

**Susana Masapanta Carrión  
J. Ángel Velázquez Iturbide**

# **Guía de Uso de la Taxonomía de Bloom en Informática**

**Número 2018-03**

**Serie de Informes Técnicos DLSI1-URJC  
ISSN 1988-8074  
Grupo Docente de Lenguajes y Sistemas Informáticos I  
Universidad Rey Juan Carlos**



## Índice

1. Introducción .....	1
Agradecimientos .....	2
Referencias .....	2
Apéndice A: Guía de uso la Taxonomía de Bloom en Informática .....	3



# Guía de Uso de la Taxonomía de Bloom en Informática

Susana Masapanta Carrión<sup>1</sup>; J. Ángel Velázquez Iturbide<sup>2</sup>

<sup>1</sup> Facultad de Ingeniería, Pontificia Universidad Católica del Ecuador, Av. 12 de Octubre 1076 y Roca, 17012184, Quito, Ecuador

<sup>2</sup> Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos, C/ Tulipán s/n, 28933, Móstoles, España

<sup>1</sup> [smmasapanta@puce.edu.ec](mailto:smmasapanta@puce.edu.ec); <sup>2</sup> [angel.velazquez@urjc.es](mailto:angel.velazquez@urjc.es)

**Resumen.** En una revisión sistemática anterior se realizó un estudio sobre las principales dificultades para el uso de la taxonomía de Bloom en la educación en informática. En este informe presentamos una guía de uso de la taxonomía revisada de Bloom en asignaturas de Informática. El objetivo de la guía es proporcionar consejos y aclarar aspectos confusos sobre el uso de la taxonomía.

**Palabras clave:** Taxonomía revisada de Bloom, educación en informática.

## 1. Introducción

En un trabajo anterior [1] hemos revisado sistemáticamente el uso de la taxonomía de Bloom en la enseñanza de la Informática. Como consecuencia, se documentaron diversas dificultades en su uso por numerosos autores. La principal dificultad se presenta al clasificar tareas o preguntas de evaluación según la taxonomía. Además, se identificó las posibles causas y soluciones a estas dificultades.

Entre las causas más relevantes tenemos: la necesidad de conocer la forma como se ha enseñado el contenido en clase (contexto educativo) antes de clasificar, la deficiente comprensión de la taxonomía debido a un conocimiento superficial o a la inexperiencia en manejo de la terminología taxonómica y la falta de ejemplos para informática. Por otra parte, la solución más frecuente es dar pautas de aplicación de la taxonomía.

En base a esta información, se realizó un estudio minucioso sobre la taxonomía revisada para elaborar un documento con consejos e información práctica para el uso de este modelo en un ambiente informático.

En este informe presentamos una guía que pretende ayudar a los profesores en el uso de la taxonomía revisada de Bloom [2] en asignaturas de informática. Las consideraciones realizadas y los ejemplos incluidos son de programación, dado el amplio uso de la taxonomía en la enseñanza de la programación y la relevancia de esta dentro de la informática. La guía propuesta se presenta en el Apéndice A.

**Agradecimientos.** Este trabajo se ha financiado con los proyectos TIN2015-66731-C2-1 del Ministerio de Economía y Competitividad de España y S2013/ICE-2715 de la Comunidad Autónoma de Madrid.

## Referencias

1. Masapanta-Carrión, S.; Velázquez-Iturbide, J.Á.: A systematic review of the use of Bloom's taxonomy in computer science education. In Proceedings 49th ACM Technical Symposium on Computing Science Education, February 2018, DOI: 10.1145/3159450.3159491
2. Anderson, L.W.; Krathwohl, D.R.; Airasian, P.W.; Cruikshank, K.A.; Mayer, R.E.; Pintrich, P.R.; Raths, R.; Wittrock, M.C.: *A Taxonomy for Learning, Teaching and Assessing. A Revision of Bloom's Taxonomy of Educational Objectives*. Pearson Education (2014)

## Apéndice A: Guía de uso la Taxonomía de Bloom en Informática

### Índice

Guía de Uso de la Taxonomía de Bloom en Informática.....	2
1. Introducción.....	2
2. La taxonomía de Bloom.....	2
2.1 Dimensión del proceso cognitivo.....	3
2.2 Dimensión del conocimiento.....	3
2.3 La matriz taxonómica.....	4
2.4 Alineamiento de objetivos, actividades de instrucción y evaluaciones.....	4
3. La taxonomía de Bloom y la informática.....	5
3.1 La terminología de la taxonomía de Bloom.....	5
3.2 Ejemplos en informática de la dimensión del proceso cognitivo.....	7
3.3 Ejemplos en informática de la dimensión del conocimiento.....	11
4. Uso práctico de la taxonomía de Bloom.....	14
4.1 Recomendaciones generales.....	14
4.2 Pasos para la clasificación de actividades de evaluación.....	15
5. Ejemplos de clasificación de ejercicios de programación.....	17
5.1 Ejemplo 1.....	17
5.2 Ejemplo 2.....	18
5.3 Ejemplo 3.....	19
5.4 Ejemplo 4.....	20
5.5 Ejemplo 5.....	21
5.6 Ejemplo 6.....	23
Referencias.....	24

# Guía de Uso de la Taxonomía de Bloom en Informática

## 1. INTRODUCCIÓN

La taxonomía de Bloom es un marco frecuentemente utilizado en educación, pero su uso no está exento de dificultades y malentendidos (Masapanta Carrión & Velázquez Iturbide, 2018). La guía aquí presentada pretende proporcionar una ayuda a los profesores de informática, aunque las consideraciones recogidas pueden aplicarse a otras disciplinas, convenientemente adaptadas. Las consideraciones realizadas y los ejemplos incluidos son de programación, dado el amplio uso de la taxonomía en la enseñanza de la programación y la relevancia de esta dentro de la informática.

La estructura de la guía es la siguiente. En la sección 2 se presenta un resumen de la taxonomía revisada de Bloom. La sección tercera trata la adaptación de la taxonomía a la informática. La sección 4 contiene diversos consejos y guías prácticas para el uso de la taxonomía. Finalmente, la sección 5 muestra el uso de la taxonomía mediante la clasificación de 5 ejercicios de programación.

## 2. LA TAXONOMÍA DE BLOOM

La taxonomía de objetivos educativos es un modelo que permite clasificar el grado de aprendizaje que se espera que alcancen los alumnos luego de un proceso educativo.

Existen dos versiones. La primera versión se conoce como taxonomía original de Bloom o simplemente taxonomía de Bloom (Bloom *et al.*, 1956). Su principal objetivo era clasificar los comportamientos que se espera que tengan los alumnos tras recibir la educación. El uso más corriente que se le ha dado ha sido clasificar los ejercicios utilizados para evaluar el aprendizaje de los alumnos. La segunda versión (Anderson *et al.*, 2001) se conoce como taxonomía revisada de Bloom. En esta se realizaron cambios importantes en la estructura y la terminología del modelo original. Además, se enfatiza su uso en la planificación docente, donde debería haber una correspondencia (“alineamiento”) entre objetivos, actividades educativas y ejercicios de evaluación. En esta guía, salvo que digamos lo contrario, nos centramos en la versión revisada de la taxonomía.

Según la taxonomía revisada, un objetivo educativo describe un comportamiento esperado del alumno y el contenido sobre el que tendrá lugar dicho comportamiento. Normalmente, un objetivo educativo se declara con una frase que empieza “el alumno será capaz de ...” o “el alumno aprenderá a...”. Por ejemplo, un objetivo puede ser “el alumno aprenderá a resumir el propósito de las subrutinas presentes en un programa”. Si analizamos la estructura de la frase, observamos que el sujeto siempre es el alumno, el verbo indica el comportamiento esperado (en el ejemplo, resumir) y un complemento indica el contenido (en el ejemplo, el propósito de una subrutina). Por brevedad, el comienzo (“el alumno”) suele omitirse.

Dados los objetivos para los que se han diseñado ambas versiones de la taxonomía, no deben usarse para evaluar el progreso del alumno. Existen otros modelos o taxonomías para medir dicho progreso, por ejemplo, la taxonomía SOLO (Biggs & Collis, 1982).

En los dos apartados siguientes veremos el desarrollo de ambos aspectos: el contenido y el comportamiento. En la taxonomía se habla de dos dimensiones, de forma que el contenido se trata en la dimensión del conocimiento y el comportamiento esperado en la dimensión del proceso cognitivo. Posteriormente veremos el tratamiento de dos cuestiones importantes en educación y relacionadas con el logro de los objetivos educativos: los objetivos instruccionales y las actividades de evaluación.



## 2.1 Dimensión del proceso cognitivo

Los procesos cognitivos proporcionan una forma de describir el rango de actividades cognitivas de los alumnos en el aprendizaje constructivista.

La taxonomía distingue 6 categorías de procesos cognitivos. Cada categoría se nombra con un verbo y, a su vez, incluye varios procesos cognitivos, resultando un total de 19 procesos (Anderson *et al.*, 2014) (ver Tabla 2). Como regla general, se recomienda fijarse más en el proceso cognitivo correspondiente a un objetivo educativo que en su categoría.

Para aquellos lectores familiarizados con la taxonomía original, se produce un cambio ya que la taxonomía original caracterizaba cada categoría con un nombre, en lugar de un verbo (así, la categoría de Conocimiento de la taxonomía original pasa a ser Recordar en la taxonomía revisada). Las categorías de Síntesis y Evaluación se cambian de orden (y se renombran con verbos, quedando Evaluar y Crear).

