

INVESTIDURA DEL PROFESOR
DR. DAVID A. PADUA
COMO “DOCTOR HONORIS CAUSA” POR LA
UNIVERSIDAD DE VALLADOLID

“LAUDATIO”

DR. DIEGO R. LLANOS FERRARIS

VALLADOLID

2017

SR. RECTOR MAGNÍFICO

MIEMBROS DEL CLAUSTRO UNIVERSITARIO
SEÑORAS Y SEÑORES

Comenzaré por manifestar que es para mí un excepcional honor y privilegio actuar como Padrino de la investidura como Doctor Honoris Causa de alguien que reúne una indiscutible talla científica y una personalidad que constituye un modelo para todos los que lo conocen. Me refiero al Dr. David Padua.

Por ello agradezco profundamente a todas las personas y órganos de la Universidad de Valladolid que han intervenido en el proceso de aceptación de la concesión de este Doctorado Honoris Causa, por su interés en reconocer públicamente la contribución del Dr. Padua a la disciplina de la programación de ordenadores. El estudio de la obra del Dr. Padua en dicho campo podría servir para ilustrar la evolución de la Informática de los últimos cuarenta años. El desarrollo de sus líneas de investigación ha ido siempre por delante de las necesidades tecnológicas del momento, lo que convierte al Dr. Padua en pionero en el estudio de los problemas asociados a la computación paralela.

* * * * *

Antes de describir algunas de las principales aportaciones del Dr. Padua a la disciplina, me gustaría describir brevemente su biografía académica. El Dr. David Padua, de origen venezolano, se graduó en Informática en la Universidad Central de Venezuela en 1973, doctorándose en la Universidad de Illinois en Urbana-Champaign en 1979, con una tesis

titulada “Multiprocesadores de alta velocidad”, bajo la dirección del Prof. David J. Kuck. Esta tesis analizó diversos problemas relacionados con la organización de multiprocesadores, entendiendo como tales a aquéllos ordenadores que disponen de varios procesadores que cooperan en la resolución de una tarea. Además de proponer nuevos mecanismos de interconexión entre la memoria y los procesadores, este trabajo sentó las bases de lo que sería una de sus principales líneas de investigación, las técnicas de paralelización automática de programas, un tema sobre el que volveremos en un momento. Las publicaciones derivadas de esta tesis han sido citadas más de mil veces desde entonces, lo que ayuda a poner en perspectiva la relevancia de esta contribución.

Tras la defensa de su tesis doctoral, el Dr. Padua trabajó cuatro años como Profesor Agregado en la Universidad Simón Bolívar de Venezuela, para volver a la Universidad de Illinois en Urbana-Champaign en 1985, primero como Profesor Asistente entre 1985 y 1990, luego como Profesor Titular hasta 1995, y desde entonces y hasta la fecha como Catedrático. Es importante destacar que la Universidad de Illinois en Urbana-Champaign es una de las más reconocidas a nivel mundial en el campo de la Informática, ocupando en la actualidad el puesto 13 en el Ranking de Shangai.

* * * * *

La prolífica labor investigadora del Dr. Padua se centra en dar respuesta a una cuestión que lleva preocupando a los investigadores en informática desde los albores de esta disciplina: cómo dividir una tarea de cálculo de modo de que varios ordenadores puedan cooperar en su resolución, finalizándola más rápidamente que si sólo dedicáramos a ella un único ordenador. Desde los albores mismos de la informática, los informáticos han buscado resolver problemas numéricos con la máxima rapidez. Resulta natural que la idea de dividir el trabajo entre dos ordenadores haya estado siempre en la mente de los investigadores, tanto los que desarrollan programas como los que se dedican al diseño de las máquinas computadoras.

El problema de poner a cooperar varios ordenadores en la resolución de una tarea se denomina “paralelización de programas”, y es mucho más complicado de lo que puede parecer a simple vista. Debe de tenerse en cuenta que, en el campo de la paralelización de programas, el rendimiento lo es todo: de nada sirve desarrollar una ingeniosa solución para que varios ordenadores colaboren en la resolución de un problema, si el tiempo requerido por ellos excede del tiempo necesario para resolver el problema utilizando un único ordenador. No se trata, por tanto, de conseguir sólo resultados correctos, sino también de obtener una mejora en relación al tiempo que tarda un único ordenador (que es lo que se denomina “ejecución secuencial” de un programa, por contraposición a su “ejecución paralela”). Aunque intuitivamente podría pensarse que dos ordenadores deberían resolver el problema con más rapidez que uno, hay un motivo que muchas veces impide que esto sea así en la práctica: el coste de las comunicaciones. Por poner un ejemplo, no es lo mismo poner a dos personas a trabajar conjuntamente en un proyecto en la misma habitación, que obligarlos a trabajar en ciudades diferentes, y

sin más comunicación entre sí que cartas manuscritas enviadas a través del correo postal. En el segundo caso, es muy probable que ambas personas pasen gran parte del tiempo escribiendo y leyendo cartas y esperando a recibir la respuesta del otro, en lugar de avanzar en la resolución del problema.

Pese a lo mucho que ha avanzado la informática en los últimos setenta años, el desarrollo de programas paralelos que permitan a varios ordenadores cooperar en la resolución de un único problema continúa siendo una tarea particularmente difícil. Esto se debe a tres motivos, que pasaremos a explicar.

El primer motivo es el siguiente: el desarrollo de programas paralelos requiere comprender muy bien el problema a resolver. Esto puede parecer una obviedad, pero debe de tenerse en cuenta que los problemas que pueden beneficiarse de una solución paralela suelen pertenecer a dominios tan dispares como la mecánica de fluidos, el cálculo de estructuras o las simulaciones en el campo de la economía experimental. En consecuencia, la persona que comprende bien el problema (es decir, el experto en el dominio de aplicación) no suele comprender los fundamentos de la programación paralela, y viceversa.

El segundo motivo por el que la paralelización de algoritmos es una tarea tan difícil incluso para ingenieros informáticos es que la obtención de una versión paralela competitiva en términos de rendimiento requiere conocer muy bien la arquitectura interna de los ordenadores y cómo desarrollar programas que aprovechen todas sus características. De nuevo, podría pensarse que todo informático conoce al detalle el funcionamiento interno de un computador. Sin embargo, esto es cada vez menos cierto: el desarrollo de los modelos de programación modernos presuponen cada vez menos conocimientos de la arquitectura interna de un computador, y se da la paradoja de que muchos programadores expertos no conocen los mecanismos exactos por los cuales los ordenadores ejecutan internamente los programas, ni mucho menos son capaces de estructurar sus programas para mejorar su velocidad de ejecución. Esta falta de conocimientos no les impide desarrollar programas eficaces, del mismo modo que un taxista no necesita conocer el funcionamiento exacto de un motor de combustión interna. Sin embargo, siguiendo con esta comparación, la programación paralela de aplicaciones de alto rendimiento es una disciplina en la que no basta ser conductor profesional, sino que requiere más bien ser piloto de carreras, es decir, no sólo ser un buen programador sino también conocer todos los detalles del ordenador a programar. Por si esto no fuera suficiente, estamos pidiendo a nuestros pilotos de carreras que sean capaces de conducir cualquier vehículo: en nuestro caso, los programadores paralelos deben ser capaces de extraer rendimiento tanto de los sistemas con varios procesadores presentes en un teléfono móvil hasta de los grandes ordenadores paralelos utilizados en Japón para predecir terremotos. Como podemos ver, la diversidad de arquitecturas de ordenadores no facilita precisamente la tarea.

Por último, tenemos el problema de las comunicaciones, mencionado anteriormente. Para que un programa paralelo funcione más rápidamente que si se ejecutara en un solo

ordenador, es necesario que las comunicaciones entre los diferentes procesadores se lleven a cabo de la manera más rápida posible. Esto depende en parte de la propia tecnología de comunicaciones, que es la que a la postre determina la rapidez en el envío y recepción de información. Sin embargo, de nuevo los programadores juegan un papel fundamental: para obtener el máximo rendimiento, debe minimizarse dos factores. En primer lugar, la necesidad de intercambiar información (lo que, volviendo a nuestro símil epistolar, lleva a un menor intercambio de cartas entre las dos personas que cooperan para resolver una tarea), y en segundo lugar a reducir el volumen de datos a intercambiar (lo que lleva a cartas más cortas, y por lo tanto más fáciles de redactar y de leer). Decidir en qué puntos del programa es necesario intercambiar datos entre los procesadores y definir el formato y volumen de los datos a intercambiar es, de nuevo, una tarea sumamente compleja.

Las tres razones expuestas (necesidad de comprensión profunda del problema a resolver, necesidad de conocimiento de la arquitectura informática subyacente, y necesidad de un buen manejo de los mecanismos de comunicaciones) hacen que la programación paralela sea tan difícil hoy día como lo era hace treinta años.

En este contexto es donde puede apreciarse mejor el trabajo investigador del Dr. Padua. La programación paralela es una tarea muy complicada, y por ello el Dr. Padua lleva más de treinta años procurando simplificarla. Con este objetivo en mente, ha trabajado a lo largo de tres líneas principales: la obtención automática de programas paralelos; la búsqueda de mecanismos que permitan encontrar con relativa facilidad los errores en un programa paralelo; y el desarrollo de nuevos lenguajes de programación paralela que permitan reducir la curva de aprendizaje, aumentando así la productividad de los desarrolladores de código paralelo.

En lo que respecta al primero de estos objetivos, la obtención automática de programas paralelos, el objetivo general de la labor investigadora del Dr. Padua pasa por el desarrollo de técnicas que faciliten la programación de esta clase de sistemas. Para simplificar esta labor, lo ideal sería que los propios ordenadores fueran capaces de paralelizar automáticamente un algoritmo. En esta línea, la labor del Dr. Padua ha consistido en desarrollar programas de ordenador que se ocuparan ellos mismos de paralelizar otros programas, con el fin de obtener versiones paralelas que se adaptaran lo mejor posible a las características de los ordenadores donde fueran a ejecutarse. Para poder beneficiarse de esta ayuda, los programadores deben primero desarrollar programas secuenciales perfectamente estructurados y claros, y luego confiar en herramientas automáticas que procesen esos programas, con el fin de adaptarlos a su ejecución en un sistema con varios procesadores. Obtener programas paralelos de manera automática no sólo reduce el tiempo y el esfuerzo necesario por parte de los programadores, sino que además estos programas obtenidos automáticamente pueden adaptarse con rapidez a ordenadores muy distintos, que como hemos dicho anteriormente es una ventaja muy importante.

Como puede imaginarse, encontrar el modo de que los ordenadores paralelicen los programas de manera automática es el Santo Grial de la computación paralela. En ese sentido, la labor pionera del Dr. Padua ha inspirado a generaciones de investigadores en el campo de la computación paralela, hasta el punto de que hoy día, gracias a su trabajo, existen lenguajes de programación y herramientas asociadas que permiten una transformación automática de muchos programas secuenciales para su ejecución en los modernos sistemas multiprocesadores.

Es interesante resaltar que la labor del Dr. Padua en el campo de la transformación automática de código comenzó mucho antes de que se popularizaran los ordenadores con varios procesadores. De hecho, comenzó antes de que existieran los ordenadores personales. En 1980, como parte de su trabajo de tesis, el Dr. Padua, junto con David Kuck y Duncan Lawrie, publicó un artículo titulado: “Sistemas multiprocesadores de alta velocidad y técnicas de generación de código”. Este trabajo contiene algunas ideas realmente novedosas en el campo del diseño de sistemas multiprocesadores, así como en el desarrollo de mecanismos automáticos de generación de código paralelo. Su tesis doctoral describe con precisión cómo son las relaciones entre los diferentes datos de una tarea repetitiva, lo que se denomina técnicamente como “análisis de dependencias en bucles”. Repito: este trabajo se presentó en 1980. Incluso a los informáticos nos cuesta poner en perspectiva la importancia de esta contribución: hoy en día, todos llevamos en nuestros bolsillos un teléfono móvil que tiene varios procesadores. Sin embargo, cuando este trabajo fue publicado en la revista IEEE Transactions on Computers, aún faltaban dos años para que IBM sacara al mercado el primer ordenador personal, el IBM XT. Mucho tiempo ha pasado desde entonces: el smartphone más modesto del mercado lleva cuatro u ocho procesadores casi mil veces más rápidos que el 8086 del IBM XT, tiene mil veces más memoria RAM y dispone de una capacidad de almacenamiento mil veces mayor. Y no digo “mil veces” como si quisiera decir “muchas veces”. Lo digo en sentido literal. Y todo por menos de cien euros. Cuando todavía faltaban dos años para el inicio de la revolución de la informática personal, el Dr. Padua y sus colegas estaban proponiendo soluciones a problemas que eran difícilmente imaginables entonces. Si ya nos cuesta a los investigadores en computación paralela explicar hoy en día a qué nos dedicamos, no quiero ni imaginar lo que le costaría al Dr. Padua explicar su trabajo a sus amigos en aquellos tiempos.

Volviendo al tema de la paralelización automática de código, existen dos momentos clave en el que dicha transformación puede producirse: durante la compilación del programa y durante su ejecución. Para comprender esto es necesario decir unas palabras sobre el desarrollo de programas. Contrariamente a lo que suele creerse, los ordenadores no son capaces de comprender sin más los programas escritos por los informáticos. Los programadores escriben los programas en los llamados “lenguajes de alto nivel”, que permiten describir tareas complejas (por ejemplo, calcular el coseno de un ángulo o aplicar una transformación a una matriz) con pocas líneas de código. Sin embargo, los ordenadores no son capaces de entender tareas tan complejas: en realidad, los ordenadores sólo son capaces de realizar operaciones aritméticas básicas, mover datos y

compararlos entre sí para tomar decisiones. El proceso de traducción del código de alto nivel desarrollado por los programadores al código de bajo nivel de los ordenadores se denomina “compilación”. Ese proceso lo realizan los compiladores, programas especializados en dicha traducción. Pues bien, una de las primeras contribuciones del Dr. Padua consistió en la propuesta de técnicas que permitirían a los compiladores no sólo obtener versiones de bajo nivel de programas secuenciales, sino además generar versiones paralelas de los mismos, para su ejecución en sistemas multiprocesadores.

Decíamos anteriormente que existen dos momentos clave para transformar los códigos secuenciales en códigos paralelos. El primero, como ya se ha indicado, es el momento de la compilación. El segundo momento es el de la propia ejecución. La idea es sencilla de explicar pero muy difícil de llevar a la práctica, y es la siguiente. Cuando el programa ya ha sido compilado y se está ejecutando, el ordenador debería de ser capaz de detectar de manera automática qué partes del programa podrían ejecutarse en paralelo, y en ese mismo instante tomar la decisión de asignar ese trabajo a otros procesadores que se encuentren libres. En 1995, el Dr. Padua publicó, junto a su discípulo, el Dr. Rauchwerger, uno de los primeros mecanismos que permitían a un ordenador ejecutar la tarea en paralelo y luego decidir si la ejecución había sido correcta. De nuevo, se trata de un trabajo adelantado a su tiempo que en este caso posibilitó el nacimiento de un nuevo campo de investigación: la llamada paralelización especulativa, en la que mi grupo de investigación lleva trabajando desde hace quince años.

Como se ha indicado más arriba, no basta con desarrollar código paralelo: también es necesario que el código desarrollado sea eficiente. El Dr. Padua ha realizado contribuciones muy notables en este campo, denominado Optimización Automática de Código. Su labor en este campo ha permitido, por ejemplo, que los programas desarrollados para cálculo intensivo dispongan de varias versiones del mismo código, eligiéndose de manera automática en tiempo de ejecución la que permita obtener un mayor rendimiento. Su contribución mejoró tanto la implementación de algoritmos existentes (como es el caso de la Transformada Rápida de Fourier y métodos de ordenación) como además permitió el almacenamiento compacto de las diferentes versiones de código disponibles, permitiendo elegir la más conveniente en cada caso.

La labor investigadora del Dr. Padua no sólo se ha centrado en la obtención automática de versiones paralelas de programas secuenciales y en su optimización: también ha realizado importantes contribuciones en el campo de la depuración de programas paralelos, es decir, de la búsqueda de errores. Es fácil imaginar que una tarea tan compleja como la programación paralela es también una tarea propensa a errores humanos. También en este campo el Dr. Padua y algunos de sus discípulos han realizado importantes contribuciones, estudiando en detalle cómo pueden intercalarse los eventos en diferentes procesadores, y como gestionar la incertidumbre derivada del hecho de que no es posible conocer en todo momento en qué orden se realizan los cálculos. Su trabajo en este campo ha supuesto una gran influencia en el desarrollo de

herramientas de apoyo a la depuración de programas, como las que ofrece la compañía Intel dentro de su conjunto de herramientas de programación paralela.

Para finalizar este recorrido por la prolífica labor investigadora del Dr. Padua, me gustaría hacer referencia a su trabajo en el desarrollo de lenguajes de programación paralela. Como hemos tenido ocasión de indicar anteriormente, la obtención de una versión paralela a partir de un algoritmo secuencial es una tarea muy difícil, tanto si la desarrolla un programador como si se deja en manos del propio ordenador para que la realice de manera automática.

En lugar de desarrollar primero un programa secuencial para luego paralelizarlo, una solución alternativa pasa por desarrollar directamente la versión paralela, a través del uso de lenguajes de programación paralelos. Con su ayuda, es el propio programador el que deja indicado qué partes del trabajo pueden distribuirse entre los diferentes procesadores y cuáles de ellas es mejor que se realicen de manera secuencial. En este campo, el Dr. Padua desarrolló uno de los primeros lenguajes de programación paralelo, una versión de Fortran que servía para programar Cedar, un ordenador desarrollado en la Universidad de Illinois en Urbana-Champaign a finales de los años 80. Este lenguaje incluía muchas de las características que más tarde fueron adoptadas por otras soluciones, como OpenMP, posiblemente el mecanismo de programación paralela más popular para los sistemas de memoria compartida.

Más recientemente, el trabajo del Dr. Padua y de sus colaboradores se ha centrado en el último problema de la programación paralela que nos quedaba por abordar: la distribución de los datos al objeto de minimizar los costes asociados a las comunicaciones. Su solución, basada en distribuir matrices de datos que han sido ordenadas jerárquicamente, permite reducir el número de comunicaciones necesarias entre los procesadores, a la vez que mantiene a mano los datos más utilizados, con la consiguiente mejora en la velocidad de ejecución. En este sentido, cabe destacar que nuestra propia labor investigadora en modelos de programación paralela, codirigida por el Dr. González Escribano, debe mucho a las soluciones de distribución de datos propuesta en el trabajo mencionado.

En los últimos años, el Dr. Padua ha trabajado en el desarrollo de mecanismos autónomos para la compilación de aplicaciones, incluyendo el uso de sistemas de aprendizaje automático. También ha desarrollado nuevas notaciones de alto nivel para programación paralela, evaluación de compiladores y la implementación eficiente de lenguajes de script, creando técnicas que mejoran el rendimiento de aplicaciones tan utilizadas como MATLAB y R.

* * * * *

Una vez que hemos finalizado este repaso a la prolífica labor investigadora del Dr. Padua, me gustaría decir unas palabras respecto al impacto que el trabajo del Dr. Padua ha tenido en su campo. En ese sentido, bastaría con mencionar que el Dr. Padua tiene un H-index de 58 según Google Scholar, con más de 11 600 citas, lo que lo convierte en

uno de los investigadores cuyos trabajos han tenido mayor impacto en la disciplina de Computación Paralela a escala mundial. El Dr. Padua ha publicado más de 170 artículos en los campos de la programación paralela, compiladores y herramientas, y el diseño de sistemas paralelos. Asimismo, ha dirigido un total de 30 tesis doctorales. Algunos de sus alumnos se cuentan entre los investigadores de más prestigio en el campo de la programación paralela.

Entre los servicios prestados por el Dr. Padua a la comunidad científica, cabe destacar que ha sido el Editor en Jefe de la Encyclopedia of Parallel Computing, una monumental obra publicada por Springer-Verlag que resume el estado del arte en una disciplina que hoy goza de la máxima importancia. También es miembro del Comité Editorial de algunas de las revistas más prestigiosas del área, como la IEEE Transactions on Parallel and Distributed Systems, la Journal of Parallel and Distributed Computing, y también la International Journal of Parallel Programming. El Dr. Padua ha servido asimismo en los Comités Científicos de más de 70 congresos internacionales.

Entre sus múltiples galardones, el Dr. Padua recibió en 1997 el “Golden Core Member”, otorgado por la IEEE Computer Society, la sociedad científica más importante en el campo de la ingeniería. Este galardón reconoce los servicios del Dr. Padua a la sociedad. También ha recibido las distinciones de “Fellow member” del IEEE (Institute of Electrical and Electronics Engineers) y de la ACM (Association for Computing Machinery), en 2000 y 2008, respectivamente. Finalmente, en 2015 el Dr. Padua recibió el “IEEE Computer Society Harry H. Goode Memorial Award”, un premio que conceden los investigadores que trabajan en el ámbito de las Tecnologías de la Información y que reconoce, entre otras cosas su importante y definitiva contribución a la teoría y práctica de los lenguajes paralelos, compiladores y herramientas que han ayudado a perfilar la tecnología que está en la base de la sociedad digital, tal y como la conocemos ahora. También ha recibido numerosos premios como docente, concedidos por votación popular entre los estudiantes del Departamento de Computer Science en la Universidad de Illinois at Urbana-Champaign en la que ha sido profesor durante las últimas décadas.

* * * * *

Hasta aquí, la descripción de los méritos científicos del Dr. Padua, que como hemos indicado lo convierten en una autoridad indiscutible en su campo del saber. Pero además, se da la circunstancia de que el Dr. Padua atesora una calidad humana difícil de encontrar hoy día, que es la que justifica el impacto que este respetado profesor tiene entre todos aquellos que han tenido el privilegio de trabajar con él. Siendo muy querido y respetado en la comunidad científica, lo es aún más entre quienes han tenido el privilegio de ser sus alumnos y colaboradores. Su actitud respetuosa, así como su disposición a escuchar y dar consejos lo distingue incluso de entre aquellos que tienen madera de líder. Esta cualidad lo eleva al punto de ser hoy día el sabio al que todos acuden a buscar consejo. Su gran generosidad y el hecho de que nunca diga "no" a prestar ayuda se suma a su larga lista de cualidades. A tenor de muchos de sus

discípulos y colaboradores, “ayudar” debe de ser uno de los verbos favoritos de Dr. Padua. Y no sólo ayuda a sus amigos y colegas, sino muy especialmente al que más lo necesita, por ejemplo a investigadores jóvenes que apenas están empezando su carrera investigadora, prestando apoyo para conseguir financiación, presentándolos a colegas a los que no conocen personalmente, o contando con ellos para participar en actividades que permita aumentar su visibilidad internacional, como participar en comités de programas de congresos o revistas, comités de evaluación, etcétera. Por poner otro ejemplo, el Dr. Padua siempre está dispuesto a acoger a investigadores de otras universidades para realizar estancias en la Universidad de Illinois. Pero las invitaciones del Dr. Padua no se quedan en un "ven a este despacho y siéntate aquí", sino que se esfuerza en que el visitante forme parte de su grupo y que desarrolle conexiones con aquéllos de sus colaboradores que trabajen en temas afines. Eso permite al visitante formar poco a poco una red de contactos formada por investigadores excepcionales.

De su vocación de ayudar al prójimo no sólo se benefician sus colegas en la disciplina, sino también sus alumnos. Por poner un ejemplo, este año el Dr. Padua se ha presentado como voluntario en su Departamento para impartir un curso de programación básica para alumnos universitarios de letras y ciencias sociales, es decir, alumnos que no han cursado matemáticas ni asignaturas tecnológicas en el bachillerato. Su Departamento a algún profesor con experiencia, dispuesto a aceptar el desafío de enseñar conceptos básicos de matemáticas y lógica que están en la base de los fundamentos de la programación. A pesar de no tener ya obligaciones docentes, él se ofreció voluntariamente a preparar el curso e impartirlo, lo que incluye el desarrollo de cientos de transparencias nuevas con contenidos básicos. El Dr. Padua, con su inmenso bagaje científico, es consciente del valor educativo y del desafío intelectual que un curso de estas características puede tener para alumnos de ese perfil. Ayudar es, sin duda, su actividad favorita.

Como resultado de su vocación de ayuda, todos los que conocen al Dr. Padua sólo tienen palabras de agradecimiento hacia él. No conocemos a nadie que lo conozca y que no lo admire profundamente. El Dr. Padua es un pionero de la programación paralela ha sabido conjugar algo muy difícil en este competitivo mundo: estar en la vanguardia en investigación y docencia; ser un ejemplo de gran mentor y amigo; y ser respetado por todos los miembros de la comunidad a la que pertenece.

* * * * *

De acuerdo con el Reglamento de Doctorado Honoris Causa de la Universidad de Valladolid, esta distinción, que es la máxima distinción académica que otorga la Universidad de Valladolid, podrá otorgarse a aquellas personas que destaquen, de manera singular, en el campo de la investigación o de la docencia, en el cultivo de las Artes y de las Letras o en aquellas actividades que tuviesen una repercusión notoria e importante, desde el punto de vista universitario, en el terreno científico, artístico, cultural, tecnológico o social. El reglamento establece que, para su concesión, debe existir una vinculación comprobable con la Universidad de Valladolid o su entorno

social. El Reglamento también indica que, en ausencia de ésta, y siempre con carácter excepcional, podrán proponerse para la distinción candidatos que por su prestigio nacional o internacional justifiquen asociar su nombre al de la Universidad de Valladolid.

Creemos que la descripción del trabajo del Dr. Padua realizada anteriormente es suficiente para acreditar el enorme prestigio internacional del que goza el candidato. Sin embargo, existe una conexión adicional entre la labor investigadora del Dr. Padua y la Universidad de Valladolid. El trabajo del Dr. Padua sentó las bases para toda la labor investigadora de mi grupo de investigación, que codirijo con el Dr. Arturo González Escribano, del Departamento de Informática de la Universidad de Valladolid. Nuestro trabajo parte de los análisis de dependencias en la paralelización de bucles establecidos por el Dr. Padua y sus discípulos, trabajos que sentaron, entre otras, las bases de la paralelización especulativa de aplicaciones, nuestro principal tema de investigación. La labor investigadora de nuestro grupo ha dado lugar hasta la fecha a más de 50 trabajos en revistas y congresos de impacto, cinco tesis doctorales defendidas y dos más en proceso de realización. Toda esta labor se asienta sobre las premisas establecidas por los trabajos pioneros del Dr. Padua.

* * * * *

En resumidas cuentas, creemos que la concesión del Doctorado Honoris Causa al Dr. David Padua complementa la vinculación científica que ya existe entre sus contribuciones y las que se desarrollan en la Escuela de Ingeniería Informática de la Universidad de Valladolid, asociando además de forma permanente el nombre de la Universidad de Valladolid y el de su Escuela de Ingeniería Informática a la figura de uno de los investigadores más importantes, conocidos y admirados a nivel mundial en el campo de la Programación Paralela.

Al aceptar David Padua formar parte del claustro de doctores de la Universidad de Valladolid, nuestra institución se ve enriquecida con uno de los científicos más sobresalientes del campo de la computación paralela, y cuya obra docente, investigadora y también humana es un modelo a seguir no sólo en el campo de los estudios de Ingeniería Informática, sino para todas las disciplinas que se imparten en nuestra Universidad.

Por todo lo expuesto proclamo y manifiesto:

**His de Causis, Peto Gradum Doctoris Honoris Causa
Domino David Padua.**

SOLICITO SE PROCEDA A INVESTIR AL EXCMO. SR. DR. D. DAVID PADUA
CON EL GRADO DE DOCTOR "HONORIS CAUSA" POR ESTA UNIVERSIDAD.