu ou conhece sobre herança em POO?

**Palavras-chave:** Avaliação; Programação; Bloom; Herança em POO.

### **Abstract**

In this article we sketch a technique drawn from social research methods that allows isolating a concept under study, so that we can divide it into its different dimensions, or even distinguish its different aspects. Each dimension will have associated a series of indicators to measure it. Also, the measures of the indicators can be adequately grouped to obtain an index (what the student knows). This method is especially useful at the moment of fixating pedagogical objectives using Bloom's taxonomy as a reference. Its application to a concrete concept is exemplified: single inheritance in OOP. In this way we can answer the question: How can we measure what a student learnt or knows about inheritance in OOP?

**Key words:** Evaluation; Programming; Bloom; OOP Inheritance.



## 1. Introdução

La evaluación en el contexto de la educación es el proceso de caracterizar lo que el alumno sabe o conoce. Las razones para realizar estas evaluaciones son muy variadas, y van desde la necesidad de conocer informalmente el progreso del aprendizaje del alumno en un curso para adecuar la metodología, a la necesidad de caracterizar el grado de experiencia del discente en una materia para ponerle una nota.

Cuando un investigador o un profesor se propone realizar un estudio o una evaluación sobre la eficacia educativa de una metodología o de una herramienta informática para la educación, lo primero que se plantea es el diseño del experimento. Una vez decidido por el método a emplear para medir el rendimiento (incremental si se contrastan los conocimientos adquiridos antes y después de recibir una lección o de usar una herramienta pedagógica y final si se mide el nivel de conocimiento actual) el paso siguiente es la preparación de las cuestiones a las que se someterá al alumno para calibrar el grado de aprendizaje del mismo. Esta última etapa es crucial para la obtención de datos indicativos del nivel del discente. Si se diseñan mal las preguntas se obtendrán resultados erróneos y/o poco significativos. Si, por el contrario, se consiguen unas cuestiones relacionadas con lo que se intenta medir, se podrán obtener conclusiones o relaciones correctas, aunque, a veces, no sean las esperadas.

Tener un método con ejemplos de cómo diseñar correctamente preguntas para obtener el nivel del alumno puede ser útil al profesor o investigador a la hora de preparar una evaluación. Existen multitud de clasificaciones de evaluaciones basadas en la taxonomía de Bloom [Scoot 03, Lister 03], pero todas ellas están basadas en las apreciaciones de los autores. De hecho, es bien sabido que adjudicar una pregunta a un nivel determinado no depende de la pregunta en sí, sino del alumno al que se le presenta. Es decir, la misma pregunta puede situarse en distintos niveles dependiendo del grado de conocimiento previo del discente. En este artículo se trata de dotar de un mayor grado de objetividad a la realización de preguntas para evaluar. Para ello se propone un método y se ejemplifica con cuestiones centradas en el aprendizaje del concepto de herencia simple en programación orientada objetos usando el lenguaje Java para implementar dicho concepto.

Por otro lado, la evaluación del conocimiento de un alumno también es complicada. De forma habitual, se realizan pruebas que constan de varios ejercicios. A estos



problemas se les puntúa de forma homogénea, si son de dificultad parecida, o de forma heterogénea a criterio del profesor/evaluador. La suma (ponderada) de los puntos da una nota numérica que sirve como indicativo del grado de conocimiento del alumno en esa materia a la vez que puede servir para establecer una clasificación entre los discentes: los de mayor nota son mejores estudiantes que los que obtienen una nota inferior. La clave para obtener una buena nota indicativa, está en evaluar la dificultad. Para realizar esta tarea de forma lo más subjetiva posible se puede usar la taxonomía de Bloom. [Bloom 56].

El artículo está dividido en las siguientes secciones: en la siguiente sección se explica la taxonomía de Bloom como marco comúnmente usado para medir y evaluar el grado de aprendizaje. En la tercera sección se introduce el método tomado de la investigación social [Garcia 03] para descomponer un concepto general en sus dimensiones, variables e indicadores. En la cuarta sección se aplica este método a un ejemplo concreto, al concepto de herencia, descomponiéndolo en todas sus dimensiones [Budd 00]. En la quinta se hacen unas reflexiones sobre la relación del método con la taxonomía de Bloom, y en la última sección se dan las conclusiones y trabajos futuros.

## 2. Taxonomía de Bloom

La educación debe estructurarse en torno a cuatro aprendizajes fundamentales que serán los pilares del conocimiento de cada individuo. El primero es aprender a conocer (dominio cognitivo), el segundo, aprender a hacer (dominio psicomotor), el tercero es aprender a vivir juntos (dominio relacional/social) y el cuarto es aprender a ser (dominio actitudinal).

Si nos centramos en la enseñanza de la programación orientada a objetos, el factor más importante es el cognitivo. El trabajo en este dominio empezó en 1948 y se terminó en 1956. El resultado se publicó en un libro donde se define una taxonomía del aprendizaje conocida comúnmente como taxonomía de Bloom [Bloom 56]. Dicho estudio establece una jerarquía de seis niveles con grado creciente de aprendizaje del alumno (ver tabla 1). Cada nivel presupone la capacitación del alumno en los niveles precedentes. Según ascendemos por la jerarquía nos encontramos un mayor grado de aprendizaje.

Se ha demostrado la bondad de esta taxonomía salvo en los dos últimos niveles. Se debate si la síntesis es más compleja que la evaluación o viceversa, o si están al



mismo nivel y se diferencian en los procesos cognitivos usados. Algunos investigadores reducen esta jerarquía a cuatro niveles: conocimiento, comprensión, aplicación y el cuarto y último de pensamiento crítico o de resolución de problemas que engloba análisis, síntesis y evaluación de Bloom [Johnson 07]. Se ha aplicado a multitud de disciplinas y materias. En concreto, se ha aplicado al diseño de cursos de programación orientada a objetos [Bennedsen 04]

Anderson y Krathwohl revisaron la taxonomía para refinarla y dividirla en dos dimensiones [Anderson 01]: la dimensión del conocimiento, basada en el nombre de la materia que se pretende enseñar y la dimensión del proceso cognitivo, fundamentada en el verbo o acción que se quiere conseguir enseñar. Así en esta categoría el primer nivel se renombra y se denomina recordar, y los dos últimos niveles se cambian de orden, pasando a ser el quinto evaluar y el sexto y último crear.

<b>Nivel de la taxonomía</b>	<b>Descripción</b>
<b>1 ó de conocimiento (o de recuerdo)</b>	El alumno es capaz de reconocer o recordar información sin que sea necesario ninguna comprensión o razonamiento sobre lo que hay tras dicha información.
<b>2 ó de comprensión.</b>	El alumno es capaz de entender y explicar el significado de la información recibida.
<b>3 ó de aplicación.</b>	El estudiante es capaz de seleccionar y usar datos y métodos para resolver una nueva tarea o un problema.
<b>4 ó de análisis</b>	El alumno es capaz de distinguir, clasificar y relacionar hipótesis y evidencias de la información dada, así como descomponer un problema en sus partes.
<b>5 ó de síntesis.</b>	El estudiante es capaz de generalizar ideas y de integrarlas para resolver o realizar algún problema que es nuevo para él.
<b>6 ó de evaluación</b>	El alumno está capacitado para comparar, criticar y evaluar métodos o soluciones para resolver un problema o para discernir la mejor entre varias soluciones.

Tabla 1. Taxonomía de Bloom



### 3. Dimensiones, indicadores e índices

Las dimensiones de un concepto son los distintos aspectos en que puede ser considerado el mismo. Las dimensiones que consideremos deben ser mensurables de forma que por medio de medidas (indicadores) se haga operativo el concepto. Generalmente hay una pérdida de información al descomponer un todo en sus partes y analizar esas partes por separado. Por ello hay que operar de forma que las medidas (indicadores) de las dimensiones reflejen lo más ajustadamente posible el concepto del que partimos [Carmona 77].