Categorías de procesos cognitivos	Procesos cognitivos
1. Recordar: Recuperar el conocimiento relevante de la memoria a largo plazo	1.1 Reconocer 1.2 Recordar
2. Comprender: Construir el significado de los mensajes de instrucción, incluida la comunicación oral, escrita y gráfica	2.1 Interpretar 2.2 Ejemplificar 2.3 Clasificar 2.4 Resumir 2.5 Inferir 2.6 Comparar 2.7 Explicar
3. Aplicar: Llevar a cabo o usar un procedimiento en una situación dada	3.1 Ejecutar 3.2 Implementar
4. Analizar: Divide el material en sus partes constituyentes y determinar cómo las partes se relacionan entre sí y con la estructura general o propósito	4.1 Diferenciar 4.2 Organizar 4.3 Atribuir
5. Evaluar: Hacer juicios basados en criterios y estándares	5.1 Comprobar 5.2 Valorar
6. Crear: Poner elementos para formar un todo coherente o funcional; organizar elementos en un nuevo patrón o estructura	6.1 Generar 6.2 Planificar 6.3 Producir

Tabla 1. Dimensión del proceso cognitivo (Anderson *et al.*, 2014)

## 2.2 Dimensión del conocimiento

Las asignaturas se planifican alrededor de cierta materia. Sin embargo, el estudio de una materia puede hacerse de diferentes formas, dando lugar a distintos tipos de conocimiento. La taxonomía distingue cuatro tipos de conocimiento, que a su vez se dividen en 11 subtipos de conocimiento. La Tabla 1 presenta un resumen de su clasificación. Más adelante se verán ejemplos de programación.

Tipos	Subtipos
A. Conocimiento factual: Los elementos básicos que los estudiantes deben conocer para estar familiarizados con una disciplina o resolver problemas en ella	Aa. Conocimiento de terminología Ab. Conocimiento de detalles y elementos específicos
B. Conocimiento conceptual: Las interrelaciones entre los elementos básicos dentro de una estructura más grande que les permite funcionar juntos	Ba. Conocimiento de clasificaciones y categorías Bb. Conocimiento de principios y generalizaciones Bc. Conocimiento de teorías, modelos y estructuras

C. Conocimiento procedimental: Cómo hacer algo, métodos de investigación y criterios para usar habilidades, procedimientos, técnicas y métodos	Ca. Conocimiento de habilidades y procedimientos específicos del tema Cb. Conocimiento de técnicas y métodos específicos del tema Cc. Conocimiento de los criterios para determinar cuándo usar procedimientos apropiados
D. Conocimiento metacognitivo: Conocimiento de la cognición en general, así como conciencia y conocimiento de la propia cognición	Da. Conocimiento estratégico Db. Conocimiento sobre tareas cognitivas, incluido el apropiado conocimiento contextual y condicional Dc. Autoconocimiento

Tabla 2. Dimensión del conocimiento (Anderson *et al.* 2014)

### 2.3 La matriz taxonómica

La “matriz taxonómica” es una tabla con dos dimensiones, correspondientes a las dos dimensiones de la taxonomía de Bloom. Cada objetivo de aprendizaje puede escribirse en la celda correspondiente a su tipo de conocimiento y de proceso cognitivo. De esta forma, la tabla permite mostrar de una forma visual el conocimiento y el proceso cognitivo de cada objetivo de aprendizaje (véase Tabla 3).

Supóngase que se desea clasificar el objetivo “el estudiante aprenderá a aplicar métodos de ordenamiento en Java”. Para ello, deben determinarse el sustantivo y el verbo más importantes. El verbo distintivo es “aplicar”, que en este contexto quiere decir “usar, utilizar”. Esta actividad corresponde a la categoría Aplicar, cuya definición es “usar un procedimiento en una situación dada”. El sustantivo es “métodos de ordenamiento en Java”. Si los métodos ya están desarrollados, el alumno debe conocer su función y seleccionar el más adecuado; por tanto, se trata de conocimiento conceptual. El objetivo quedará ubicado en la celda B3 de la matriz taxonómica, que es la intersección del conocimiento conceptual con el proceso de Aplicar. Sin embargo, si los métodos de ordenamiento no están desarrollados, el alumno debe desarrollarlos, siendo conocimiento procedimental. En este caso, el objetivo se colocará en la celda C3. Este ejemplo muestra algo que se resaltaré más adelante (apartado 5.1), la necesidad de conocer el contexto en el cual se enmarcan objetivos, actividades o ejercicios para su adecuada clasificación.

**RECUERDE:** El verbo generalmente describe el proceso cognitivo intentado. El sustantivo generalmente describe el conocimiento que los estudiantes deben adquirir o construir.

Dimensión del conocimiento	Dimensión del proceso cognitivo					
	1 Recordar	2 Comprender	3 Aplicar	4 Analizar	5 Evaluar	6 Crear
A. Factual						
B. Conceptual						
C. Procedimental						
D. Metacognitivo						

Tabla 3. Estructura de la matriz taxonómica

### 2.4 Alineamiento de objetivos, actividades de instrucción y evaluaciones

El uso de la tabla taxonómica no se limita a los objetivos de aprendizaje. Para que unos objetivos educativos se alcancen, no basta con enunciarlos, sino que deben realizarse actividades de instrucción encaminadas a alcanzar dichos objetivos y posteriormente el profesor realizará pruebas de evaluación para comprobar si se han alcanzado. La tabla permite colocar en sus celdas objetivos, actividades y evaluaciones.

Si en una celda coinciden un objetivo, una o más actividades de instrucción y al menos una evaluación, hay un alto grado de alineamiento. Sin embargo, las celdas que solo tengan un objetivo, una actividad de instrucción o una evaluación, tienen un bajo alineamiento. En caso de encontrar un bajo alineamiento, el profesor debe realizar cambios en los objetivos, las actividades de instrucción o las evaluaciones para aumentar el grado de alineamiento.

### 3. LA TAXONOMÍA DE BLOOM Y LA INFORMÁTICA

En este apartado, se busca dar algunas pautas que faciliten la comprensión y la aplicación de la taxonomía en la enseñanza de la informática. La primera parte incluye los principales términos de la taxonomía de Bloom que pueden producir confusiones porque su significado es distinto del que tienen en informática. Las dos partes siguientes definen los tipos de conocimiento y los procesos cognitivos en un ambiente informático. Se ha puesto énfasis en las definiciones de las subcategorías y para cada una de ellas se dan ejemplos. Si bien esta guía puede ser aplicada a cualquier asignatura de informática, se han dado ejemplos de programación para lograr una mejor comprensión. Además, se han incluido algunos consejos para la aplicación de esta guía y notas aclaratorias extraídas del libro de Anderson *et al.* (2014).

#### 3.1 La terminología de la taxonomía de Bloom

Algunos términos usados en la versión revisada de la taxonomía tienen un significado distinto al usual. Por ejemplo, el proceso “Explicar”, incluido en la categoría “Comprender”, se refiere a construir un modelo mental del funcionamiento de un artefacto. Sin embargo, en el lenguaje cotidiano “explicar” tiene este y otros significados, por ejemplo, dar a conocer algún suceso o acontecimiento a otra persona. Esta polisemia es una fuente potencial de malentendidos y equívocos. Además, algunos nombres de procesos cognitivos tienen un significado distinto del usual en informática, lo que añade una dificultad añadida al usarlo en un contexto de enseñanza de la informática. Por tanto, un uso relativamente seguro de la taxonomía de Bloom exige una comprensión del significado de cada término de la taxonomía.

Para las definiciones en español se ha utilizado como referencia el diccionario de la Real Academia de la Lengua (RAE)<sup>1</sup> y se toman las definiciones más habituales. Para las interpretaciones de los términos informáticos también se usan las definiciones más frecuentes. Se ha tomado como referencia Wikipedia<sup>2</sup> y el diccionario informático del portal de Alegsa<sup>3</sup>.

Comenzamos por la dimensión de proceso cognitivo. Presentamos una lista de los principales verbos que pueden inducir a confusión, en orden creciente de nivel y subnivel de la taxonomía:

- Interpretar (proceso cognitivo de la categoría Comprender). Para la taxonomía, “interpretar” es “cambiar de una representación (oral, escrita, numérica, gráfica, etc.) a otra”, por ejemplo, de gráficos a palabras, de palabras a código, o de palabras a palabras (parafrasear). En su uso popular, la palabra “interpretar” puede tener otro significado, “dar un significado a algo” o “resumir”. En informática, “la interpretación es la realización de un paso intermedio entre el texto escrito y su significado. Y en programación interpretar es usar un intérprete para ejecutar líneas de código escritas en un lenguaje de programación”<sup>4</sup>.

