

Investidura DOCTOR HONORIS CAUSA DAVID A.PADUA

Discurso DAVID A. PADUA

Magnífico y Excelentísimo Señor Rector
Excelentísimo Señores Vicerrectores
Ilustrísima Señora Secretaria General
Ilustrísimo Señor Director de la Escuela de Ingeniería Informática
Miembros del Claustro Universitario,
Señoras y Señores

Agradezco inmensamente al Señor Rector, a las Autoridades Universitarias y al Consejo de Gobierno de la Universidad de Valladolid por este gran honor. Para mí, no hay galardón más importante que un Doctorado Honoris Causa de una de las universidades con más historia del mundo. Mi agradecimiento también a la Junta de la Escuela de Ingeniería Informática de Valladolid, a los profesores de la Escuela, y particularmente al Profesor Diego Llanos y al Director de la Escuela, Prof. Benjamín Sahelices y por la nominación. Finalmente, quiero agradecer también al Profesor Diego Llanos por su muy generoso Laudatio.

Fui uno de los primeros alumnos de la carrera de Computación en Venezuela. Los grandes centros de computación de las compañías petroleras de entonces y la presencia de numerosos científicos de otras tierras, especialmente de Argentina, hicieron posible la creación tan temprana de un Departamento de Computación en Venezuela y le dio a este departamento los grandes maestros a los que estoy tan agradecido. La verdad es que no sabía mucho de la disciplina y escogí la carrera en gran parte por recomendaciones, sin tener mucha base para la decisión. Ahora, muchos años después, puedo decir que fue una elección afortunada por la gran influencia de la informática en nuestro mundo, por los apasionantes temas que estudia, y porque ser parte de esta profesión me permitió observar de cerca uno de los desarrollos tecnológicos más impresionantes en la historia de la humanidad.

Lo primero que viene a la mente cuando hablamos de este desarrollo es el casi increíble progreso de la velocidad de cálculo. En 1965, Gordon Moore observó que el número de transistores en circuitos integrados se había duplicado cada año desde la invención de estos dispositivos y predijo que el crecimiento exponencial continuaría. Esta predicción, conocida hoy como la Ley de Moore, ha sido cumplida fielmente por la industria electrónica durante más de 50 años. La continua reducción del tamaño de los transistores, que hizo y sigue haciendo posible el aumento de la densidad de los circuitos integrados, también hizo posible el aumento de la velocidad de cálculo de los procesadores y una dramática reducción en el consumo de energía. Un reportaje de Intel preparado en celebración de los cincuenta años de la Ley de Moore¹ nos dice que comparados con los procesadores de hace cuarenta años los de hoy son 3.500 veces más rápidos, consumen 90.000 veces menos energía, y cuestan 60.000 veces menos por transistor. Y que, si la tecnología automotriz se hubiera desarrollado de la misma manera, los automóviles se desplazarían a 500.000 kilómetros por hora, podrían recorrer 850.000

¹ <http://www.intel.com/content/www/us/en/silicon-innovations/moores-law-technology.html>

kilómetros con solo un litro de gasolina, y costarían 4 céntimos de euro. No solo los procesadores se hicieron cada vez más rápidos, sino que ha sido posible crear ordenadores con un número cada vez más grande de procesadores. Hoy en día, hay ordenadores con millones de procesadores como Sequoia en los Estados Unidos y Sunway TaihuLight en la República Popular China. La combinación de la Ley de Moore, que nos ha dado procesadores cada vez más rápidos, con el aumento en el número de procesadores ha resultado en sistemas con una capacidad de cálculo que es difícil de concebir: más de 125.000 billones de operaciones por segundo y estos son de los billones europeos o sea que hablamos de más de 125.000 millones de millones de operaciones por segundo.

Además de mejorar en su velocidad de cálculo, los ordenadores también han evolucionado en la capacidad de resolver problemas difíciles. En efecto, se ha aprendido mucho de cómo realizar con gran eficacia los cálculos numéricos que son tan importantes para las ciencias naturales y las ingenierías. Así, poderosos métodos de cálculo ejecutados por ordenadoras son ubicuos en el estudio de todas las ciencias como la química, la física, la meteorología, y la biología y en los diseños en todas las ingenierías, incluyendo los de automóviles, aeronaves, plantas nucleares y redes eléctricas. Poderosos métodos de cálculo junto con el aumento de velocidad de los ordenadores han hecho posible la creación de programas que con gran precisión representan la naturaleza y las construcciones de los ingenieros y son hoy en día un componente fundamental de la investigación y el desarrollo.

También ha habido gran progreso en el desarrollo de métodos para la manipulación de símbolos. Dos ejemplos de este increíble progreso son los triunfos de sistemas de IBM primero en ajedrez, sobre Garry Kasparov en 1997, y, en 2011, en el concurso de preguntas y respuestas *Jeopardy!*. Los ganadores de ambos eventos fueron máquinas y programas desarrollados por IBM. Estos logros demostraron que los ordenadores pueden superar al hombre en actividades donde el éxito ha sido considerado como demostración de gran inteligencia y erudición. El progreso en el desarrollo de estos métodos ha continuado, y de hecho se ha acelerado en los últimos años, con avances en aprendizaje automático y robótica y fascinantes logros como vehículos autónomos y traducción automática en tiempo real. Particularmente importante ha sido el desarrollo de las técnicas del aprendizaje automático que han permitido la resolución de problemas muy diferentes usando las mismas estrategias.

Una tercera dimensión de progreso de los ordenadores, igualmente impresionante, es el desarrollo de las tecnologías de redes. Estas, sin lugar a dudas, caracterizan la sociedad de nuestros tiempos con el correo electrónico, el comercio electrónico, los buscadores de información en Internet, y las redes sociales. El impacto de las redes sobre la sociedad es universalmente reconocido. Se dice por ejemplo que los políticos de los tiempos de John Kennedy tenían que entender la televisión pero que en estos tiempos de Barak Obama y Donald Trump los políticos tienen que entender las redes sociales y en especial Twitter. El impacto de las redes sobre la difusión del conocimiento se ve en la frecuencia con la que se usan los buscadores de Internet y en la disponibilidad a lo largo y ancho del planeta de toda clase de información. Hoy, por ejemplo, los artículos en revistas científicas se consiguen en línea y las grandes salas que contenían estanterías de revistas han sido reemplazadas en muchas bibliotecas por salas de lectura. Hoy solo queda el recuerdo de los paquetes de artículos que los profesores jóvenes enviaban por correo regular para promocionar su trabajo y de las tarjetas postales enviadas de Cuba y de los países de Europa del Este pidiendo copias de publicaciones científicas.

Estas tres áreas son solo una pequeña parte de los logros de la informática. La lista completa es muy numerosa como se hace evidente si observamos alrededor y recordamos temas como la productividad

en la empresa, el impacto de la robótica en la industria, las herramientas computarizadas de la medicina de hoy, y los teléfonos inteligentes. Con esta lista interminable es fácil de ver por qué quienes hemos tenido la suerte de participar en el estudio y la enseñanza de la informática en las últimas décadas hemos vivido en un mundo mágico de grandes logros, de numerosos beneficios para la humanidad, y de esperanzas en un futuro que traerá beneficios que hoy no imaginamos, así como muchos no imaginábamos los beneficios que traería el Internet.

Cuando nos preguntamos que ha hecho posible todo esto debemos reconocer en primer lugar a los pioneros de la informática cuya visión y creatividad creo esas técnicas y dispositivos que definen nuestros tiempos. No son héroes populares, pero deberían serlo. Por otra parte, viendo el progreso desde otra perspectiva, observamos que los logros de la informática son inesperados e incomprensibles. En una clase magistral de 1959, Eugene Wigner nos hablaba de la irrazonable eficacia de las matemáticas en las ciencias naturales². La informática de hoy se beneficia de esta capacidad de las matemáticas, así como también de la irrazonable eficacia del aprendizaje automático³. Repitiendo a Wigner podemos decir que la eficacia de la informática es sin duda “un regalo maravilloso que ni entendemos ni merecemos”. Lo que nos queda es estar agradecidos por un Universo cuyas leyes han permitido tantos logros.

* * * * *

Junto al avance de la velocidad de cálculo de las máquinas, uno de los logros más importantes en informática fue la introducción de los lenguajes de programación de alto nivel. Este no es un tema que se discuta mucho fuera de los círculos de la informática, pero tiene gran importancia ya que estos lenguajes han hecho posible la larga lista de contribuciones que acabamos de esbozar. Al ejecutar programas, los ordenadores son guiados por secuencias de operaciones muy elementales tales como sumas, multiplicaciones, y comparaciones de valores. Es posible escribir directamente los programas en el lenguaje de esas secuencias elementales, conocido como *lenguaje de máquina*. De hecho, en los primeros tiempos, todos los programas fueron escritos en este lenguaje. Sin embargo, el lenguaje de máquina es muy difícil de usar, hasta el punto de que hubiera sido prácticamente imposible escribir a mano los cientos de millones de operaciones elementales que integran los programas más importantes de hoy.

Esta dificultad llevó al desarrollo de lenguajes de alto nivel que son notaciones más compactas y por lo tanto más fáciles de usar que las secuencias de instrucciones elementales del lenguaje de máquina. Los lenguajes de alto nivel ayudan al programador y facilitan, y en muchos casos hacen posible, el desarrollo de programas complejos. Sin embargo, los ordenadores no pueden ejecutar programas en lenguaje de alto nivel y por lo tanto ellos deben ser convertidos a lenguaje de máquina por una clase especial de programas llamados compiladores. Así que el propósito de los compiladores es traducir programas en lenguaje de alto nivel a programas en lenguaje de máquina

² E. Wigner, *The Unreasonable Effectiveness of Mathematics in the Natural Sciences*, **Comm. Pure and Applied Mathematics**, vol. 13, no. 1, 1960, pp. 1–14.

³ A. Halevy, P. Norvig, and F. Pereira. 2009. *The Unreasonable Effectiveness of Data*. **IEEE Intelligent Systems** 24, 2 (March 2009), 8-12.

En 1956, un equipo de IBM dirigido por John Backus diseñó FORTRAN, uno de los primeros lenguajes de alto nivel, y programó su compilador, el primer compilador comercial⁴. La palabra FORTRAN es un acrónimo derivado de la frase “traductor de fórmulas”. En FORTRAN, las fórmulas se escriben de la manera que aprendimos en los cursos de matemáticas. El compilador de FORTRAN traduce cada fórmula a una secuencia de operaciones elementales. La conversión de fórmulas y de otras partes del lenguaje es un proceso difícil y era aún más difícil en 1956 cuando se sabía mucho menos de métodos de traducción. Saul Rosen en un artículo escrito en 1967, diez años después del lanzamiento del primer compilador de FORTRAN, decía que “Al principio la gente pensó que nunca se completaría. Después, cuando fue puesto a prueba, con muchos defectos y muchas partes importantes sin terminar, muchos pensaron que nunca funcionaría.”⁵ Afortunadamente, para IBM y para todos nosotros, el compilador fue mejorando gradualmente hasta que se hizo lo suficientemente robusto como para ser usable.

Este éxito requirió gran habilidad e ingenio porque no solamente era necesario traducir correctamente sino también traducir a programas que fueran eficientes, que ejecutaran a alta velocidad. Un programa de alto nivel se puede convertir a lenguaje de máquina de muchas maneras diferentes, unas más rápidas que otras. Muchos creían que los compiladores no podrían producir programas en lenguaje de máquina suficientemente rápidos y que los programas para ser realmente eficientes tenían que ser escritos por seres humanos directamente en lenguaje de máquina, o como se dice en nuestra disciplina tenían que ser escritos *a mano*. John Backus narraba en un artículo de 1979⁶ que la meta que se habían trazado para garantizar la aceptación del lenguaje era que el compilador convirtiera a programas en lenguaje de máquina que fueran al menos la mitad de rápidos que los programas escritos a mano. No sabemos qué tan cerca llegaron de esa meta, pero no pueden haber estado muy lejos de ella ya que el compilador de FORTRAN fue eventualmente aceptado universalmente.

Sesenta años después, podemos ver la magnitud de la contribución de John Backus y su equipo. El éxito de su lenguaje FORTRAN abrió el camino a la era de la informática y permitió el desarrollo de la inmensa variedad de programas complejos que definen la informática de hoy. FORTRAN fue el primero de una numerosa colección de lenguajes de alto nivel que incluye COBOL, Lisp, ALGOL, PL/1, Pascal, Prolog, C, C++, Java, JavaScript, Python, R y cientos más. Por si fuera poco, FORTRAN es todavía usado con mucha frecuencia, especialmente en programas de alto rendimiento para el cálculo de modelos matemáticos. El equipo de John Backus estaba integrado por solo seis personas y trabajaron en un área de la que no se habla mucho, pero la contribución de este pequeñísimo equipo ha sido la base del trabajo de millones de profesionales de la informática y ha influenciado las vidas de miles de millones de usuarios.

* * * * *

La informática es apasionante como tema de investigación en parte porque es una disciplina joven cuya edad se mide en décadas y no en siglos o milenios. Una manifestación de esta juventud es que nuevas subáreas de la informática, de gran importancia e interés, surgen constantemente. Por ejemplo, seguridad, minería de datos y aprendizaje automático son algunas de las áreas que han crecido en importancia en los últimos años y como resultado han capturado el interés de muchos investigadores. Además, hay todavía mucho por hacer y mucho que debe hacerse aun en las áreas que ya tienen más de

⁴ David Padua. *The Fortran I Compiler*. **Computing in Science and Engineering** 2, 1 (January 2000), 70-75.

⁵ Saul Rosen. *Programming systems and languages: a historical survey*. **Proceedings of the April 21-23, 1964, spring joint computer conference (AFIPS '64 (Spring))**. ACM, New York, NY, USA, 1-15.

⁶ John W. Backus. *The History of FORTRAN I, II and III*. **IEEE Annals of the History of Computing** 1(1): 21-37 (1979)

sesenta años como diseño de ordenadores y métodos de compilación. Avances en estas áreas medulares de la informática son cada vez más difíciles, pero al mismo tiempo son de inmensa importancia, ya que el progreso de la informática descansará sobre estos avances.

En el área de diseño de máquinas es necesario continuar aumentando la velocidad de cálculo. A pesar de lo mucho que se ha hecho, aún más es necesario para reducir el tiempo de respuesta y hacer posibles cálculos más precisos. Cálculos de más alta fidelidad son necesarios para lograr avances en diseños de naves aeroespaciales y de automóviles, de sistemas de generación de energía, de tratamientos médicos más eficaces, de sistemas embebidos y de teléfonos celulares inteligentes. En muchos casos el aumento en fidelidad de cálculo requerirá nuevos métodos que serán posibles solo con nuevos niveles de poder computacional⁷. Sin embargo, estos nuevos niveles no serán fáciles de obtener. Mientras en los primeros tiempos al aumentar la densidad de transistores se aumentaba la velocidad de ejecución de los procesadores, hoy esta relación ya no es verdadera y el mecanismo más eficaz para el aumento de la velocidad de cálculo es aumentar el número de procesadores dentro de cada ordenador. Gracias a la Ley de Moore, que todavía se cumple, podemos hoy aumentar cada vez más el número de procesadores sin aumentar mucho el costo de las máquinas. Observamos este aumento en toda clase de máquinas, desde los teléfonos celulares inteligentes hasta las máquinas más poderosas. Para estas últimas, nuestra comunidad se ha trazado la meta de desarrollar en la próxima década máquinas que ejecuten a exaescala, es decir un trillón de operaciones por segundo. Aquí nuevamente el término tiene la acepción europea de millones de millones de millones de operaciones por segundo.

Para continuar aumentando el número de procesadores en los ordenadores será necesario solventar dos dificultades. La primera es el crecimiento de consumo de energía que resulta del aumento del número de procesadores y de la comunicación entre ellos. Para que máquinas más poderosas se hagan una realidad será necesario avances en electrónica, en la organización interna de los ordenadores y el desarrollo de métodos computacionales que en lugar de enfocarse solamente en la reducción del número de operaciones aritméticas eviten comunicaciones tal vez a expensa de realizar cálculos adicionales.

La segunda dificultad es que alcanzar un uso eficaz de máquinas con numerosos procesadores requiere programas de gran complejidad. La razón es que la eficacia depende de que las operaciones se distribuyan de forma balanceada entre los procesadores. Además, ya que las operaciones de un programa no pueden realizarse en cualquier orden, al distribuir las operaciones de cálculo es necesario añadir operaciones adicionales de sincronización para garantizar que ninguna operación se ejecute antes de tiempo. Ocuparse de la distribución y sincronización requiere un gran esfuerzo de programación y aumenta la probabilidad de errores que luego hay que corregir. Reducir este costo adicional de programación es de gran importancia ya que el costo de programación es de por sí alto hoy en día. El crecimiento de este costo es un impedimento a la popularización de las máquinas con numerosos procesadores y por lo tanto un obstáculo al avance de la velocidad de cálculo.

La mejor manera de reducir el costo de la programación de máquinas con múltiples procesadores es desarrollar herramientas para facilitar su programación y así hacer posible su aceptación. Junto con mis estudiantes y colaboradores, he dedicado mucho esfuerzo al desarrollo de estas herramientas. Nuestro

⁷ U.S. Department of Energy. The Opportunities and Challenges of Exascale Computing. Office of Science, Washington, D.C., 2010; <http://science.energy.gov/~media/ascr/ascac/pdf/>

trabajo construye sobre bases creada por mis excelentes maestros y mentores en la Universidad de Illinois, donde hice mis estudios de doctorado. Fue entonces que comencé mi trabajo en el desarrollo de compiladores, la herramienta de programación usada con más frecuencia. Hemos desarrollado una variedad de técnicas que hacen posible que los compiladores hagan más que simplemente descomponer las fórmulas en secuencias de instrucciones de máquina, sino que también identifique cuales operaciones que pueden ejecutarse simultáneamente, las organice en unidades de cálculo, y finalmente inserte las sincronizaciones necesarias. Aunque nos gustaría que todo el trabajo lo hiciera el compilador, esta meta no se ha alcanzado todavía. Los compiladores son de gran ayuda, pero hoy el programador que busca alta velocidad de ejecución debe evaluar lo que produce el compilador y a menudo continuar el proceso de transformación a mano. Así que hay todavía la oportunidad y la necesidad de avances. Quizás lo más importante que permanece sin desarrollar son métodos para la identificación por parte del compilador de secuencias optimas de transformación de programas. Poder identificar estas secuencias garantizaría la optimalidad del producto de los compiladores y así evitaría la necesidad de trabajo adicional tan frecuente hoy en día.

Tambien hemos trabajado en el desarrollo de herramientas que hacen posible evitar prácticamente todo el trabajo de programación. Estas herramientas generan automáticamente programas eficientes a partir de descripciones generales. Los métodos desarrollados han sido usados por la industria para la generación de rutinas de biblioteca a una fracción del costo que requeriría un desarrollo manual. La estrategia usada por estas herramientas es la de explorar el espacio de todos los métodos que pueden resolver el problema. Esta exploración tiene que hacerse inteligentemente porque el número de posibles métodos es a menudo inmenso y una exploración exhaustiva podría tardar años.

Una tercera clase de herramientas son colecciones de módulos que implementan operaciones fundamentales que pueden combinarse naturalmente para crear computaciones eficientes. Nuestra visión es que los programadores desarrollen sistemas haciendo uso de una variedad de herramientas de tal manera que algunas partes sean programadas usando lenguajes de alto nivel tradicionales y luego transformados por compiladores poderosos, otras generadas a partir de descripciones generales, y finalmente otras representadas como combinación de módulos pre-fabricados.

Mucho se ha logrado en las últimas décadas y entendemos mejor el proceso de compilación, de generación y cuales abstracciones son de más utilidad para la programación, pero más debe hacerse en el desarrollo de esas herramientas porque los programadores de hoy o bien deben resignarse a una eficiencia subóptima o con demasiada frecuencia deben avocarse a meticulosamente cambiar sus programas para mejorar su eficiencia. Estoy convencido de que esta situación mejorará con el tiempo y que eventualmente ningún programador podrá hacerlo mejor a mano que la combinación de herramientas. Así como las maquinas vencen en juegos como ajedrez y go y en juegos de preguntas y respuestas también las máquinas podrán vencer en la producción de programas de máxima velocidad. El resultado de esta situación será permitir que continúe el rápido avance de la velocidad de cálculo ya que nuevas clases de ordenadores no aumentarían el costo de programación.

* * * * *

Los grandes logros en la informática están cambiando nuestro mundo a pasos cada vez más rápidos. Es la responsabilidad de nuestra sociedad reaccionar para beneficiarse de la mejor manera posible de este desarrollo. En particular, por razones de justicia social, se debe facilitar el acceso universal a máquinas e interconexiones. Mucho se ha avanzado en esta dirección en los últimos años gracias a la reducción del

precio de los ordenadores personales y la creciente disponibilidad de acceso a redes de ordenadores. Pero es necesario hacer más. Afortunadamente hay razones para ser optimistas. Numerosas iniciativas de la industria, los gobiernos, y las universidades se enfocan en este problema. Ejemplos incluyen el “Connectivity Lab” de Facebook cuya meta es hacer más asequible el acceso a la internet en el mundo subdesarrollado y el programa “One Laptop Per Child” del Instituto Tecnológico de Massachusetts que tiene como misión contribuir a la educación de niños haciendo los computadores personales más asequibles.

También el desarrollo de la informática tiene consecuencias para la enseñanza universitaria. El resultado de la proliferación de los ordenadores y los avances en la resolución de problemas es que una fracción muy importante de los empleos del futuro requerirá la habilidad de programar. Esta predicción esta respalda por la intuición colectiva que se manifiesta por ejemplo en la demanda que tienen hoy en día los cursos de programación. Esta demanda es tan grande que, a pesar de la extensa contratación de profesores, numerosos estudiantes se gradúan de universidades en los Estados Unidos sin haber satisfecho su deseo de registrarse en cursos de programación debido a restricciones de espacio. Creo que las universidades no solamente deben aumentar su capacidad de enseñanza de la programación para satisfacer la demanda actual, sino que deben tener como meta dar educación en informática a tantos estudiantes como sea posible en todas las disciplinas. Una meta podría ser que todo graduado universitario tenga una sólida preparación en informática. Esto servirá no solamente para satisfacer las necesidades que hoy se conocen, sino que al aumentar el número de profesionales con educación en informática se abrirían nuevas direcciones de desarrollo. No soy de los que cree que los avances in informática conducirán al desempleo masivo. Creo que más bien el trabajo rutinario será una cosa del pasado, que habrá grandes oportunidades de desarrollar la creatividad, y que los sistemas de informática, de reemplazar al hombre, serán herramientas que aumentarán la productividad y la creatividad.

Finalmente, y quizás sorprendentemente, los avances en informática claman por más y mejor educación en las humanidades. El desarrollo de las redes sociales ha hecho posible un grado de interacción entre los hombres sin precedentes. Cada vez más participamos en una conversación universal. Esto que tiene mucho de positivo, al mismo tiempo crea dificultades. Las noticias fluyen sin mucho control. Lo verdadero y lo falso se mezclan y saber lo que es real y lo que es falso requerirá una gran capacidad de análisis crítico. Sostener la democracia y evitar desastres trágicos en este mundo de comunicaciones instantáneas requerirá de ciudadanos altamente educados. Hoy resuena con más fuerza que nunca las palabras de Bolívar grabadas bajo su estatua en la Universidad que lleva su nombre: “Un pueblo ignorante es instrumento ciego de su propia destrucción.” Es nuestra misión en las universidades educar para la democracia con la historia, la literatura, y la filosofía, y de desarrollar en nuestros estudiantes la habilidad de expresar ideas claramente y de discernir. Estoy seguro que nuestras universidades seguirán estando a la altura de esta misión. Después de todo, según el himno de la Universidad de la que recibí mi primer título, las universidades son casas que vencen las sombras. Y las vencen porque fueron construidas por la sabiduría, según el lema de la Universidad que me dio mi último título.