Ciencias de la Computación
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS UNIVERSIDAD DE CHILE
11

REVISTA DEL DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN DE LA UNIVERSIDAD DE CHILE

Bits DE CIENCIA

SEGUNDO SEMESTRE 2014

SOCIEDAD CHILENA DE CIENCIA DE LA COMPUTACIÓN: ORÍGENES, FUNDACIÓN (1984) Y PRIMEROS AÑOS | Juan Álvarez

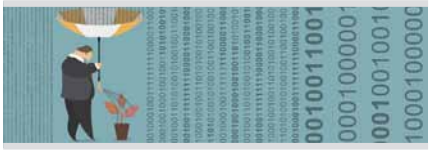
BIG DATA: UNA PEQUEÑA INTRODUCCIÓN | Aidan Hogan

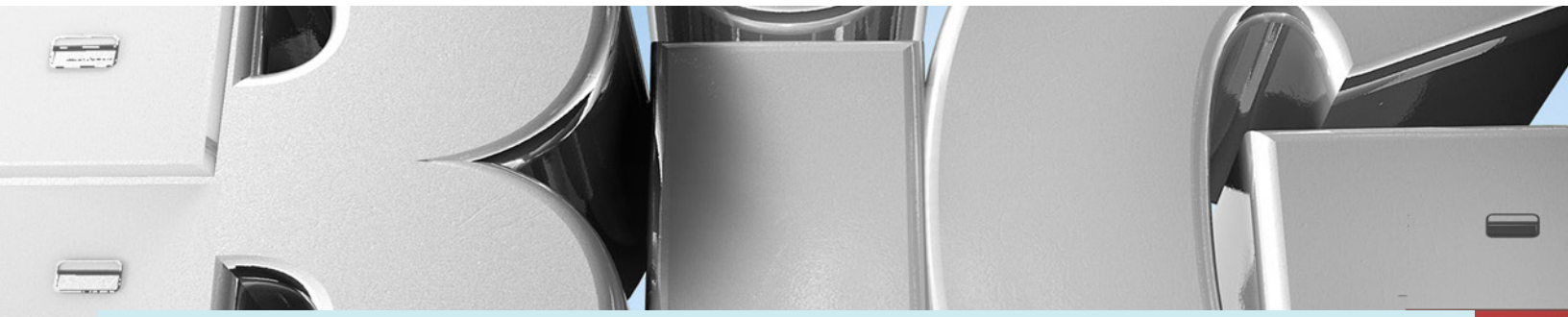
Bits DE CIENCIA

Revista Bits de Ciencia N°11
ISSN 0718-8005 (versión impresa)
www.dcc.uchile.cl/revista
ISSN 0717-8013 (versión en línea)

Departamento de Ciencias de la Computación
Avda. Beauchef 851, 3º piso,
Santiago, Chile
837-0459 Santiago
www.dcc.uchile.cl
Fono 56-2-29780652 | Fax 56-2-26895531
revista@dcc.uchile.cl

CONTENIDOS





Buzzword es el anglicismo que se refiere a aquellas palabras que utilizamos para impresionar o que están de moda, pero cuyo significado a veces ni siquiera podemos precisar. Big data, diluvio de datos, etc. parecen buenos ejemplos de esto: todo el mundo las ha escuchado, muchos las ocupan dudosamente para justificar proyectos de millones de dólares, conseguir posiciones académicas o simplemente aumentar las posibilidades de que un paper no tan sólido sea aceptado en la conferencia a la que aspiran, pero pocos, quizá muy pocos, son capaces de entender los componentes de este diluvio de datos y precisar las características que hacen diferente a este problema de todos los que hemos enfrentado anteriormente.

Porque lo cierto es que el diluvio de datos –es decir, la producción masiva de datos que supera la capacidad de las instituciones de manejarlos y de los investigadores de entenderlos– es un problema del todo real, y ser capaz de extraer información desde grandes volúmenes de información es quizá el más relevante desafío que enfrenta la Computación hoy en día. Este problema del Big Data –que se caracteriza por las cuatro V's de Volúmen, Velocidad, Variedad y Veracidad– nos sitúa a nosotros, la gente de Informática, en el ojo de este huracán que requiere de nuevos algoritmos, estructuras de datos, formas de almacenamiento, lenguajes, heurísticas y otros para su solución.

En este número de la Revista Bits hemos decidido darle una mirada a fondo al tema del Big Data y convocar a nuestros expertos nacionales a que nos desmenucen la problemática relacionada, desde sus fundamentos a sus aplicaciones. En particular:

1. Aidan Hogan hace un amplio resumen del problema del Big Data, de sus desafíos y de muchas de las técnicas que se utilizan para abordarlos.

2. Renzo Angles nos cuenta de temas de benchmarking para Big Data y la Web Semántica.

3. El grupo de Big Data de Inria Chile nos presenta su trabajo en visualización de grandes volúmenes de información y sus aplicaciones a la Astronomía.

4. Juan Reutter nos hace un resumen de los modelos de procesamiento de información para Big Data.

5. Varios expertos nacionales en transportes, business inteligence, astronomía y medicina nos cuentan de los problemas relacionados con el diluvio de datos en sus respectivas áreas.

Además, en la Sección de Investigación Destacada presentamos un artículo de Valeria Fionda, Claudio Gutiérrez y Giuseppe Pirrò sobre navegación semántica de datos en la Web. Por otro lado, en la sección de Computación y Sociedad presentamos el artículo de Juan Álvarez sobre los orígenes de la Sociedad Chilena de Ciencias de la Computación (SCCC) y un artículo del Profesor Aldo Migliaro sobre la historia del Departamento de Ingeniería Informática de la PUC de Valparaíso.

Finalmente, el alumno de Doctorado Guillermo Cabrera nos entrega un interesante survey sobre el manejo de datos en astronomía, y la profesora Bárbara Poblete nos presenta el grupo de investigación Prisma.

Espero que disfruten la Revista que, como es costumbre, hemos preparado con especial dedicación para nuestros lectores. Si tienen algún comentario, sugerencia o reclamo por favor envíenlo al correo **revista@dcc.uchile.cl**

PABLOBARCELÓ
Editor General
Revista Bits de Ciencia

NAVEGANDO EL CONOCIMIENTO EN LA WEB

En los últimos años la Web ha evolucionado desde un repositorio inicialmente sólo de documentos a una Web donde además hay datos, y donde los objetos descritos por datos estructurados se hallan interconectados. Existen diversos esfuerzos por transformar las partes relevantes de la Web en una base de datos enorme. Esto significa esencialmente publicar, describir, organizar e interconectar los datos existentes en la Web.

VALERIAFIONDA

Investigadora Fellow en el Departamento de Matemáticas y Computación de la Universidad de Calabria, Italia. Hizo su MSc en Computing Engineering en 2006 y su Ph.D. en Matemáticas y Computación en 2010 en la Universidad de Calabria. Sus intereses de investigación incluyen Bioinformática, E-commerce, Algoritmos para Manipulación de Grafos, Minería de Procesos y Web Semántica. Ha publicado artículos en conferencias y journals de primer nivel como WWW, IJCAI, CIKM, AIJ.

fionda@mat.unical.it

CLAUDIOGUTIÉRREZ

Profesor Asociado Departamento de Ciencias de la Computación, Universidad de Chile. Ph.D. Computer Science, Wesleyan University; Magíster en Lógica Matemática, Pontificia Universidad Católica de Chile; Licenciatura en Matemáticas, Universidad de Chile. Áreas de investigación: Fundamentos de la Computación, Lógica aplicada a la Computación, Bases de Datos, Semántica de la Web, Máquinas Sociales.

cgutierr@dcc.uchile.cl

GIUSEPPEPIRRÒ

Investigador postdoctoral senior en el Instituto WeST, Universidad de Koblenz, Alemania. Antes fue Profesor Asistente en la Universidad Libre de Bolzano, Italia y postdoctorado en Inria Grenoble, Francia. Obtuvo su Ph.D. en Computer Engineering en el Departamento DEIS, Universidad de Calabria, Italia, en 2009. Sus intereses de investigación se enfocan en la Web Semántica, Lenguajes de Consulta para Grafos, Inteligencia Artificial y Sistemas Distribuidos. Ha escrito más de 40 papers publicados en conferencias y journals de primer nivel como WWW, ISWC, AAAI, CIKM, CCGRID, TWEB y FGCS.

pirro@uni-koblenz.de

Estas nuevas tendencias pavimentan el camino para servicios que –aprovechando las técnicas de bases de datos sobre datos estructurados e interconectados a escala planetaria– puedan darle más valor a la Web descubriendo nuevo conocimiento. Nuevos desafíos de investigación emergen en este nuevo mundo debido a nuevas realidades: la creación y manipulación de los datos intrínsecamente descentralizada; la carencia de esquemas superimpuestos; el desconocimiento de la topología de los recursos y sus conexiones; la existencia de enormes volúmenes de información cubriendo áreas y dominios diferentes. Este grafo de datos estructurados –ilimitado y distribuido– es conocido hoy como **la Web de los Datos**.

Entre las iniciativas de esta nueva Web, podemos mencionar la extracción de información desde datos Web textuales o semiestructurados [23] y comunidades que comparten conocimiento (e.g., Wikipedia); proyectos como las *Web tables* de Google que aspiran a recobrar pequeñas bases de datos relacionales de las tablas HTML [10]; las iniciativas YAGO [22] y DBpedia [7] para construir bases de conocimiento que recolecten datos sobre entidades y sus relaciones desde fuentes Web; y desde el mundo de los datos abiertos y la Web Semántica, la iniciativa que intenta la organización e interconexión sistemática de datos semánticos estructurados en la Web, llamada *Linked Open Data* [9].

Una herramienta esencial para descubrir fuentes de datos, información y conocimiento en este grafo gigante es la navegación, el procedimiento de ir, guiado por un mapa, desde lo conocido a lo desconocido en algún espacio. Por ejemplo, considere cómo la navegación desde el nodo *Stanley Kubrick* en DBpedia hasta el nodo *A Clowckwork Orange* nos permite descubrir conocimiento adicional tal como los actores que actuaron en la película.

Este artículo discute cómo realizar navegación de conocimiento en la Web de los datos y los desafíos que acarrea para el manejo de los datos. A modo de ilustración, nos concentraremos en el escenario de *Linked Open Data* (LOD).

NAVEGAR VERSUS CONSULTAR

La herramienta tradicional para consultar datos estructurados es un lenguaje de consultas. Consultar implica acceder a la fuente de datos para satisfacer una necesidad expresada en algún lenguaje formal. La topología de la fuente de los datos está fija a priori. Por otro lado, la navegación consiste en la especificación de partes de las fuentes de datos que tienen que ser "exploradas" o descubiertas para encontrar datos relevantes. Por lo tanto, cuando la topología del espacio de fuentes de datos a ser consultados es (desde un punto de vista práctico) ilimitada, desconocida en su identidad, y dinámica, como en el caso de la Web de Datos, la navegación se vuelve necesaria. En el mundo del manejo de datos, la noción de navegación se volvió popular con el lenguaje de consultas XQuery, cuya filosofía subyacente es permitir la especifica-

→

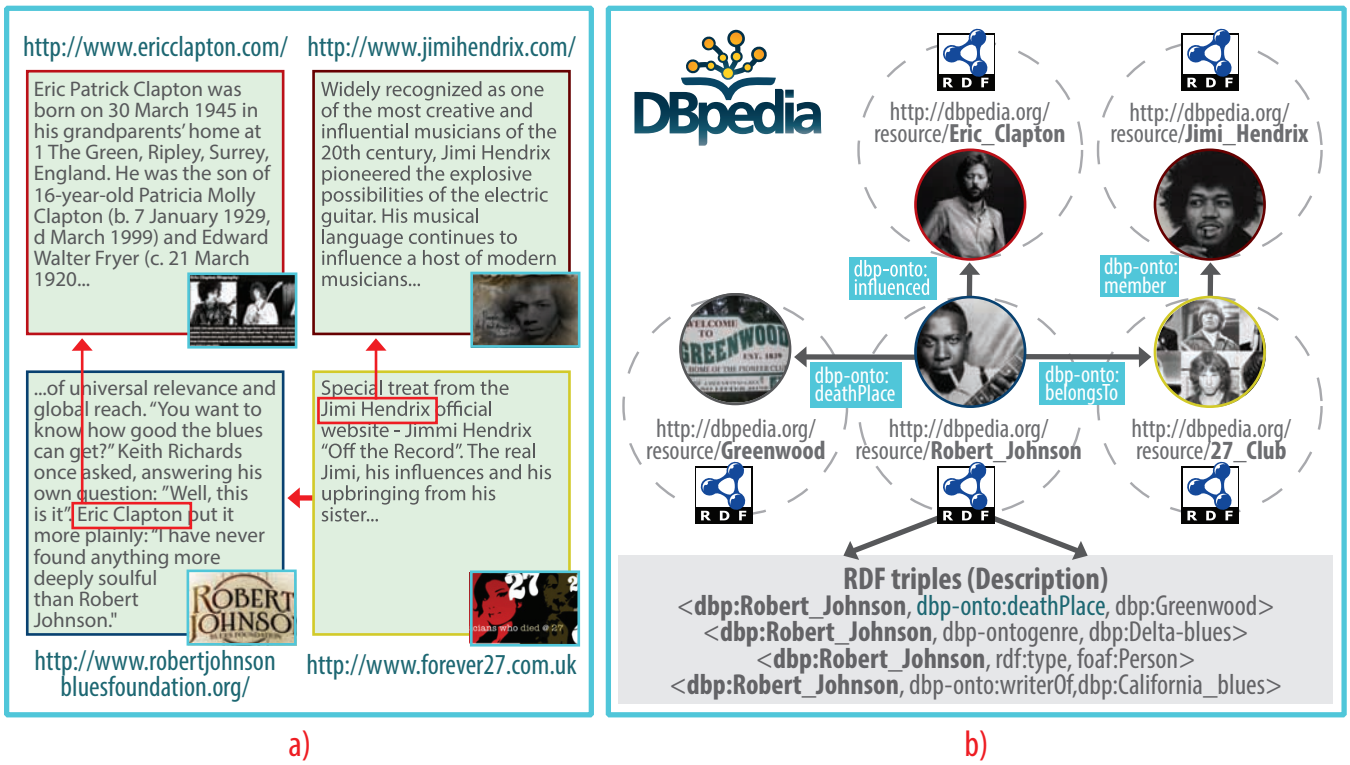


FIGURA 1.
EXTRACTO DE WIKIPEDIA Y SU CONTRAPARTE DBPEDIA.

ción de una ruta en un grafo (árbol) de datos, en oposición a los lenguajes de consulta clásicos (e.g., SQL) que representaban afirmaciones lógico/algebraicas. Por lo tanto, la navegación es una función complementaria de consulta, especialmente cuando se trata con grafos. En particular, en la Web *navigare necesse est, vivere non est necesse.*

Trabajar con navegación en la Web crea varios desafíos. Primero, el grafo de la Web está formado por un gigantesco número de fuentes de datos interconectadas que no están disponibles *off-the-shelf.* Por otro lado, la disponibilidad de datos estructurados en cada fuente sugiere que la navegación puede y debe ser productivamente combinada con consultas. De hecho, satisfacer una necesidad de información en este escenario incluye dos facetas que debieran combinarse: el descubrimiento de fuentes relevantes de datos vía navegación; la obtención de respuestas precisas desde esas fuentes a través de consultas.

LA WEB DE LOS DATOS

La Web tradicional estaba basada en tres pilares: identificadores únicos para nombrar recursos (URIs), el protocolo HTTP para intercambiar información, y el lenguaje HTML para describir páginas. La Web de datos extiende este modelo de dos maneras. Primero, se extiende el alcance de las URIs para nombrar nuevos tipos de recursos tales como objetos del mundo real (personas, lugares, equipos de fútbol) y conceptos abstractos (deporte, filosofía, geografía) [14]. Descripciones de tales recursos pueden ser obtenidas de la misma forma que para los documentos tradicionales, esto es, dereferenciando sus URIs asociados a través del protocolo HTTP. Segundo, se introducen *links* semánticos, en vez de los *links* a secas de la Web tradicional, de manera de poder describir y relacionar entidades significativamente.

La **Figura 1** muestra un extracto de Wikipedia y su contraparte DBpedia. Mientras el primero está basado en documentos y sus *hiperlinks*, el segundo está basado en recursos y descripciones semánticas. En la **Figura 1**, cada círculo tachado representa una fuente de datos, identificada por una URI, que contiene la descripción del recurso en el lenguaje formal RDF. Por ejemplo, en la fuente de datos asociada con el cantante Robert Johnson, hay un triple RDF que nos dice que Johnson falleció en la ciudad de Greenwood. Observe el *link* semántico entre los recursos correspondientes expresado a través de la propiedad **dbp-onto:deathPlace** (la que es definida en la ontología de DBPedia).

El conocimiento generado distribuidamente puede de esta forma ser interconectado, de tal forma de crear una Web de Datos donde el valor de las piezas individuales de datos aumenta en la medida que más preguntas pueden ser respondidas y más gente pueda acceder a ellos. Los que publican los datos mantienen miles de fuentes de datos que cubren diversos dominios, tales como conocimiento general (e.g., DBpedia, Freebase/Google, YAGO), información geoespacial (e.g., Geonames), información del sector público (e.g., los Gobiernos de EE.UU., UK, Chile, etc.) y así sucesivamente. Iniciativas como la de microformatos y RDFa están haciendo difusa la distinción entre Web de Documentos y Web de Datos al permitir insertar "meta-información" (triples RDF) a nivel de páginas Web.

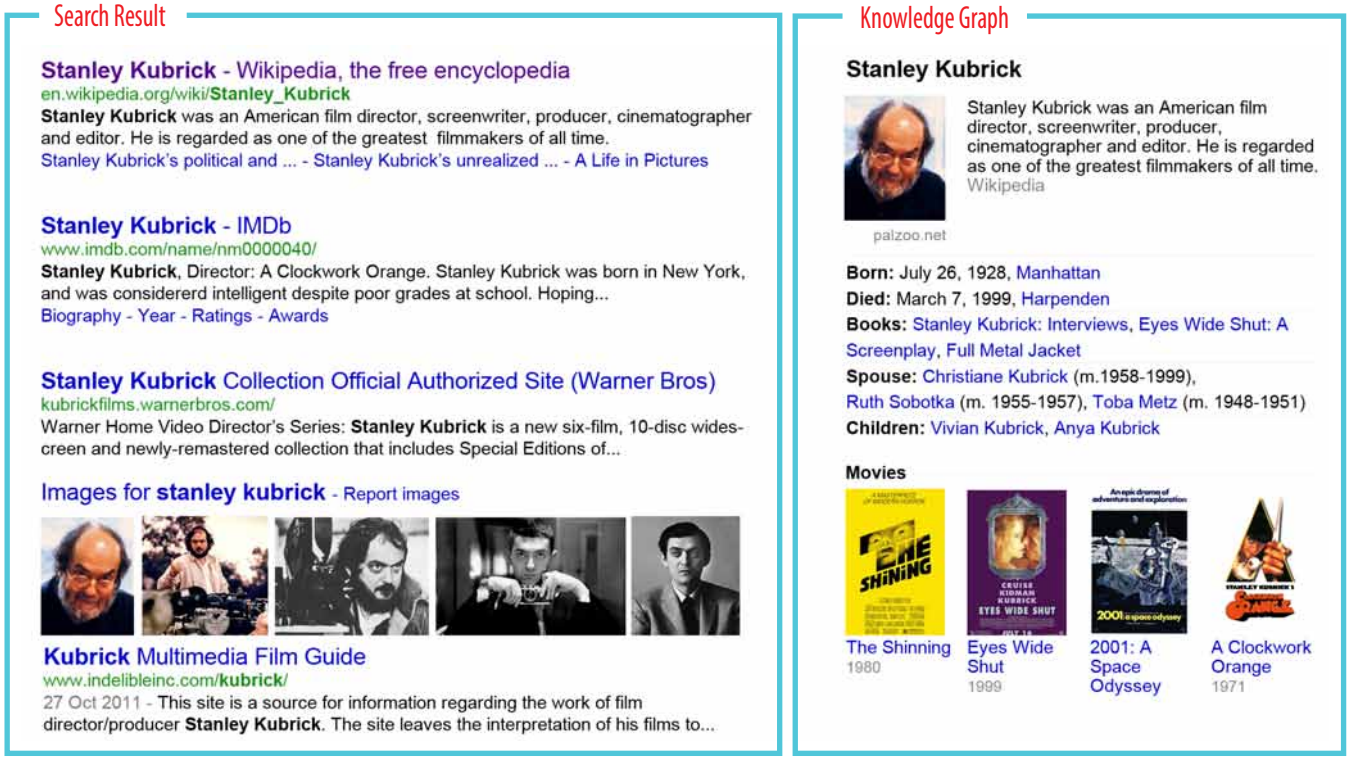


FIGURA 2.
RESULTADOS DE BÚSQUEDA DE STANLEY KUBRICK EN GOOGLE.

CONECTANDO LOS PUNTOS

Aplicaciones tales como *Facebook Graph* (FG) [2] y *Google's Knowledge Graph* (KG) [3] reconocen la importancia de enriquecer el paradigma de búsqueda tradicional al tratar a los grafos semánticos como ciudadanos de primera clase. Los nodos de FG almacenan información sobre entidades junto con relaciones con otras entidades. Para acceder FG, Facebook provee de una interfaz de búsqueda basada en el lenguaje natural, donde los usuarios pueden preguntar, por ejemplo, "gente que vive en mi ciudad". KG de Google, con el lema *"Things not Strings"*, resalta la necesidad de considerar una palabra ya no meramente como un string, sino como identificadores para entidades del mundo real (e.g., personas) en el grafo. Para dar un ejemplo, la **Figura 2** muestra los resultados de una búsqueda de Stanley Kubrick en Google. La parte izquierda reporta el clásico conjunto plano de páginas Web que contienen esa *keyword*. El KG que se muestra en la parte derecha lista las películas de Kubrick, la ciudad donde nació y así sucesivamente, con *links* a las entidades correspondientes en el KG. Al seguir cada uno de esos *links* es posible "descubrir" nuevo conocimiento. FG y KF ofrecen un atisbo de las potencialidades de la navegación sobre una Web de entidades interconectadas. En particular, estos sistemas intentan tratar una de las limitaciones de la búsqueda tradicional: el descubrimiento del *unknown unknown*, esto es, algo que uno no sabe que no sabe.

EL MÉTODO LOD

Sistemas como FG y KG utilizan modelos propietarios, grafos construidos de forma ad-hoc, y actualmente no pueden procesar peticiones complejas de navegación incluyendo la cobertura de múltiples fuentes de datos distribuidas. LOD adopta una filosofía diferente. Links semánticos son introducidos mediante un lenguaje estándar de descripción semántica llamado Resource Description Framwork (RDF). Este es un lenguaje simple y extensional basado en triples de la forma sujeto-predicado-objeto. Por ejemplo, el triple RDF (Kubrick, birthPlace, Manhattan) expresa que Stanley Kubrick nació en Manhattan. Una colección de triples RDF puede ser pensado como un grafo etiquetado, el cual debido a su uso de taxonomías estándar, vocabularios y ontologías expresadas en lenguajes formales tales como RDF(S), SKOS [8] u OWL [18], permite describir, integrar e intercambiar conocimientos de áreas de dominio específico. A las entidades se les asignan identificadores (en la forma de URIs) y se generan descripciones distribuidas y autónomas que pueden ser parte de cualquier fuente de datos. Por lo tanto, usando las palabras de Tim Berners-Lee, cualquiera puede decir cualquier cosa sobre cualquier cosa y publicar donde quiera. **La Figura 3** muestra un extracto de datos incluyendo información sobre Stanley Kubrick, David Lynch, Quentin Tarantino y las películas Path of Glory y A.I. Cada rectángulo gris tachado representa una fuente de datos que contiene un conjunto de triples RDF (representados con el mismo color) que pueden ser obtenidos al dereferenciar la URI correspondiente (representada como un círculo tachado). Por ejemplo, al dereferenciar Stanley Kubrick en DBpedia uno obtiene, entre otras cosas, el triple `<dbpo: Stanley Kubrick, dbpo: birthPlace, dbpo: Manhattan>`. Interesantemente, la información ha sido obtenida desde tres fuentes de datos diferentes.

→

dbpo: <http://dbpedia.org/ontology/>
dbp: <http://dbpedia.org/>
lmdb: <http://linkedmdb.org/>
owl: <http://www.w3.org/2002/07/owl/>
fb: <http://rdf.freebase.com/ns/>
rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
foaf: <http://xmlns.com/foaf/spec/>

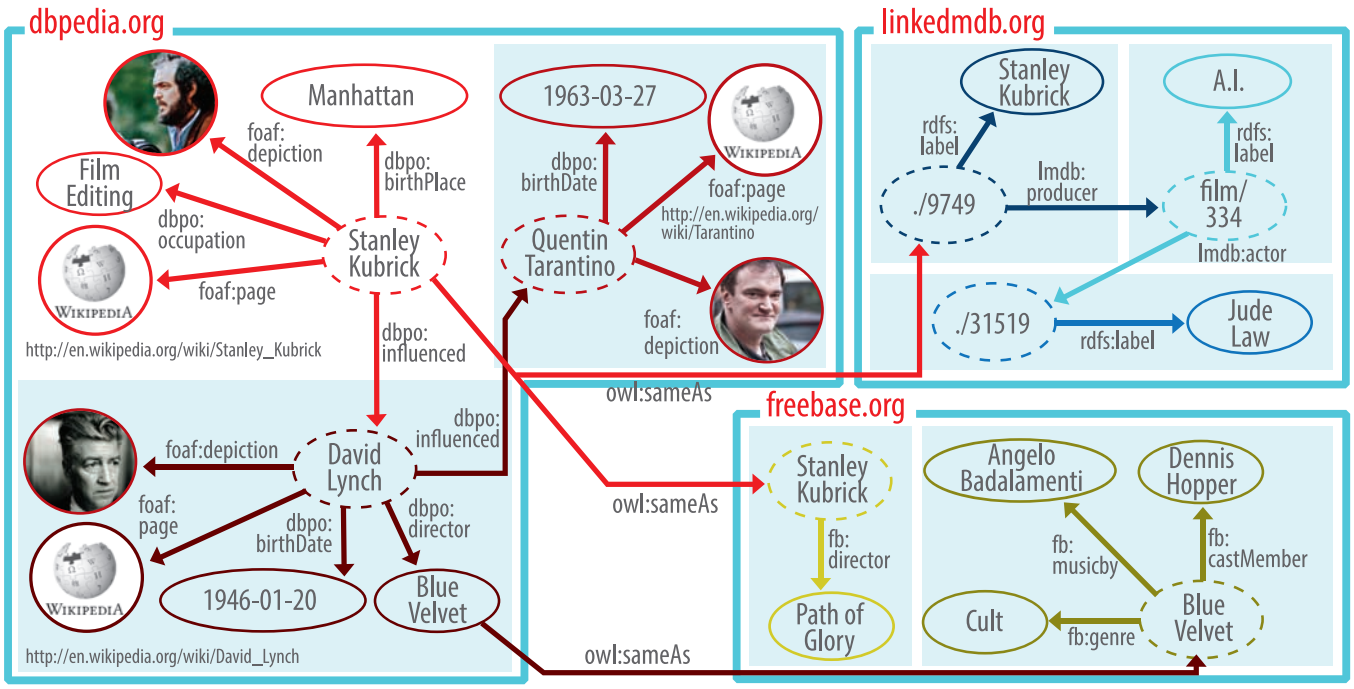


FIGURA 3.
EXTRACTOS DE DATOS DE DBPEDIA.

La interconexión entre los *datasets* permite juntar datos de múltiples fuentes. En la Web de Datos, el predicado `owl:sameAs` es usado para unir dos URIs que identifican la misma "cosa". Por ejemplo, el recurso Stanley Kubrick en DBpedia, que provee conocimiento general, está unido con aquel en LinkedMDB, el cual es una fuente especializada de conocimiento sobre películas, directores y actores. Esto permite descubrir conocimiento adicional, como el hecho que Jude Law actuó en la película A.I. de la cual Stanley Kubrick fue el productor.

CONSULTANDO LA WEB DE DATOS

Para acceder la información en la Web de Datos no es suficiente contar con un lenguaje de consultas. Como ya mencionamos, aunque las iniciativas como KG enriquecen los resultados de las búsquedas, sus capacidades de responder consultas son aún limitadas. Por ejemplo, no es posible expresar peticiones simples como "encuentre los directores italianos influenciados directa o indirectamente por Stanley Kubrick" o "encuentre los actores de menos de cincuenta años que hayan actuado en películas de Kubrick". Para consultar grafos RDF, existe un lenguaje de consulta estándar, llamado SPARQL [21]. Se parece a SQL enriquecido con un protocolo para consultar datos estructurados en la Web. Las debilidades de SPARQL son las mismas que las de cualquier lenguaje cuando enfrenta una topología que no está fija y de creación distribuida. Por tanto, no es ninguna sorpresa que varios lenguajes de consulta y navegación de grafos hayan sido propuestos para solucionar parcialmente la navegación semántica en la Web.

La especificación (y recuperación) de colecciones de sitios fue tempranamente tratada por herramientas como `wget`, que permite recolectar recursivamente información de sitios Web ("*crawling*"). El problema es que, aparte de no ser declarativa, esta herramienta se halla restringida solo a funcionalidades sintácticas. También existen *crawlers* semánticos como LDSpider [17] y técnicas de indexamiento como SWSE [16] y Sindice [19]. Estos métodos se enfocan en el *crawling* eficiente y masivo para construir repositorios centralizados y, por tanto, no se concentran en los lenguajes declarativos para especificar navegación sobre las fuentes de datos en la Web. Otras iniciativas optaron por enriquecer el lenguaje de consulta SPARQL con operadores navegacionales [20,6,12] mediante la consideración de un conjunto fijo de fuentes de datos (típicamente un único grafo RDF) pero con navegación a escala Web. Finalmente, otros métodos extienden el alcance de las consultas SPARQL mediante operadores de navegación [13] que permiten descubrir fuentes de datos relevantes para la consulta, aunque carecen de maneras de especificar la navegación. Es importante recalcar que ninguno de los métodos mencionados arriba incorpora acciones, una funcionalidad que, como veremos, se vuelve importante al momento de navegar.

NAVEGACIÓN EN LA WEB DE DATOS

Generalmente hablando, un lenguaje de navegación de grafos (ver [24] para un reciente survey) permite encontrar pares de nodos que están conectados por una secuencia de etiquetas

de arcos que satisfacen cierto patrón. La forma más común de especificar un patrón (o expresión navegacional) es vía una expresión regular sobre el alfabeto de etiquetas, comenzando la navegación en un nodo semilla. En el ejemplo previo sobre KG, el nodo semilla es Stanley Kubrick (o, más precisamente, su identificador interno en el KG). Sin embargo, el mecanismo de navegación estaba limitado: no era posible especificar patrones para seleccionar información relevante. KG sólo provee un conjunto de nodos relacionados (e.g., la película The Shining) desde los cuales es posible continuar la navegación manualmente. La Web de Datos, gracias al lenguaje de descripción RDF, permite la navegación semántica de grafos a escala Web.

En lo que sigue, presentamos un lenguaje navegacional para la Web de Datos llamado NautiLOD [11]. Este lenguaje está basado en expresiones regulares sobre predicados RDF entremezclados con tests del tipo ASK en SPARQL realizados sobre la descripción RDF de los recursos. Las expresiones regulares permiten expresar necesidades de información complejas que requieren de navegación a través de los nodos del grafo, mientras los tests permiten la selección de fuentes de datos relevantes desde las cuales continuar la navegación. NautiLOD también presenta un mecanismo que gatilla acciones (e.g., enviar mensaje de notificación) usando los datos encontrados durante la navegación. Discutiremos ejemplos del mundo real y presentaremos una implementación en la herramienta `swget`, la cual está disponible online[1].

NAUTILOD

Para dar un esbozo de las potencialidades del lenguaje NautiLOD, mostraremos algunos ejemplos considerando el extracto de datos mostrado en la **Figura 3**. Una presentación detallada de la sintaxis será provista en la siguiente sección.

Imagine que queremos descubrir todo lo que se predica sobre Stanley Kubrick en distintas fuentes de datos. La idea es considerar los arcos `<owl:sameAs>` que comienzan en el identificador de Kubrick en DBpedia. Se chequean entonces todos los triples de la forma `<dbpo: Stanley Kubrick, owl: sameAs, v>`, y se seleccionan todos los `v`'s encontrados. Finalmente, por cada uno de esos `v`'s, se retornan todas las URIs `w` de los triples de la forma `<v, p, w>` que se encuentren en la fuente de datos de `v`.

Lo anterior puede ser especificado utilizando NautiLOD mediante la expresión:

```
<dbpo:Stanley Kubrick>
  <owl:sameAs>/ <_>
```

Aquí, la expresión `<_>` denota un wildcard para los predicados RDF. Al evaluar esta expresión comenzando desde la URI dbpo: Stanley Kubrick obtenemos todas las distintas representaciones de Stanley Kubrick provistas por dbpedia.org, freebase.org y linkedmdb.org. Desde estos nodos, la expresión `<_>` se puede instanciar en cualquier predicado. El resultado de la evaluación es `{lmdb:/film/334, fb:Path of Glory}`. Esto remarca cómo NautiLOD es capaz de tratar con fuentes de datos distribuidos y dinámicas. Un ejemplo más complejo que realiza acciones sobre los datos se presenta a continuación. En él intentaremos encontrar aquellas películas (y sus alias) cuyo director tiene más de cincuenta años y ha sido influenciado, ya sea directa o indirectamente, por Stanley Kubrick. Cada vez que uno de esos directores es detectado enviaremos su sitio Wiki por email.

Esta especificación trata sobre caminos de *influencia* y aliases como en el ejemplo anterior; tests (expresados en NautiLOD usando consultas del tipo ASK en SPARQL) sobre la fuente de datos asociada con la URI dada (si alguien influenciado por Kubrick es encontrado, verifica si él/ella es de la edad apropiada); y acciones a realizarse usando datos de la fuente. La especificación en NautiLOD que expresa el ejemplo mencionado en el párrafo anterior es:

```
<dbpo:Stanley Kubrick>
(<dbpo:influenced>)+[Test]/Act/
<dbpo:director>/<owl:sameAs>
```

En la expresión, el símbolo + denota que uno o más niveles de influencia son aceptables, e.g., obtenemos directores como David Lynch o Quentin Tarantino. Desde este conjunto de recursos, la restricción en la edad forzada por la consulta ASK es evaluada en la fuente de datos asociada con cada uno de los recursos ya matcheados. Este filtro deja en este caso solo `dbpo: DavidLynch`. En este momento, sobre los elementos del conjunto (un elemento en este caso), la acción enviará vía email la página Wiki (obtenida de la consulta SELECT).

La acción **sendEmail**, implementada por un procedimiento de programación ad-hoc, no influencia el proceso de navegación. Por tanto, la evaluación continuará desde la `URI u = dbpo: DavidLynch`, al navegar la propiedad dbpo:director (encontrada en el conjunto de datos al dereferenciar `u`). Es posible hacer esto, por ejemplo, al seguir el triple `<u, dbpo: director, dbpo: BlueVelvet>`. Luego, desde `dbpo: blueVelvet`, la parte final de la expresión es evaluada. El resultado de la evaluación es: (1) el conjunto `{dbpo: BlueVelvet, fb: BlueVelvet}`, esto es, los datos acerca de la película BlueVelvet que se obtienen de dbpedia.org y freebase.org; (2) el conjunto de acciones realizadas, en este caso un email enviado.

LA SINTAXIS DE NAUTILOD

NautiLOD provee un mecanismo para declarativamente: (1) definir expresiones de navegación; (2) permitir el control semántico de la navegación; (3) realizar acciones como efectos colaterales a partir de los caminos de navegación. El reducto navegacional del lenguaje se basa en expresiones regulares de caminos, casi de la misma forma que algunos lenguajes de consulta para la Web y XPath. El control semántico es realizado a través de tests existenciales usando consultas SPARQL del tipo ASK sobre fuentes de datos RDF. Este mecanismo permite orientar la navegación basado en la información presente en cada nodo del camino. Finalmente, se gati- →

1 http://swget.wordpress.com

llan acciones de acuerdo a decisiones basadas en la especificación original y a la información local encontrada en cada fuente de datos.

La sintaxis del lenguaje está definida con respecto a la gramática mostrada en la Tabla XX. El lenguaje está basado en expresiones de camino (i.e., **`path`**), esto es, concatenación de expresiones de caso base construidas sobre predicados (i.e., **`pred`**), tests (i.e., **`tests`**) y acciones (i.e., **`action`**). Las expresiones complejas son disyunciones de expresiones: (1) expresiones que utilizan un número de repeticiones utilizando las funcionalidades de las expresiones regulares, y (2) expresiones con tests.

Los bloques sobre los que se construye una expresión NautiLOD son:

1. Predicados: el caso base de **`pred`** puede ser un predicado RDF o el wildcard `<_>` que denota cualquier predicado.
2. Expresiones Test: un `test` denota una expresión de consulta. Su caso base es una consulta ASK en SPARQL.
3. Expresiones Action: un `action` es una especificación procedural de un comando (e.g., enviar un mensaje de notificación, un comando GET, etc.), el cual obtiene sus parámetros de la fuente de datos alcanzada durante la navegación. Es un efecto colateral, esto es, no influencia el proceso subsiguiente de navegación.

Si se restringe a (1) y (2), NautiLOD puede ser visto como un lenguaje declarativo para describir porciones de la Web de Datos, i.e., un conjunto de URIs que se ajusta a cierta especificación semántica. Las expresiones NautiLOD son evaluadas contra la Web de Datos comenzando en una segunda URI `u`. El significado de una expresión NautiLOD es un conjunto de URIs más un conjunto de acciones gatilladas por la evaluación. Para una discusión más comprensiva de la semántica y complejidad del lenguaje el lector puede referirse a Fionda et al. 2012 [11].

SWGET

El comando `swget` es una implementación Java de NautiLOD. La herramienta está libremente disponible en el sitio web swget (http://swget.wordpress.com) donde también se discuten otros ejemplos y una descripción detallada de sus principios de funcionamiento. La aplicación se halla disponible tanto como una herramienta de línea de comando y como una GUI. Un ejemplo de la herramienta `swget` GUI se muestra en la **Figura 4**. La evaluación de una expresión puede ser visualizada de distintas maneras (e.g., como conjunto de URIs, como grafo RDF). En la Figura 4 se muestra la visualización del grafo, donde las descripciones de los diferentes recursos visitados durante la navegación están interconectados. Es posible seleccionar partes específicas del grafo al filtrar predicados, buscar predicados o nodos específicos, y expandir los nodos proveyendo un valor del radio a partir desde un nodo objetivo.

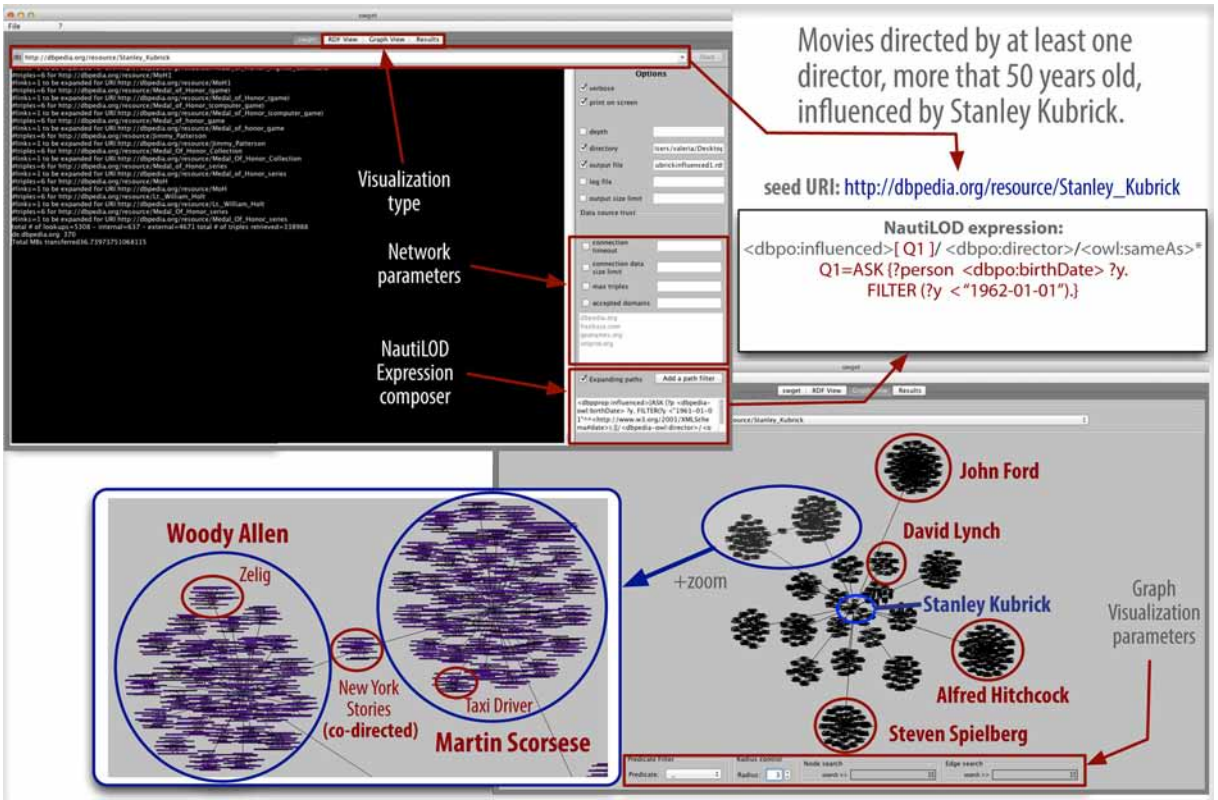


FIGURA 4.
EJEMPLO DE LA HERRAMIENTA SWGET (HTTP://SWGET.WORDPRESS.COM).

CONCLUSIÓN

EL CONOCIMIENTO ESTRUCTURADO COMO GRAFO IMPREGNA NUESTRA VIDA COTIDIANA. INICIATIVAS COMO FACEBOOK OPEN GRAPH Y GOOGLE KNOWLEDGE GRAPH SON BUENOS EJEMPLOS DE ESTA TENDENCIA. SIN EMBARGO, ELLOS FALLAN A LA HORA DE CAPTURAR UN ASPECTO DEL CONOCIMIENTO ESTRUCTURADO COMO GRAFO: LA POSIBILIDAD DE LA NAVEGACIÓN DECLARATIVA. EN ESTE ARTÍCULO HEMOS DEMOSTRADO LA IMPORTANCIA DE TENER UN LENGUAJE DECLARATIVO PARA LA NAVEGACIÓN AUTOMÁTICA EN LA WEB. DADO QUE MILES DE FUENTES DE DATOS SEMÁNTICAS, DISTRIBUIDAS E INTERCONECTADAS ESTÁN HOY EN DÍA DISPONIBLES, LA NAVEGACIÓN SE VUELVE NAVEGACIÓN SEMÁNTICA. EL LENGUAJE NAUTILOD Y SU IMPLEMENTACIÓN EN SWGET DEMUESTRAN LA POSIBILIDAD DE LA NAVEGACIÓN SEMÁNTICA Y AUTOMÁTICA A ESCALA WEB. NUESTRA EXPERIENCIA INDICA LA NECESIDAD DE CONTAR CON METADATOS SEMÁNTICOS ESTANDARIZADOS, LENGUAJES DE NAVEGACIÓN EXPRESIVOS PARA ESPECIFICAR CONOCIMIENTO EN LA WEB, ASÍ COMO LENGUAJES PARA ESPECIFICAR ACCIONES. POR ELLO PODEMOS DECIR: ¡UNA NUEVA ERA PARA LA WEB ESTÁ EMERGIENDO! ■

BIBLIOGRAFÍA

[1] DBLP Bibliography Database: http://dblp.l3s.de/d2r/

[2] Facebook Graph: https://www.facebook.com/about/graphsearch.

[3] Google Knowledge Graph: http://www.google.com/insidesearch/features/search/knowledge.html

[4] Google Scholar: http://scholar.google.com.

[5] Microsoft Academic Research: http://academic.research.microsoft.com

[6] F. Alkhateeb, J.-F. Baget, and J. Euzenat. Extending SPARQL with Regular Expression Patterns (for querying RDF). Journal of Web Semantics, 7(2):57-73, 2009.

[7] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. The Semantic Web, pages 722-735, 2007.

[8] S. L. Bechhofer and A. Miles. SKOS Simple Knowledge Organization System, 2009.

[9] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. IJSWIS, 5(3):1-22, 2009.

[10] M.J. Cafarella, A. Halevy, and J. Madhavan. Structured data on the web. Communications of the ACM, 54(2):72-79, 2011.

[11] V. Fionda, C. Gutiérrez, and G. Pirrò. Semantic Navigation on the Web of Data: Specification of Routes, Web Fragments and Actions. In World Wide Web Conference, pages 281{290. ACM, 2012.

[12] S. Harris and A. Seaborne. SPARQL 1.1 Query Language, 2010.

[13] O. Hartig, C. Bizer, and J.C. Freytag. Executing SPARQL Queries over the Web of Linked Data. In International Semantic Web Conference, pages 293-309, 2009.

[14] T. Heath and C. Bizer. Linked Data: Evolving the Web into a Global Data Space. Morgan & Clay-pool, 2011.

[15] J. E. Hirsch. An Index to Quantify an Individual's Scientific Research Output that Takes into Account the Effect of Multiple Coauthorship. Scientometrics, 85(3):741-754, 2010.

[16] A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, and S. Decker. Searching and browsing linked data with swse: The semantic web search engine. Web Semantics: Science, Services and Agents on the World Wide Web, 9(4):365-401, 2011.

[17] R. Isele, A. Harth, J. Umbrich, and C. Bizer. LDspider: An OpenSource Crawling Framework for the Web of Linked Data. In Poster- International Semantic Web Conference, 2010.

[18] D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language, 2004.

[19] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: A document-oriented lookup index for open linked data. Int. J. of Metad., Semant. and Ontolog., 3(1), 2008.

[20] J. Pérez, M. Arenas and C. Gutiérrez. nSPARQL: A Navigational Language for RDF. Journal of Web Semantics, pages 255 - 270.

[21] Eric Prud'hommeaux and Andy Seaborne. SPARQL 1.1 Query Language, 2008.

[22] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO - A Core of Semantic Knowledge. In World Wide Web Conference, pages 697-706. ACM, 2007.

[23] G. Weikum, G. Kasneci, M. Ramanath, and F. M. Suchanek. Database and information-retrieval methods for knowledge discovery. Communications of the ACM, 52(4):56-64, April 2009.

[24] P. T. Wood. Query Languages for Graph Databases. SIGMOD Record, 41(1):50-60, 2012.

SOCIEDAD CHILENA DE CIENCIA DE LA COMPUTACIÓN:
ORÍGENES, FUNDACIÓN (1984) Y PRIMEROS AÑOS

Este año 2014 se conmemoran treinta años desde la fundación, en 1984, de la Sociedad Chilena de Ciencia de la Computación (SCCC), iniciativa que representó la culminación de una concatenación de esfuerzos previos de colaboración entre académicos de distintas universidades. La creación de la SCCC tuvo antecedentes directos (1975-1984) e influencias indirectas de los comienzos de la computación universitaria chilena (1962-1974). En sus primeros años (1984-1989), la Sociedad desarrolló múltiples actividades que permitieron la consolidación y reconocimiento académico de la disciplina en Chile y en el extranjero.

JUANÁLVAREZ

Académico Departamento de Ciencias de la Computación, Universidad de Chile. Master of Mathematics (Computer Science), University of Waterloo. Ingeniero de Ejecución en Procesamiento de la Información, Universidad de Chile. Junto a su labor como docente, trabaja en reconstruir la Historia de la Computación en Chile.

jalvarez@dcc.uchile.cl

INTRODUCCIÓN

Los orígenes de la Sociedad Chilena de Ciencia de la Computación (SCCC) se encuentran e identifican con el inicio de la computación universitaria chilena. En efecto, en los primeros computadores, administrados por los centros de computación de las universidades pioneras en el área, se desarrollaron aplicaciones de cálculo científico y de ingeniería. La necesidad de difundir y utilizar la computación, en los sectores público y privado de la industria y los servicios, estimuló la creación de las primeras carreras universitarias de programación e ingeniería de ejecución.

Las aplicaciones crecieron en envergadura y complejidad requiriendo tecnología e ingeniería nacional sustentada en investigación científica y tecnológica. Los ingenieros e investigadores, que trabajaban en los centros de computación y en otras especialidades, convergieron a mediados de los setenta creándose los primeros departamentos de Ciencia de la Computación. Algunos de estos académicos obtuvieron postgrados en el extranjero y contribuyeron a profesionalizar las tareas de investigación y docencia. Simultáneamente se crearon los primeros pre y postgrados académicos y, casi una década después, las primeras carreras de Ingeniería Civil en Computación.

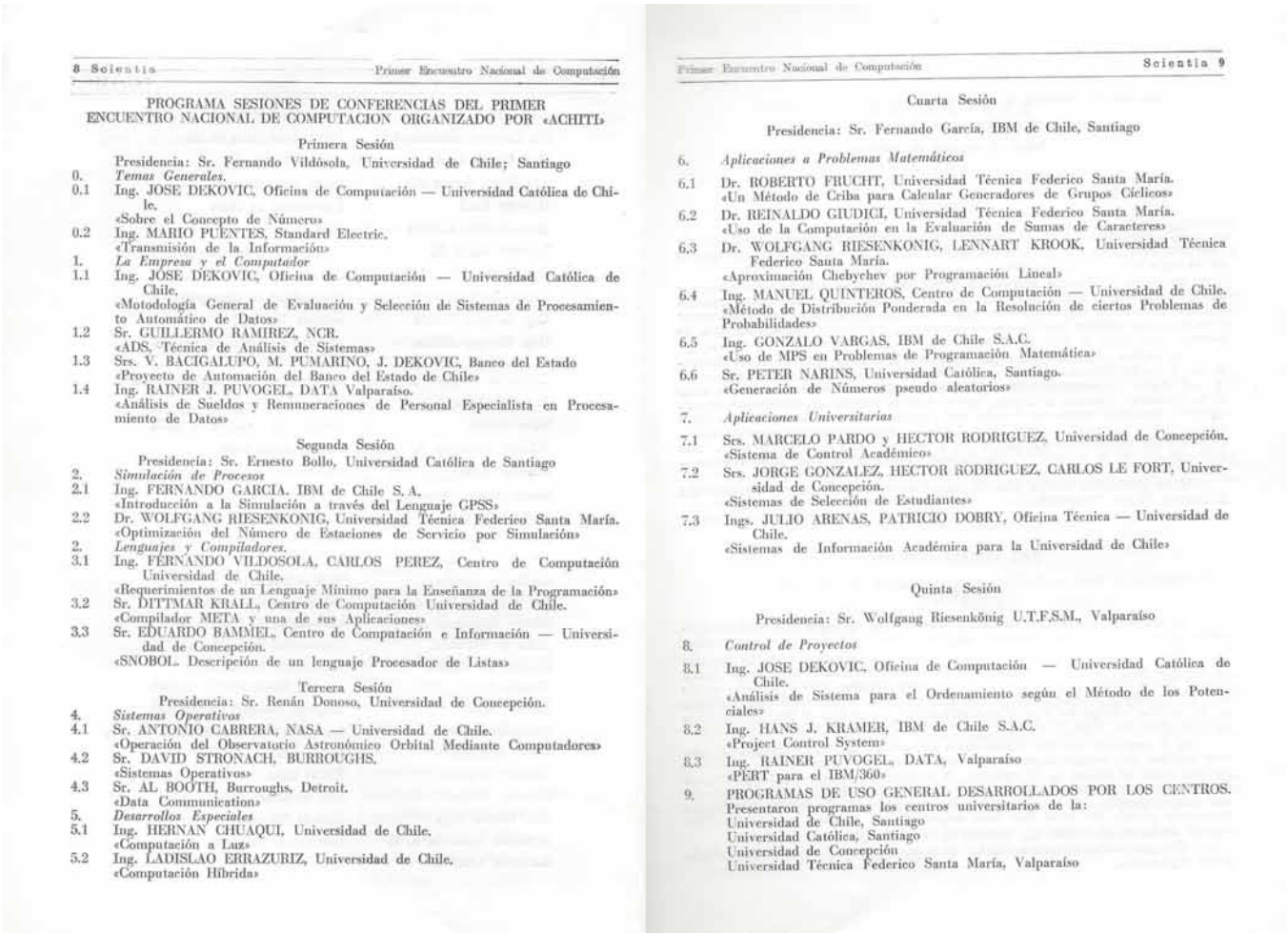
La necesidad de coordinación y colaboración entre los investigadores y académicos de las distintas universidades se tradujo en reuniones, encuentros, seminarios y conferencias nacionales e internacionales. De este intercambio surgió la necesidad de la creación de una asociación que los reuniera y representara, la que finalmente tomó la forma de una sociedad científica.

A continuación se presenta el camino que condujo a la creación de la SCCC. En primer lugar se revisa los comienzos de la computación universitaria chilena (período 1962-1974). En seguida se analiza la década 1975-1984 que incluye los antecedentes e influencias más directas para la creación de la Sociedad. La sección siguiente presenta la gestación y fundación de la SCCC en el año 1984. Finalmente, se revisa el primer lustro de vida de la Sociedad (1984-1989).

EVENTOS Y ASOCIACIONES INICIALES (1962-1974)

Los primeros computadores se instalaron en las universidades chilenas a comienzos de los sesenta [1]. En 1962 se instaló el computador alemán ER-56 en la Universidad de Chile (UCH). En los años siguientes llegaron tres computadores IBM-1620: a la Universidad Católica (UC) en 1963, a la Universidad Técnica Federico Santa María (UTFSM) en 1964 y a la Universidad de Concepción (UdeC) en 1966. Adicionalmente,

→



8 Scientia — Primer Encuentro Nacional de Computación

PROGRAMA SESIONES DE CONFERENCIAS DEL PRIMER ENCUENTRO NACIONAL DE COMPUTACION ORGANIZADO POR «ACHITI»

Primera Sesión

Presidencia: Sr. Fernando Vildósola, Universidad de Chile; Santiago

0. *Temas Generales.*
0.1 Ing. JOSE DEKOVIC, Oficina de Computación — Universidad Católica de Chile.
«Sobre el Concepto de Número»
0.2 Ing. MARIO PUENTES, Standard Electric.
«Transmisión de la Información»
1. *La Empresa y el Computador*
1.1 Ing. JOSE DEKOVIC, Oficina de Computación — Universidad Católica de Chile.
«Metodología General de Evaluación y Selección de Sistemas de Procesamiento Automático de Datos»
1.2 Sr. GUILLERMO RAMIREZ, NCR.
«ADS, Técnica de Análisis de Sistemas»
1.3 Srs. V. BACIGALUPO, M. PUMARINO, J. DEKOVIC, Banco del Estado
«Proyecto de Automación del Banco del Estado de Chile»
1.4 Ing. RAINER J. PUVOGEL, DATA Valparaíso.
«Análisis de Sueldos y Remuneraciones de Personal Especialista en Procesamiento de Datos»

Segunda Sesión

Presidencia: Sr. Ernesto Bollo, Universidad Católica de Santiago

2. *Simulación de Procesos*
2.1 Ing. FERNANDO GARCIA, IBM de Chile S. A.
«Introducción a la Simulación a través del Lenguaje GPSS»
2.2 Dr. WOLFGANG RIESENKONIG, Universidad Técnica Federico Santa María.
«Optimización del Número de Estaciones de Servicio por Simulación»
2. *Lenguajes y Compiladores.*
3.1 Ing. FERNANDO VILDOSOLA, CARLOS PEREZ, Centro de Computación Universidad de Chile.
«Requerimientos de un Lenguaje Mínimo para la Enseñanza de la Programación»
3.2 Sr. DITTMAR KRALL, Centro de Computación Universidad de Chile.
«Compilador META y una de sus Aplicaciones»
3.3 Sr. EDUARDO BAMMEL, Centro de Computación e Información — Universidad de Concepción.
«SNOBOL. Descripción de un lenguaje Procesador de Listas»

Tercera Sesión

Presidencia: Sr. Renán Donoso, Universidad de Concepción.

4. *Sistemas Operativos*
4.1 Sr. ANTONIO CABRERA, NASA — Universidad de Chile.
«Operación del Observatorio Astronómico Orbital Mediante Computadores»
4.2 Sr. DAVID STRONACH, BURROUGHS.
«Sistemas Operativos»
4.3 Sr. AL BOOTH, Burroughs, Detroit.
«Data Communication»
5. *Desarrollos Especiales*
5.1 Ing. HERNAN CHUAQUI, Universidad de Chile.
«Computación a Luz»
5.2 Ing. LADISLAO ERRAZURIZ, Universidad de Chile.
«Computación Híbrida»

Primer Encuentro Nacional de Computación — Scientia 9

Cuarta Sesión

Presidencia: Sr. Fernando García, IBM de Chile, Santiago

6. *Aplicaciones a Problemas Matemáticos*
6.1 Dr. ROBERTO FRUCHT, Universidad Técnica Federico Santa María.
«Un Método de Criba para Calcular Generadores de Grupos Cíclicos»
6.2 Dr. REINALDO GIUDICI, Universidad Técnica Federico Santa María.
«Uso de la Computación en la Evaluación de Sumas de Caracteres»
6.3 Dr. WOLFGANG RIESENKONIG, LENNART KROOK, Universidad Técnica Federico Santa María.
«Aproximación Chebychev por Programación Lineal»
6.4 Ing. MANUEL QUINTEROS, Centro de Computación — Universidad de Chile.
«Método de Distribución Ponderada en la Resolución de ciertos Problemas de Probabilidades»
6.5 Ing. GONZALO VARGAS, IBM de Chile S.A.C.
«Uso de MPS en Problemas de Programación Matemática»
6.6 Sr. PETER NARINS, Universidad Católica, Santiago.
«Generación de Números pseudo aleatorios»
7. *Aplicaciones Universitarias*
7.1 Srs. MARCELO PARDO y HECTOR RODRIGUEZ, Universidad de Concepción.
«Sistema de Control Académico»
7.2 Srs. JORGE GONZALEZ, HECTOR RODRIGUEZ, CARLOS LE FORT, Universidad de Concepción.
«Sistemas de Selección de Estudiantes»
7.3 Ings. JULIO ARENAS, PATRICIO DOBRY, Oficina Técnica — Universidad de Chile.
«Sistemas de Información Académica para la Universidad de Chile»

Quinta Sesión

Presidencia: Sr. Wolfgang Riesenkönig U.T.F.S.M., Valparaíso

8. *Control de Proyectos*
8.1 Ing. JOSE DEKOVIC, Oficina de Computación — Universidad Católica de Chile.
«Análisis de Sistema para el Ordenamiento según el Método de los Potenciales»
8.2 Ing. HANS J. KRAMER, IBM de Chile S.A.C.
«Project Control System»
8.3 Ing. RAINER PUVOGEL, DATA, Valparaíso
«PERT para el IBM/360»
9. PROGRAMAS DE USO GENERAL DESARROLLADOS POR LOS CENTROS.
Presentaron programas los centros universitarios de la:
Universidad de Chile, Santiago
Universidad Católica, Santiago
Universidad de Concepción
Universidad Técnica Federico Santa María, Valparaíso

FIGURA 1.
PROGRAMA PRIMER ENCUENTRO NACIONAL DE COMPUTACIÓN.

en 1964 la Universidad Técnica del Estado (UTE) recibió un computador Datatron. Todas las máquinas utilizaban tecnología de transistores ("segunda generación") y para su administración, operación y difusión se crearon los centros de computación. La docencia formal se limitó a asignaturas (o parte de ellas) para carreras de ingeniería.

Contemporáneamente, surgieron iniciativas de colaboración entre universidades. En enero de 1962, se registra comunicación entre el Decano Frucht de la UTFSM y Santiago Friedmann, director del Centro de Computación de la U. de Chile, con el propósito de crear un centro regional de computación, iniciativa que lamentablemente no prosperó [2]. En el mismo año, las universidades Santa María y Católica colaboraron a propósito que ambas adquirieron computadores IBM-1620 [3].

En septiembre de 1962, desde la U. de Chile, Friedmann convocó a una reunión que "fijará las bases para la creación de un Instituto Chileno de Investigación Operativa y Computación" [4]. La reunión se efectuó el 2 y 3 de octubre en el Auditorium de IDIEM y resolvió, sin embargo, crear una "Asociación Chilena de Computación" [5] que se ocuparía de:

a) constituir comités de trabajo;
b) organizar reuniones periódicas, conferencias, seminarios;
c) editar publicaciones;
d) establecer comités regionales;
e) adherir al Centro Internacional de Cálculo y mantenerse en relación directa con entidades similares de los demás países;
f) promover el intercambio de profesores y estudiantes, el otorgamiento de becas y la venida de expertos que traigan su experiencia y conocimientos.

De los trabajos preparativos para la formación de la Asociación (como elaboración de los estatutos) se encargará un comité organizador de cinco santiaguinos, los señores Claro, Dekovic, Guillermo González, Grandjot y Riquelme; en provincias colaborarán los señores Phagouapé (Concepción) y Frucht (Valparaíso).

En 1967 se fundó oficialmente la "Asociación Chilena de Computación y Tratamiento de la Información" (ACHITI) "en una acción conjunta de las Universidades que contaban con computadores". El objetivo principal fue "posibilitar el intercambio de experiencias en el amplio campo constituido por la tecnología de la información, sus fundamentos y sus aplicaciones". La ACHITI se afilió a la International Federation for Information Processing (IFIP) y organizó el "Primer Encuentro Nacional de Computación" que se realizó entre el 11 y 14 de diciembre de 1968 en la UTFSM [6]. En el evento se presentaron 28 trabajos (7 de ellos con versión completa en las actas): 7 de UCH, 4 de UTFSM, 4 de UC, 3 de UdeC, 3 de IBM, y 7 de otras instituciones (**Figura 1**).

Entre el 6 y 12 de septiembre de 1967, con el auspicio de UNESCO, la Universidad de Concepción organizó el "Primer Simposio Latinoamericano de centros académico-científicos de computación" [7]. Los temas tratados estuvieron relacionados con los centros: estructura, docencia, investigación, medio universitario, industria, campo social. Se presentaron 40 trabajos de varios países, desde México por el norte hasta Argentina y Chile por el sur. En la oportunidad se discutieron las bases para la creación de un "Instituto Latinoamericano de Ciencias de la Información y Computación", nombrándose un directorio provisorio encabezado por Phenix Ramírez de la U. de Concepción y Sergio Beltrán de la UNAM de México.

Por otra parte, y en el contexto de los procesos de reforma universitaria de fines de los sesenta, se crearon las primeras carreras universitarias de programación de tres años de duración en la U. de Chile (1968), U. de Concepción (1970) y U. Católica (1971). Seguidamente, se crearon carreras de Ingeniería de Ejecución de cuatro años en la U. de Chile (1971) y en la U. Técnica del Estado (1972). Paralelamente, la Empresa Nacional de Computación (EMCO), creada en 1968, capacitó programadores y analistas de sistemas [8].

Los profesores de estas primeras carreras fueron mayoritariamente ingenieros de otras especialidades que trabajaban en los centros de computación universitarios y en algunas empresas públicas. Para entonces, las universidades y EMCO contaban con computadores de tercera generación y de propósito general, lo que permitió apoyar la formación de profesionales y tecnólogos para desempeñarse indistintamente en computación científica y computación administrativa.

El 29 de agosto de 1969 en la UC se celebró la Primera Reunión de los Centros Universitarios de Computación promovida por Ernesto Bollo (UC), René Peralta (U. de Chile), José Durán (UTE), Renán Donoso (U. de Concepción), Juan Ignacio Cahís (UC) y Wolfgang Riesenkönig (UTFSM). El 12 de diciembre de 1970 se creó formalmente la "Asociación Chilena de Centros Universitarios de Computación" (ACUC) conformada inicialmente por centros de las universidades Católica, de Chile, Técnica del Estado, de Concepción, Santa María, Católica de Valparaíso, Austral y del Norte [9]. Posteriormente, se incorporó ECOM (el nuevo nombre de EMCO) en carácter de invitado. Años después, en 1975, ECOM y las universidades de Chile, Católica, Técnica del Estado y de Concepción crearon un "Plan Nacional de Capacitación" (PLANACAP) para ayudar a superar el déficit de especialistas [10].

Este recuento de iniciativas en los inicios de la computación universitaria culminó en 1974 con el "Panel de Discusión sobre Tópicos de Computación" organizado por la Universidad Católica de Valparaíso y coordinado por Aldo Migliaro. Inicialmente el Panel tuvo carácter regional, pero en 1975 [11] y 1976 [12] alcanzó cobertura nacional. Posteriormente, desde 1977, se transformó en internacional dando origen, en 1979, al "Centro Latinoamericano de Estudios en Informática" (CLEI) que en 2014 realizó la edición N° 40 de su conferencia anual.

Cabe señalar que en el 2° Panel de Discusión (1975), José Durán (de la UTE) presentó los estatutos de la "Asociación Chilena de Tratamiento de la Información" (ACTI) que establecía que los miembros ordinarios debían "ser egresados de una carrera universitaria que tenga no menos de cinco años de estudios" y "desempeñar dentro de sus actividades profesionales labores relacionadas en forma importante con la Computación y el Tratamiento de la Información". Considerando que entonces las únicas carreras universitarias de la especialidad duraban menos de cinco años, la ACTI estuvo orientada a los profesionales de otras carreras (principalmente ingeniería) que se dedicaron a la Computación en la etapa inicial de la disciplina. [13]

EVENTOS Y CONGRESOS PREVIOS A LA FUNDACIÓN DE LA SCCC (1975-1984)

Las iniciativas e hitos mencionados en la sección anterior estuvieron principalmente organizados y coordinados en los centros de computación que fueron creados en torno a los primeros computadores y promovieron su utilización en las diversas áreas universitarias, especialmente en las disciplinas de ingeniería y ciencia. La experiencia acumulada y el desarrollo independiente del área gatilló la creación de departamentos académicos centrados específicamente en la disciplina de Computación con labores de docencia, investigación y extensión, según los principios propiciados por la reforma universitaria.

En 1975 se crearon los primeros departamentos de Ciencia de Computación en la U. de Chile [14], UTFSM [15] y UTE. Los dos primeros impartieron programas de Magíster en Ciencias de la Computación y el tercero una Licenciatura en Matemáticas y Ciencia de la Computación. Posteriormente, se crearon departamentos similares en las universidades Católica [16] y de Concepción [17]. Todos estos departamentos se dedicaron a la investigación y la docencia de pre y postgrado. Inicialmente sus académicos eran ingenieros de otras especialidades que gradualmente fueron obteniendo grados de Magíster y Doctor en el extranjero.

→

NOMBRE	UNIVERSIDAD	TITULO DEL TRABAJO
Berry, Daniel	UCLA	Specification of programs (13 páginas)
Campos, Iván Estrin, Gerald	UFMG UCLA	SARA Aided Design of Software for Concurrent Systems (12p)
Hesse, Wolfgang	U. Munich	A Wide Spectrum Language in action on systematic program development (14p)
Lauterbach, Carlos	UC	Concepts in Data Abstractions (50p)
Lucena, Carlos	PUCRJ	Proyecto de Programas: Idéias Correntes e Perspectivas (33p)
Lucena, Carlos Pequeño, Tarcisio	PUCRJ	Program Derivation Using Data Types: A Case Study (19p)
Pequeño, Tarcisio Lucena, Carlos	PUCRJ	An Axiomatic Method for Data Type Specificaction and its use in Program Verification (18p)
Cowan, D.D. Lucena, Carlos	UW PUCRJ	A Data-Directed Approach to Program Construction (37p)
Setzer, V.W.	USP	Program Development by transformations applied to Relational Data-Base Queries (22)

TABLA 1.
PRIMER SEMINARIO INTERNACIONAL DE INVIERNO (1978).

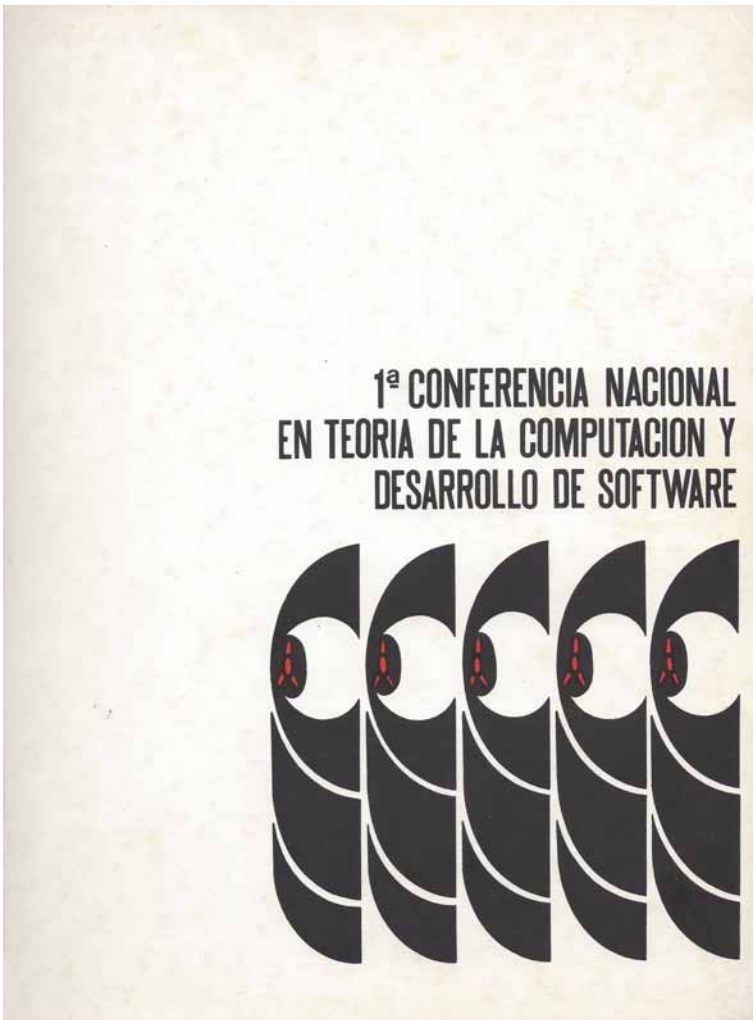


FIGURA 2.
PRIMERA CONFERENCIA NACIONAL.

Por otra parte, a comienzos de los ochenta se impuso una nueva legislación universitaria que fomentó el mercado educacional y la competencia por los recursos. Al mismo tiempo, redujo las dos universidades estatales nacionales al ámbito de Santiago (de hecho la UTE cambió su nombre a U. de Santiago de Chile - USACH) y sus sedes dieron origen a las universidades estatales regionales. Adicionalmente, la legislación quitó el carácter estrictamente universitario a varias carreras, entre otras a las ingenierías de ejecución del área. Esta situación, gatilló y/o coincidió con la iniciativa de crear carreras de Ingeniería Civil de seis años de duración en la UTFSM, U. de Chile, U. de Concepción, U. Católica y U. de Santiago (a cargo del nuevo Departamento de Ingeniería Informática creado en 1982) [18].

El acelerado desarrollo del área motivó a PLANACAP y a la Universidad Católica a organizar, entre el 26 y 29 de julio de 1978, el "I Seminario Internacional de Invierno sobre Desarrollo de Software Confiable" con el propósito de "dar a conocer y utilizar en nuestro medio las más modernas técnicas de diseño, construcción y evaluación de sistemas de computación" [19]. El evento contó con ocho investigadores invitados y se publicaron actas con sus trabajos (**Tabla 1**).

Al año siguiente, entre el 23 y 29 de agosto de 1979, se realizó el "II Seminario Internacional de Invierno sobre Ingeniería de Software" por ser "ésta una de las areas de más reciente desarrollo en computación" [20]. Los expositores fueron Daniel Berry de la UCLA, Peter Wegner de la Brown University y Arndt von Staa de la PUCRJ. El Seminario de Invierno continuó en los años 1980 y 1981. Posteriormente, entre el 2 y 6 de agosto de 1982, la U. Católica realizó el "I Seminario de Invierno en Ciencia de la Computación" sobre el tema de "Evaluación de Sistemas Computacionales" con los expositores Domenico Ferrari y Alan Smith de la U. de California-Berkeley, Ashok Agrawala de la U. de Maryland, Jeffrey Buzen de Boston, y los chilenos Luis Felipe Cabrera de la UC y Juan Muñoz de la empresa SID [21].

I Conferencia Nacional (1979)			II Conferencia Nacional (1980)		
NOMBRE	UNIVER.	TÍTULO	NOMBRE	UNIVER.	TÍTULO
Dezerega, G. Pino, J.	UC UCH	Comité Organizador	Dezerega, G. Pino, J.	UC UCH	Comité Organizador
Mujica, S. Pérez, H. Piquer, A. Rodríguez, H	UC UTFSM UCH UdeC	Comité de Programa	Guzmán, M. Meléndez, H. Pereda, A. Pino, J.	UCH UC UdeC UCH	Comité de Programa
Ovalle, A. Mujica, S.	UC	Experiencia en cursos introductorios a la computación (1página)	Poblete, P.	UCH	SIETE: Aspectos de su implementación (6p)
Pérez, H.	UTFSM	Herramientas para la docencia en Programación de Sistemas Operativos (17p)	Asuar, J.	UCH	Generación y control de sonidos musicales por medio de un computador (1p)
Gutiérrez, J. Muñoz, C.	Banco Osorno	Selección dinámica: alternativa para extraer información en bases de datos (17p)	Berry, D. Berry, O.	UCLA	The programmer-Client Interaction in Arriving at Program Specifications: Abstract Data Typing, Strong Typing and Jewish Motherhood (20p)
Guerra, L. Mahn, R. Pérez, H. Pérez, G. Barbera, A.	UTFSM Texas-Chile UCH	Desarrollo pragmático de una base de datos farmacológica y toxicológica (14p)	Olivos, J.	UCH	Sobre el cálculo simultáneo de varios multinomios (1p)
Mujica, S. Fuller, D. Fernández, C.	UC	Lenguajes de Computación Paralela (16p)	Oyarzún, F.	UCH	Assembler usado como metalenguaje (1p)
Leiss, E.	U. Kentucky	Representing regular languages by expressions (6p)	Krell, E. Meléndez, H. Mujica, S.	UC	Proposición de un Currículum para una Carrera de alto nivel en Ciencia de la Computación (8p)
Bornscheuer, G.	UCV	Una breve introducción al software science (12p)	Chuaqui, R.	UC	La Lógica Matemática como Método de Mecanización de los Razonamientos (2p)
Hernández, R. Piquer, A. Poblete, P.	UCH	Mejoras a la interfaz con el usuario del sistema CMS bajo VM/370 (6p)	Mujica, S.	UC	Sobre el Algoritmo Coke-Kasami-Younger Concurrente (6p)
Wong, J.	U. Waterloo	Analytic modelling of computer Networks (1p)	Poblete, P.	UCH	Estrategias de Implementación de un Sistema de Recuperación de la Información (5p)
Pérez, H.	UTFSM	Conceptos de Leng.de alto nivel para aplicaciones de proc. distribuido (1p)	Meléndez, H.	UC	Un Modelo de Generación de Código (12p)
Van Buer, D. Mujica, S.	UCLA UC	An architecture for distributed processes (10p)	Mujica, S.	UC	Sistemas de Computación para Oficinas (1p)
Pino, J.	UCH	Un algoritmo generalizado para puntos centrales en árboles (1p)	Olivos, J.	UCH	On the Vectorial Addition Chains(16p)
Lauterbach, C.	UC	Lenguajes de definición de datos (5p)			
Pino, J. Piquer, A. Poblete, P.	UCH	BIRDS: Bibliographical Information Retrieval and Dissemination System (12p)			
Leiss, E.	U. Kentucky	New directions in statistical database Security (17p)			
Cabrera, L.	UC	Query Processing (12p)			
Cabrera, L.	UC	Maintaining multiple copy consistency (7p)			

TABLA 2.
CONFERENCIAS NACIONALES DE 1979 Y 1980.

NOMBRE	UNIVERSIDAD	TITULO DEL TRABAJO
Berry, O. King, R.	South California	An Abstract Data Type Approach to Database Design (15p)
Sampaio, A. Parsaye-Ghoni	U.Federal Para' UCLA	OBJ Specification and Testing of Generalized Hardware BBs (52p)
Farrán, Y. Stanton, M.	UdeC PUCRJ	Programacao Concorrente Usando Monitores – Uma Implementacao (12p)
De Oliveira, A. Mendes, V.	Brasilia	Projeto de Fabricacao de Módulos de Memória Semiconductora Utilizando Circuitos Integrados de Alta Densidade (15)
Sabani, C. Rockenback, L.	UFRGS	O Uso de Dialogos na Interacao Homem-Máquina (30p)
Leiss, E.	U. Houston	On the security of Randomized Databases: A Simulation (24p)
Linden, N. Berry, D.	Israel UCLA	Parametrization and Abstract Data Types in a Program Design Language: The Design of Software Development Processor (30p)
Cabrera, L. Meyer, M.	UC California-Berkeley	The INGRES Protection System (28p)
Cavalcanti, A.	PUCRJ	Um estudo comparativo de Métodos de Especificacao de Sistemas Automatizados (31p)
Veloso, P.	PUCRJ	Methodical Specification of Abstract Data Types via Rewrite Rules (21p)
De Castilho, J. Furtado, A. Veloso, P.	PUCRJ	A Formal Approach to the Specification and Design of Database Applications (15p)
Araya, A.	UCH	Un Sistema Basado en Reglas para Implementar Sistemas de Inteligencia Artificial (14p)
Barahona, F.	UCH	On the Computational Complexity of Certains Physic Models (3p)
Mujica, S. Pinto, J.	UC	On Asynchronous Hardware Design (27p)

TABLA 3.
PRIMERA CONFERENCIA INTERNACIONAL EN CIENCIA DE LA COMPUTACIÓN (1981).

Los encuentros y seminarios motivaron a las universidades de Chile y Católica a organizar conjuntamente una conferencia orientada a los investigadores chilenos. Así, el 23 y 24 de agosto de 1979 en la U. de Chile, se realizó la "I Conferencia Nacional en Teoría de la Computación y Desarrollo Software" (**Figura 2**) con 17 trabajos aceptados (incluyendo 4 extranjeros)[22]. Al año siguiente, desde el 4 al 7 de agosto de 1980 en la U. Católica, se realizó la "II Conferencia Nacional en Sistemas de Computación" con 12 trabajos aceptados (entre ellos 1 extranjero) [23]. De los 29 trabajos aceptados en 1979 y 1980, 10 fueron de académicos de la UC (35%), 9 de la UCH (31%) y 3 de la UTFSM (10%) (**Tabla 2**).

El éxito de las dos conferencias nacionales, y el interés que despertaron fuera de Chile, aconsejaron organizar la "I Conferencia Internacional en Ciencia de la Computación". El evento, co-organizado por las universidades Católica y de Chile, se realizó entre el 24 y el 27 de agosto de 1981 en la casa Central de la U. Católica con 14 trabajos aceptados (**Tabla 3**) por un Comité de Programa integrado por dos extranjeros y dos chilenos [24].

La Conferencia Internacional continuó realizándose anualmente. En sus cuatro primeras ediciones, entre los años 1981 y 1984, se aceptaron 54 trabajos: 10 de Chile (19%), 18 de Brasil (33%), 10 de Estados Unidos (19%), 3 de Canadá, 3 de Israel, 2 de Francia, y 8 de otros países (**Tabla 4**). Cabe señalar que en 1984 se utilizó por primera vez un afiche de difusión con un cuadro de pintura chilena: "Abrir el cielo y encontrar la vida" de Roberto Matta (ver imagen principal de este artículo).

FUNDACIÓN DE LA SCCC (1984)

Diversas razones motivaron la creación de la SCCC. Por una parte, el área tenía ya dos décadas de desarrollo y de acelerada evolución. En segundo lugar, los departamentos disciplinares contaban con una cantidad significativa de académicos, algunos con doctorados obtenidos en el extranjero y otros con maestrías internacionales y nacionales. Tercero, la exitosa realización de las conferencias, primero nacionales y luego internacionales, requerían de un soporte institucional que garantizara su continuación. En síntesis, la disciplina precisaba de un organismo que la representara y difundiera ante las instituciones públicas y privadas, nacionales y extranjeras, profesionales y científicas, educacionales e industriales. En otras palabras, se necesitaba una cara visible de la computación chilena ante la sociedad.

Sucesivas conversaciones, correos y reuniones realizadas durante el otoño de 1984, principalmente entre académicos de las universidades de Chile, Católica y de Santiago, resultaron en la creación de la "Sociedad Chilena de Ciencia de la Computación" [25]. La Ley de Asociaciones Gremiales —creada en 1981 para obligar a los colegios profesionales a terminar con su naturaleza única, afiliación obligatoria y control de la ética— no pareció el marco legal adecuado. En consecuencia, e inspirados en los estatutos de una sociedad de matemáticas aplicadas, se decidió crear una sociedad científica.

El 5 de octubre de 1984 se llevó a efecto la reunión convocada para acordar los estatutos de la SCCC. Por unanimidad de los asistentes se acordó la constitución de una Corporación, Persona Jurídica de Derecho Privado sin fines de lucro, denominada Sociedad Chilena de Ciencia de la Computación cuyos estatutos se transcribieron (manualmente) al libro de Actas [26]. En su Artículo 3° estableció:

La finalidad de la Corporación es la de estimular la investigación en el campo de la Computación, la divulgación de esta disciplina y el contacto con las personas que tengan como ocupación la práctica de esta ciencia.

Para el cumplimiento de estos objetivos tendrá las siguientes funciones:

a) Organizar reuniones científicas periódicas y fomentar la publicación de los trabajos de investigación presentados.

b) Apoyar la formación de grupos de socios interesados en desarrollar áreas específicas de la Ciencia de la Computación.

c) Propender al mejoramiento de la enseñanza de la Ciencia de la Computación en todos los niveles y a lo largo del país.

d) Mantener relaciones con otras sociedades científicas o profesionales de Chile y el extranjero y ayudar al intercambio de informaciones.
e) Asesorar a los organismos gubernamentales o internacionales en asuntos o problemas de carácter científico en los casos que le sea requerido.
f) En general, ejecutar todos los actos que sean necesarios para el cumplimiento de sus fines.

La corporación tendrá en todo caso, carácter estrictamente científico, no pudiendo proponerse fines gremiales ni de lucro y deberá manifestarse ajena a toda discriminación política, religiosa, racial y de sexo.

Respecto de los socios, el artículo 7° estableció la existencia de cuatro categorías:

a) Socios Activos: *Serán aquellos que suscriban la escritura de constitución de la corporación y los que habiendo realizado un trabajo de incorporación que haya sido aceptado en la Conferencia anual de la Sociedad, soliciten su ingreso mediante una presentación escrita dirigida al presidente de la corporación.*

b) Socios Adherentes: *Serán aquellos que cumplan con el siguiente procedimiento:*

1. Presentar una solicitud escrita firmada por el solicitante, declarando en ella que conoce y promete cumplir las exigencias del presente estatuto.

2. Que esta solicitud sea aprobada por el directorio.

c) Socios Honorarios: *Esta es una categoría honorífica reservada para distinguidos científicos, chilenos o extranjero, que hayan aportado una contribución valiosa al desarrollo de la Ciencia de la Computación en Chile. Será facultad de la asamblea de socios otorgar esta distinción, a propuesta del directorio.*

d) Socios Institucionales: *Esta es una categoría reservada a empresas o instituciones que deseen colaborar en la realización de los objetivos de la Sociedad. Cada socio institucional deberá designar una persona para que lo represente. Este tipo de socios tendrá los derechos y obligaciones de un socio adherente.*

Por su parte, el artículo 1° transitorio estableció:

El Directorio Provisorio estará integrado por las siguientes personas: Señor Pedro Hepp Kuschel, como presidente; Señor Patricio Poblete Olivares, secretario; Señor Edgardo Krell Goldfard, tesorero; Señor Jorge Olivos Aravena y Señor Yussef Farrán Leiva, directores.

El "Acta de la Sesión N°1/84" fue firmada por 18 asistentes: Pedro Hepp, Patricio Poblete, Jorge Olivos, Pablo Alliende, Rafael Hernández, José Benguria, Ernesto Azorín, Edgardo Krell, Marcelo Pardo, Juan Carlos Cockbaine, Horacio Meléndez, Sergio Mujica, David Fuller, Miguel Nussbaum, Francisco Aurtenechea, Juan Carlos Cordero, Hugo Bórquez, Óscar Mimica. Por razones prácticas los 18 firmantes eran académicos santiaguinos: 7 de la U. Católica, 6 de la U. de Chile y 5 de la U. de Santiago. Sin embargo, la composición de la directiva reflejó diversidad institucional y amplitud nacional: Hepp (UC), Poblete (UCH), Krell (USACH), Olivos (UCH), Farrán (UdeC) y Mujica (USACH) como director suplente (**Figura 3**).

La Revista Informática subtituló una entrevista a Pedro Hepp como "La nueva alianza para el progreso que en el campo computacional busca hacer expeditos los caminos para juntar el talento disperso y acortar la brecha que hoy separa a Chile de naciones desarrolladas" [27]. El presidente de la SCCC afirmó que "nos asociamos para vencer cien años de soledad" y "si marchamos unidos podremos sumar el talento y servir en la mejor forma las necesidades del país". Por otra parte, se consignó que "suman aproximadamente 50 los especialistas que tienen un postgrado en Ciencia de la Computación realizado en el extranjero. Unos 10 a 12 de ellos tienen el grado de Doctor. La mayor parte de ellos está en las universidades. Hay uno solo que se cambió al sector privado, donde creó su propia empresa. A nivel Magíster, en cambio, existen varios que están en la empresa privada".

LOS PRIMEROS AÑOS DE LA SCCC (1984-1989)

La primera actividad de la directiva fue lograr la legalización de la SCCC lo que implicó obtener, tanto una escritura pública (diciembre 1984), como la autorización y firma de los ministerios de Interior y Justicia. Para financiar estos trámites se consiguió apoyo económico de las principales empresas del área (SONDA, IBM, Burroughs, Data General) [28].

En concordancia con sus objetivos, la SCCC recibió solicitudes y apoyó diversos eventos. De hecho, en 1984 patrocinó un Seminario en la U. Católica sobre software de desarrollo (casos LINC y DUNGA), de un Curso en la U. de Chile sobre Unix y C, y de una Escuela de Verano de la U. de Chile que se realizó en Valdivia. [28]

En 1985, la SCCC coordinó los esfuerzos para implementar una red universitaria: "Hoy la red está funcionando en una etapa experimental,

→

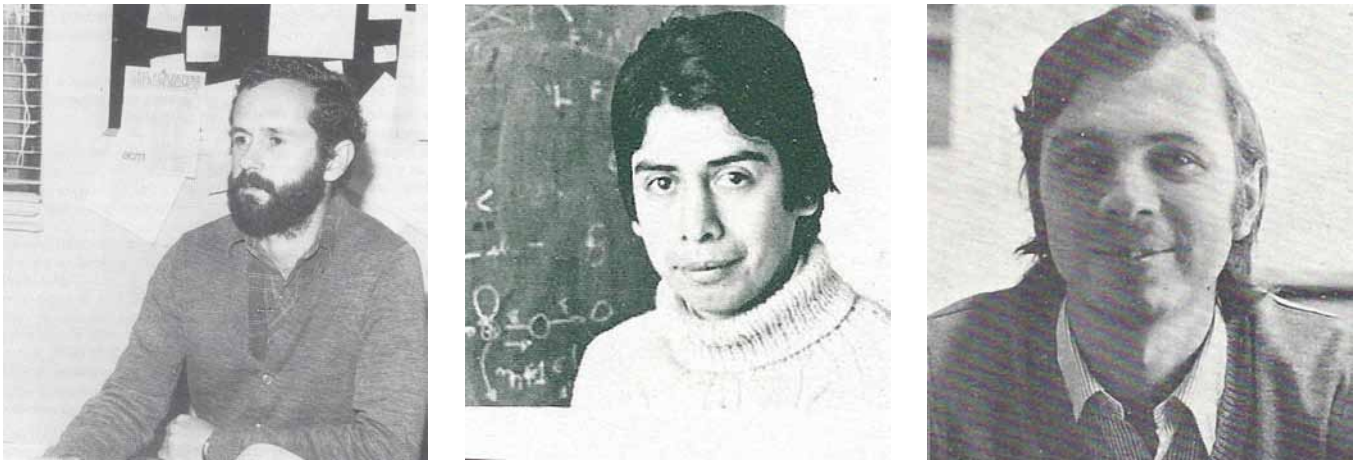


FIGURA 3.
DE IZQUIERDA A DERECHA: PEDRO HEPP (PRESIDENTE), PATRICIO POBLETE (SECRETARIO), EDGARDO KRELL (TESORERO).

conectando equipos que corren el sistema operativo UNIX de las universidades Católica, de Chile y de Santiago (…) En un futuro cercano, la mayoría de las universidades dispondrán de líneas dedicadas al computador, de modo que las llamadas podrán realizarse a cualquier hora (…) También se espera poder extender esta red a universidades de provincia, que ya han mostrado interés, y en el futuro contar con una conexión internacional con USENET" [29].

Respecto de la Conferencia Internacional, se apreció un crecimiento en la cantidad de trabajos chilenos, sumando 10 en las conferencias de 1985 [30] (**Figura 4**) y 1986 [31] en comparación con 10 entre 1981 y 1984 (**Tabla 4**). Consecuentemente, la lista de socios activos aumentó a 31 personas, incorporándose académicos e investigadores con trabajos aceptados en la Conferencia. Lamentablemente, se detectó también una baja considerable de los trabajos de Brasil, atribuida a situaciones internas de ese país relativas al financiamiento de los viajes.

En julio de 1986 se eligió la directiva para el período 1986-1988, quedando conformada por Patricio Poblete (UCH, presidente), Pedro Hepp (UC, secretario), José Pino (UCH, tesorero), Horacio Meléndez (USACH, director), Pedro Osses (UTFSM, director), Yussef Farrán (UdeC, director suplente) [32]. Durante este período, la Conferencia Internacional, que se realizó en conjunto con el Taller de Ingeniería de Sistemas, creció notoriamente en la cantidad de integrantes del comité de programa (9 en 1987 y 10 en 1988), en el número de trabajos presentados y aceptados (19 en 1987 y 28 en 1988) y en la cantidad de países participantes (14 en 1988) (**Tabla 4**).

En 1988, la SCCC registraba 41 socios activos, 19 adherentes y 2 institucionales (SONDA, IBM) [33]. El lento crecimiento, y la sugerencia de varias personas, motivó una modificación de los estatutos que permitió asociarse presentando una solicitud que debía ser aprobada por el directorio, eliminándose la condición de la aceptación de un trabajo en la Conferencia Internacional [34]. En consecuencia, en 1990 la SCCC registraba 72 socios: 57 activos y 15 inactivos [35].

En julio de 1988, la directiva para el período 1988-1990 quedó conformada por José Pino (UCH, presidente), Ignacio Casas (UC, secretario), Jorge Olivos (UCH, tesorero), Jaime Navón (UC, director), Yussef Farrán (UdeC, director), Horacio Meléndez (USACH, director suplente) [36] y se dedicó especialmente a organizar la IX Conferencia para el mes de julio de 1989. Considerando que José Pino era simultáneamente presidente del CLEI, la IX Conferencia se realizó en conjunto con la XV Conferencia Latinoamericana de Informática. Se presentaron 103 trabajos y fueron aceptados 66 de 16 países: 37 de investigación y 29 de aplicaciones [37] [38].

En los primeros años, y de acuerdo a sus objetivos fundacionales, la SCCC estableció relaciones con instituciones profesionales y científicas, internacionales y nacionales. En 1988 la Sociedad ya era miembro de CLEI y tenía representación en IFIP. En el ámbito nacional, la SCCC se incorporó como sociedad afiliada al "Instituto de Injenieros de Chile" [39] e interactuó ocasionalmente con el Colegio de Ingenieros con un anteproyecto sobre acreditación de carreras de Ingeniería en Computación [40] y un informe sobre el desarrollo informático nacional [41].

FIGURA 4.
ACTAS 5TA. CONFERENCIA.

N° Y FECHA	C. ORGANIZADOR	COMITÉ DE PROGRAMA		PAPERS
Primera 1981, 24-27 Agosto Organiza: U. Católica, U. de Chile	Pino, J. Mujica, S. Hernández, R. Peralta, R.	Berry, D. Pereda, A. Lucena, C. Olivos, J.	UCLA UdeC PUCRJ UCH	14 Chile: 5 Brasil: 6 EE.UU.: 2
Segunda 1982, 9-11 Agosto Organiza: U. de Chile, U. Católica	Benguria, J. Pino, J. Cabrera, L. Fuller, D.	Berry, D. Kerchberg, L. Leiss, E. Smallberg, D.	UCLA U. South California U. Houston UCLA	13 Chile: 1 Brasil: 4 EE.UU.: 6
Tercera 1983, 20-23 Junio Organiza: U. Católica	Cabrera, L. Araya, A. Pinto, J.	Leiss, E. Ferrari, D. Munro, I. Fernández, E.	U. Houston U. C.Berkeley U. Waterloo U. Miami	11 Chile: 1 Brasil: 2 EE.UU.: 2
Cuarta 1984, 25-27 Junio Organiza: U. de Chile		Gonnet, G. Chang, E. Leiss, E. Mendelzon, A.	U. Waterloo U. Victoria U. Houston U. Toronto	16 Chile: 3 Brasil: 6 Canadá: 2
Quinta 1985, 15-17 Julio Organiza: U. Católica	Hepp, P. Fuller, D. Eterovic, Y. Straub, P.	Piquer, A. Atkinson, M. Cabrera, L. García-Molina, H. Gonnet, G. Schkolnick	UCH U. Glasgow UC U. Princeton U. Waterloo IBM NY	10 Chile: 6 EE.UU.: 3 Alemania: 1
Sexta 1986, 28-30 Julio Organiza: USACH	Meléndez, H. Cockbaine, J. Contreras, F. Vilches, L. Pino, J.	Gonnet, G. Coffman, E. Flajolet, Ph. Hepp, P. Leiss, E. Nievergelt, J. Olivos, J.	U. Waterloo Bell Labs INRIA UC U. Houston U. North Carolina UCH	16 Chile: 4 EE.UU.: 3 Canadá: 3 Brasil: 1
Séptima 1987, 4-6 Agosto Organiza: U. de Chile (junto con X Taller Ingeniería Sistemas)	Olivos, J. Álvarez, J. Pino, J. Poblete, P.	García-Molina, H. Azorín, E. Bartels, R. Flajolet, Ph. Furtado, A.L. Jensen, D. Munro, J.I. Vidart, J. Vuillemin, J.	U. Princeton UCH U. Waterloo INRIA PUCRJ U. Carnegie Mellon U. Waterloo ESLAI INRIA	19 Chile: 3 EE.UU.: 6 Francia: 3 Brasil: 2 Argentina: 2 Venezuela: 2
Octava 1988, 4-8 Julio Organiza: U. Católica (junto con XI Taller Ingeniería Sistemas)	Casas, I. Aurtenechea, F. Campos, A. Disset, L. Hepp, P. Navón, J. Pinto, J.	Mendelzon, A. Batory, D. Bocca, J. Gonnet, G. Montanari, U. Pazos, J. Pietrasanta, A. Sanz, J. Scolnick, H. Sevcik, K.	U. Toronto U. Texas-Austin ECRC-Alemania U. Waterloo U. P. Madrid U. Pisa IBM Hawthorne IBM Almaden UBA U.Toronto	28 Chile: 1 Brasil: 4 EEUU: 7 Argentina: 4 España: 2 Alemania: 2 (14 países)
Novena 1989, 10-14 Julio Organiza: U. de Chile Con XV Conferencia Latinoamericana de Informática (junto con XII Taller Ingeniería Sistemas)	Poblete, P. Canales, M. Echeverría, L. Olivos, J. Pino, J.	Gonnet, G. Brasky, B. Cabrera, L. Díaz, J. Flajolet, Ph. Marsland, T. Veloso, P. Vidart, J.	U. Waterloo U.C. Berkeley IBM Almadem U.P. Cataluña INRIA U. Alberta PUCRJ ESLAI	66 Chile: 5 Brasil: 13 Argentina: 10 EE.UU.: 8 Francia: 7 Canadá: 5 (16 países)

TABLA 4:
RESUMEN CONFERENCIA INTERNACIONAL (1981-1989).

→

CONCLUSIONES

LA PRIMERA ETAPA DE LA COMPUTACIÓN UNIVERSITARIA (1962-1974) TRANSITÓ, DESDE LOS PRIMEROS COMPUTADORES Y LOS CENTROS DE COMPUTACIÓN, HACIA LAS PRIMERAS CARRERAS, LAS PRIMERAS ORGANIZACIONES Y LOS PRIMEROS CONGRESOS. UNA SEGUNDA ETAPA (1975-1984) COMENZÓ CON LA CREACIÓN DE LOS PRIMEROS DEPARTAMENTOS DE CIENCIA DE LA COMPUTACIÓN QUE DESARROLLARON INVESTIGACIÓN Y DOCENCIA DE PREGRADO (INGENIERÍA) Y DE POSTGRADO (MAGÍSTER). AL FINAL DEL PERÍODO SE ORGANIZARON SEMINARIOS DE INVIERNO Y CONFERENCIAS NACIONALES Y SE CULMINÓ CON LAS CUATRO PRIMERAS EDICIONES DE LA CONFERENCIA INTERNACIONAL.

LA FUNDACIÓN DE LA SOCIEDAD CHILENA DE CIENCIA DE LA COMPUTACIÓN EN 1984 REPRESENTÓ, POR LO TANTO, LA CULMINACIÓN DE UNA TAREA COLECTIVA Y COLABORATIVA QUE COMENZÓ CON LOS INICIOS DE LA COMPUTACIÓN UNIVERSITARIA EN CHILE. SU CREACIÓN ESTÁ EN UNA LÍNEA DE CONTINUIDAD CON LOS ESFUERZOS ANTERIORES, Y AL MISMO TIEMPO REPRESENTÓ UN CAMBIO CUALITATIVO IMPORTANTE: SU PERMANENCIA EN EL TIEMPO LE PERMITIÓ CUMPLIR EL DESEO DE SUS FUNDADORES DE REPRESENTAR NACIONAL E INTERNACIONALMENTE A LA DISCIPLINA. EL PERÍODO INICIAL DE LA SCCC FUERON AÑOS DE FRUCTÍFERO APRENDIZAJE. EN EFECTO, PARA CUMPLIR PLENAMENTE LOS OBJETIVOS EXPRESADOS EN SUS ESTATUTOS, LA SOCIEDAD FUE CAPAZ DE MODIFICAR SU INSTITUCIONALIDAD CON EL PROPÓSITO DE AMPLIARSE Y CONGREGAR A UNA MAYOR CANTIDAD DE ESPECIALISTAS. POR OTRA PARTE, LA SCCC EMPRENDIÓ INICIATIVAS, COMO LA RED UNIVERSITARIA, QUE TERMINARON POR ADOPTARSE GRACIAS A SUS MÉRITOS TÉCNICOS Y ACADÉMICOS.

LA CONFERENCIA INTERNACIONAL, QUE RESULTÓ DE LA EVOLUCIÓN DE SEMINARIOS Y DE CONFERENCIAS NACIONALES, FUE (Y ES) EL EVENTO MÁS IMPORTANTE Y VISIBLE DE LA SCCC. PARA MUCHOS ACADÉMICOS, CHILENOS Y EXTRANJEROS, CONSTITUYÓ UN ESPACIO CONFIABLE Y EXIGENTE PARA PUBLICAR SUS TRABAJOS DE INVESTIGACIÓN. PARA ESTE EFECTO, LA SOCIEDAD SE PREOCUPÓ SIEMPRE DE CONFORMAR LOS COMITÉS DE PROGRAMA CON DESTACADOS INVESTIGADORES EXTRANJEROS Y NACIONALES QUE GARANTIZARON, TANTO LA CALIDAD DE LOS TRABAJOS ACEPTADOS, COMO LA DIFUSIÓN Y RECONOCIMIENTO DEL EVENTO EN CHILE Y EN EL EXTERIOR.

A DIFERENCIA DE LA SITUACIÓN DE OTROS PAÍSES O DE OTRAS ÁREAS, EN CHILE RESULTA DIFÍCIL IDENTIFICAR PERSONAJES CLAVES E IMPRESCINDIBLES EN LOS COMIENZOS DE LA SCCC, Y DE LA DISCIPLINA EN GENERAL. EL NACIMIENTO Y CRECIMIENTO DE LA SOCIEDAD FUE FRUTO DEL TRABAJO CONJUNTO DE MUCHAS PERSONAS, CUYOS NOMBRES SE PUEDEN ENCONTRAR EN LAS DIRECTIVAS, EN LA LISTA DE SOCIOS FUNDADORES, EN LOS COMITÉS ORGANIZADORES Y DE PROGRAMA DE LOS EVENTOS REALIZADOS EN LOS AÑOS PREVIOS Y POSTERIORES A LA FUNDACIÓN.

EN SÍNTESIS, LA SCCC LLEGÓ A SER UN SÍMBOLO DEL ESPÍRITU DE COLABORACIÓN Y DE SERVICIO PÚBLICO DE SUS INTEGRANTES, QUE SUPERÓ LAS PRESIONES AMBIENTALES QUE INCENTIVABAN LA INDIVIDUALIDAD, LA NO COOPERACIÓN Y LA COMPETENCIA. POR EL CONTRARIO, Y EN CONSISTENCIA CON UNA CULTURA Y TRADICIÓN DE TRABAJO CONJUNTO, LA SOCIEDAD REALIZÓ UN VALIOSO APORTE AL PROGRESO, CONSOLIDACIÓN Y RECONOCIMIENTO DE LA DISCIPLINA Y CIENCIA DE LA COMPUTACIÓN EN EL PAÍS Y EN LA REGIÓN. ■

AGRADECIMIENTOS

Si bien la responsabilidad del artículo es exclusiva del autor, varias personas contribuyeron significativamente a su realización. En primer lugar, mis agradecimientos a Patricio Poblete, primer secretario de la SCCC, por confiarme sus carpetas y archivadores de la época. Gracias también a Edgardo Krell, José Pino y José Benguria por las conversaciones que me permitieron completar información que no encontré en fuentes escritas.

Y si de fuentes escritas se trata, mi especial agradecimiento a María Cecilia Cornejo, responsable del archivo de la UTFSM, por permitirme acceder a cartas y documentos de los inicios de la computación en Chile. Gracias también a Jocelyn Symmonds por facilitarme el libro de actas de la SCCC y a Rosa Leal, y los funcionarios de la Biblioteca de la Escuela de Ingeniería, por facilitar mi trabajo. Finalmente, al editor general Pablo Barceló y a todo el Comité Editorial y Periodístico de la Revista Bits, que permiten publicar estudios históricos de nuestra disciplina.

REFERENCIAS

[1] Álvarez, Juan; Gutiérrez, Claudio. "History of Computing in Chile, 1961-1982: Early years, Consolidation and Expansion". IEEE Annals of the History of Computing. Vol 34 n°3. July-September 2012.

[2] Frucht, Roberto. "Carta a Santiago Friedmann Director Centro de Computación U. de Chile". Archivo UTFSM. 17 de enero de 1962.

[3] Frucht, Roberto. "Carta a Julio Hirschmann, Vicerrector Investigaciones UTFSM". Archivo UTFSM. 9 de octubre de 1962.

[4] Friedmann, Santiago. "Carta a Rector de la UTFSM". Archivo UTFSM. 13 de septiembre de 1962.

[5] Frucht, Roberto; Olavarría, Humberto. "Informe sobre las reuniones de consulta para creación de Instituto Chileno de Investigación Operativa y Computación". Archivo UTFSM. 10 de octubre de 1962.

[6] UTFSM. "Scientia". Año xxxvi, N° 132. Enero-junio de 1969.

[7] Riesenkönig, Wolfgang. "Informe sobre el primer Simposio Latinoamericano de Computación". 1967.

[8] Pardo, Mario. "A training programme in electronic computation in a developing country". World Conference on Computer Education". August 1970.

[9] ACUC. "Revista de la Asociación Chilena de Centros Universitarios de Computación". N°1, julio de 1973.

[10] Silva, Fernando. "Formación de recursos humanos en procesamiento de datos". Boletín ECOM, Vol.1 N°7. Enero-marzo 1976.

[11] UCV-Centro de Ciencias de Computación e Información. "Actas II Panel de Discusión sobre tópicos de Computación 1975". Enero 1975.

[12] ECOM. "Actas Tercer Panel de Discusión sobre tópicos de Computación y Expodata". Boletín Vol.1 N°7. Enero-marzo 1976.

[13] Durán, José. "Estado actual y proyecciones de ACTI (Asociación Chilena de Tratamiento de la Información)". