

Relatos, Mapas Conceptuales y Arquitectura de Computadores

Julio Ortega, Mancia Anguita
Departamento de Arquitectura y Tecnología de Computadores
E.T.S.I.I.T., Universidad de Granada
[jortega,manguita}@ugr.es](mailto:{jortega,manguita}@ugr.es)

Resumen. Este trabajo parte de la hipótesis de que la historia de una disciplina y, más concretamente, los relatos de episodios de su desarrollo contribuyen a conformar los mapas conceptuales que permiten fijar los conceptos esenciales, y a motivar a los estudiantes en el estudio de dicha disciplina. Aquí se proponen algunos episodios relacionados con la evolución de los computadores para incluir en un curso de Arquitectura de Computadores, junto con referencias que permitan acceder a información relativa a los mismos.

Palabras clave: Arquitectura de Computadores, Historia de los computadores, Mapas conceptuales, Motivación

Abstract. This paper stems from the hypothesis that the history of the development of a discipline and, more specifically, the storytelling of episodes of that development contribute to build conceptual maps that make it easy a persistent memorization of the essential concepts along with the motivation of the students towards these concepts. Here, we also propose a set of histories related with the computer evolution to be included in a Computer Architecture course.

Keywords: Computer architecture, Conceptual maps, History of computers, Motivation

1 Introducción

La inclusión de relatos acerca de la forma en que se han planteado problemas esenciales en el desarrollo de una disciplina, y de cómo han ido proponiéndose soluciones, puede incrementar la motivación para el estudio de dicha disciplina. En [1] se analiza la utilidad de la historia de la computación como herramienta pedagógica y la necesidad de incluir asignaturas de este tipo en los planes de estudios. La historia contribuye a que los estudiantes se hagan preguntas fuera de lo que es la dinámica de razonamiento propio de la asignatura y la contemplan dentro del contexto

del desarrollo global de la informática descubriendo relaciones con otras asignaturas de su currículo. Se contribuye así a construir mapas que interrelacionan elementos y facilitan su retención y a ampliar su comprensión del contexto y su pensamiento crítico. La historia puede mejorar la comprensión de las tendencias del desarrollo del hardware y el software al relacionar los computadores con su dimensión humana.

Otra de las razones más aparentes para conocer la historia del desarrollo de una tecnología es la de tener en cuenta los errores cometidos para no repetirlos, y fomentar la capacidad de construir a partir de las aportaciones que han tenido éxito en el pasado. También se puede poner de manifiesto la importancia de disponer de una tecnología suficientemente madura para poder plasmar ideas avanzadas, y la de reflejar las estrategias sociales que han dado lugar a nuevos avances como el papel de los pioneros y emprendedores, el de la cooperación y el trabajo en grupo, etc. En muchos casos, no se tiene una información completa de las respuestas que las empresas o las instituciones han dado a problemas o retos que se han ido planteando, pudiéndose fomentar en este caso la discusión en clase y la curiosidad.

La referencia a los protagonistas más relevantes en el desarrollo de los elementos que se han constituido en los conceptos propios de la disciplina contribuye a definir una comunidad con la que el estudiante puede empezar a identificarse. En esta red social va a encontrar personalidades que pueden definir modelos y trayectorias para su actividad como ingeniero. El acercamiento histórico a la disciplina permite interrelacionar conceptos técnicos y su incidencia social contribuyendo a ubicar el papel del ingeniero dentro de su entorno. Surge así una faceta de motivación digna de consideración, que puede contribuir de manera relevante a mejorar el acercamiento del estudiante a la disciplina en cuestión.

Partiendo de la utilidad de incluir referencias históricas en las asignaturas, no obstante es importante tener en cuenta la forma en que esto se lleva a cabo. Presentar una serie de hechos distribuidos en el tiempo sin más, sería contemplada por los estudiantes como un ejercicio más de memoria que tendría un efecto negativo. A este respecto consideramos acertado el uso del relato o la narración histórica, tal y como se propone en [2], donde se hace referencia a la importancia de esta estrategia dado que los relatos son intrínsecamente interesantes dada nuestra mayor disposición a recordar narraciones más que argumentos lógicos. La idea es aprovechar la tendencia a recordar historias para que los estudiantes también recuerden la información técnica relacionada con la disciplina involucrada en el relato.

Considerando las claves que proporcionan algunas de las teorías de la motivación más relevantes propuestas hasta el momento, se proponen algunas estrategias docentes basadas en el uso de relatos de la evolución de los computadores para motivar el estudio de las asignaturas de tecnología, estructura y arquitectura de computadores, en particular la asignatura de Arquitectura de Computadores, y mejorar la asimilación de los correspondientes mapas conceptuales. Si los estudiantes consideran los contenidos de estas asignaturas asequibles y útiles para su formación como Ingenieros Informáticos es más probable que la Ingeniería de Computadores sea contemplada por algunos como su profesión.

En este artículo se considera una asignatura de introducción a la Arquitectura de Computadores. En la Sección 2 se proporcionan los detalles relativos a los contenidos y competencias de la asignatura que se imparte en el Grado de Ingeniería Informática de la Universidad de Granada, así como un mapa conceptual para la misma, y en la

Sección 3 se considera el modelo de aprendizaje y la importancia de la motivación, junto con la utilidad de los relatos históricos. La Sección 4 presenta, a partir de referencias a una serie de artículos, relatos de la evolución de los computadores y se muestra su relación con los conceptos de la asignatura. Finalmente, la Sección 5 incluye las conclusiones del artículo y la Sección 6 proporciona las referencias.

