

El enfoque motivado de abajo hacia arriba aplicado a las materias Fundamentos de Informática y Arquitectura de computadoras

Aldo J. Benítez ^{a,b}, Roberto N. Schuster ^{a,c}, Hernán F. Kisiel ^a, Noelia I. Bouchuk ^a, Lucas D. Corbalán ^a, Conrado A. Faleiro Da Silva ^a, Carlos A. Kornuta ^{a,d}, Roberto E. Carballo ^{a,b*}

^a Facultad de Ingeniería, Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina.

^b GIDE, FI-UNaM, Oberá, Misiones, Argentina.

^c LABSE, FI-UNaM, Juan Manuel de Rosas 325, Oberá, Misiones, Argentina

^d Universidad Gastón Dachari, Salta esquina Colón, Posadas, Misiones, Argentina

e-mails: robertocarballo@fio.unam.edu.ar, javierbenitez@fio.unam.edu.ar

Resumen

El objetivo de este trabajo es divulgar la experiencia del equipo de cátedra de Fundamentos de Informática y Arquitectura de Computadoras de la carrera de ingeniería en computación de la FIO, en la adaptación del enfoque de enseñanza denominado “motivado de abajo hacia arriba” para estos cursos. Este enfoque ha sido propuesto originalmente por el prof. Yale Patt para la enseñanza del curso introductorio de computación para estudiantes de ciencias de la computación e ingeniería en computación. Esperamos con este trabajo poder contribuir a la planificación de materias similares de otras facultades o universidades, así como también extender la aplicación del enfoque, en lo que sea posible, a otras áreas de la enseñanza de la Ingeniería.

Palabras Clave – *Motivated Bottom-Up Approach, pedagogía para cursos introductorios en computación, fundamentos de informática, arquitectura de computadoras.*

1 Introducción

Diversos planes de estudio de carreras de ingeniería de computación, así como también ingeniería en sistemas, ciencias de la computación o afines, tienen cursos de arquitectura de computadoras en sus diseños curriculares, siempre con el objetivo de comprender el funcionamiento de las computadoras (ejecución secuencial, una instrucción a la vez, en definitiva: lo que se conoce como modelo de ejecución de von Neumann), y en casos específicos como el de ingeniería en computación, con mayor profundidad para conocer los principios con los que se diseñan y construyen los sistemas computacionales modernos (*pipelining*, super escalar, fuera de orden, para nombrar algunos – conceptos) [1].

Debido a estas diferentes necesidades existen diversas propuestas [2-5], las cuales permiten graduar la cantidad de conceptos “pertenecientes” al hardware y/o al software, y que se busca que los estudiantes aprendan a lo largo del cursado.

El presente trabajo expone los detalles en la adaptación del enfoque de enseñanza “motivado de abajo hacia arriba” [6], para la carrera de ingeniería en computación de la FIO. Este enfoque ha sido propuesto por el profesor Yale Patt para la enseñanza del curso introductorio de computación para estudiantes de ciencias de la computación e ingeniería en computación en la Universidad de Michigan [7], y actualmente aplicado en la Universidad de Texas Austin donde actualmente se

encuentra trabajando. Además, es posible establecer que este enfoque es aplicado por el profesor Onur Mutlu en el ETH Zürich [8], para un curso de Diseño Digital y Arquitectura de Computadoras.

Para el caso de nuestros estudiantes, los conceptos fundamentales con los que se construyen las computadoras modernas se encuentran abordados en las materias fundamentos de informática y arquitectura de computadoras, por lo que la adaptación del enfoque se debe hacer para ambos cursos por separado.

Los argumentos por los cuales utilizar este enfoque es que posibilita introducir a los estudiantes de forma progresiva a los fundamentos con los que se diseñan y construyen las computadoras modernas, presentando inicialmente una visión global y contextualizada de los temas, para luego adentrarse en los detalles más específicos. Un ejemplo de visión global en los sistemas de computación son las distintos tipos de procesadores existentes en la actualidad, las CPU (*Central Processing Unit* - unidad central de procesamiento), GPU (*Graphic Processing Unit* - unidad de procesamiento gráfico), TPU (*Tensor Processing Unit* - unidad de procesamiento de tensores) y otros aceleradores, mientras que ejemplos de detalles específicos son las compuertas lógicas y como estas están construidas por transistores con tecnología CMOS (*Complementary Metal-Oxide Semiconductor*, semiconductor complementario de óxido metálico), detallándose más adelante cual es la profundidad con la que se abordan estos temas en el cursado.

