

## Enfrentando desafíos de la currícula en Informática: Concurrencia, Paralelismo, Cloud Computing, Multicores y GPUs

Armando De Giusti<sup>1,2</sup>, Marcelo Naiouf<sup>1</sup>, Laura De Giusti<sup>1</sup>, Franco Chichizola<sup>1</sup>, Adrián Pousa<sup>1</sup>, Victoria Sanz<sup>1,2</sup>, Ismael Rodríguez<sup>1</sup>, Enzo Rucci<sup>1,2</sup>,

<sup>1</sup> Instituto de Investigación en Informática LIDI (III-LIDI) – Facultad de Informática – UNLP

<sup>2</sup> Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

Argentina

{degiusti, mnaiouf, ldgiusti, francoch, apousa, vsanz, ismael, erucci}@lidi.info.unlp.edu.ar

### Resumen

Se presenta y analiza el esquema curricular utilizado para la enseñanza de los temas relacionados con concurrencia y paralelismo en las carreras de Licenciatura de la Facultad de Informática de la UNLP, considerando el cambio tecnológico de las arquitecturas actuales. El enfoque utilizado, siguiendo los lineamientos internacionales y los desarrollados por la Red de Universidades con Carreras de Informática en Argentina, es la incorporación temprana de los temas básicos de concurrencia y paralelismo en el inicio de las carreras, profundizando y aplicando los temas en espacios curriculares relacionados, obligatorios y optativos, de años superiores. Finalmente, se exponen algunos resultados concretos de los cambios curriculares implementados en los últimos tres años.

**Palabras clave:** Concurrencia, Paralelismo, Currícula, GPU, Cloud computing.

### 1. Introducción

La Concurrencia ha sido un tema central en el desarrollo de la Informática y los mecanismos de expresión de procesos concurrentes que cooperan y compiten por recursos ha estado en el núcleo curricular de los estudios de Informática desde la década del 70, en particular a partir de los trabajos fundacionales de Hoare, Dijkstra y Hansen [HOA78][HOA85][DIJ65][DIJ78][HAN77]. Estos conceptos se enseñaron tradicionalmente partiendo de la disponibilidad de un único procesador central, que podía explotar

parcialmente la concurrencia de un dado algoritmo, en función de la arquitectura física disponible (incluso con hardware específico como los coprocesadores, los controladores de periféricos o esquemas vectoriales que replicaban las unidades de cómputo aritmético-lógico).

El paralelismo, entendido como “concurrencia real” en la que múltiples procesadores pueden operar simultáneamente sobre múltiples threads o hilos de control en el mismo instante de tiempo, resultó durante muchos años una posibilidad limitada por la tecnología de hardware disponible [HWA84][HWA93][DAS89].

En las currículas informáticas clásicas [ACM68][ACM78][ACM99] aparecían los conceptos de concurrencia en diferentes áreas (Lenguajes, Paradigmas, Sistemas Operativos) y se omitía casi totalmente el tratamiento del paralelismo, salvo al plantear los conceptos de sistemas distribuidos.

La aparición del lenguaje ADA [OLS83] a mediados de los 80 marca un hito en la evolución del tema, ya que especifica claramente en un lenguaje real los diferentes mecanismos de expresión de la concurrencia y al mismo tiempo deja clara la posibilidad de asociar los procesos (“tasks” en ADA) a diferentes procesadores físicos.

Las nuevas arquitecturas de los procesadores, que integran múltiples “cores” o núcleos en un procesador físico [GEP06][MCC08][GPG] han producido un notorio impacto en el desarrollo de la Informática, obligando a replantear el “modelo base” de un procesador. Esto ha llevado a reemplazar el formato de “máquina de Von Neuman” [GOL72] con un

solo hilo de control, por un esquema que integra múltiples “cores” cada uno con uno o más hilos de control y varios niveles de memoria accesible en forma diferenciada [AMD09].

Al mismo tiempo, los cambios tecnológicos han producido una evolución de los temas de mayor interés en Informática, fundamentalmente por las nuevas aplicaciones que se desarrollan a partir de disponer de arquitecturas y redes de comunicación de mayor potencia y menor costo [DEG13][HOO13].