2 Arquitectura de Computadores y sus mapas conceptuales

En [3] se describen los contenidos y las estrategias docentes de una asignatura de introducción a la Arquitectura de Computadores. Concretamente, se trata de una asignatura de grado que se imparte en el Grado en Ingeniería Informática de la Universidad de Granada. La Figura 1 muestra un mapa conceptual que relaciona los principales conceptos y contenidos que se abordan en dicha asignatura. Para construir dicho mapa se puede partir de uno de los modelos más frecuentemente utilizados para evaluar las prestaciones de un procesador (por ejemplo en [4]). Se trata de un modelo sencillo, y con un nivel de detalle suficiente para nuestro propósito:

$$T_{\text{tarea}} = NI \times CPI \times T_{\text{ciclo}} = NI \times \left(\frac{CPI}{f} \right)$$

donde T_{tarea} es el tiempo que tarda en ejecutarse un programa, NI es el número de instrucciones máquina que lo componen, CPI es el número medio de ciclos por instrucción, y T_{ciclo} el periodo de reloj del procesador (inverso de la frecuencia, f). Así, el estudio de la arquitectura del computador debe referirse a todos aquellos aspectos que determinan los valores de NI , CPI , y f , y su interrelación. Así, NI depende del repertorio de instrucciones y del compilador; CPI viene determinado por el repertorio de instrucciones y la organización del computador; y f depende de las prestaciones que proporcione la tecnología y de la organización del computador. Como se ha indicado, la organización del computador afecta tanto al valor de CPI como a f , ocasionando que realmente la métrica importante sea el cociente CPI/f . Así, por ejemplo, se puede reducir el valor de CPI utilizando frecuencias menores.

El estudio de la organización del computador implica considerar las características de los distintos subsistemas que lo integran (procesador, memoria, dispositivos periféricos y de almacenamiento), tanto desde el nivel de arquitectura como desde el de microarquitectura; así como la interconexión de dichos subsistemas (jerarquía de buses), y los flujos mutuos de datos y control (comunicación y sincronización) que permiten el funcionamiento del computador. Por tanto, desde una perspectiva de ingeniero, el estudio de la Arquitectura del Computador debe incluir las interacciones y compromisos entre los distintos componentes del computador para conseguir el nivel de prestaciones requerido. Esto nos lleva a estudiar las distintas alternativas de mejora de prestaciones, teniendo en cuenta la relación entre las características de los elementos de una arquitectura y el nivel de prestaciones alcanzado, a través de distintos modelos.

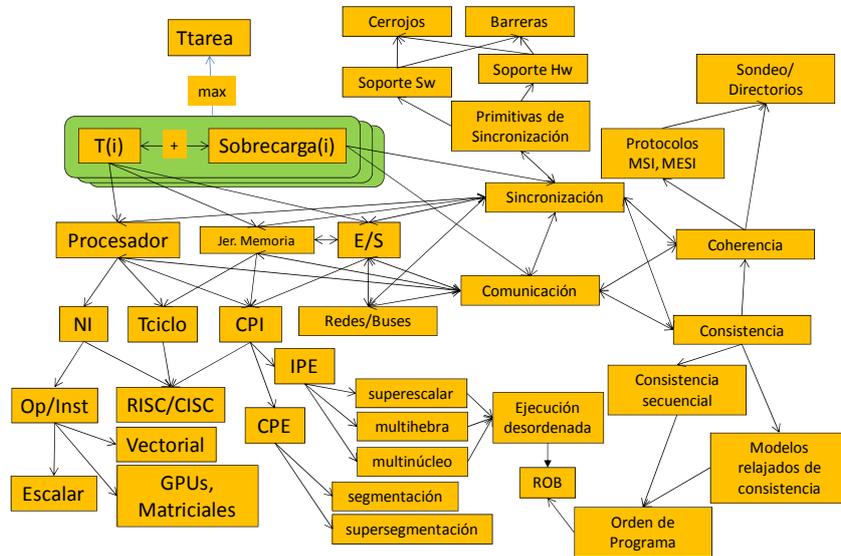


Figura 1. Un mapa conceptual para una asignatura de introducción a la Arquitectura de Computadores

La arquitectura traduce a capacidades reales y prestaciones las posibilidades que ofrece la tecnología de integración y fabricación de circuitos integrados. La forma en que esto se ha llevado a cabo a lo largo de la evolución de los computadores (tanto en el pasado como en un futuro previsible) se puede resumir en paralelismo y localidad. El paralelismo, aplicado a distintos niveles, permite reducir el tiempo por la realización simultánea de operaciones, y la localidad de los datos permite reducir las latencias de acceso a los mismos por parte del procesador, que evita accesos a niveles de la jerarquía de memoria más profundos y lentos. No obstante, tanto para implementar el paralelismo como para mejorar la localidad hay que utilizar recursos, sobre cuya distribución debe decidir el diseñador, en base a la comprensión de los efectos posibles en las prestaciones del sistema. Esta interesante interacción paralelismo-localidad se da en el computador a todas las escalas, tiene efectos en los elementos y problemas que se plantean (por ejemplo la coherencia de cache, o la necesidad de estructuras de comunicación entre procesadores, procesador y memoria, etc.), y permite exponer los conceptos esenciales al diseñar/evaluar/utilizar un computador. Por lo tanto, tienen un papel central en el estudio de la arquitectura del computador.

El paralelismo se puede aprovechar de formas diferentes (paralelismo funcional y paralelismo de datos) y a niveles muy distintos (paralelismo entre instrucciones, paralelismo entre hebras, entre procesos, etc.), y su estudio, junto con los problemas que plantea su aprovechamiento y las alternativas para resolverlos debe incluirse dentro de los contenidos de la materia. El paralelismo entre instrucciones (ILP) se ha aprovechado a través de procesadores segmentados en los que se han introducido mejoras como la posibilidad de enviar a ejecutar varias instrucciones al mismo tiempo, la ejecución desordenada, el procesamiento especulativo, etc. Se puede

reflejar esta situación en la expresión del tiempo de ejecución, si se introduce en ella el número medio de ciclos entre emisiones de instrucciones (CPE, Ciclos por cada Emisión), y el número medio de instrucciones que se emiten cada vez (IPE). Así, el valor de CPI sería igual a CPE/IPE. En un procesador segmentado (no superescalar), CPE=IPE=1, y CPI=1, en uno superescalar debería ocurrir que CPE=1 y IPE>1 (cada ciclo se emiten varias instrucciones), y en un procesador no segmentado se tendría que IPE=1 y CPI=CPE>1 (se emite una instrucción cada ciclo máquina constituido por varios ciclos de reloj).

$$T_{tarea} = NI \times \left(\frac{CPE}{IPE} \right) \times T_{ciclo} = NI \times \left(\frac{CPE}{IPE} \right) * \left(\frac{1}{f} \right)$$

En los repertorios de instrucciones de tipo vectorial o SIMD, cada instrucción codifica varias operaciones. Por lo tanto, en lugar de tener que considerar NI como en una arquitectura escalar, donde al codificar cada instrucción una operación se tendría que NI = Noper (Noper es el número de operaciones a realizar en el programa), el número de instrucciones ejecutadas será Noper/Op_instr y se tendrá:

$$T_{tarea} = \left(\frac{Noper}{Op_instr} \right) \times CPI \times T_{ciclo}$$

En el caso de un procesador vectorial Op_instr > 1 y el número de instrucciones es menor que el de operaciones, en un repertorio escalar, Op_instr = 1 y, como se dijo NI=Noper.

Si se utilizan varios procesadores en el computador, y se distribuye el trabajo entre ellos, la expresión que se utilizó al comienzo de esta sección para el tiempo de ejecución de un programa debe modificarse. Ahora, al dividir el trabajo entre P procesadores, se tendría:

$$T_{tarea} = \max_{i=1, \dots, P} \{ T_{tarea(i)} + O_i \}$$

donde $T_{tarea(i)}$ es el tiempo correspondiente a la ejecución de la parte de tarea asignada al procesador i-ésimo, y O_i es la sobrecarga (overhead) ocasionada básicamente, por los tiempos de comunicación entre los procesadores y los tiempos de espera para la sincronización. Las causas de esta sobrecarga (redes de conexión, modelos y costo de las comunicaciones, los problemas de coherencia, etc.), y las alternativas para reducir sus efectos deben incorporarse a los contenidos de la materia. Así pues, habría un segundo grupo de contenidos relacionados con el incremento de prestaciones de las arquitecturas.

3 Los relatos históricos y la motivación

En [5] se revisan distintas teorías de la motivación para justificar el diseño de estrategias docentes que incrementen el interés y el rendimiento de los estudiantes en una materia. En la Tabla 1 se resumen algunas de las estrategias que se discuten en

[5], relacionándolas con las distintas teorías de la motivación analizadas en dicho trabajo.

Tabla 1. Estrategias docentes según distintas teorías de la motivación

Teoría de la motivación	Estrategias docentes
Objetivos (ser competente, alcanzar los objetivos)	<ul style="list-style-type: none"> • Despertar la curiosidad planteando preguntas sobre aplicaciones prácticas de la materia que se irán respondiendo a lo largo de la asignatura • Evitar referencias que puedan convertir a las calificaciones en objetivos de aprendizaje
Atribuciones (causas internas, dinámicas, y controlables)	<ul style="list-style-type: none"> • Proponer relaciones de problemas de dificultad creciente, comenzando con problemas de resolución muy directa a partir de las leyes o expresiones matemáticas estudiadas en la asignatura • Evaluar el esfuerzo: proponer varias actividades voluntarias y tener en cuenta el número de actividades completadas.
Expectativa-Valor (metas alcanzables y relevantes)	<ul style="list-style-type: none"> • Mostrar que los conceptos que se abordan en la asignatura son importantes para ser profesionales competentes. • Realizar pruebas de evaluación sobre habilidades concretas o contenidos específicos tratados en clase con detalle.
Autodeterminación (autonomía, competencia, socialización)	<ul style="list-style-type: none"> • Plantear prácticas y ejercicios individuales que pongan de manifiesto la utilidad de los contenidos de la asignatura. • Promover el trabajo cooperativo en equipo • Describir posibilidades profesionales que abren las competencias alcanzadas con los contenidos de la asignatura

Así, desde el punto de vista de la motivación, los objetivos de las asignaturas (1) deben ser relevantes para la profesión, y (2) deben contemplarse como asequibles a través de un esfuerzo razonable. No es “bajar el listón” (ver los 10 mandamientos de la docencia según el profesor Yale Patt [6]), sino buscar que los estudiantes asimilen y sientan que controlan el aprendizaje de la asignatura, y sean capaces de abordar conceptos o problemas complejos que, en muchos casos, son los que les hacen ser conscientes de la utilidad de la asignatura para su futuro profesional, los que les convencen de que el esfuerzo realizado les han permitido avanzar en su conocimiento del computador y de aspectos de su profesión que no hubieran ni imaginado si no es por la asignatura estudiada.

Según un modelo coevolutivo de las transformaciones tecnológicas aplicado al ámbito del computador digital [7], en las innovaciones y los logros de los ingenieros y científicos (von Neumann, Eckert y Mauchly, Wilkes, Amdahl, Cray, etc.) y de las empresas (IBM, Burroughs, CDC, DEC, Cray, etc.) están enraizadas formas de trabajo y tecnologías previas, y son parte de un proceso evolutivo de cambio social y tecnológico que no está aislado de su contexto histórico. Lo nuevo no aparece aleatoriamente sino que se crea en organizaciones y sistemas sociales a partir de las capacidades tecnológicas disponibles, las ideas sobre la necesidad de ciertos productos, y las posibilidades de fabricación y uso de los mismos. Un ingeniero no explora todas las posibles soluciones a un problema sino que enfoca su búsqueda a partir de sus nociones acerca de los problemas relevantes y de las formas plausibles de abordarlos según lo que se denomina el régimen o paradigma tecnológico dominante, definido a partir del conocimiento científico existente, el trabajo práctico

del ingeniero, las tecnologías de producción, las características de los productos, y las costumbres, habilidades, e infraestructuras de los usuarios y de las instituciones. Los ingenieros desarrollan su actividad a través de proyectos que tendrán como resultado productos o procesos que competirán con otras alternativas en el momento futuro en que se comercialicen. Según esto, todo ingeniero debe tener presente un modelo de la dinámica de su campo, y para construirlo es importante conocer qué factores han determinado los cambios en el pasado.

Tabla 2. Referencias utilizadas para construir relatos históricos y su relación con los contenidos abordados en la asignatura Arquitectura de Computadores

Contenidos	Referencias para los relatos
Prestaciones del procesador	"A Tale of two processors: revisiting the RISC-CISC debate" C. Isen, L. John, E. John, 2009 [8] "RISC: Back to the future?" C. G. Bell [9]
Clasificación de las arquitecturas	"A Historical overview of Computer Architecture" R. E. Smith [10] "Creating the computer industry" E.W. Pugh; W. Aspray, 1996 [11]
Multicomputadores y supercomputadores	"The social limits of Speed: The development and use of supercomputers" B. Elzen; D. Mackenzie, 1994 [12]
Ley de Amdahl	"Computer Architecture and Amdahl's Law" G.M. Amdahl, 2013 [13]
Multiprocesadores y multicomputadores	"Ultracomputers a Teraflops before its time" G. Bell [14]
Paralelismo entre instrucciones	"The Pentium Chronicles: Introduction". R.P. Colwell, 2006 [15]

4 Algunos episodios relevantes en Arquitectura de Computadores

En esta sección se proponen varios artículos que pueden servir de ayuda para conformar mapas conceptuales en los temas de un programa de introducción a la Arquitectura de Computadores. Como se ha indicado en la Sección 2, una asignatura de este tipo comprende temas dedicados a la evaluación de prestaciones, la clasificación de las arquitecturas y la introducción a la programación paralela. En cuanto a las arquitecturas paralelas, se presentan los conceptos fundamentales del paralelismo ILP y el procesamiento multihebra: planificación de instrucciones, comunicación y sincronización, coherencia y consistencia de memoria, etc. En la Tabla 2 se recogen las referencias que pueden utilizarse para construir relatos relacionados con los conceptos fundamentales que se abordan. Se incluyen artículos con información acerca de la controversia RISC/CISC [8,9], y del momento en que una arquitectura CISC se implementó de forma eficiente a través de una microarquitectura superescalar (microarquitecturas P6 de Intel) [15]. También hay

artículos que describen cómo van surgiendo distintas arquitecturas a partir de planteamientos que priman la compatibilidad con diseños previos o la obtención de mejores prestaciones a cualquier precio. Así, se considera el nacimiento y desarrollo de las arquitecturas vectoriales [12] y de los multiprocesadores [14], y la controversia relativa al uso de varios procesadores o mejora de la capacidad del procesador que se produce a partir de la ley de Amdahl [13]. Textos frecuentemente utilizados como el libro de Arquitectura de Hennessy y Patterson [4] incluyen secciones que revisan el desarrollo histórico de los conceptos y las implementaciones que se abordan en sus distintos temas. Esas fuentes proporcionan información de la sucesión en el tiempo de las distintas arquitecturas y sus principales características. Lo que aquí se propone son una serie de referencias para construir relatos sobre el desarrollo ciertos episodios clave en el avance de las arquitecturas de computador incluyendo algunas anécdotas e información acerca del entorno en el que se van sucediendo los principales conceptos estudiados en la asignatura.

A continuación se resumen parte de los contenidos de los artículos que se proponen, y se indican algunas de las razones por las que resultan interesantes para la asignatura y pueden servir para motivar a los estudiantes.

A historical overview of Computer Architecture (R.E. Smith) [10]. Este extraordinario artículo presenta una historia de la arquitectura de computadores que incluye los orígenes de la disciplina. Se describen con cierto detalle los episodios correspondientes al desarrollo de los primeros computadores y se analiza cómo van surgiendo los diferentes tipos de máquinas, desde las que buscan mejorar las prestaciones ante todo (los supercomputadores) hasta los que introducen el coste como un elemento importante a la hora de evaluar el computador (los minicomputadores y los microcomputadores). No se limita a describir detalles técnicos de las máquinas, sino que se relata el proceso según el que se fueron introduciendo distintos elementos clave de la arquitectura, desde las mejoras en los circuitos aritméticos, hasta la incorporación de distintas tecnologías de memoria, la necesidad de lenguajes de programación de alto nivel, y la llegada de los microprocesadores, entre otros. Este trabajo puede utilizarse tanto en los temas de introducción en los que se presenta el concepto de arquitectura y se describen los distintos tipos de máquinas, como para poner ejemplos del proceso de diseño y fabricación de computadores de cada uno de ellos.

The social limits of Speed: The development and use of supercomputers (B. Elzen; D. Mackenzie) [12]. Proporciona detalles del desarrollo de los primeros supercomputadores comparando dos tendencias, representadas inicialmente por CDC y por IBM. Por un lado está la búsqueda de las máquinas más rápidas en cada momento, y por otro la necesidad de mantener la compatibilidad con diseños previos de la compañía. La primera opción es la defendida por Seymour Cray, uno de los ingenieros más populares y relevantes en el desarrollo de arquitecturas de computador innovadoras, que inició su carrera en CDC. La información que se proporciona en este artículo permite construir relatos acerca del desarrollo de máquinas como la CDC6600 y la CDC7600 y contraponer su filosofía de diseño a la seguida por IBM con sus System 360. El CDC6600 debía ser de 15 a 20 veces más rápido que su predecesor el CDC1604, y se partió de un diseño totalmente novedoso que buscaba la mejora deseada en dos fuentes: (1) el uso de circuitos fabricados por Fairchild

Semiconductor que eran cinco veces más rápidos que los utilizados en el CDC1604, y (2) el diseño de una arquitectura de computador novedosa con varias unidades funcionales que podían funcionar en paralelo y que sería responsable de un factor de mejora de tres o cuatro. El nuevo CDC6600 apareció en Octubre de 1964 y fue capaz de atraer gran parte del mercado antes dominado por los IBM7090 e IBM7094. Una alternativa de diseño diferente la representa el IBM System 360, que también apareció en 1964. Este computador utilizaba un repertorio de instrucciones más complejo que buscaba satisfacer los requisitos de aplicaciones tanto de carácter científico, como los de otras propias del ámbito comercial donde, por ejemplo, era necesario manejar grandes cantidades de datos sin tener que realizar operaciones matemáticas costosas sobre ellos. Con estas máquinas IBM introdujo la idea de separar las características de la arquitectura de la máquina que determinaban el nivel de prestaciones y el repertorio de instrucciones, que mantenía la compatibilidad entre máquinas diferentes. IBM fue capaz de doblar las prestaciones del CDC6600 en 1967, con el IBM System360/91, pero dos años después el CDC7600 dobló a su vez las del computador de IBM (el ciclo de reloj había pasado de 100 ns a 27.6 ns). Por tanto, los relatos del desarrollo de los CDC6600 y CDC7600 y del IBM System360 permiten poner de manifiesto, la existencia de distintas filosofías de diseño, las orientadas a mejorar la velocidad a cualquier precio frente a las que pretenden mantener la compatibilidad con los diseños previos. En ambos casos hay que tener en cuenta las capacidades que ofrece la tecnología disponible, y el uso que la arquitectura puede hacer de ella, con o sin cambios en el repertorio de instrucciones (según el mantenimiento de la compatibilidad sea o no un objetivo).

Otro relato importante que se puede encontrar en [12] es el del diseño del Cray-1 y los principales supercomputadores vectoriales que le siguieron. En 1972, S. Cray fundó Cray Research Inc. a partir de 500.000 dólares aportados por el propio Cray, otros 500000 aportados por CDC, y 1.500.000 aportados por otros 14 inversores (entre ellos, algunos amigos de S. Cray). Es importante resaltar que en el nombre de la empresa aparece la palabra “investigación”, poniendo de manifiesto su importancia en el desarrollo de máquinas cada vez más potentes.

Computer Architecture and Amdahl's Law (G.M. Amdahl) [13]. El número de diciembre de 2013 de la revista IEEE Computer Magazine tiene como tema central los orígenes, el impacto, y la situación actual de las leyes de la computación. Concretamente se incluyen artículos relacionados con las leyes de Moore, Amdahl, Metcalfe, Grosch, y Makimoto. El artículo correspondiente a la ley de Amdahl esta escrito precisamente el autor de dicha ley, Gene M. Amdahl. Se trata de un trabajo muy útil e interesante porque ofrece una descripción de primera mano del sentido de dicha ley, y una información difícil de encontrar acerca de la evolución del trabajo de un ingeniero en el desarrollo de los primeros computadores. Tras graduarse en física en 1948, después del periodo correspondiente a la Segunda Guerra Mundial en el que había enseñado electrónica en la Marina, empezó investigando en física nuclear y a partir de ahí, se plantea la posibilidad de mejorar la forma en que se realizaban los cálculos necesarios para resolver los problemas que se planteaban en su línea de trabajo. Amdahl relata cómo va implicándose cada vez más en problemas relacionados con el diseño de computadores. En 1952 lee su tesis doctoral titulada “The Logical Design of an Intermediate Speed Digital Computer”, que tuvo que ser

leída en otra Universidad distinta dado que en la suya no había nadie capaz de evaluarla, y dio lugar al computador WISC (Wisconsin Integrally Synchronized Computer), construido en 1955.

Después participa en el diseño del IBM 709 (más concretamente en el diseño de los canales de E/S) aunque inicialmente su tarea en IBM era la simulación de redes neuronales artificiales y su aplicación al reconocimiento de caracteres. El artículo describe bastantes detalles acerca de su etapa como ingeniero en IBM, en particular su participación en el diseño de la familia IBM System 360 y de cómo tuvo acceso a bastantes datos acerca del uso de computadores de IBM, desde el 704 al 7094, y llegó a ciertas conclusiones respecto a la velocidad de cómputo que se podría conseguir en función del tamaño de la memoria en esas máquinas, y de una relación que establecía que por cada instrucción ejecutada se necesitaba 1 bit de E/S. Una de las secciones del artículo se dedica, como no puede ser de otra forma, a la ley de Amdahl que, según cuenta el propio G. Amdahl, surgió a partir una presentación y del artículo correspondiente para la Spring Joint Conference de 1967. Se trataba de comparar la potencia de un computador basado en un procesador de altas prestaciones con las de un computador paralelo, como era el ILLIAC IV, un computador SIMD presentado en la misma conferencia por D. Slotnick. Parece ser que el debate fue bastante intenso, y posiblemente, esto contribuyó a dar más publicidad a la correspondiente controversia. En [13], Amdahl se arrepiente de haber participado en una discusión que, en algunos momentos pareció referirse más a las personas que defendían cada alternativa que a la comparación de las propias alternativas. También habla de las propuestas que han mostrado los límites de aplicación de su ley, pero hace referencia a que, a través de ella se puso de manifiesto las exigencias que los multicomputadores deberían satisfacer respecto al aprovechamiento del paralelismo. De hecho termina indicando que Massively Parallel, una compañía dedicada a la fabricación de computadores paralelos, lo contrató para su comité de asesores.

The Pentium Chronicles: Introduction (R.P. Colwell) [15]. Este artículo del IEEE Computer es un resumen de la introducción del libro del mismo título publicado en 2006 por R. P. Colwell, responsable del equipo de diseño de la microarquitectura P6 de Intel. Esta ha sido una microarquitectura bastante relevante ya que permitió a Intel disponer de microprocesadores con características verdaderamente superescalares manteniendo su repertorio de instrucciones CISC compatible con los microprocesadores anteriores. La idea de incluir una etapa de decodificación que traduce las instrucciones CISC de los programas a un repertorio interno de microoperaciones con un formato RISC constituye una idea que parece bastante sencilla y directa pero que plantea sus dificultades desde el punto de vista de la implementación hardware. Esta misma tendencia, pero en el ámbito del software, es la que se ha aplicado en el desarrollo de arquitecturas novedosas posteriores como el Crusoe de Transmeta.

La propia descripción que hace Colwell del inicio del proyecto es reveladora, ya que pone de manifiesto el planteamiento de los objetivos de mejora de prestaciones a alcanzar (doblar las prestaciones del microprocesador anterior, el Pentium, incluso antes de que haya salido al mercado), y la organización del equipo de desarrollo. Se trata de un artículo ideal para incluir en los temas dedicados a los procesadores ILP, y

también se puede usar en aquellos en los que se hable de las alternativas RISC y CISC.

RISC: Back to the future? (C. G. Bell) [9]. La controversia que se produjo en la década de los 80 entre estas dos filosofías de diseño del repertorio de instrucciones máquina ilustra muy bien cómo, en ingeniería, razonamientos plausibles desde un punto de vista lógico pueden no verse corroborados por la realidad y hacen necesario un análisis cuantitativo y una evaluación experimental que refleje las características de las aplicaciones, tal y como se plantea en el libro de Arquitectura de Computadores de Hennessy y Patterson. En este artículo de Gordon Bell [9] la propuesta de las arquitecturas RISC se sitúa en el contexto del aumento de la capacidad de integración que hace posible diseños más complejos de microarquitecturas, y la inclusión caches en los microprocesadores, con lo que se puede reducir el tiempo de acceso a memoria. Además de proporcionar información acerca de los inicios de los RISC, también se ilustran los principios que caracterizan estos procesadores y se hace referencia a máquinas que pueden considerarse precursoras de esta filosofía de diseño.

En esta controversia RISC/CISC se ha puesto de manifiesto la importancia del compromiso entre el nivel de prestaciones que proporciona una alternativa y la compatibilidad con el software existente y cómo nuevas mejoras tecnológicas promueven microarquitecturas novedosas que anulan diferencias de prestaciones previas. En [8] se pone de manifiesto este proceso y cómo las discusiones sobre la superioridad de un repertorio RISC respecto a uno CISC fueron desapareciendo a lo largo de los 90, a la par que desaparecían muchos microprocesadores basados en esta filosofía, que ofrecían unos niveles de prestaciones considerables, mientras que los microprocesadores CISC copaban prácticamente todo el mercado, incluyendo el de los computadores de altas prestaciones. Concretamente, en [8] se hace referencia a un estudio de 1991, donde Bhandarkar y Clark [16] comparaban el VAX (repertorio CISC) y el MIPS (repertorio CISC), encontrando un factor de mejora del 2.7 para el MIPS respecto al VAX. Después se indica que, en un estudio de 1997 [17] en el que se comparaban el Alpha 21064 con el Intel Pentium, todavía se ponía de manifiesto la superioridad de la arquitectura RISC en algunos programas del SPEC CPU95 (entre un 5% y un 200% de mejora). Finalmente, en [8] se muestra como esa diferencia se redujo, comparando procesadores como el POWER5+ y el Xeon 5160. Dado que las técnicas más agresivas para extraer el paralelismo entre instrucciones se pueden implementar en microarquitecturas con repertorios CISC, la controversia RISC/CISC desapareció, como se pone de manifiesto en el mercado actual de microprocesadores.

Ultracomputers, a Teraflops before its time (G. Bell) [14]. Este trabajo proporciona una descripción detallada de la aparición de arquitecturas que buscan alcanzar los mayores niveles de prestaciones posibles en cada momento. Es interesante porque presenta las alternativas que se planteaban en ese momento para fabricar máquinas capaces de alcanzar el Teraflops (el artículo es de 1992, prevé que se alcanzará en 1995, pero hasta 1998 no se consiguió, con el ASCI Red según el TOP500), y se contempla la tendencia hacia máquinas paralelas con nodos basados en microprocesadores comerciales frente a las que utilizaban procesadores vectoriales como los computadores SX3 de NEC o los Cray C90, o los SIMD de Thinking Machines como el CM2. Así, este artículo es una fuente muy útil para generar discusiones sobre lo acertado de algunos de los juicios y previsiones del artículo, y se

puede comprobar hasta qué punto unas máquinas han sido más o menos exitosas. También se puede encontrar información acerca de precios, tamaños de las memorias, frecuencias de reloj, y velocidades pico de los principales computadores paralelos del momento, análisis de la escalabilidad de los distintos tipos de computadores. Se hace referencia a la mayoría de los computadores paralelos del momento, ya clasificados como multiprocesadores (de memoria compartida y memoria distribuida) y multicomputadores, y se ofrecen bastantes explicaciones relativas a la importancia de la jerarquía de memoria y comparaciones con otros computadores de altas prestaciones del momento. Se trata, pues, de un artículo muy adecuado para construir relatos de los temas que aborden el estudio de los multiprocesadores y los multicomputadores.

Creating the computer industry (E.W. Pugh; W. Aspray) [11]. En la innovación en el ámbito de los computadores, la industria ha contribuido de forma decisiva. Si una motivación fundamental para la construcción de los primeros computadores electromecánicos de los 40 fue su previsible utilidad en el desarrollo armamentístico imprescindible en el contexto de la segunda guerra mundial, pronto quedó clara las posibilidades de negocio que se abrían, aunque inicialmente ligadas a aplicaciones demandadas por el sector público. En este artículo de E.W. Pugh y W. Aspray [11] se pone de manifiesto cómo la interacción entre tecnología, fabricantes y clientes, determinó la evolución del computador desde los primeros dispositivos electromecánicos hasta los computadores modernos.

Un aspecto importante para el desarrollo de la industria de los computadores fue su desclasificación, a pesar de que estaban inicialmente ligados a la seguridad militar. Esto se debió, según el artículo, a que en realidad el ENIAC no pudo poner de manifiesto su utilidad militar en la segunda guerra mundial, y a que von Neumann expuso las cuestiones relacionadas con el diseño de computadores más en un ámbito académico ligado con los estudios neurológicos que en un entorno de ingeniería convencional. En realidad, el computador se veía como una calculadora de propósito general más que como una máquina para la guerra y había cierto interés por parte del ejército y de la Universidad de Pennsylvania de dar publicidad a los logros en electrónica que se habían alcanzado. En 1946, Eckert y Mauchly dejaron la Universidad de Pennsylvania para fundar una compañía dedicada a la fabricación y venta de computadores. El primero de ellos fue el UNIVAC, utilizado en 1951 por la oficina del Censo de los EE.UU., pero Eckert y Mauchly ya habían vendido su compañía a Remington Rand para conseguir el dinero y la experiencia empresarial que necesitaban. El UNIVAC apareció en la televisión para predecir los resultados de las elecciones presidenciales de 1952 y, durante un tiempo, llegaron a identificarse los términos UNIVAC y computador, que se contemplaba ya como una máquina electrónica capaz de realizar tareas de procesamiento de información entre las que las tareas de computación eran solo un subconjunto. El diseño del computador de programa almacenado fue el que hizo posible la transición de las tecnologías electromecánicas a las tecnologías electrónicas en los computadores: las instrucciones y los datos de los programas podían ser utilizados por los circuitos electrónicos con la rapidez necesaria a costes razonables. Otros elementos esenciales en el desarrollo del computador fueron las memorias de núcleo de ferrita, la tecnología de almacenamiento magnético, los circuitos de semiconductores, y los sistemas

operativos. A medida que empiezan a haber clientes en ámbitos de negocio distintos al de la defensa, la posición de los fabricantes como suministradores de máquinas en estos sectores se hizo decisiva. Así, IBM que tenía alrededor del 90% del mercado de los equipos de tarjetas perforadas fue capaz de absorber gran parte del mercado de los computadores, a diferencia de lo que ocurrió con Sperry Rand que, a pesar de su tamaño, tenía una menor participación en el negocio de los equipos de tarjetas perforadas, y una falta de liderazgo centralizado y fuerte. Igual sucedió a RCA y GE, mientras que empresas más pequeñas como CDC, DEC, Burroughs y NCR encontraron mejores enfoques. A comienzos de los 60 ya se puede considerar que había aparecido el computador moderno y que existía un volumen de clientes sensibles al coste del computador que promovió el desarrollo de productos más estandarizados. En este contexto se anuncia el IBM S/360 que no es capaz de satisfacer todos los requerimientos del mercado en un momento en que la mejora en coste y prestaciones de las tecnologías del computador y la aparición de una gran diversidad de aplicaciones complejas dieron lugar a que apareciesen nuevas oportunidades para los fabricantes de computadores (los supercomputadores). Este artículo ofrece bastante información de las empresas pioneras en la fabricación de computadores, y anécdotas acerca de la intervención de las empresas en el desarrollo de los computadores, y de la interacción entre tecnología y demanda de aplicaciones con el propio desarrollo de la arquitectura de computadores.

5 Conclusiones

La generación de mapas conceptuales para los contenidos de una asignatura permite integrar el conocimiento adquirido de forma interrelacionada, contribuyendo a mejorar la memorización de dichos conceptos. La capacidad para avanzar en la comprensión de una asignatura y la resolución de problemas planteados en la disciplina mejora la motivación hacia la misma. Por tanto, la utilización de relatos acerca del proceso histórico que ha generado el cuerpo de conocimientos de una asignatura contribuye a crear relaciones entre los conceptos y las circunstancias en las que se han desarrollado.

Además de aprovechar la facilidad para memorizar relatos, al poner de manifiesto la dimensión social aparejada al desarrollo de nuevas propuestas, se dota a los conceptos de una serie de características emotivas que contribuyen a la motivación, y con ello facilitan la asimilación de los conceptos. En relación con estos planteamientos, en este artículo se han propuesto y comentado una serie de artículos de los que se pueden extraer esos relatos, y se ha indicado su relación con los mapas conceptuales de la asignatura que, como se ha dicho, pueden contribuir a mejorar el aprendizaje de los estudiantes.

6 Referencias

1. Implagliazzo, J.; Lee, J.A.N.: "Using computing history to enhance teaching". *History of Computing in Education*, pp. 165-176, 2004.

2. Papadimitriou, C.: "Mythematics: In Praise of Storytelling in the teaching of Computer Science and Math". Invited Editorial, ACM inroads, Vol. 35, No.4, pp. 7-9. Diciembre, 2003.
3. Ortega, J.; Anguita, M.: "Arquitectura de Computadores en seis créditos ECTS". Enseñanza y Aprendizaje de Ingeniería de Computadores, pp.13-25 (<http://hdl.handle.net/10481/20528>), 2012.
4. Hennessy, J.L.; Patterson, D. A.: "Computer Architecture. A Quantitative Approach". Elsevier Science & Technology, 2011 (5ª Edición).
5. Ortega, J.; Anguita, M.; Damas, M.; González, J.: "Motivación para la Ingeniería de Computadores". Enseñanza y Aprendizaje de Ingeniería de Computadores, pp.3-22, (<http://hdl.handle.net/10481/26351>), 2013.
6. Patt, Y.: "Ten Commandments for Good Teaching". (<http://users.ece.utexas.edu/~patt/Ten.commandments/>), 2013.
7. Ende, J. van den; Kemp, R.: "Technological transformations in history: how do computer regimes grow out of existing computing regimes". Research Policy, 28, pp.833-851, 1999.
8. Isen, C.; John, L.; John, E.: "A Tale of two processors: revisiting the RISC-CISC debate". Springer-Verlag LNCS 5419, pp. 57-76, 2009.
9. Bell, C.G.: "RISC: Back to the Future?". DATAMATION, pp.76-82. Junio, 1987.
10. Smith, R.E.: "A Historical overview of Computer Architecture". Annals Hist. Computer, 10, pp. 277-303, 1989.
11. Pugh, E. W.; Aspray, W.: "Creating the Computer Industry". IEEE Annals of the History of Computing, Vol. 18, No.2, pp. 7-17, 1996.
12. Elzen, B.; Mackenzie, D.: "The social limits of Speed: The development and use of Supercomputers". IEEE Annals of the History of Computing, Vol. 16, No.1, pp. 46-61, 1994.
13. Amdahl, G.M.: "Computer Architecture and Amdahl's Law". IEEE Computer, pp.38-46. Diciembre, 2013.
14. Bell, G.: "Ultracomputers, a Teraflops before its time". Communications of the ACM, Vol.35, No.8, pp.27-47. Agosto, 1992.
15. Colwell, R.P.: "The Pentium Chronicles: Introduction". IEEE Computer, pp.49-54. Enero, 2006.
16. Bhandarkar, D.; Clark, D.W.: "Performance from architecture: comparing a RISC and a CISC with similar hardware organization". Proc. of ASPLOS, pp. 310-319, 1991.
17. Bhandarkar, D.: "RISC versus CISC: a tale of two chips". ACM SIGARCH Computer Architecture News, Vol. 25, No.1, pp. 1-12, 1997.