---

<sup>1</sup> <http://www.rae.es/>

<sup>2</sup> [https://es.wikipedia.org/wiki/Wikipedia\\_en\\_espa%C3%B1ol](https://es.wikipedia.org/wiki/Wikipedia_en_espa%C3%B1ol)

<sup>3</sup> <http://www.alegsa.com.ar/Diccionario/diccionario.php>

<sup>4</sup> <http://www.alegsa.com.ar/Dic/interpretar.php>

- Explicar (proceso cognitivo de la categoría Comprender). La definición dada en la taxonomía es “construir y usar un modelo causa-efecto de un sistema”, que es una definición diferente de usada en español y en informática. Los autores de la taxonomía revisada avisan de que un profesor puede usar la palabra “explicar” cuando no quiere decir “construir un modelo causal” sino queriendo decir “interpretar o resumir” (Anderson *et al.*, 2014, pág. 16).
- Ejecutar (proceso cognitivo de la categoría Aplicar). En la taxonomía revisada de Bloom, “ejecutar” es “usar algoritmos y habilidades en una tarea familiar”, produciendo un resultado predeterminado. En informática, la palabra “ejecutar” indica “la acción de iniciar la carga de un programa o de cualquier archivo ejecutable”<sup>5</sup>, acción que realiza el computador. Aunque la ejecución de un programa puede producir un resultado predeterminado, en un caso la aplicación la realiza el alumno mientras que en el otro, la realiza el ordenador. Los esfuerzos cognitivos de “lanzar” la ejecución de un programa y de realizar paso a paso un procedimiento son muy distintos.
- Implementar (proceso cognitivo de la categoría Aplicar). En la taxonomía, se define como “aplicar técnicas y métodos a una tarea no familiar”. Este proceso cognitivo suele ir asociado a los procesos de comprender y crear. Si bien las definiciones de implementar en español (“poner en funcionamiento o aplicar métodos, medidas, etc., para llevar algo a cabo”) y en informática (“la realización de una especificación técnica o algoritmos como un programa, componente software, u otro sistema de cómputo”<sup>6</sup>) concuerdan con lo expresado en Bloom, hay que tener cuidado con algunas interpretaciones de la palabra implementar dentro del ambiente informático. Por ejemplo, en programación un algoritmo especificado en pseudocódigo se implementa en forma de código de un lenguaje de programación”<sup>7</sup>. Según la taxonomía, esta tarea es más acorde con el proceso de interpretar que de implementar.
- Analizar (categoría cognitiva). Las definiciones del español y de Bloom son muy similares. Para Bloom, es “dividir el material en sus partes constituyentes y determinar cómo estas partes se relacionan entre sí y con la estructura o el propósito general”. Pero en informática el término analizar comúnmente se relaciona con técnicas de análisis. Por ejemplo, el análisis de algoritmos busca determinar la eficiencia de un algoritmo. Esta tarea corresponde al nivel de aplicación (proceso cognitivo de implementar).
- Evaluar (categoría cognitiva). La palabra “evaluar” en informática puede tener algunas interpretaciones distintas a la definición de la taxonomía, “hacer juicios en base a criterios y estándares”. Por ejemplo, si se pide evaluar la expresión “ $2*8/6+9-5$ ”, en realidad se pide obtener un resultado mediante un proceso de cálculo bien determinado (es decir, el proceso cognitivo Ejecutar).
- Generar (proceso cognitivo de Crear). En la taxonomía se define como “representar el problema para sugerir varias soluciones o hipótesis que cumplan ciertos criterios”. En informática, comúnmente significa “producir o dar lugar a algo”, por ejemplo, generar números aleatorios. En este ejemplo, se podría usar una función ya existente, sin implicar un proceso creativo por parte del estudiante, que correspondería a la categoría Aplicar.

Veamos la dimensión del conocimiento:

- Conocimiento factual y Conocimiento conceptual. Según la taxonomía, el tipo Conocimiento factual engloba los elementos básicos que los estudiantes deben conocer para estar familiarizados con una disciplina o resolver problemas en ella. Por tanto, contiene los conceptos de la disciplina (subtipo Terminología). El tipo Conocimiento Conceptual incluye las interrelaciones entre los

<sup>5</sup> <http://www.alegsa.com.ar/Dic/ejecutar.php>

<sup>6</sup> <https://es.wikipedia.org/wiki/Implementaci3n>

<sup>7</sup> <http://www.alegsa.com.ar/Dic/implementacion.php>

elementos básicos dentro de una estructura más grande que les permite funcionar juntos. Por tanto, incluye las interrelaciones entre los conceptos de la disciplina.

- Conocimiento procedimental – conocimiento de habilidades y algoritmos específicos de la materia. En la taxonomía, se define un algoritmo como una secuencia de pasos que se realizan en un orden fijo y el resultado, también. Aunque esta definición coincide parcialmente con los algoritmos informáticos, hay que tener en cuenta que incluye cualquier procedimiento que cumpla estas condiciones, como una receta de cocina o un baile concreto. En la informática, tendrían la calificación de algoritmos otros procesos como realizar una traza de un programa.

### 3.2 Ejemplos en informática de la dimensión del proceso cognitivo

Repetimos en este apartado el mismo proceso que en el apartado anterior, pero centrándonos en la dimensión del proceso cognitivo. Cabe mencionar que la taxonomía proporciona un pequeño número de ejemplos extraídos de la informática, que citamos.

1. Recordar: “recuperar conocimiento relevante de la memoria a largo plazo”. A pesar de su sencillez, recordar conocimiento es esencial para el aprendizaje significativo y para la resolución de problemas, ya que dicho conocimiento se utiliza en tareas más complejas. Si el profesor se centra en el aprendizaje de memoria, la instrucción la evaluación se centra en recordar elementos o fragmentos de conocimiento, frecuentemente aislados de ningún contexto. Sin embargo, cuando se busca aprendizaje significativo, recordar conocimiento se integra dentro de una tarea mayor. También se da recuerdo cuando el alumno hace una tarea bajo condiciones parecidas a aquellas en las que aprendió el material. Esta categoría tiene asociados dos procesos cognitivos:

**1.1 Reconocer:** “recuperar conocimiento relevante de la memoria a largo plazo para compararlo con información presente”.

Ejemplos: reconocer los diferentes elementos de un programa presentado al alumno; reconocer los pasos de desarrollo de un programa (edición, compilación, etc.).

**1.2 Recordar:** “recuperar conocimiento relevante de la memoria a largo plazo cuando se le urge a ello”. Normalmente se pide mediante una pregunta.

Ejemplos: indicar el orden de precedencia de los operadores aritméticos; escribir la sintaxis de la instrucción ‘for’.

---

CITA: “Si las tareas de evaluación van a detectar procesos cognitivos de orden superior, deben exigir que los estudiantes no puedan responderlas correctamente confiando solo en la memoria.” (pág. 71).

---

2. Comprender: “se dice que los alumnos *Comprenden* cuando son capaces de construir un significado a partir de mensajes instruccionales, incluyendo comunicaciones orales, escritas y gráficas, independientemente de cómo se hayan presentado a los alumnos: en clase, en libros o en monitores de ordenador”. Es la categoría más amplia entre los objetivos educativos que buscan que el conocimiento no sólo se retenga, sino que se transfiera a nuevos problemas. Está compuesta por siete procesos cognitivos:

**2.1 Interpretar:** “*Interpretar* sucede cuando un alumno es capaz de convertir información de una representación a otra. *Interpretar* puede conllevar convertir palabras en palabras (p.ej., parafrasear), diagramas en palabras, palabras en diagramas, números en palabras, palabras en números, notas musicales en tonos, y así sucesivamente.”

Ejemplos: convertir un diagrama de flujo en un programa o viceversa; expresar en lenguaje natural los pasos de un segmento de código o un diagrama.

**2.2 Ejemplificar:** “*Ejemplificar* sucede cuando el alumno da un ejemplo concreto de un concepto o un principio general, identificando sus características definitorias. *Ejemplificar* conlleva identificar las características definitorias del concepto o principio general (...) y usar estas características para seleccionar o construir un ejemplo concreto (...).”

Ejemplos: escribir un ejemplo de instrucción iterativa; seleccionar el caso que representa un ejemplo de herencia.

**2.3 Clasificar:** “*Clasificar* sucede cuando un alumno reconoce que algo (p.ej. un ejemplo concreto) pertenece a una categoría determinada (p.ej. un concepto o principio). *Clasificar* conlleva detectar las características relevantes o los patrones que “se ajustan” al ejemplo concreto y al concepto o principio.”

Ejemplo: indicar si las siguientes afirmaciones son adecuadas para un archivo de texto o para un archivo binario; identificar la técnica de diseño de algoritmos usada para el diseño del siguiente algoritmo que resuelve el siguiente problema.

---

CITA: “*Clasificar* es un proceso complementario a *ejemplificar*” (pág. 72).

---

**2.4 Resumir:** “*Resumir* sucede cuando un alumno propone una única frase en representación de cierta información mostrada o que resume un tema. *Resumir* conlleva construir una representación de la información, como el significado de una escena en una obra, y elaborar un resumen de ella, como determinar el tema o los aspectos principales.”

Ejemplo: "En informática, un objetivo podría ser aprender a resumir el propósito de varias subrutinas en un programa. Un ítem de evaluación presenta un programa y le pide al alumno que escriba una frase que describa el objetivo que cada fragmento del programa cumple dentro del programa completo." (Anderson *et al.*, 2014, pág. 73).

**2.5 Inferir:** “*Inferir* conlleva encontrar un patrón dentro de una serie de ejemplos. *Inferir* sucede cuando un alumno es capaz de resumir un concepto o principio que representa a un conjunto de ejemplos mediante la identificación de las características relevantes de cada ejemplo y, más importante, mediante la observación de sus relaciones.”

Ejemplo: Una variable  $x$  debe tomar sucesivamente los valores 0, 2, 4 en cada iteración de un bucle; se pide construir una instrucción iterativa, basada en una variable de control  $i$ , que imprima esos valores. Dado el análisis de complejidad de varios algoritmos con una estructura sintáctica similar, inferir un esquema de análisis de complejidad que los represente.

---

CITAS: (pág. 74)

“El proceso de *inferir* conlleva hacer comparaciones entre instancias en el contexto de todo el conjunto. (...) Esto es un ejemplo de *ejecutar*, que es un proceso de cognitivo asociado a *Aplicar*. *Inferir* y *ejecutar* a menudo se usan juntos en tareas cognitivas.”

“(...) *inferir* es distinto de *atribuir* (un proceso cognitivo asociado con *Analizar*).”

---

**2.6 Comparar:** “*Comparar* conlleva detectar similitudes y diferencias entre dos o más objetos, sucesos, ideas, problemas o situaciones (...).”

Ejemplo: indicar las diferencias existentes entre los métodos constructores de una clase y el resto de los métodos; comparar las características de algoritmos voraces, heurísticos y aproximados.

---

CITA: “Cuando se usa conjuntamente con *inferir* (p.ej., primero abstraer una regla a partir de una situación familiar) e *implementar* (p.ej., después aplicar la regla a una situación menos familiar), *comparar* puede contribuir al razonamiento por analogía.” (pág. 75)

---

**2.7** Explicar: “Explicar sucede cuando un alumno es capaz de construir y usar un modelo de causa-efecto de un sistema. (...) Una explicación completa conlleva construir un modelo de causa-efecto, incluyendo las partes principales de un sistema o los sucesos principales de la cadena, y usar el modelo para determinar cómo un cambio en una parte del sistema o un «enlace» en la cadena produce un cambio en otra parte.”

Ejemplos: explicar una propiedad o resultado que puede deducirse de un fragmento de código sin ejecutarlo, como cuántas vueltas da un bucle, qué imprime en pantalla, cuál es valor de determinadas variables tras ejecutar el código o qué efecto tendría que el índice que controla el bucle se incrementara de dos en dos.

**3.** Aplicar: “*Aplicar* conlleva utilizar procedimientos para realizar ejercicios o resolver problemas. Por tanto, *Aplicar* está estrechamente relacionado con *Conocimiento procedimental*.” La categoría Aplicar consta de dos procesos cognitivos:

**3.1** Ejecutar: “Al *ejecutar*, el alumno lleva a cabo un procedimiento de forma rutinaria para afrontar una tarea familiar (es decir, un ejercicio). La familiaridad de la situación frecuentemente proporciona pistas suficientes para guiar en la elección del procedimiento adecuado a usar. *Ejecutar* está más frecuentemente asociado con el uso de habilidades y algoritmos que con técnicas y métodos.”

Ejemplos: determinar el resultado producido por un fragmento de código con unos valores concretos, realizando para ello su ejecución paso a paso (es decir, su “traza”), p.ej. como qué imprime en pantalla o cuál es valor de determinadas variables; escribir un algoritmo siguiendo las instrucciones dadas en el enunciado, sin que exista una traducción “una a una” de las instrucciones dadas con el código final; resolver un sistema de ecuaciones recurrentes para calcular la complejidad de un algoritmo recursivo.

**3.2** Implementar: “*Implementar* sucede cuando un estudiante selecciona y usa un procedimiento para realizar una tarea no familiar.” cuando se requiere seleccionar un procedimiento para usarlo en una tarea desconocida. Los procedimientos están más relacionados con técnicas y métodos antes que habilidades y algoritmos. El rango de procedimientos es muy variado, desde un procedimiento que debe seleccionarse para afrontar una nueva situación, pasando por un procedimiento que haya que modificar hasta el caso en que no exista ningún procedimiento adecuado y haya que elaborarlo a partir de conocimiento conceptual.

Ejemplo: analizar la complejidad de un algoritmo; analizar la usabilidad de un programa.

---

CITA: “Debido a que se requiere una selección, los estudiantes deben comprender el tipo de problema encontrado, así como la gama de procedimientos disponibles. Por lo tanto, *implementar* se usa conjuntamente con otras categorías de procesos cognitivos, como *Comprender* y *Crear*.” (pág.78)

---

**4.** Analizar: “*Analizar* conlleva descomponer un material en sus partes constituyentes y determinar cómo estas partes se relacionan entre sí y con la estructura global.” “Las categorías de *Comprender*, *Analizar* y *Evaluar* están entrelazadas y frecuentemente se usan de forma iterativa al realizar tareas cognitivas.” Incluye tres procesos cognitivos:

**4.1** Diferenciar: “*Diferenciar* conlleva distinguir las partes de una estructura global basándose en su relevancia o importancia. *Diferenciar* sucede cuando un alumno distingue entre información relevante e irrelevante, o entre información importante y no importante, y entonces atiende a la información relevante o importante.”

Ejemplo: identificar las partes relevantes de un fragmento de código para las preguntas que se formulan; agrupar instrucciones del mismo orden complejidad en pasos que simplifiquen el análisis de complejidad de un algoritmo.

- 4.2 Organizar:** “*Organizar* conlleva identificar los elementos de una comunicación o situación y reconocer cómo encajan en una estructura coherente. Al *organizar*, un alumno construye conexiones sistemáticas y coherentes entre partes de la información presentada.”

Ejemplo: determinar el orden de ejecución de los operadores infijos presentes en una expresión; determinar el orden en que deben colocarse varias instrucciones para que hagan lo que se espera (problemas de Parsons).

**RECUERDE:** Al examinar el código, los estudiantes deben analizar léxica, sintáctica y semánticamente, por lo que es necesario alguna actividad de **Analizar (Organizar y a veces Diferenciar)**.

---

CITA: (pág. 81)

“*Organizar* generalmente aparece junto a **diferenciar**. El estudiante primero identifica los elementos relevantes o importantes y después determina la estructura completa en la que los elementos encajan. *Organizar* también puede aparecer junto a *atribuir*, en cuyo caso se centra en determinar la intención o el punto de vista del autor.”

---

- 4.3 Atribuir:** “*Atribuir* sucede cuando un alumno es capaz de determinar el punto de vista, sesgos, valores o tendencias que subyacen a las comunicaciones. *Atribuir* conlleva un proceso de deconstrucción, en el que el alumno determina las intenciones del autor del material presentado.”

Ejemplos: determinar las razones de haber tomado ciertas decisiones de codificación; determinar las razones de haber implementado un algoritmo con ciertas estructuras de datos; analizar el estilo de programación usado en un programa.

- 5. Evaluar:** “Evaluar se define como hacer juicios basados en criterios y niveles de calidad. Los criterios más frecuentemente usados son calidad, eficacia, eficiencia y coherencia.” Incluye dos procesos cognitivos:

- 5.1 Comprobar (*checking*):** “*Comprobar* conlleva verificar si existen incoherencias internas o falacias en una operación o producto.”

Ejemplos: decidir qué segmento de código completa un programa incompleto en una pregunta de tipo *skeleton code*<sup>8</sup>; verificar si un programa satisface su especificación.

---

CITA:

“Cuando se combina con *planificar* (un proceso cognitivo de la categoría Crear) e *implementar* (un proceso cognitivo de la categoría Aplicar), comprobar conlleva determinar cómo de bien está funcionando el plan.” (pág. 83)

---

- 5.2 Valorar (*critiquing*):** “*Valorar* conlleva juzgar un producto u operación basándose en criterios y niveles de calidad diseñados externamente. Al *valorar*, un alumno identifica las características positivas y negativas de un producto y hace un juicio basado, al menos parcialmente, en dichas características. *Valorar* está en el corazón de lo que se ha llamado pensamiento crítico.”

---

<sup>8</sup> En una pregunta de tipo *skeleton code*, se entrega un fragmento de código o un algoritmo con algunas líneas sin completar y se pide completar esas líneas para que funcione apropiadamente.



Ejemplos: valorar si un algoritmo resuelve eficientemente un problema; valorar la calidad de la aproximación que realiza un algoritmo heurístico a la solución óptima de un problema; valorar un reporte técnico teniendo en cuenta su organización y estilo de escritura.

6. Crear: “*Crear* conlleva combinar diversos elementos para formar un todo coherente y funcional. Los objetivos clasificados como *Crear* piden a los alumnos construir nuevos productos mediante la reorganización mental de algunos elementos o partes en un patrón o estructura que antes no existía.” “Para algunas personas, la creatividad es la producción de productos inusuales, frecuentemente como resultado de alguna habilidad especial. *Crear*, tal y como se usa aquí, sin embargo, aunque incluye objetivos que piden una producción original, también se refiere a objetivos que pidan una producción que todos los alumnos pueden hacer.” *Crear* está asociado con tres procesos cognitivos:

- 6.1 Generar: “*Generar* conlleva representar el problema e idear alternativas o hipótesis que cumplen ciertos criterios. Frecuentemente, la forma de representar inicialmente un problema sugiere posibles soluciones; sin embargo, si se redefine o se llega a una nueva representación del problema, puede sugerir diferentes soluciones.”

Ejemplo: “En una tarea de usos, un alumno debe enumerar todos los usos posibles de un objeto, como «¿cuáles son los usos posibles de la World Wide Web?»” (pág. 87); proponer distintas formas de resolver un problema, sin entrar en sus detalles; idear criterios de selección razonables que puedan ser la base para un algoritmo voraz que resuelva un problema de optimización.

---

CITA: “*Generar* se usa aquí en un sentido restringido. *Comprender* también requiere procesos generativos que hemos incluido en *traducir, ejemplificar, resumir, inferir, clasificar, comparar y explicar*. Sin embargo, el objetivo de *Comprender* suele ser convergente (es decir, alcanzar un único significado). Por el contrario, el objetivo de *generar* dentro de *Crear* es divergente (es decir, alcanzar varias posibilidades).” (pág. 86)

---

- 6.2 Planificar: “*Planificar* conlleva concebir un método que cumple los criterios de un problema, es decir, desarrollar un plan para resolver el problema. (...) Al *planificar*, un alumno puede determinar subobjetivos o dividir una tarea en subtareas que deben realizarse para resolver el problema.”

Ejemplo: diseñar un algoritmo que resuelva un problema; diseñar una secuencia de pasos en la interacción del usuario para realizar una función de un programa.

- 6.3 Producir: “*Producir* conlleva realizar un plan, que cumple ciertas especificaciones, para resolver un problema dado.”

Ejemplo: implementar un algoritmo en un lenguaje de programación dado; concretar los diálogos y otras técnicas de interacción necesarias para la interacción del usuario en una función de un programa.

---

CITA (pág. 87):

“(...) los objetivos en la categoría *Crear* pueden o no incluir la originalidad o la unicidad en su especificación. Lo mismo ocurre con los objetivos de *producir*. *Producir* puede requerir la coordinación de los cuatro tipos de conocimiento.”

---

### 3.3 Ejemplos en informática de la dimensión del conocimiento

En este subapartado se dan ejemplos de los tipos de conocimiento factual, conceptual y procedimental en informática, sobre todo ejemplos de programación. El tipo metacognitivo no es considerado, por ser independiente de la materia. Para cada tipo o subtipo, damos su definición según Anderson *et al.*

(2014) y después damos ejemplos tomados preferentemente de programación. La taxonomía no proporciona ningún ejemplo informático en la dimensión del conocimiento.

**A.** Conocimiento factual: “los elementos básicos que los alumnos deben conocer para estar familiarizados con una disciplina o resolver problemas en ella”. Este tipo de conocimiento se divide en dos subtipos:

**Aa.** Conocimiento de terminología: “conocimiento verbal y no verbal sobre una determinada materia. Está relacionado con el vocabulario técnico”. Forman el vocabulario con el que se entienden los expertos de una disciplina. Este vocabulario es el resultado de un acuerdo (tácito) entre los expertos de la disciplina. La terminología incluye hechos, datos y conceptos.

Ejemplos: Los conceptos de programación “instrucción”, “asignación” y “operador infijo” y su simbología en distintos lenguajes de programación. Tres ejemplos de Java<sup>9</sup> son el delimitador de fin de instrucción ‘;’, el símbolo de asignación ‘=’ y el operador de igualdad ‘==’. Otros ejemplos son las palabras clave o las formas válidas de escribir caracteres en un lenguaje de programación.

**RECUERDE:** Cuando se espera que un alumno sea capaz de reconocer los elementos que aparecen en un programa, nos referimos a este tipo de conocimiento.

**RECUERDE:** El subtipo “conocimiento de terminología” agrupa hechos (como las palabras claves “while” y “for”) y conceptos (como el concepto de bucle).

**Ab.** Conocimiento de detalles y elementos específicos: “conocimiento sobre sucesos, lugares, gente, fechas, fuentes de información y similares”. No son el resultado de ningún acuerdo entre expertos, pero enriquecen el conocimiento de la disciplina.

Ejemplos: los precursores de la informática Ada Lovelace y Alan Turing; URLs y blogs donde puede encontrarse información sobre Java; el nombre del intérprete de Java (Java Virtual Machine, JVM); entornos de programación (como BlueJ). También entran en esta categoría los elementos concretos de un programa dado (variables, métodos, etc.). Un último ejemplo es el rango de valores del tipo de datos ‘int’; obsérvese que mientras que el nombre del tipo de datos es terminología, el rango de valores son detalles adicionales.

**RECUERDE:** El subtipo “conocimiento de detalles y elementos específicos” no es imprescindible, pero enriquece el conocimiento de una materia. Cada profesor hará una elección distinta de la cantidad de conocimiento de este subtipo a incluir.

**B.** Conocimiento conceptual: “conocimiento de categorías y clasificaciones y las relaciones entre ellas”. Este tipo de conocimiento no incluye hechos y conceptos aislados, sino que representa el conocimiento que una persona tiene de cómo se organiza una disciplina.

**Ba.** Conocimiento de clasificaciones y categorías: “las categorías, clases, divisiones y ordenamientos específicos que se hacen en diferentes materias”. Las clasificaciones y categorías se diferencian de la terminología y los elementos en que relacionan elementos específicos. Anderson *et al.* (2014, pág. 49) reconocen que a veces es difícil distinguir entre el conocimiento de clasificaciones y categorías (Ba) y conocimiento factual (A).

---

<sup>9</sup> Por brevedad en las explicaciones, usaremos siempre ejemplos tomados del mismo lenguaje de programación (hemos elegido Java).

Ejemplos: la diferencia entre tipos de datos primitivos y definidos por el usuario; las distintas clases de bucles; las partes de un método; tipos de recursividad (lineal, múltiple, anidada, etc.); la relación entre los conceptos de clase, superclase y subclase;

**RECUERDE:** La comprensión de la estructura de un fragmento de código remite a este tipo de conocimiento.

**Bb.** Conocimiento de principios y generalizaciones: “se usan para estudiar fenómenos o resolver problemas en la disciplina”. Se basan en clasificaciones y categorías. Expresan definiciones, explicaciones o propiedades.

Ejemplos: las características de una definición recursiva bien fundada; la declaración de métodos estáticos y dinámicos en una clase; los principales tipos abstractos de datos (pila, cola, árbol, grafo, etc.); las principales técnicas de diseño de algoritmos (divide y vencerás, vuelta atrás, etc.).

**Bc.** Conocimiento de teorías, modelos y estructuras: “incluyen conocimiento de principios y generalizaciones junto con sus relaciones, presentando una visión clara, rotunda y sistemática de un fenómeno problema o materia complejo”. Se basa en principios y generalizaciones, de forma que se relacionan de alguna manera para formar una teoría, modelo o estructura. Sin embargo, los principios y generalizaciones de la categoría Bb no tienen por qué estar relacionados de ninguna forma.

Ejemplos: la especificación estática de un lenguaje de programación (léxico, sintaxis y semántica estática); la teoría de los tipos abstractos de datos; el lenguaje de modelado orientado a objetos UML; la teoría de gramáticas formales; la teoría de la complejidad computacional.

- C.** Conocimiento procedimental: “el conocimiento de habilidades, algoritmos, técnicas y métodos que son específicos de una materia o disciplina”. Frente a los conocimientos factual y conceptual, que representan el “qué” del conocimiento, el conocimiento procedimental representa el “cómo” del conocimiento. “El conocimiento procedimental puede expresarse como una serie o secuencia de pasos, en conjunto conocidos como procedimiento”. En otras palabras, los conocimientos factual y conceptual tratan de “productos”, frente al conocimiento procedimental, que trata de “procesos”.

**Ca.** Conocimiento de habilidades y algoritmos específicos de una materia. Incluye procedimientos que generalmente producen un resultado fijo. A veces, los pasos del procedimiento se siguen en un orden establecido, mientras que en otros casos debe decidirse qué paso realizar a continuación. Ejemplos: conocimiento de la semántica operacional de un lenguaje de programación; conocimiento del modelado de un problema a resolver algorítmicamente como cabecera de método; conocimiento del método para analizar la complejidad de un algoritmo recursivo.

**RECUERDE:** Se utiliza este subtipo de conocimiento siempre que se realiza una traza o prueba de escritorio de un algoritmo. Puede usarse para muy diversas tareas: determinar la causa de un error en tiempo de ejecución, conocer el valor devuelto por una llamada a un método, etc.

**RECUERDE:** Debe distinguirse entre un método para realizar una tarea de programación (es un proceso y por tanto conocimiento procedimental propio de la informática) y un algoritmo que resuelve un problema (es un producto y por tanto conocimiento factual o conceptual).

**Cb.** Conocimiento de técnicas y métodos específicos de una materia. Son métodos generales que constan de varios pasos, pero cuyo diseño puede variar dependiendo de variados factores. Por tanto, su resultado no es fijo.

Ejemplo: conocimiento de métodos de prueba de programas; conocimiento de la metodología del análisis de complejidad de algoritmos; conocimiento de las técnicas de especificación e implementación de tipos abstractos de datos; conocimiento de metodologías de modelado orientado a objetos; conocimiento de métodos de evaluación de la usabilidad de un programa.

**RECUERDE:** Normalmente corresponden a este subtipo de conocimiento las técnicas y métodos informáticos de carácter abierto conocidos como “metodologías”.

**Cc.** Conocimiento de criterios para determinar cuándo utilizar procedimientos apropiados: “además de conocer procedimientos específicos de la materia, se espera que los alumnos sepan *cuándo* usarlos, lo que frecuentemente conlleva conocer las formas en que se han usado en el pasado”. Los criterios de aplicación suelen ser más sencillos que los propios procedimientos, pero son importantes para un uso adecuado.

Ejemplo: conocimiento de las condiciones para que el análisis de complejidad de un algoritmo exija distinguir entre casos mejor, peor y medio; conocimiento de las condiciones bajo las que un problema puede resolverse con alguna técnica de diseño de algoritmos; conocimiento de criterios para determinar un método adecuado de evaluación de la usabilidad de un programa.

## 4. USO PRÁCTICO DE LA TAXONOMÍA DE BLOOM

En este apartado se dan varios consejos y pautas para usar la taxonomía de Bloom en asignaturas de informática. Comenzamos con varios consejos generales y continuamos con una recomendación de los pasos a seguir para clasificar tareas de evaluación usando la taxonomía de Bloom.

### 4.1 Recomendaciones generales

La taxonomía de Bloom presenta numerosas dificultades para su uso (Masapanta Carrión & Velázquez Iturbide 2018). Su naturaleza es variada, desde malentendidos sobre la misma a interpretaciones equivocadas. Algunas de estas dificultades ya fueron señaladas por sus autores (Bloom *et al.*, 1956; Anderson *et al.*, 2014), pero no se han tenido en cuenta. Algunas ya se han comentado en los apartados anteriores.

Veamos algunas aclaraciones destacadas:

- La familiarización y uso de la taxonomía de Bloom lleva tiempo. Igual que nadie presupone que aprender una taxonomía de seres vivos se aprende inmediatamente, tampoco lo es el aprendizaje de nuestra taxonomía. Recuérdese que propone 11 subtipos de conocimiento y 19 procesos cognitivos.
- Aunque la taxonomía da una jerarquía de categorías de procesos cognitivos de creciente complejidad, no debe tomarse como una jerarquía estricta. Pueden encontrarse procesos cognitivos de una categoría que son más complejos que otros procesos de una categoría superior. Por ejemplo, es más complejo el proceso de explicar algo (correspondiente a la categoría Comprender) que ejecutar un procedimiento fijo (correspondiente a la categoría Aplicar).
- La clasificación de una actividad de evaluación según la dimensión del proceso cognitivo debe tener en cuenta el nivel de conocimientos previo de los alumnos y la posible realización de actividades similares en la asignatura. En ambos casos, es probable que el alumno se mantenga en la categoría de Recordar, pero si el contenido es desconocido para el alumno, probablemente necesite Analizar, Implementar y Producir, ya que el estudiante debe tomar partes del conocimiento aprendido para generar la solución.
- Al intentar clasificar un objetivo, actividad de instrucción o actividad de evaluación, es preferible centrarse en el proceso cognitivo que en la categoría del proceso cognitivo.

- La clasificación de un objetivo educativo según la dimensión del proceso cognitivo no debe hacerse de forma automática por el verbo. En ciertos casos, el verbo principal puede tener más de una posibilidad de clasificación. Un ejemplo dado por Anderson *et al.* (2014, pág. 14) es el siguiente: “Los estudiantes podrán listar tres razones para el surgimiento del comunismo en Europa del Este”. El verbo “listar” puede ser clasificado como Recordar (una lista dada por el profesor) o como Analizar (varias fuentes de información y elaborar una lista). A esto hay que añadir el distinto significado de varios verbos de procesos cognitivos entre la taxonomía de Bloom y el uso común o el uso informático (véase el apartado 3.1 de esta guía). Por ejemplo, el verbo “explicar” tiene un uso diferente en la taxonomía con respecto al uso común y “analizar” la complejidad de un algoritmo consiste en aplicar un método de análisis de algoritmos. También es causa de confusiones la ambigüedad de muchos verbos, como “estimar” (Anderson *et al.*, 2014, pág. 106). En este caso, es necesario considerar la combinación verbo-sustantivo para una clasificación adecuada, por ejemplo, “estimar el producto de dos números” (Anderson *et al.*, 2014, págs. 106-107).
- La forma de clasificar un objetivo educativo, actividad de instrucción o actividad de evaluación varía entre las tres categorías inferiores de procesos cognitivos (Recordar, Comprender y Aplicar) y las tres categorías superiores (Analizar, Evaluar y Crear). En objetivos correspondientes a las categorías inferiores, el sustantivo tomado como tipo de conocimiento es el asociado al verbo del proceso cognitivo. Por ejemplo, “Dar un ejemplo de bucle for cuya ejecución no termine”. Sin embargo, en objetivos correspondientes a las categorías superiores, el sustantivo es el criterio utilizado para realizar el proceso cognitivo del verbo. Por ejemplo, “Evaluar la calidad de un algoritmo según su estilo de programación”.
- En general, un objetivo, actividad de instrucción o actividad de evaluación implica varios procesos cognitivos, salvo que sea muy sencillo. “Aunque hemos descrito los procesos cognitivos individualmente, probablemente se usarán de forma coordinada para facilitar aprendizaje escolar significativo. La mayoría de las tareas académicas requieren el uso coordinado de varios procesos cognitivos, así como varios tipos de conocimiento. Por ejemplo, para resolver un problema matemático en palabras, un alumno puede implicarse en:
  - *interpretar* (comprender cada frase del problema);
  - *recuperar* (recuperar el *Conocimiento factual* necesario para resolver el problema);
  - *organizar* (construir una representación coherente de la información clave del problema, es decir, *Conocimiento conceptual*);
  - *planificar* (diseñar un plan para la solución);
  - *producir* (llevar a cabo el plan, es decir, *Conocimiento procedimental*)” (Anderson *et al.* 2014, pág. 89).

#### 4.2 Pasos para la clasificación de actividades de evaluación

La clasificación de actividades de evaluación es una tarea compleja y ambigua, donde es normal que diferentes profesores realicen clasificaciones distintas. Con el objetivo de reducir esta disparidad de criterios, proponemos un procedimiento consistente en varios pasos, aunque puede simplificarse. Varios fueron señalados de forma implícita por los autores de la taxonomía revisada.

##### 0) Descomponer el enunciado en subejercicios indivisibles.

Frecuentemente, el enunciado de un ejercicio es compuesto, esto es, contiene varios subejercicios que resolver. Cada subejercicio se clasificará por separado, siguiendo los pasos siguientes. Esta situación puede verse en el Ejemplo 1 del Apartado 5.

1) Identificar las actividades cognitivas de un subejercicio indivisible.

El profesor debe determinar la secuencia de actividades cognitivas que espera que realicen los alumnos para resolver cada subejercicio identificado anteriormente. Una actividad cognitiva es una actividad catalogable de forma indivisible como proceso cognitivo. La resolución de un ejercicio puede conllevar la realización de una o varias actividades cognitivas, correspondientes a uno o varios procesos cognitivos.

Hay que tener cuidado en que los alumnos pueden resolver un mismo problema de varias formas, dependiendo de la experiencia que tengan sobre dicha actividad y su contenido. En algunos casos, puede distinguirse las actividades cognitivas esperadas de un alumno principiante y las esperadas de un alumno experto.

2) Clasificar cada actividad cognitiva.

Cada actividad cognitiva identificada anteriormente debe clasificarse en la matriz taxonómica, lo cual significa que debe clasificarse según proceso cognitivo y subtipo de conocimiento. Para la clasificación en subtipo de conocimiento, se identifica el sustantivo más representativo de la actividad cognitiva. Para la clasificación en proceso cognitivo, se identifica el verbo principal de la actividad cognitiva. Recordemos que no hacerse una clasificación automática del verbo, sino que debe analizarse.

El resultado de ambas clasificaciones debe incluirse en la celda correspondiente de la matriz taxonómica. Si no se está usando la matriz, pueden escribirse en un formato que permita identificar la categoría y subcategoría en cada dimensión. Una forma sencilla de escribir dicha clasificación es:

Categoría de proceso cognitivo (Proceso cognitivo) – Tipo de conocimiento (Literal del subtipo) donde el literal del subtipo será una sola palabra, por simplificar.

Por ejemplo, una clasificación es:

Comprender (Explicar) – Conceptual (Clasificaciones)

donde “Clasificaciones” es una simplificación de “Conocimiento de clasificaciones y categorías”.

**RECUERDE:** La identificación del sustantivo es distinto entre las categorías de Recordar, Comprender y Aplicar, por un lado, y Analizar, Evaluar y Crear, por otro.

Hay que hacer una distinción importante entre ambos grupos de categorías. Las categorías Recordar, Comprender o Aplicar presentan dos características propias. En primer lugar, es probable que haya una correspondencia directa entre la categoría de proceso y el tipo de conocimiento en objetivos clasificados como Recordar, Comprender o Aplicar: recordar conocimiento factual, comprender conocimiento conceptual y aplicar conocimiento procedimental. En segundo lugar, el sustantivo al que se refiere el verbo es el complemento directo de la frase.

Sin embargo, las categorías Analizar, Evaluar y Crear presentan otras características. En primer lugar, el tipo de conocimiento es menos predecible. En segundo lugar y más importante, el sustantivo al que se refiere el verbo no es el complemento directo sino un complemento circunstancial. Los objetivos correspondientes a estas categorías deben tener la estructura “los alumnos podrán hacer algo *basándose en o en función de* criterios”. Por ejemplo, “los alumnos serán capaces de verificar si un algoritmo es correcto [con respecto a su especificación]”. En el

ejemplo, tenemos el proceso cognitivo Chequear (de la categoría Evaluar). Sin embargo, el tipo de conocimiento no es la respuesta calculada ni el propio algoritmo analizado (en ambos casos, conocimiento factual, de subtipo detalles específicos), sino los criterios utilizados para realizar la evaluación (conocimiento conceptual, de subtipo criterios). Hemos encerrado entre corchetes los criterios usados porque probablemente no se escriban en un primer momento, pero deberían hacerse explícitos al clasificar el objetivo.

CITA (resaltado en el texto original):

"Considere, por ejemplo, *evaluar conocimiento conceptual*. Por lo general, no pretendemos que los alumnos aprendan a *criticar (Evaluar)* un conjunto de criterios (*Conocimiento conceptual*). En su lugar, intentamos que aprendan a *criticar algo basándose en o en función de* los criterios. (...) Por lo tanto, *evaluar conocimiento conceptual* se convierte en esencia en *evaluar [basándose en] conocimiento conceptual o evaluar [en función de] conocimiento conceptual*." (pág. 107).

### 3) Identificar los procesos cognitivos principales (discrecional).

Como hemos visto en el paso 2, algunas actividades de evaluación constarán de varias actividades cognitivas que probablemente corresponderán a varios procesos cognitivos. El profesor decidirá que procesos cognitivos son los más importantes para la evaluación.

A continuación, se dan algunas recomendaciones:

- a) Seleccionar aquellos procesos cognitivos que reflejen de mejor manera la intención de la pregunta de evaluación (problema).
- b) Elegir los procesos cognitivos que el profesor considere más importantes en su evaluación.
- c) Considere que en cualquier actividad en la que se proporcione código habrá que realizar alguna tarea de recordar el lenguaje de programación o de analizar el código. Para simplificar el profesor puede establecer estas actividades como implícitas permitiéndole enfocarse en las más significativas.

## 5. EJEMPLOS DE CLASIFICACIÓN DE EJERCICIOS DE PROGRAMACIÓN

En este apartado se exponen algunos ejemplos de uso de la guía para clasificar ejercicios. Aunque la taxonomía revisada enfatiza la planificación de la docencia y la comprobación de alineamiento entre objetivos educativos, actividades instruccionales y pruebas de evaluación, nos centramos en la clasificación de pruebas de evaluación por lo extendido de su uso, al menos en la literatura (Masapanta Carrión y Velázquez Iturbide 2018).

Los ejemplos son preguntas usadas en evaluaciones de programación en primer año de la carrera de informática y se han extraído de artículos, que se citan. Los ejemplos se presentan sin ningún orden en especial. Cada ejemplo incluye una breve indicación sobre el contexto educativo y la experiencia de los alumnos. En algunos ejemplos, se hacen dos análisis distintos, dependiendo de la maestría de los alumnos en la materia. La clasificación se realiza siguiendo los pasos anteriores.

### 5.1 Ejemplo 1

Tomado de Shuhidan *et. al.* (2009).

Contexto educativo: Los estudiantes han visto esta información previamente.

*Which one of the following statements is true?*

- a) *An abstract class must have an abstract method*
- b) *All methods in an abstract class must themselves be abstract*
- c) *An abstract class cannot define instance variables*
- d) *An abstract class cannot be instantiated*

PASO 1, identificar las actividades cognitivas:

- 1ª “Identificar la propiedad que caracteriza a una clase abstracta”.

PASO 2, clasificar las actividades cognitivas:

- 1ª El proceso de identificación depende de que los alumnos hayan visto este tipo de actividad previamente. Se afirma que los alumnos han visto esta información previamente, por lo que no es necesaria ninguna clase de comprensión, sino simplemente su recuerdo. Por otro lado, el sustantivo es “propiedad que caracteriza a una clase abstracta”, que es un conocimiento conceptual asociado con un tipo específico de clase. Por tanto, es conocimiento sobre clasificaciones y categorías. La clasificación resultante es:

Recordar (Reconocer) – Conceptual (Clasificaciones)

En este caso, sólo hay un proceso cognitivo que caracteriza la prueba de evaluación.

## 5.2 Ejemplo 2

Tomado de Thompson *et. al.* (2008).

Contexto educativo: En clase los estudiantes han trabajado con ejemplos similares.

```
public class Circle
{
    private int diameter;
    private int xPosition;
    private int yPosition;
    private String color;
    private boolean isVisible;
    public Circle()
    {
        diameter = 30;
        xPosition = 20;
        yPosition = 60;
        color = "blue";
        isVisible = false;
    }
    //code removed for brevity
}
```

*Write a constructor that would allow the location, colour, and diameter of the circle to be set. Show how this constructor would be used to create a circle at x = 200 and y = 400, with colour blue, and diameter = 90.*

PASO 0, descomponer el enunciado en subejercicios indivisibles:

- a) Escribir un constructor que permita fijar la posición, color y diámetro del círculo.
- b) Usar este constructor para crear un círculo en x=200 e y=400, de color azul y diámetro 90.

### Subejercicio (a): Declarar un constructor



PASO 1, identificar las actividades cognitivas:

Analizando el problema, proponemos la siguiente forma de resolver el apartado (a). Reescribimos la primera actividad para facilitar su posterior clasificación, pero no es necesario para las otras dos.

- 1ª El alumno debe ser capaz de identificar el léxico y la sintaxis propia del lenguaje de programación utilizado (Java). Esta actividad puede expresarse como “Reconocer las partes del código”.
- 2ª El estudiante debe analizar el código y determinar su estructura.
- 3ª El alumno debe declarar un constructor *Circle* con cuatro parámetros (usando dicho concepto).

PASO 2, clasificar cada actividad cognitiva:

- 1ª “Reconocer el código”. “Reconocer algo” corresponde al nivel cognitivo de Recordar. Por otro lado, identificar la sintaxis, el léxico y la semántica de un lenguaje de programación corresponde al tipo de conocimiento Factual (terminología). Por tanto, su clasificación queda:  
Recordar (Reconocer) – Factual (Terminología)
- 2ª Se analizan las distintas partes del código para determinar “cómo funcionan por separado y en conjunto” (proceso cognitivo Organizar). El análisis se realiza utilizando el conocimiento de la forma de estructurar clases en Java, que corresponde al subtipo de conocimiento de “principios y generalizaciones”. La clasificación queda:  
Analizar (Organizar) – Conceptual (Generalizaciones)
- 3ª Se pide declarar un constructor. Se puede considerar que es una tarea familiar que puede realizarse usando dicho concepto.  
Aplicar (Ejecutar) – Conceptual (Clasificaciones)

PASO 3, identificar los procesos cognitivos principales:

El enunciado pide que el alumno demuestre que sabe crear el constructor de una clase. Por tanto, el proceso cognitivo más importante es el correspondiente a Aplicar. Las otras dos actividades cognitivas (Reconocer y Analizar) son necesarias para poder resolver el problema. Por tanto, queda al criterio del profesor tenerlas o no en cuenta para valorar el rendimiento del alumno en estos procesos cognitivos del alumno.

### **Subejercicio (b): Realizar una llamada al constructor**

PASO 1, identificar las actividades cognitivas:

Si este subejercicio no estuviera precedido por el subejercicio (a), se iniciaría su resolución con las dos actividades cognitivas identificadas en (a), es decir, reconocer el código y analizarlo para determinar su estructura. Sin embargo, estas tareas ya se han realizado, por lo que solamente se realiza una actividad cognitiva, la correspondiente a escribir una llamada al constructor *Circle* con cuatro parámetros concretos.

PASO 2, clasificar las actividades cognitivas:

Es similar a la actividad cognitiva de la parte (a), pero utilizando el concepto de “creación de objeto” o “llamada a constructor”, en lugar de “declaración de constructor”. Por tanto:

Aplicar (Ejecutar) – Conceptual (Clasificaciones)

Para este subejercicio, sólo hay un proceso cognitivo implicado.

## **5.3 Ejemplo 3**

Tomado de Lister (2000).

Contexto educativo: En clase, los estudiantes han trabajado con ejemplos similares. Existe otra posibilidad de clasificación, dependiendo de la experiencia de los alumnos.

*What output is produced by the following Java code:  
for (int i = 0; i<=3; i+=2) System.out.print (i+" " " ");*

- a) 0 1 2
- b) 0 1 2 3
- c) 0 2
- d) 0 2 3
- e) 0 2 4.

Este problema, tiene en común con el anterior que incluye un fragmento de código que hay que leer y comprender antes de responder a la pregunta. Por tanto, el alumno realizará las dos actividades cognitivas identificadas y que escribiremos sin comentarlas. En los siguientes ejercicios también serán las dos primeras actividades cognitivas, pero no las identificaremos por sencillez.

PASO 1, identificar las actividades cognitivas:

- 1ª “Reconocer el código”.
- 2ª “Analizar la estructura del código”.
- 3ª “Ejecutar el código, guardando un registro de los valores impresos por pantalla”.

PASO 2, clasificar las actividades cognitivas:

- 1ª  
Recordar (Reconocer) – Factual (Terminología)
- 2ª  
Analizar (Organizar) – Conceptual (Generalizaciones)
- 3ª El verbo de esta actividad es ejecutar. En este caso, coincide con el significado informático de “ejecutar”: aplicar la semántica operacional del lenguaje Java para determinar la salida producida. Se trata de una tarea familiar y que siempre debe producir el mismo resultado. Por otro lado, la semántica operacional del lenguaje es un “algoritmo” específico:  
Aplicar (Ejecutar) – Procedimental (Algoritmos)

PASO 3, identificar los procesos cognitivos principales:

Según lo solicitado en la pregunta, el alumno debe haber aplicado correctamente el procedimiento para seleccionar la respuesta correcta. Por tanto, el proceso cognitivo más relevante es el tercero:

Aplicar (Ejecutar) – Procedimental (Algoritmos)

**OTRA POSIBLE CLASIFICACIÓN:**

Es posible concebir que el problema se resuelva realizando otras actividades cognitivas. El alumno puede intentar “comprender el comportamiento del código” (es decir, cómo funciona el bucle) en lugar de realizar una ejecución “ciega”. En este caso, tras reconocer y analizar la organización del código, la tercera actividad consistiría en desarrollar un modelo de comportamiento del código. Por tanto, la tercera actividad cognitiva la clasificamos como:

Comprender (Explicar) – Conceptual (Generalizaciones)

## 5.4 Ejemplo 4

Tomado de Whalley *et. al.* (2006).

Contexto educativo: Suponemos que los estudiantes han trabajado con ejemplos similares en clase.

*In plain English, explain what the following segment of code does:*

```
bool bValid = true;
for (int i = 0; i < iMAX-1; i++)
{
    if (iNumbers[i] > iNumbers[i+1])
    {
        bValid = false;
    }
}
```

PASO 1, identificar las actividades cognitivas:

- 1ª “Comprender lo que realiza el código”.
- 2ª “Resumir en inglés, el efecto del código”.

PASO 2, clasificar las actividades cognitivas:

- 1ª El estudiante debe desarrollar una representación mental o modelo de lo que realiza el código (Explicar). Dicho modelo se definirá en término de los “conceptos” identificados tras el análisis del código, expresando una generalización, que es un conocimiento conceptual (generalizaciones, Bb):  
Comprender (Explicar) – Conceptual (Generalizaciones)
- 2ª El alumno debe “resumir” el modelo en lenguaje natural. La clasificación es:  
Comprender (Resumir) – Conceptual (Generalizaciones)

PASO 3, identificar los procesos cognitivos principales:

El problema pide al estudiante que escriba cuál es la finalidad del algoritmo entregado, por lo que el proceso cognitivo más importante es el correspondiente a resumir:

Comprender (Resumir) – Conceptual (Generalizaciones)

No obstante, el profesor puede considerar también importante el proceso cognitivo de explicar porque implica que el estudiante comprende y da sentido al código.

## 5.5 Ejemplo 5

Tomado de Whalley *et. al.* (2006).

Contexto educativo: Los estudiantes no han visto ejemplos similares.

*The following segment of code was intended to add the elements of the array iNumbers, from left to right, until the sum of those elements is greater than the value stored in the variable iLimit:*

```
int iNumbers[iMAX] = {...some values here..};
int iLimit = 3;
int iIndex = 0;
int iSum = 0;
while((iSum <= iLimit) && (iIndex < iMAX))
{
    iIndex = iIndex + 1;
    iSum = iSum + iNumbers[iIndex];
}
```

*The code was intended to finish with the variable `iIndex` containing the first position in the array where the sum exceeds `iLimit`, and `iSum` containing the sum of the array elements from `iNumbers[0]` to `iNumbers[iIndex]` inclusive.*

*However the given code is buggy. For example, if `iNumbers` has the values {2, 1, 4, 5, 7}, `iIndex` should be 2 and `iSum` should be 7. Instead, after the above segment of code is executed, `iIndex` equals 2 and `iSum` equals 5.*

*The bug in the above code can be fixed by:*

- a) Replacing `iIndex = 0` with `iIndex = -1`*
- b) Replacing `iSum = 0` with `iSum = -1`*
- c) Replacing `iSum <= iLimit` with `iSum < iLimit`*
- d) Moving `iIndex = iIndex + 1` from above*  
*`Sum = iSum + iNumbers[iIndex]` to below it*

Este problema es una tarea compleja, que será resuelta de forma distinta por un experto (p.ej. un profesor) y por un principiante (un alumno). Suponemos que el alumno realiza en secuencia las distintas actividades del enunciado, aunque esto sólo influye en el orden de las actividades cognitivas que realiza.

PASO 1, identificar las actividades cognitivas:

- 1ª “Comprender la especificación”.
- 2ª “Analizar la estructura del código”.
- 3ª “Comprender el segmento de código para ver su efecto”.
- 4ª “Realizar la prueba de escritorio del ejemplo”.
- 5ª “Determinar los errores lógicos del programa”.
- 6ª Para cada solución, “valorar si cumple con la especificación dada”.

PASO 2, clasificar las actividades cognitivas:

- 1ª El estudiante debe construir el “modelo mental” de este comportamiento (Explicar). Por otro lado, el sustantivo es “la especificación” que permite abstraer “el comportamiento del código”, por lo que es conocimiento conceptual correspondiente a una generalización. Por tanto, tenemos:
  - Comprender (Explicar) – Conceptual (Generalizaciones)
- 2ª “Analizar la estructura del código”. Al igual que en ejemplos anteriores:
  - Analizar (Organizar) – Conceptual (Generalizaciones)
- 3ª Esta actividad es similar a “Comprender lo que realiza el código”, ya que el estudiante desarrolla una representación mental de la finalidad del programa:
  - Comprender (Explicar) – Conceptual (Generalizaciones)
- 4ª Una prueba de escritorio corresponde al proceso cognitivo de Aplicar (Ejecutar). El procedimiento utilizado es la semántica operacional del lenguaje, un conocimiento de tipo procedimental (habilidades y algoritmos).
  - Aplicar (Ejecutar) – Procedimental (Algoritmos)
- 5ª “Determinar errores lógicos en un programa”. El verbo principal “determinar” es un verbo asociado al nivel de análisis. Sin embargo, el análisis se ha realizado en el paso 2, siendo necesario ahora identificar las partes erróneas del código. Por tanto, debe construirse un modelo del comportamiento erróneo, que corresponde al proceso cognitivo Explicar. Por otro lado, el sustantivo es “errores lógicos en un programa”, que exige que el estudiante exprese qué falla mediante una generalización, que es un subtipo de conocimiento conceptual.

### Comprender (Explicar) – Conceptual (Generalizaciones)

Observación: Un experto probablemente no necesita este paso, sino que al estudiar el código ya habrá deducido que no se corresponde con la especificación.

- 6ª Para cada solución, “valorar si cumple con la especificación dada”. En principio, cada solución debería someterse a dos procesos análogos al 2º y 3º pero, dado que ya se ha realizado con la solución original, tienen poco peso. La actividad principal es “valorar si cada solución cumple con la especificación dada”.

### Evaluar (Comprobar) – Conceptual (Generalizaciones)

PASO 3, identificar los procesos cognitivos principales:

Este problema conlleva cierta complejidad para el alumno y también para quien lo clasifique usando la taxonomía de Bloom. Se tienen 6 actividades cognitivas, que dependiendo de la intencionalidad de la pregunta o del profesor pueden variar los procesos cognitivos relevantes. Por ejemplo, si consideramos el objetivo de la pregunta (escoger la respuesta correcta) los procesos cognitivos de las actividades cognitivas quinta y sexta son los más importantes. Sin embargo, otro profesor también puede considerar importantes otras actividades cognitivas, como la tercera o incluso la primera.

## 5.6 Ejemplo 6

Tomado de Lister (2000).

Contexto educativo: Los estudiantes no han visto este tipo de pregunta antes.

```
public static int maxPos (int[] y, int first, int last) {
  /* Returns the position of the maximum element in the subsection of the array "y", starting at
  * position "first" and ending at position "last" */
  int bestSoFar=first;
  ***missing code goes here ***
  return bestSoFar;
}

a) for (int i=last; i>first; i--)
    if (y[i] < y[bestSoFar])
        bestSoFar = i;
b) for (int i=first+1; i<=last; i--)
    if (y[i] < y[bestSoFar])
        bestSoFar = i;
c) for (int i=last; i>first; i--)
    if (y[i] < bestSoFar)
        bestSoFar = i;
d) for (int i=last; i>first; i--)
    if (y[i] > y[bestSoFar])
        bestSoFar = i;
e) for (int i=first+1; i<=last; i--)
    if (y[i] > bestSoFar)
        bestSoFar = i;
```

PASO 1, identificar las actividades cognitivas:

- 1ª “Comprender la especificación del problema y el fragmento de código proporcionado”.
- 2ª Para cada opción, “analizar su segmento de código”.
- 3ª Para cada opción, “comprender su segmento de código”
- 4ª Para cada opción, “evaluar si cumple con lo solicitado en el enunciado”.

PASO 2, clasificar las actividades cognitivas:

- 1ª “Comprender la especificación del problema y el fragmento de código proporcionado”. Su proceso cognitivo es el mismo que se ha explicado en otros ejemplos:  
Comprender (Explicar) – Conceptual (Generalizaciones)
- 2ª Para cada opción, “analizar su segmento de código”:  
Analizar (Organizar) – Conceptual (Generalizaciones)
- 3ª Para cada opción, “comprender su segmento de código”:  
Comprender (Explicar) – Conceptual (Generalizaciones)
- 4ª Para cada opción, “evaluar si cumple con lo solicitado en el enunciado”.  
Evaluar (Comprobar) – Conceptual (Generalizaciones)

PASO 3, identificar los procesos cognitivos principales:

Si se considera que se debe seleccionar la respuesta correcta, el proceso más importante es: Evaluar (Comprobar) – Conceptual (Generalizaciones). No obstante, este paso es trivial una vez que se ha comprendido el comportamiento de su fragmento de código. Por tanto, es probable que el profesor decida que el proceso cognitivo más importante es: Comprender (Explicar) – Conceptual (Generalizaciones).

Obsérvese también que hay otras formas de proceder. Si fuera un alumno avanzado, podría concebir primero la solución y después compararla con las proporcionadas para elegir la que fuera igual.

## REFERENCIAS

- Anderson, L.W., Krathwohl, D.R., Airasian, P.W., Cruikshank, K.A., Mayer, R.E., Pintrich, P.R., Raths, R., Wittrock, M.C. (2014) *A Taxonomy for Learning, Teaching and Assessing. A Revision of Bloom's Taxonomy of Educational Objectives*, Londres, Gran Bretaña: Pearson Education.
- Biggs, J.B., Collis, K.F. (1982) *Evaluating the Quality of Learning: The SOLO Taxonomy (Structure of the Observed Learning Outcome)*, Academic Press.
- Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill W.H., Krathwohl D.R. (1956) *Taxonomy of Educational Objectives: Handbook I: Cognitive Domain*, Longman.
- Lister, R. (2000) “On blooming first year programming, and its blooming assessment”, *Proceedings of the Australasian Conference on Computing education ACM*, pp. 158-162.
- Masapanta Carrión, S., Velázquez Iturbide, J.Á. (2018) “A systematic review of the use of Bloom’s taxonomy in computer science education”, *Proceedings of the 49th ACM Technical Symposium on Computing Science Education, SIGCSE’18*, ACM Press, págs. 441-446. DOI [10.1145/3159450.3159491](https://doi.org/10.1145/3159450.3159491).
- Shuhidan, S., Halmilton, M., D'souza, D. (2009) “A taxonomic study of novice programming summative assessment”, *Proceedings of the Eleventh Australasian Conference on Computing Education*, vol. 95, pp. 147-156.
- Thompson, E., Luxton-Reilly, A., Whalley, J.L., Hu, M., Robbins, P. (2008) “Bloom's taxonomy for CS assessment”, *Proceedings of the tenth conference on Australasian computing education*, vol. 78, pp. 155-161.
- Whalley, J.L., Lister, R., Thompson, E., Clear, T., Robbins, P., Kumar, P.K., Prasad, C. (2006) “An Australasian study of reading and comprehension skills in novice programmers, using the bloom and SOLO taxonomies”, *Proceedings of the 8th Australasian Conference on Computing Education*, vol. 52, pp. 243-252.