Esto ha llevado a que las recomendaciones curriculares internacionales [ACM04][ACM08][ACM13] mencionen la necesidad de tratar los temas de concurrencia y paralelismo en etapas tempranas de la formación del alumno, dado que todas las arquitecturas y sistemas reales con los que trabajará son esencialmente paralelos.

Sin embargo, aquí aparece uno de los problemas importantes, ya que la programación paralela (y los conceptos fundamentales de concurrencia) resulta más compleja para un alumno en las etapas iniciales de su formación. Es necesario contar con nuevas estrategias que permitan abordar tempranamente el tema [CAR03][DEG12a] y profundizarlo a lo largo de la carrera.

El trabajo está estructurado de la siguiente manera: la Sección 2 presenta el esquema curricular que se utiliza en las carreras de Licenciatura de la Facultad de Informática. La Sección 3 describe y analiza cada uno de los espacios curriculares involucrados. En la Sección 4 se presentan algunos resultados y conclusiones.

## 2. Esquema curricular

Los planes de estudio de las carreras de Licenciatura en Informática y Licenciatura en Sistemas de la Facultad de Informática están en continua evolución, manteniendo los conceptos fundamentales aunque buscando incorporar los cambios que se producen en una disciplina tan dinámica. Los planes se

encuentran estructurados en las líneas curriculares establecidas por la Red de Universidades Nacionales con Carreras de Informática (RedUNCI), esto es: 1) Algoritmos y Lenguajes, 2) Ciencias Básicas Generales y Específicas, 3) Ingeniería de Software, Bases de Datos y Sistemas de Información, 4) Arquitectura, Redes y Sistemas Operativos, y 5) Aspectos Sociales y Profesionales de la Informática.

Considerando la currícula actual (Plan 2015), los conceptos relacionados con concurrencia y paralelismo se encuentran en los diferentes trayectos, aunque en mayor medida en 1) y 4), en los siguientes espacios curriculares:

- En la asignatura “Taller de Programación”, dictada en el segundo semestre del primer año e incorporada en la última modificación de plan, los alumnos reciben una introducción a diferentes paradigmas (procedural, objetos y concurrente). Asimismo, a fin de reforzar los conceptos, realizan experimentación en máquina.
- En el segundo semestre del tercer año se encuentra “Programación Concurrente”, en la que se abordan en profundidad los temas y problemas clásicos de concurrencia (comunicación y sincronización sobre memoria compartida y memoria distribuida), y se realiza una breve introducción a la programación paralela. Esta asignatura se complementa con un curso de “Introducción a los sistemas operativos” en el cual se focaliza en temas de scheduling de procesos.
- En el primer semestre del cuarto año se dicta “Sistemas Paralelos”, asignatura que avanza sobre los conceptos mencionados para tratar su aplicación en arquitecturas reales del tipo multicore y cluster de multicore, y la evaluación de rendimiento orientado a High Performance Computing (HPC). Se complementa con

“Programación Distribuida”, dictada en el segundo semestre.

- En “Sistemas Paralelos” se introducen los conceptos de GPU y Cloud computing, los cuales son tomados como base para las materias optativas “Taller de Programación sobre GPUs” y “Cloud Computing. Aplicaciones en Big Data”. En la primera se profundiza sobre el modelo de programación en arquitecturas many-core como las GPU, realizando diferentes aplicaciones paralelas. En la segunda se tratan aspectos de despliegue de infraestructura virtual de cluster en Clouds, estudio de aplicaciones de HPC sobre Cloud Computing y tratamiento masivo de datos utilizando la infraestructura ofrecida por el Cloud.

### 3. Descripción y Análisis de Implementación de las Asignaturas

Como se expresó, un problema importante es encontrar una manera adecuada de introducir los conceptos básicos de concurrencia y paralelismo, ya que la programación en este paradigma resulta mucho más compleja para el alumno. Esto ocurre principalmente por la necesidad de pensar un problema compuesto por comportamientos individuales de procesos a lo que se suma la interacción entre los mismos. Por este motivo, es necesario contar con nuevas estrategias que permitan abordar tempranamente el tema [CAR03] [DEG12a] [DEG14a], profundizando y experimentándolo a lo largo de la carrera. A continuación se detalla cómo se aborda el tema en las diferentes asignaturas.

#### 3.1 Taller de Programación. Concurrencia en 1er año

La utilización de herramientas interactivas para la enseñanza de conceptos fundamentales a alumnos de un curso CS1 [ACM04][ACM08][ACM13](Taller de Programación en nuestro caso) se ha vuelto

una práctica generalizada, considerando los aspectos motivacionales y las actividades habituales de los jóvenes que ingresan a la Universidad. En general viven rodeados de pantallas táctiles, smartphones, tablets, etc y su interacción con las mismas es permanente [DEG13]. Por este motivo, se ha elegido brindarles un entorno gráfico e interactivo que permita reducir la brecha que tradicionalmente existió entre la abstracción y la posibilidad de ver gráficamente la aplicación de los conceptos estudiados [AMD09][HOO13].

El entorno CMRE (Concurrent Multi Robot Environment) es una herramienta desarrollada en el III-LIDI, destinada a introducir los conceptos de concurrencia y paralelismo. Surge como una evolución del entorno Visual da Vinci [CHA97] cuyo objetivo principal era resolver problemas donde se especifica el comportamiento de un único robot, el cual puede moverse en una ciudad compuesta por 100 avenidas (verticales) y 100 calles (horizontales) y es capaz de distinguir objetos (flores y papeles) y realizar operaciones con los mismos (juntarlos y/o depositarlos bajo ciertas condiciones); asimismo el robot puede “contar” e “informar” resultados.

CMRE [DEG12b][DEG12c][DEG14b] agrega a las funcionalidades del Visual Da Vinci la posibilidad de definir múltiples robots en la ciudad, los cuales deben ser programados para evitar conflictos y eventualmente cooperar. Cuenta con un enfoque visual e interactivo combinado con el empleo de robots físicos para la demostración de los conceptos y ejemplos de desarrollo. En cuanto a los aspectos de concurrencia, el entorno permite:

- Declarar “procesadores” o robots virtuales que representan los “cores” de una arquitectura multiprocesador real. Los robots pueden cooperar y/o competir.
- El modelo de ambiente (“ciudad”) en la que desarrollan sus tareas admite áreas privadas, parcialmente compartidas y totalmente compartidas. En un área privada sólo puede moverse un robot, en

un área parcialmente compartida se especifica el conjunto de robots que pueden moverse y en un área totalmente compartida todos los robots pueden moverse dentro de ella. Si se instancia un sólo robot en un área que abarque toda la ciudad, se repite el esquema del Visual Da Vinci.

- Cuando dos o más robots están en un área compartida (parcial o totalmente), compiten por el acceso a las esquinas del recorrido y a los recursos que allí existan. Para esto deben sincronizar a fin de evitar conflictos.
- Cuando dos o más robots (en un área común o no) desean intercambiar información (datos o control) deben hacerlo por mensajes explícitos.
- La sincronización se resuelve por un mecanismo equivalente a un semáforo binario.
- La exclusión mutua puede generarse con la declaración de las áreas alcanzadas por cada robot. Acceder a otras áreas de la ciudad, así como salir de ellas no está permitido.
- Todo el modelo de ejecución es sincrónico y permite la existencia de un reloj virtual de ciclos, que a su vez permite asignar tiempos específicos a las operaciones, simulando la existencia de una arquitectura heterogénea.
- En el entorno, cada robot tiene asociado un estado, en el que muestra el contenido de su bolsa (cantidad de flores y papeles en el modelo), esquina donde se encuentra situado, estado actual: si se encuentra ejecutando, esperando la llegada de un mensaje, o esperando por la liberación de una esquina.

A los puntos anteriores se le incorpora la comunicación en tiempo real a través de Wi-Fi con robots físicos de la línea Lego Mindstorms EV3 [LEGa][LEGb] de modo de poder ejecutar algoritmos paralelos en el entorno, con efecto directo en los robots físicos que replican sobre el terreno el comportamiento definido por los algoritmos. Este modelo de

demostración facilita la comprensión de determinados problemas por parte del alumno, tales como los conceptos de fairness, deadlock o inanición [AND00].

Desde 2013 se realizaron experiencias con un total de 40 grupos de 3 alumnos que utilizaron CMRE para desarrollar los trabajos finales de la materia. Los ejercicios elegidos se basan en los típicos problemas de concurrencia que se abordan en la bibliografía clásica para entender y explicar los conceptos básicos. Los resultados que se obtuvieron fueron sumamente positivos, ya que dichos alumnos del primer año de la carrera a partir de entrevistas manifestaron que entendieron y aplicaron los conceptos sin mayores dificultades, resultándoles una experiencia motivadora. Asimismo, propusieron posibles modificaciones y funcionalidades a agregar, las cuales son tenidas en cuenta en el desarrollo de mejoras al entorno. Para 2015, la utilización del entorno se extenderá a todos los estudiantes del Taller.

### 3.2 Programación concurrente

Dado que el impacto de la concurrencia se refleja en diferentes ámbitos de la disciplina tales como las arquitecturas, los sistemas operativos, los lenguajes y el diseño y desarrollo de aplicaciones, se impone que los futuros profesionales sean capaces de desarrollar soluciones que utilicen adecuadamente la tecnología disponible con fundamentos teóricos firmes. En este sentido, la asignatura “Programación Concurrente” aborda la profundización de los conceptos introducidos en el Taller de primer año y su aplicación en problemas de mayor complejidad.

Para la enseñanza de la concurrencia es posible utilizar dos enfoques diferentes:

1. Partir de un lenguaje en particular y mostrar la aplicación de los conceptos con las herramientas que posee el mismo.

2. Brindar los conceptos fundamentales independientemente del lenguaje de programación.

Desde hace más de 20 años se optó por el segundo enfoque, ya que permite independizarse de la tecnología y fortalecer el aprendizaje para poder aplicarlo en cualquier lenguaje.

Entre los objetivos de la asignatura se encuentran:

- Brindar los conceptos fundamentales de Concurrencia en software.
- Analizar la semántica y sintaxis para especificar concurrencia.
- Analizar el concepto de sistema concurrente compuesto por la arquitectura, el sistema operativo y los algoritmos.
- Estudiar la sincronización de procesos concurrentes por memoria compartida y mensajes, incluyendo las herramientas correspondientes.
- Vincular la concurrencia en software con los conceptos de procesamiento distribuido y paralelo.
- Desarrollar estudios de casos con diferentes lenguajes/ herramientas para concurrencia.

En la práctica de la materia se resuelven ejercicios usando comunicación y sincronización por memoria compartida y mensajes. En el caso de memoria compartida se trabaja con: variables compartidas, semáforos y monitores. Para las prácticas que abordan la sincronización por mensajes se emplea: pasaje de mensajes asíncronos (PMA), pasaje de mensajes sincrónicos (PMS), RPC y rendezvous.

Cabe mencionar que los alumnos tienen la posibilidad de realizar un trabajo sobre máquina donde se apliquen los conceptos vistos en la materia para la aprobación de la misma.

### 3.3 Sistemas paralelos

La evolución tecnológica de los procesadores ha impuesto el procesamiento paralelo [GRA03]. El objetivo principal de esta asignatura es formar al alumno (que ya tiene conocimientos previos de concurrencia y sus aplicaciones) en los fundamentos de los sistemas paralelos, los paradigmas de programación y las métricas de rendimiento asociadas. Se parte del conocimiento adquirido por los alumnos en “Programación Concurrente” desde el punto de vista de los algoritmos, para aplicarlos sobre arquitecturas paralelas como Multicores y Cluster de Multicores.

En el curso se analizan las características principales de las diferentes arquitecturas y la metodología para diseñar las aplicaciones paralelas a fin de que se adapten a las mismas para obtener el mejor rendimiento de los sistemas paralelos, orientados a High Performance Computing (HPC).

Entre los objetivos específicos de la asignatura se encuentran:

- Conocer las arquitecturas de procesamiento paralelo de propósito general y cómo afectan en el desarrollo de las aplicaciones.
- Aprender las diferentes herramientas (lenguajes/librerías) para desarrollar aplicaciones paralelas, tanto para memoria compartida como para mensajes.
- Analizar los diferentes paradigmas de resolución de sistemas paralelos y la forma de evaluar el rendimiento de este tipo de aplicaciones.
- Desarrollar aplicaciones que permitan adaptar el software al hardware para obtener el mejor rendimiento del sistema paralelo.
- Introducir a los alumnos en temas puntuales como el uso de arquitecturas específicas que no son de propósito general, tales como GPUs y Cloud Computing.

- Plantear conceptos generales de green computing relacionados con el consumo energético en multiprocesadores.

La práctica de la materia se lleva a cabo resolviendo todos los trabajos sobre máquina. Para esto se utilizan algunas de las arquitecturas paralelas con que cuenta la Facultad: sala con un cluster de 20 multicores con 4 cores cada uno, y acceso a un cluster de 16 multicores con 8 cores cada uno (Blade).

Sobre estas arquitecturas se desarrollan algoritmos con diferentes características usando los distintos modelos de comunicación/sincronización: memoria compartida, pasaje de mensajes y la combinación de ambos. En todos los casos se evalúan y comparan los rendimientos obtenidos con el fin de determinar la mejor solución a cada tipo de problema.

### 3.4 Taller de Programación de GPUs

La evolución de las GPU ha estado marcada por la industria de los videojuegos y la computación gráfica en general. Sin embargo, desde hace unos años comenzaron a ser utilizadas para resolver distintos problemas ajenos al dominio gráfico (álgebra lineal, tratamiento de secuencias de caracteres, algoritmos de criptografía, ordenación, etc), dando lugar al concepto de GPGPU (General Purpose GPU). En este aspecto, han mostrado un alto rendimiento [PSD11] además de un bajo consumo energético [RPSD12] y a menor costo con respecto a otras arquitecturas paralelas.

Entre los objetivos de la asignatura se encuentran:

- Profundizar el conocimiento de las arquitecturas tipo GPU y su programación.
- Analizar los modelos de resolución de problemas específicos.
- Comparar su rendimiento con arquitecturas paralelas tradicionales.

- Optimización de algoritmos en GPU para maximizar el rendimiento.

Los alumnos ingresan al taller con conocimientos de concurrencia y paralelismo y la capacidad para resolver problemas concurrentes de distinta naturaleza. Con la evolución de las prácticas los alumnos aprenden a resolver esos problemas utilizando las GPU.

La materia adoptó un enfoque de modalidad taller, donde se vinculan fuertemente la teoría y el desarrollo práctico. Las clases teórico/prácticas se llevan a cabo en una sala de PCs equipada con GPU, donde el alumno puede realizar el desarrollo y la experimentación solicitada en los trabajos prácticos. Esta metodología permite contrastar rápidamente los resultados obtenidos con los contenidos teóricos.

### 3.5 Cloud computing. Aplicaciones en Big Data

Los avances en las tecnologías de virtualización y cómputo distribuido han dado origen al paradigma de Cloud Computing, que se presenta como una alternativa a los tradicionales sistemas de Clusters y Grids para ambientes de HPC y también como un esquema de procesamiento masivo de datos y de servicios remotos de gran impacto en la Informática actual. Esta tecnología provee una abstracción de los recursos de hardware permitiendo ejecutar simultáneamente múltiples instancias de Software en máquinas virtuales sobre un único hardware físico (centralizado o distribuido).

Son objetivos de la asignatura:

- Profundizar el conocimiento de la tecnología Cloud y su utilización.
- Analizar los problemas principales en las aplicaciones sobre Cloud Computing.
- Estudiar los fundamentos de los problemas de Big Data.

- Resolver problemas de Big Data, utilizando arquitecturas Cloud.

Cloud Computing proporciona grandes conjuntos de recursos físicos y lógicos (infraestructura, plataformas de desarrollo, almacenamiento y/o aplicaciones), fácilmente accesibles y utilizables por medio de una interfaz de administración web, con un modelo de arquitectura “virtualizada”. Estos recursos son proporcionados como servicios (“As a Service”) y pueden ser dinámicamente reconfigurados para adaptarse a una carga de trabajo variable (escalabilidad), logrando una mejor utilización y evitando el sobre o sub dimensionamiento (elasticidad). [THO13] [ROT13].

Al mismo tiempo uno de los campos de aplicación más importante para las arquitecturas Cloud es el de “big data” en el cual la capacidad de procesamiento se aplica al tratamiento masivo de información (numérica, textual, imágenes, señales, etc.), buscando extraer determinado conocimiento. [MAY12] [FUR13].

La práctica de la materia se lleva a cabo realizando todos los trabajos experimentales sobre el Cloud público de Amazon Web Services (AWS) [AMA15a]. Para esto la asignatura cuenta con una beca de investigación y educación proporcionada por AWS. A los alumnos se les brinda el acceso vía WEB, por medio de un usuario con perfil de administrador. Sobre AWS, se despliegan instancias de máquinas virtuales configuradas en Cluster, se ejecutan aplicaciones de HPC y se realiza tratamiento masivo de datos utilizando Amazon EMR (Amazon Elastic MapReduce) [AMA15b].

#### 4. Resultados y Conclusiones

Se han expuesto las diferentes acciones que la Facultad de Informática de la UNLP ha tomado para la incorporación de los nuevos temas relacionados con los cambios tecnológicos en las arquitecturas de los

procesadores. Naturalmente en este trabajo sólo se analiza la línea de Concurrencia y Paralelismo, contemplando los cambios curriculares en las asignaturas existentes y también los nuevos cursos optativos que se han implementado.

Debemos reconocer la complejidad de la incorporación de los nuevos contenidos, que se combina con el requerimiento de recursos humanos formados en temas de Concurrencia, Paralelismo, Cloud Computing, Big Data y programación sobre Multicores y GPUs. El esfuerzo actual está centrado en este punto, a través de la Especialización y Maestría en Computación de Alto Desempeño y una serie de cursos específicos en el Doctorado en Ciencias Informáticas.

#### Bibliografía

- [ACM04] ACM/IEEE-CS Joint Task Force on Computing Curricula. “Computer Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering”. Report in the Computing Curricula Series. 2004.
- [ACM08] ACM/IEEE-CS Joint Interim Review Task Force. “Computer Science Curriculum 2008: An Interim Revision of CS 2001”. Report from the Interim Review Task Force. 2008.
- [ACM13] ACM/IEEE-CS Joint Task Force on Computing Curricula. “Computer Science Curricula 2013”. Report from the Task Force. 2013.
- [ACM68] ACM Curriculum Committee on Computer Science. “Curriculum ‘68: Recommendations for the undergraduate program in computer science”. Communications of the ACM, 11(3):151-197. 1968.
- [ACM78] ACM Curriculum Committee on Computer Science. “Curriculum ‘78: Recommendations for the undergraduate program in computer science”. Communications of the ACM, 22(3):147-166. 1979.

- [ACM99] ACM Two-Year College Education Committee. "Guidelines for associate-degree and certificate programs to support computing in a networked environment". New York: The Association for Computing Machinery. 1999.
- [AMA15a] Amazon Elastic Compute Cloud (AmazonEC2). <http://aws.amazon.com/es/ec2>. Abril 2015.
- [AMA15b] Amazon Elastic MapReduce (AmazonEMR). <http://aws.amazon.com/es/elasticmapreduce>. Abril 2015.
- [AMD09] AMD. "Evolución de la tecnología de múltiple núcleo". <http://multicore.amd.com/es-ES/AMD-Multi-core/resources/Technology-Evolution>. 2009.
- [AND00] Andrews G. "Foundations of Multithreaded, Parallel, and Distributed Programming". Addison Wrsley, 2000.
- [CAR03] Carr S., Mayo J., Shene C. "Threadmentor: a pedagogical tool for multithreaded programming". ACM Journal of Educational Resources, 3:1-30, 2003.
- [CHA97] Champredonde, Raul, y Armando De Guisti. "Design and Implementation of Visual Da Vinci". Proceeding del III Congreso Argentino en Ciencias de la Computación, 1997.
- [DAS89] Dasgupta S. "Computer Architecture. A Moder Synthesis. Volume 2: Advanced Topics". Jhon Wilet & Sons. 1989.
- [DEG12a] De Giusti A. E., Frati F. E., Leibovich F., Sánchez M., De Giusti L. C., Madoz M. C. "Concurrencia y Paralelismo en CS1: la utilización de un Lenguaje Visual orientado". Proceeding del VII Congreso de Tecnología en Educación y Educación en Tecnología. 2012
- [DEG12b] De Giusti L. C., Frati F. E., Leibovich F., Sánchez M., Madoz M. C. "LMRE: Un entorno multiprocesador para la enseñanza de conceptos de concurrencia en un curso CS1". Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología. Págs. 7 - 15. 2012.
- [DEG12c] De Giusti A. E., Frati F. E., Sánchez M., De Giusti L. C. "LIDI Multi Robot Environment: Support software for concurrency learning in CS1". Proceeding of IEEE International Conference on Collaboration Technologies and Systems. Pág. 294-298. 2012.
- [DEG13] De Giusti A. E. "El cambio tecnológico como motor de la Investigación en Informática". Conferencia inaugural del Workshop de Investigadores en Ciencia de la Computación (WICC). 2013.
- [DEG14a] De Giusti, L., Leibovich, F., Sanchez, M., Chichizola, F., Naiouf, M., De Giusti, A. "Desafíos y herramientas para la enseñanza temprana de Concurrencia y Paralelismo". Congreso Argentino de Ciencias de la Computación (CACIC), 2014.
- [DEG14b] De Giusti, A., De Giusti L., Leibovich, F., Sanchez, M., Rodriguez Eguren, S. "Entorno interactivo multirrobot para el aprendizaje de conceptos de Concurrencia y Paralelismo". Congreso Tecnología en Educación, Educación en Tecnología. 2014.
- [DIJ65] Dijkstra E. W. "Solution of a problem in concurrent programming control". Communications of the ACM, 8(9):569, 1965.
- [DIJ78] Dijkstra E. W. "Finding the Correctness Proof of a Concurrent Program". In Program Construction, International Summer Schoo, Friedrich L. Bauer and Manfred Broy (Eds.). Springer-Verlag, 24-34, 1978.
- [FUR13] Furth, Escalante "Handbook of Data Intensive Computing". Springer 2013.
- [GEP06] Gepner P., Kowalik M.F. "Multi-Core Processors: New Way to Achieve High System Performance". In: Proceeding of International Symposium on Parallel Computing in Electrical Engineering 2006 (PAR ELEC 2006). Pags. 9-13. 2006.
- [GOL72] Goldstine H. H. "The Computer". Princeton University Press, 1972.
- [GPG] GPGPU. "General-Purpose Computation on Graphics Processing Units". <http://gpgpu.org>.
- [GRA03] Grama A., Gupta A., Karypis G., Kumar V. "Introduction Parallel Computing". Pearson Addison Wesley, 2nd Edition, 2003.
- [HAN77] Hansen P. B. "The Architecture of Concurrent Processes". Prentice Hall, 1977.
- [HOA78] Hoare C. "Communicating Sequential Processes". Communications of the ACM, 21(8): 666-677, 1978.

- [HOA85] Hoare C. "Communicating Sequential Processes". Prentice Hall, 1985.
- [HOO13] Hoonlor A., Szymanski B. K., Zaki M. J., Thompson J. "An Evolution of Computer Science Research". Communications of the ACM. 2013.
- [HWA84] Hwang K., Briggs F. A. "Computer Architecture and Parallel Processing". McGraw Hill, 1984.
- [HWA93] Hwang K. "Advanced Computer Architecture: Parallelism, Scalability, Programmability". McGraw Hill, 1993.
- [KIR13] Kirk, Hwu "Programming massively parallel processors". Elsevier 2013.
- [LEGa] "Lego Education". <http://www.legoeducation.us/eng/characteristics/ProductLine~LEGO%20MINDSTORMS%20Education%20EV3>.
- [LEGb] Lego. "LEGO Mindstorms EV3 Announced". <http://brickextra.com/2013/01/10/lego-mindstorms-ev3-announced/>
- [MAY12] Mayer-Schonberger, Cukier "Big Data". Houghton Mifflin Harcourt Publishing 2012.
- [MCC08] McCool M. "Scalable Programming Models for Massively Parallel Multicores". Proceedings of the IEEE, 96(5): 816–831, 2008.
- [OLS83] Olsen E. W., Whitehill S. B. "Ada for Programmers". Prentice Hall, 1983.
- [PSD11] Pousa A., Sanz V., De Giusti A. "Performance Analysis of a Symmetric Cryptographic Algorithm on Multicore Architectures". Congreso Argentino de Ciencias de la Computación (CACIC), 2011.
- [ROT13] Rothon J. "Cloud computing explained: Implementation handbook for enterprises". Recursive Press 2013.
- [RPSD12] Romero F., Pousa A., Sanz V., De Giusti A. "Consumo Energético en Arquitecturas Multicore. Análisis sobre un Algoritmo de Criptografía Simétrica". Congreso Argentino de Ciencias de la Computación (CACIC). 2012.
- [THO13] Thomas Erl "Cloud Computing. Concepts, Technology & Architecture". Prentice Hall 2013.