Un indicador es algo que da señal o cuenta de algo concretándolo. Se puede refinar esta definición diciendo que un indicador es la medida estadística de un concepto o de una dimensión de un concepto o de una parte de aquélla, basado en un análisis teórico previo e integrado en un sistema coherente de medidas semejantes, que sirve para describir el concepto es estudio [Carmona 77]. Los indicadores deben tener dos propiedades fundamentales: estar relacionados con el concepto o dimensión del que tratan de ser indicación y ser expresión numérica, cuantitativa, de la dimensión que reflejan. Existen distintos tipos de indicadores [De Miguel 67], de los que tomaremos únicamente los descriptivos, que son aquellos que tratan de poner de manifiesto o explicar la posible regularidad existente en un conjunto de datos.

Un índice es una medida obtenida por la agrupación adecuada de varios indicadores. Si los indicadores que se usan pertenecen todos a una misma dimensión, ese índice representará numéricamente la dimensión medida (ver Figura 1,  $i_1$ ). Pueden existir índices de indicadores de varias dimensiones [García 03], y este índice representará al conjunto de dimensiones ( $i_4$ ). Un índice también puede representar al concepto en estudio ( $i_c$ ). En este artículo se aplican estos conceptos al estudio de la herencia simple en programación orientada a objetos.

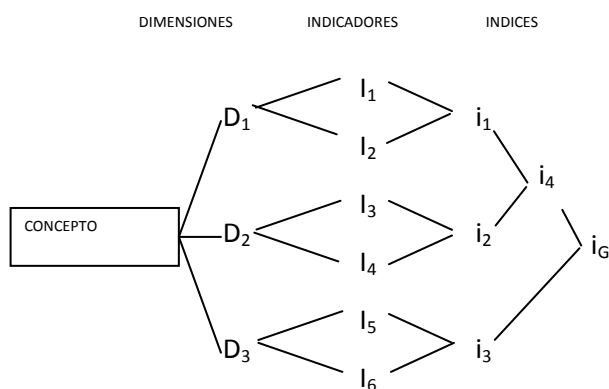


Figura 1. Dimensiones, indicadores e índices.

#### 4. Aplicación del método al concepto de herencia simple

Nuestro interés se centra en poder evaluar el conocimiento del alumno. Si echamos un vistazo a los exámenes de una asignatura de POO, esta evaluación, en la actualidad, se suele hacer de manera vaga. Las cuestiones planteadas al alumno suelen englobar diversos conceptos y diversos métodos que se aplican sobre esos conceptos. El alumno debe usar todo su conocimiento y su pericia para resolver problemas y por tanto para superar el examen con éxito. Para intentar evitar que el discente falle la resolución de un problema por el desconocimiento de una parte de él, proponemos la división de los conceptos en sus dimensiones. De esta forma, un alumno podrá conocer qué es en lo que realmente anda flojo y el profesor podrá valorar, de una manera más objetiva, el conocimiento global del alumno. En concreto, vamos a ejemplificar este método con el concepto de herencia simple.

En general, una asignatura o materia experimental, podemos dividirla en dos partes. Primero, la parte teórica esencial, donde se enseñan los conocimientos (terminologías, conceptos, hechos, clasificaciones, etc.) y los métodos (leyes, modos, formas, etc.) propios de la materia en cuestión. Segundo, la parte práctica, donde se aplican los conocimientos teóricos para resolver problemas.



#### 4.1. Dimensiones

La herencia en programación orientada objetos, como se comenta en el párrafo anterior, se puede dividir en dos dimensiones: ( $D_1$ ) la parte teórica relacionada con el concepto y ( $D_2$ ) la parte práctica de dicho concepto. Podemos refinar a su vez  $D_2$  en dos facetas o subdimensiones: ( $D_{21}$ ) el diseño de clases con herencia y ( $D_{22}$ ) su implementación en un lenguaje concreto.

- 1) El conocimiento del concepto ( $D_1$ )
- 2) El uso de la herencia en POO: ( $D_2$ )
  - a. En el diseño de clases con herencia ( $D_{21}$ )
  - b. En la implementación de la herencia ( $D_{22}$ )

#### 4.2. Indicadores

Construir indicadores no es sencillo. Hay que tener en cuenta que precisa un trabajo previo de diseñar lo que vamos a investigar, de extraer sus características y sus problemas. Esto nos dará un boceto inicial que se refina con la experiencia. Es decir, normalmente realizar indicadores será un proceso iterativo, que se basa en los conocimientos del investigador, en la observación de experimentos previos y de experiencias actuales, que guiarán esta construcción.

Los indicadores propuestos, tras un estudio del concepto de herencia y de la experiencia docente como profesor de la asignatura Programación Orientada a Objetos, para cada una de las dimensiones explicadas son:

##### Conocimiento del concepto $D_1$

Como indicadores del conocimiento del concepto propondremos:

- Grado de recuerdo del concepto de herencia. ( $I_1$ )
- Grado de comprensión del concepto y sus implicaciones. ( $I_2$ )





## Uso del concepto $D_2$

### *Diseño de clases con herencia $D_{21}$*

Un indicador de cómo el alumno es capaz de aplicar el conocimiento de la herencia, para realizar un diseño puede ser:

Medida del uso correcto de la herencia en el diseño ( $I_3$ )

### *Implementación de clases con herencia (con un lenguaje de POO) $D_{22}$*

El indicador de cómo el discente es capaz de usar el conocimiento de la herencia para resolver un problema mediante un programa puede ser:

Medida de la utilización de instrucciones y mecanismos de la herencia en un lenguaje de programación concreto. ( $I_4$ )

### 4.2.1. Ejemplos

Los ejemplos sirven de guía práctica para los investigadores que quieran usar esta técnica. En este caso, se muestra una única pregunta por indicador, pero hay que tener en cuenta que para que la obtención de datos sea correcta, debemos dar un número de preguntas suficiente. Esto permitirá conseguir unas medidas más precisas y minimizar el efecto no deseable que pueden producir otras variables ocultas (como puede ser el diseño poco adecuado o poco cuidadoso de las preguntas).

#### INDICADOR $I_1$

Las preguntas para el indicador ( $I_1$ ) del grado de recuerdo del concepto de herencia deben ser tipo test o preguntas semiabiertas (con respuestas muy acotadas), formuladas de manera similar a como se presenta el concepto con la metodología impartida o en la herramienta utilizada. Estas preguntas han de ser puntuadas para obtener un valor numérico. Un ejemplo de pregunta para este indicador es:

#### *Ejemplo ( $I_1$ ):*

Si B hereda de A entonces B incorpora la estructura (atributos) y comportamiento (métodos) de la clase A. Por lo tanto, es FALSO que B pueda:



- a) añadir nuevos atributos
- b) redefinir métodos
- c) añadir nuevos métodos
- d) eliminar atributos.

#### INDICADOR I<sub>2</sub>

Para el indicador (I<sub>2</sub>) del grado de comprensión de la herencia en POO, se deben plantear preguntas que el alumno pueda responder razonando sobre la información recibida. En este caso las preguntas pueden ser tipo test o preguntas semiabiertas. El número y el contenido de las preguntas deben ser tal que permitan al investigador o al profesor tener la suficiente confianza en que el alumno que obtenga una puntuación media-alta comprende lo que es y cómo funciona la herencia. Bajo nuestro punto de vista, los contenidos a cubrir en estas preguntas deberían ser:

- Necesidad de la herencia. Hacer ver que las clases no son suficientes para conseguir reutilización y extensibilidad.
- Relaciones entre clases: extensión, especialización y combinación.
- Herencia como solución a: representar esas relaciones y posibilitar la creación de una clase a partir de otra.
- Consecuencias de la herencia: incorporación de atributos y métodos de la clase padre, sustitución de objetos de subclases.
- Posibilidades de la herencia: adicionar nuevos atributos y métodos, redefinición de métodos (refinando o reemplazando).
- Tipos de herencia simple:
  - Especialización (la clase hija es un subtipo de la clase padre)
  - Especificación (la clase padre define comportamientos que son implementados por la clase hija)
  - Construcción (la clase hija usa el comportamiento proporcionado por la clase padre pero no es un subtipo de ella)



- Extensión (la clase hija añade nuevas funcionalidades a la clase padre, pero sin cambiar el comportamiento heredado)
- Limitación (la clase hija restringe el uso de algún comportamiento de la clase padre)

Un ejemplo de pregunta para este indicador es:

*Ejemplo (I<sub>2</sub>):*

Si un Rectángulo es un tipo especial de Polígono, el tipo de herencia que usaríamos para diseñar la clase Rectángulo a partir de la clase Polígono será:

- a) Herencia por construcción
- b) Herencia por especialización
- c) Herencia por especificación
- d) Herencia por limitación

### INDICADOR I<sub>3</sub>

El indicador (I<sub>3</sub>) de la medida del uso correcto de la herencia en el diseño podemos obtenerlo generando preguntas donde el alumno se tenga que enfrentar al diseño de una aplicación nueva, sin referencias anteriores. Para detectar la herencia durante el diseño, el discente debe ser capaz de extraer información del enunciado y ver si existen clases con comportamientos comunes (generalización) o si una clase es un caso especial de otra (especialización). Un ejemplo de pregunta en este indicador es:

*Ejemplo (I<sub>3</sub>):*

Sean los tres siguientes conceptos geométricos:

Línea recta (infinita en ambas direcciones); Rayo (sale de un punto fijo y es infinito en una dirección); Segmento (es una porción de una línea con dos puntos finales fijos)



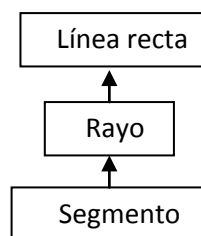
¿Cómo podría representar esos tres conceptos en una jerarquía de herencia?

#### INDICADOR $I_4$

El indicador ( $I_4$ ) de medida de la utilización de instrucciones y mecanismos de la herencia en un lenguaje concreto, debe dar la destreza del alumno en el uso del lenguaje para implementar los distintos tipos de herencia. Para ello, las preguntas de este indicador deben estar orientadas a la escritura de un programa (o fragmento) con un lenguaje de programación particular.

*Ejemplo ( $I_4$ ):*

Dada la siguiente jerarquía de herencia:



¿Cómo podría implementar en JAVA estas tres clases, sabiendo que una línea recta se puede identificar por dos puntos que atraviesa, un rayo es una recta con un punto inicial (o final) y un segmento es una recta limitada por dos puntos (inicial y final)?

### 4.3 Índices

En primer lugar vamos a fijar un índice para cada indicador. El índice ( $i_1$ ) para el indicador ( $I_1$ ) puede ser calculado de la siguiente forma:

- Si las preguntas son tipo test con cuatro opciones, de las cuales solo una es correcta, se calcula sumando a para las respuestas acertadas, restando b para las falladas (para eliminar el azar) y cero para las no contestadas. En general, el índice ( $i_1$ ) se calculará aplicando la siguiente fórmula:



$$i_1 = \frac{a*n - m*b}{t} \quad (i_1)$$

Siendo n el número de respuestas acertadas y m el número de respuestas falladas y t el número de preguntas.

- Si las preguntas son semiabiertas, la puntuación dependerá del criterio del profesor, pudiendo obtenerse un índice similar al anterior.

Este índice ( $i_1$ ) caracterizará el grado de recuerdo del concepto de herencia que el alumno posee.

Para el índice ( $i_2$ ) se puede hacer un análisis similar al anterior, caracterizando de nuevo la dimensión que representa el indicador ( $l_2$ ).

$$i_2 = \frac{a*n - m*b}{t} \quad (i_2)$$

Uniendo ambos índices podemos formar el índice ( $i_{1_2}$ ) para sacar conclusiones sobre la dimensión  $D_1$ . Este índice se puede definir como la media de los índices  $i_1$  e  $i_2$ . Por lo tanto, tendremos un valor indicativo de esta dimensión.

$$i_{1_2} = \frac{i_1 + i_2}{2} \quad (i_{1_2})$$

Análogos razonamientos se pueden hacer para los índices ( $i_3, i_4$ ). Teniendo en cuenta que en ambos se plantean preguntas abiertas, se puede puntuar cada respuesta de forma que se admitan cinco posibles valores (por ejemplo: perfecto, bien, regular, bastantes fallos, mal), asignando un peso a cada categoría (por ejemplo: perfecto = 4, bien = 3, regular = 2, bastantes fallos = 1, mal = 0). En este caso, los índices se podrían calcular con la fórmula:



$$i_3 = \frac{v_1 * p_1 + v_2 * p_2 + v_3 * p_3 + v_4 * p_4 + v_5 * p_5}{t}$$
$$i_4 = \frac{v_1 * p_1 + v_2 * p_2 + v_3 * p_3 + v_4 * p_4 + v_5 * p_5}{t}$$

Siendo  $v_1$  el número de respuestas perfectas,  $v_2$  el número de respuestas bien,  $v_3$  el número de respuestas regulares,  $v_4$  el número de respuestas con bastantes fallos,  $v_5$  el número de respuestas mal contestadas,  $p_i$  el peso de cada categoría y  $t$  el número de preguntas.

El índice ( $i_{3_4}$ ) construido como la media aritmética de  $i_3$  e  $i_4$  caracterizará la dimensión del uso de la herencia es sus dos vertientes (diseño e implementación).

$$i_{3_4} = \frac{i_3 + i_4}{2} \quad (i_{3_4})$$

## 5. Reflexiones y experiencias

Observando las dimensiones obtenidas usando el método de investigación social, se puede ver cierto paralelismo con los niveles de la jerarquía de Bloom. La dimensión  $D_1$  (conocimiento del concepto) junto con sus indicadores se puede asociar a los dos primeros niveles de Bloom (recuerdo y comprensión).  $D_2$  (uso del concepto) se puede asociar a los dos niveles siguientes (aplicación y análisis). Debido a esta relación se puede intentar sacar un índice general ( $i_G$ ) que sea la media de  $i_{1_2}$  e  $i_{3_4}$ .

$$i_G = \frac{i_{1_2} + i_{3_4}}{2} \quad (i_G \text{ Ver.1})$$

Con este índice obtendríamos teóricamente el grado de sabiduría del alumno respecto del concepto de herencia (en cuanto a su conocimiento y a su uso). Una forma de lograr interpretar dicho valor ( $i_G$ ) podría ser: Primero, restamos el mayor y el menor grado posible para obtener el intervalo. Segundo, ese intervalo se divide en cuatro fragmentos (no tienen porqué ser iguales). Un valor dentro del fragmento



menor correspondería a un alumno con poco grado de conocimiento y un valor dentro del intervalo mayor correspondería a un alumno con un alto grado de conocimiento del concepto de herencia.

Pero aquí surge nuestra duda: ¿se puede usar ese índice para medir el grado de conocimiento según la taxonomía de Bloom? O formulada desde otra perspectiva ¿el alumno que obtenga un mayor índice general ( $i_G$ ) se encuentra en un nivel más alto de la taxonomía de Bloom? Nuestra experiencia dice que no. En varios experimentos con la enseñanza de la programación se ha encontrado a alumnos que son capaces de (intuitivamente) resolver problemas del nivel de aplicación en la taxonomía de Bloom teniendo bajo porcentaje de aciertos en las preguntas de los niveles inferiores [Hernan 08] y al contrario [Lathinen 07]. No es difícil imaginar en un alumno que haya estudiado y aprendido los conceptos de memoria, que comprenda las distintas relaciones que pueden dar lugar a la herencia simple, y que obtenga una puntuación del índice general mayor que otro alumno que sepa medianamente aplicar e implementar esas relaciones pero no recuerde el nombre de dichas relaciones y por tanto, obtenga un valor menor de dicho índice.

La contradicción encontrada puede dar lugar a reformular el cálculo del índice general, añadiendo pesos  $p_{1_2}$  e  $p_{3_4}$  a cada índice  $i_{1_2}$  e  $i_{3_4}$  de forma que el índice  $i_{1_2}$  tenga menos peso que el índice  $i_{3_4}$ . De esta manera podemos compensar los casos presentados en el párrafo anterior.

$$i_G = \frac{i_{1_2} * p_{1_2} + i_{3_4} * p_{3_4}}{2} \quad (i_G \text{ ver. 2})$$

## 6. Conclusiones y trabajos futuros

En este artículo se ha presentado un nuevo método, obtenido de la investigación social, para realizar evaluaciones de una forma más objetiva y de esta manera determinar el grado de conocimiento de un alumno respecto al concepto en estudio, dotándole de una nota numérica que mide su capacidad para cada nivel de Bloom. También se ha deducido y refinado un índice general, que da una idea de la competencia del alumno en este tema. El método consiste en la descomposición del concepto en sus dimensiones, se ha obtenido sus indicadores y los índices para medir cada aspecto. Además se ha ejemplificado esta metodología aplicándola al concepto de herencia en programación orientada a objetos. Estos



ejemplos pueden servir de guía al investigador que quiera aplicar esta técnica a sus experimentos. La taxonomía de Bloom del aprendizaje ha demostrado su validez en multitud de estudios pero sus problemas para aplicarla [Fuller 07] han hecho que los investigadores en Informática Educativa busquen otras alternativas, como la que se presenta aquí.

Como trabajo futuro queda el desarrollo completo de una batería de preguntas para este concepto y su posterior evaluación del método en un experimento programado con alumnos de la asignatura de Programación Orientada a Objetos de segundo curso de Ingeniería Informática de la Universidad. De esta forma poder analizar los datos conseguidos y obtener unas conclusiones que verifiquen este método de forma experimental.

## **7. Agradecimientos**

Este trabajo se ha financiado parcialmente con el proyecto TIN2008-4103 del Ministerio de Ciencia e Innovación.





## Referencias bibliográficas

- Anderson, L & Krathwohl, D. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Ed. Longman.
- Bennedsen, J. & Caspersen, M. (2004). *Teaching object-oriented programming: Toward teaching a systematic programming process. Proceedings of the 18th European Conference on Object-Oriented Programming (ECOOP'04)*.
- Bloom, B. S. & Engelhart, M. D., Furst, E. J.; Hill, W.H., & Krathwohl, D. R.(1956). *Taxonomy of educational objectives: the classification of educational goals, handbook I, the cognitive domain*, David McKay Company, Inc, New York.
- Budd, T. (2000). *Understanding object-oriented programming with Java*. Ed Addison-Wesley.
- Carmona Guillén, J. L. (1977). *Los indicadores sociales hoy*, Ed. CIS, Madrid.
- De Miguel, A. (1967). *Tres estudios para un sistema de indicadores sociales (Estudio primero)*, Fundación Foessa, Madrid, Ed. Euramérica.
- Eisner, E. W. (2000) B. Bloom. *Perspectivas: Revista trimestral de educación comparada*, XXX (3), pp. 423-432.
- Fuller, U., Johnson, C. G., Ahoniemi, T., Cukierman D., Hernán Losada I., Jackova J., et al. (2007). *Developing a computer science-specific learning taxonomy*. ACM SIGCSE Bulletin 34(4), ACM Press, New York. pp 152-170.
- García Ferrando, M. (2003) *El análisis de la realidad social métodos y técnicas de investigación*. Ed. Alianza.
- Hernan Losada, I., Lázaro Carrascosa C. A., Velázquez Iturbide, J. A., (2008). An educative application based on Bloom's taxonomy for the learning of inheritance in oriented-object programming. In *Computers and Education: Toward educational change and innovation*. Ed. Springer London. pp 157-166.
- Johnson, C. & Fuller, U. (2007). Is Bloom's taxonomy appropriate for computer science? *Proceedings of the Sixth Baltic Sea Conference on Computing Education Research*, Ed. University of Uppsala, Stockholm. pp 120-123.
- Lahtinen, E. (2007). A categorization of novice programmers: a cluster analysis study. *Proceedings of the 19th annual Workshop of the Psychology of Programming Interest*



*Group*, Joensuu, Finland, July 2-6, 2007, Sajaniemi, J. and Tukiainen, M., Eds. University of Joensuu Department of Computer Science and Statistics, Joensuu, Finland, pp. 32-41.

Lister, R. & Leaney, J. (2003). First year programming: Let all the flowers bloo. In *Proc. of the 5th Australasian Computer Education Conference (ACE2003)*, pp 221-230.

Scott, T. (2003). Bloom's taxonomy applied to testing in computer science classes. *Journal of Computing Sciences in Colleges*. Vol 19, Issue 1, pp 267-274.