De acuerdo a palabras del Prof. Yale Patt, lo que busca su enfoque es evitar “el método tradicional de introducir la informática a los estudiantes a través de un curso de lenguaje de alto nivel” [7], el profesor argumenta que “los estudiantes no tienen comprensión de cómo funciona una computadora y, por lo tanto, se limitan a memorizar patrones con la esperanza de poder aplicar esos patrones a la aplicación en cuestión”.

Lo que encontramos con el equipo de cátedra es que hay distintas formas de introducir a los estudiantes a los detalles más específicos, así como también es posible “saltarse” algunos temas para presentarlos más adelante cuando realmente se los utilice. Exponiendo nuestra experiencia a lo largo de 4 años del cursado de las materias de fundamentos de informática y arquitectura de computadoras, pretendemos aportar a la planificación de materias similares de otros planes de estudio de carreras de ingeniería en computación, así como también abrir la puerta a la aplicación del enfoque “motivado de abajo hacia arriba” a otras áreas de enseñanza de la ingeniería.

Lo que resta del documento está organizado de la siguiente forma, 2.1 Jerarquía de Transformación, 2.2 Etapas de motivación, 2.3 punto de partida para la enseñanza y aprendizaje de los fundamentos, 2.4 ejemplo de extensión a otra área y 3 Conclusiones.

2 Adaptación del enfoque motivado de abajo hacia arriba

2.1 Jerarquía de Transformación

El funcionamiento de los sistemas computacionales puede entenderse mediante una jerarquía de transformación, la cual transforma el problema que se quiere resolver mediante computación, hasta los electrones que “hacen el trabajo” de computar. En el medio se encuentran distintos niveles de transformación, que también se los puede denominar niveles de abstracción [4]. En la Fig. 1 se puede observar un ejemplo de jerarquía de transformación para los sistemas computacionales, que es la adoptada originalmente por el prof. Yale Patt, aunque hoy en día existen múltiples variantes [9, 10].

La idea principal de esta jerarquía es que bien abajo se encuentran los transistores, que son los dispositivos que forman las compuertas lógicas, las cuales pueden realizar operaciones lógicas básicas como AND, OR y NOT.

Con estas compuertas se pueden construir circuitos más complejos, capaces de realizar aritmética, además de circuitos con memoria.

Estos componentes se conectan de determinada forma para ejecutar instrucciones, llegando así a la microarquitectura del procesador.

Esta microarquitectura tendrá que obedecer a la ISA (*Instruction Set Architecture*, Arquitectura del Conjunto de Instrucciones), la cual además de especificar el conjunto de instrucciones del procesador, determina que tipos de datos tendrán soporte en la computadora, cual es el espacio de memoria de trabajo, como se direcciona la memoria (*byte addressable* o *word addressable*), entre otras cuestiones, que en definitiva determinan como el programador (o compilador) ven el hardware de la computadora, siendo el motivo por el cual a este nivel se le dice arquitectura de la máquina.

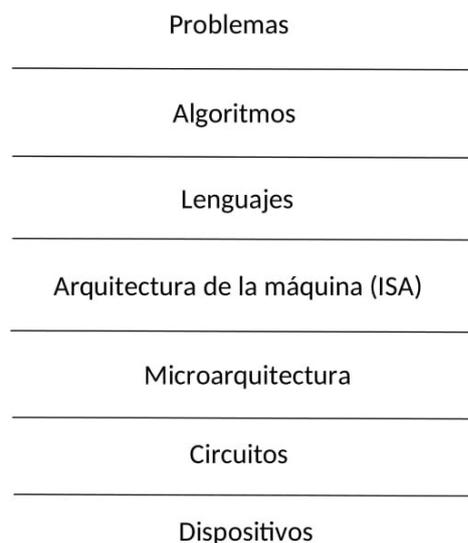


Fig. 1 Jerarquía de Transformación [4].

Arriba de la ISA se puede situar el lenguaje de alto nivel con el que se programa, por ende el compilador o interpretador que generará el código máquina a partir de código escrito en alto nivel como C o Python. Por encima del lenguaje se encontrarán los algoritmos, que son desarrollados para resolver el problema que se quiere solucionar.

Esta jerarquía de transformación es la que le da significado a parte del nombre del enfoque del prof. Patt, refiriéndose a que se enseña “de abajo hacia arriba” construyendo uno a uno cada uno de los niveles de transformación de forma progresiva.

La materia Fundamentos de Informática abarcaría desde dispositivos hasta la ISA, aunque no se culminan todos los temas de microarquitectura e ISA en esta materia, retomándolos al inicio de Arquitectura de Computadoras.

Si bien esta jerarquía establece un cierto orden con el que se podrían ir dando los temas a lo largo del cursado, en la siguiente sección describimos como abordar para mantener la motivación de los estudiantes, presentándoles contexto para comprender mejor la relación y necesidad de cada uno de los temas a la hora de construir los conocimientos de forma sólida.

2.2 Etapas de Motivación

El núcleo del enfoque “motivado de abajo hacia arriba” reside en que se comienza desde las bases del funcionamiento de los circuitos digitales, para ir construyendo componentes que conforman los circuitos de los procesadores modernos. Esto por sí solo puede causar que los estudiantes no encuentren el sentido de los temas fundamentales, hasta estar muy adentrado en el curso, por lo que el profesor Patt propone que se haga “motivado” el progreso de los temas, presentando una visión general del sistema, para luego adentrarse de a poco en los puntos específicos del temario que se debe enseñar. Inclusive, no solamente se encontrará una etapa de motivación, sino varias, donde por ejemplo se anidaría sobre la visión general, otra exploración general sobre una parte específica que se va a enseñar, para luego recién meterse “de lleno” en los temas específicos.

De acuerdo a [6], en la Fig. 2 se puede encontrar una secuencia a seguir para el curso introducción a la computación, donde se observan las etapas de motivación separadas por líneas de subrayado, siendo el orden en que se dan los temas indicado de acuerdo al número que acompaña cada título, mientras que mediante flechas se indica cómo se va progresando de abajo hacia arriba, coincidiendo el orden con el que progresa la jerarquía de transformación de la Fig. 1.

Como se puede deducir de la Fig. 2, el curso introductorio para el cual el prof. Patt propuso el enfoque motivado de abajo hacia arriba, tiene como objetivo enseñar los fundamentos de funcionamiento de los sistemas computacionales, particularmente los que siguen el modelo de ejecución de von Neumann, hasta llegar a programación con lenguajes de alto nivel (lo que figura en el punto 11 como HLL, *High Level Language*).

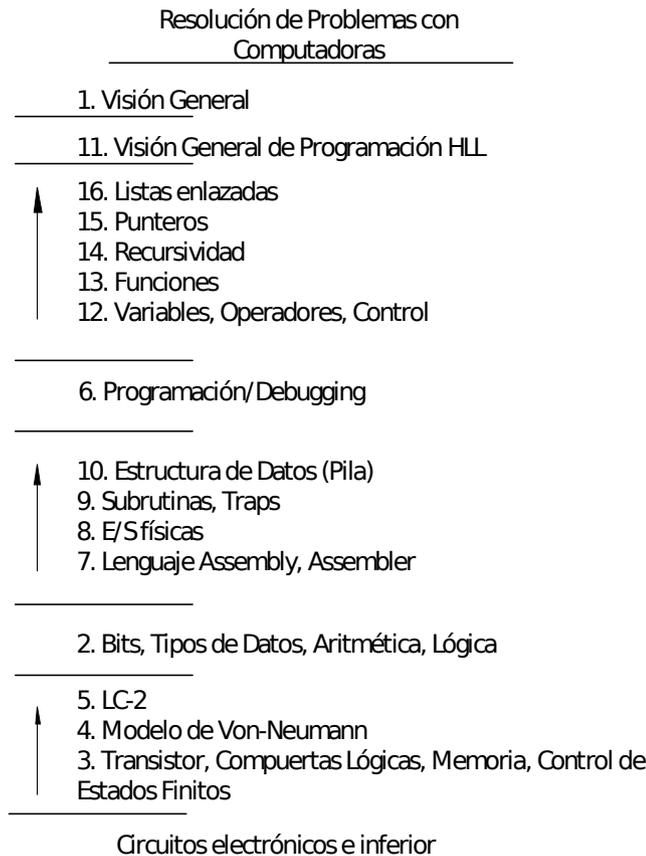


Fig. 2 Secuencia de temas de acuerdo al enfoque motivado de abajo hacia arriba para el curso Introducción a la computación [6].

De acuerdo a nuestra experiencia hemos podido abarcar todas las secciones desde 2 hasta 7 en el curso de Fundamentos de Informática, dejando las secciones 8, 9 y 10 para tratarlas en Arquitectura de Computadoras. Los puntos restantes son enseñados en el la materia Algoritmos y Estructuras de Datos de la carrera de Ingeniería en Computación.

Como ejemplos para la motivación inicial dada en la visión general, se puede establecer la necesidad por la cual computar, cuales son algunos de los problemas actuales que requieren solución en el campo de la computación, cual es el rol en la sociedad de las y los ingenieros en computación.

En cuanto al tiempo para dedicarle a las etapas de motivación, estas dependerán del tiempo que se tenga para abordar todos los temas de la planificación. En nuestro caso nos tomamos por lo general una clase para hacer la motivación inicial, menos de una clase para las primeras etapas intermedias, y hasta 2 clases para la motivación de la construcción de una computadora básica capaz de ejecutar cualquier algoritmo.

2.3 Punto de partida para la enseñanza y aprendizaje de los fundamentos

En el fondo de la pila de la jerarquía de transformación se tienen los dispositivos, donde se encuentran los transistores con los que se construyen los protagonistas del siguiente nivel de abstracción, las compuertas lógicas, estas compuertas estarían en el nivel de “circuitos” de acuerdo a [4].

El prof. Patt propone empezar el aprendizaje de los estudiantes con los transistores, mostrando solamente su comportamiento como interruptor, dejando afuera todo lo correspondiente a la física de semiconductores que permite comprender detalles de cómo operan estos dispositivos, ocultando de esta forma el nivel de abstracción “física de semiconductores” que estaría abarcado por debajo del nivel “dispositivos” en la Fig. 1.

Los autores consideramos que empezar directamente por el transistor como interruptor puede ser complicado para que un estudiante le encuentre el sentido en el funcionamiento de la computadora, ya que se requieren una comprensión previa de conceptos para empezar a ver como estos transistores permiten armar circuitos que computan, tales como el sumador o restador aritmético. Por este motivo proponemos empezar por el sistema de numeración binario, donde los estudiantes que finalizan la escuela secundaria ya tienen una formación en manipulación de símbolos, manejando inclusive álgebra convencional. De esta forma, el punto 2 en la Fig. 2 es realmente donde empieza el aprendizaje de los estudiantes en Fundamentos de Informática, y no es solamente una etapa de motivación.

El siguiente tema en Fundamentos de Informática es compuertas lógicas, trabajando con estas directamente en forma matemática a través de expresiones lógicas, con la cual mediante las reglas de álgebra de Boole y los teoremas de De Morgan se pueden construir desde las funciones lógicas básicas a circuitos para suma aritmética. Todo el tratamiento hasta aquí es puramente matemático, no hay una noción física sobre los voltajes con los que operan las compuertas, ni lo que sucede internamente en estas para lograr los 0s y 1s.

En esta etapa es importante que los estudiantes tomen contacto con problemas básicos que les permitan ir ganando confianza en la resolución de problemas. En ambas materias durante todo el cursado se trabajan problemas típicos de cada tema en cuestión, estos ayudan a los estudiantes a desarrollar habilidades de resolución para las asignaturas posteriores.

Finalizada esta etapa los estudiantes entienden como la lógica se puede utilizar para sumar bits y números binarios, y ya estarían listos para ver que hay dentro de las compuertas lógicas.

En la siguiente etapa se enseña como los transistores en configuración CMOS conforman las distintas funciones lógicas básicas (NAND, NOR, NOT, AND, OR). Para empezar este tema sí realizamos una etapa de motivación, presentando una visión global del comportamiento de la electricidad y estableciendo analogías con el flujo de corriente en un caño de agua.

La importancia de por qué tener un conocimiento inicial en electricidad y lo que físicamente ocurre con los circuitos con transistores, reside en que el estudiante comprenda dos factores fundamentales en el funcionamiento de las computadoras, el efecto del calentamiento de los “chips”

y la demora en la respuesta para obtener un resultado (retardos de propagación de señales), estableciendo con esto último los principios para determinar las causas que demoran la obtención de resultados por parte de las computadoras, lo cual luego se utiliza en Arquitectura de Computadoras para determinar la frecuencia de operación del reloj del procesador.

Una vez aprendidos los temas iniciales se puede seguir construyendo hacia arriba, pasando por el modelo de ejecución de von Neumann, con el cual se dará una visión general del funcionamiento de las computadoras (etapa de motivación), de esta forma se introduce la necesidad de tener memoria en un sistema computacional, con lo cual luego se construyen a partir de las compuertas lógicas ya conocidas, los componentes de memoria desde los *latches* hasta los registros y memoria RAM.

El tema máquinas de estados finitos no se aborda en Fundamentos de Informática por una cuestión de tiempo y de utilidad del tema, ya que no se lo aplica a los circuitos necesarios para construir las partes que siguen en el desarrollo de los temas. Si es abordado en Arquitectura de Computadoras, debido a su utilidad en ejecución multiciclo y predicción de saltos.

La etapa final del curso de Fundamentos de Informática cierra con el camino de datos de un procesador de un solo ciclo, pero para esto motivamos las características que tiene que tener un procesador para poder ejecutar cualquier algoritmo, esto lo hacemos dividiendo en tres prototipos, aunque con los dos primeros podemos establecer los principios para computar (el primero establece los componentes y conexiones que tiene que tener un procesador para procesar datos, mientras que el segundo establece la ejecución condicional dándole las características de una máquina Turing completa [11]), el tercer prototipo introduce la arquitectura *load-store* presente en todos los procesadores RISC (*Reduced Instruction Set Computer* – Computadora de Conjunto de Instrucciones Reducido) modernos.

Luego de esta motivación se introducen los conceptos de ISA y microarquitectura, construyendo a partir de la ISA de la LC-3 un procesador de un solo ciclo. Al finalizar el curso los estudiantes pueden construir un procesador básico “en papel”, lo cual luego se retomará en Arquitectura de Computadoras para modelar en un software denominado Logisim.

Resumiendo la secuencia de temas del dictado de Fundamentos de Informática, en la Fig. 1 se presenta la Jerarquía de conocimientos para esta materia, debiendo notar que no sigue necesariamente el mismo orden que la jerarquía de transformación.

En cuanto a los temas específicos de la materia Arquitectura de Computadoras, estos comienzan con una revisión de los conceptos de lógica combinacional y circuitos con memoria, para luego abordar máquinas de estados finitos. Antes de esto se puede hacer una motivación inicial, pero no será del mismo tipo que en Fundamentos de Informática por contar con estudiantes formados (Arquitectura de computadoras es correlativa con la regularidad de Fundamentos de Informática).

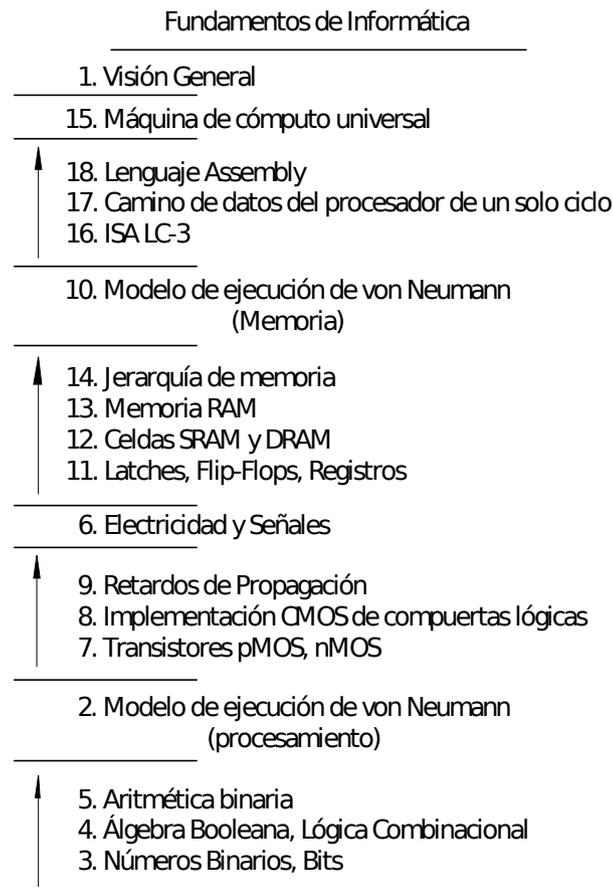


Fig. 3 Secuencia de temas de acuerdo al enfoque motivado de abajo hacia arriba adaptado para la materia Fundamentos de Informática.

Una propuesta es presentar las herramientas que se utilizan hoy en día en la industria para diseñar y modelar sistemas computacionales, presentando así los lenguajes de descripción de hardware (HDL: *Hardware Description Language*). Vincular esto con el modelado en Logisim es necesario para que los estudiantes puedan darle sentido a lo que van a hacer específicamente en la materia, permitiéndoles ver una aplicación directa en la práctica profesional, aunque el objetivo de modelar componentes y procesadores en Logisim sea el aprendizaje de los temas.

Los temas siguientes progresan de la siguiente forma: procesador multiciclo, pipelined, excepciones precisas, ejecución fuera de orden, VLIW (*Very Long Instruction Word* - arquitectura de palabra de instrucción muy larga), SIMD (*Single Instruction Multiple Data* – una instrucción aplicada a múltiples datos), GPU, arreglos sistólicos, caches y organización de memorias. De estos temas solamente hasta excepciones precisas se incluye en los laboratorios de la materia, los cuales son seis en total, todos enfocados en el modelado en Logisim.

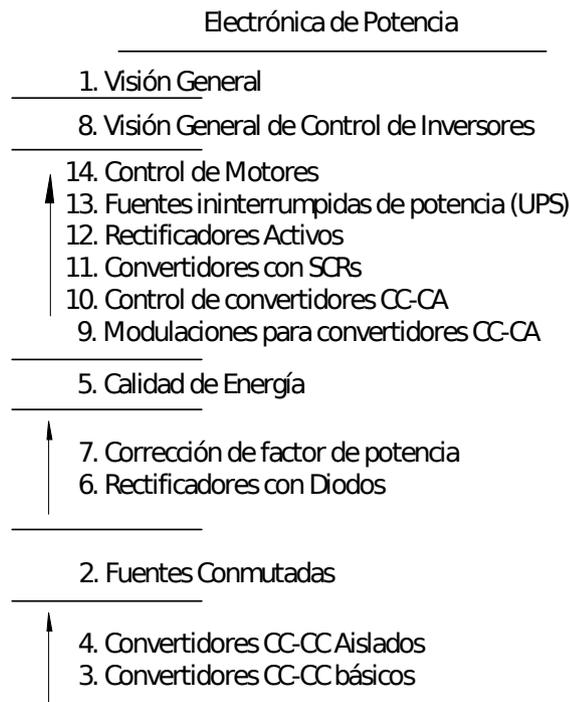


Fig. 4 Jerarquía de conocimiento para la materia Electrónica de Potencia de la carrera de Ingeniería electrónica de la FIO.

2.4 Ejemplo de extensión a otra área

Si bien la mayoría de las áreas del conocimiento no presentan una jerarquía en el funcionamiento como lo hacen los sistemas computacionales, si es posible jerarquizar el conocimiento de acuerdo a la dificultad que tendrá el estudiante, o de acuerdo a la utilidad inmediata que se pueda dar de ciertos temas (como por ejemplo el caso mencionado de máquinas de estados finitos). De hecho, en muchos planes de estudio y planificaciones de materias existe una jerarquía en el orden con el que se arman los cronogramas de clases, por lo que a partir de esto podemos delimitar dos pasos a seguir para extender el enfoque motivado de abajo hacia arriba a otras materias

- 1) Definir una secuencia de temas que progrese de menor a mayor dificultad, teniendo en cuenta la relevancia y utilidad de cada tema para el desarrollo de los conceptos posteriores. Esta estructura proporcionará una guía clara para presentar los contenidos de manera progresiva y comprensible para los estudiantes.
- 2) Identificar oportunidades para introducir etapas de motivación que despierten el interés y la curiosidad de los estudiantes en los temas que se abordarán. Estas etapas pueden incluir ejemplos prácticos, aplicaciones reales o proyectos que conecten los conceptos teóricos con situaciones cotidianas o desafíos profesionales.

A modo de ilustración presentamos como ejemplo en la Fig. 4 una Jerarquía de Conocimiento para la materia de Electrónica de Potencia de la carrera de Ingeniería Electrónica de la FIO.

Esta jerarquía de conocimiento permite establecer una progresión gradual desde conceptos básicos hasta temas más avanzados en Electrónica de Potencia. Los puntos de motivación se pueden integrar estratégicamente en cada etapa para brindar contexto de las aplicaciones que tienen los distintos convertidores estáticos de potencia, mostrando su lugar en la industria, así como también líneas de investigación actuales que buscan mejorar las características de rendimiento, volumen o costo.

3 Conclusiones

La adaptación del enfoque de enseñanza "motivado de abajo hacia arriba" permitió establecer una estrategia efectiva en la enseñanza de las materias Fundamentos de Informática y Arquitectura de Computadoras en la carrera de Ingeniería en Computación de la FIO. Este enfoque, adaptado de la propuesta original del profesor Yale Patt, logra introducir a los estudiantes gradualmente en los fundamentos con los que se diseñan las computadoras modernas, consiguiendo que estos construyan sus conocimientos desde lo que van aprendiendo en cada nivel de abstracción, achicando las dificultades que podrían tener al arrancar de niveles de abstracción mayores, como lo que ocurre si se comienza directamente con un lenguaje de programación de alto nivel.

Además, indicamos un procedimiento general para extender este enfoque a otras áreas de enseñanza, introduciendo un ejemplo de jerarquía de conocimiento para la materia Electrónica de Potencia de la FIO, e indicando las etapas de motivación para enriquecer la experiencia de aprendizaje.

Referencias

- [1] J. P. Shen and M. H. Lipasti, *Modern processor design: fundamentals of superscalar processors*: Waveland Press, 2013.
- [2] M. Guzdial and B. Ericson, *Introduction to computing and programming in python*: Pearson New York, NY, 2016.
- [3] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach*: Elsevier, 2011.
- [4] S. Patel and Y. Patt, *Introduction to Computing Systems: from bits & gates to C & beyond*: McGraw-Hill Professional, 2019.
- [5] P. Jamieson, D. Davis, and B. Spangler, "The Mythical Creature Approach-A Simulation Alternative to Building Computer Architectures," in *FECS*, 2010, pp. 23-28.
- [6] Y. N. Patt, "Teaching and teaching computer architecture: two very different topics: (some opinions about each)," presented at the Proceedings of the 2003 workshop on Computer architecture education: Held in conjunction with the 30th International Symposium on Computer Architecture, San Diego, California, 2003.
- [7] <https://www.ozyegin.edu.tr/en/events/50018671/prof-yale-patt-will-give-a-talk-about-%22the-correct-first-course-for-serious-students-of-computing%22-in-ozu>.
- [8] <https://safari.ethz.ch/courses/>.
- [9] S. Harris and D. Harris, *Digital Design and Computer Architecture, RISC-V Edition*: Morgan Kaufmann, 2021.
- [10] <https://www.youtube.com/watch?v=vgPFzblBh7w>.
- [11] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem," *J. of Math*, vol. 58, p. 5, 1936.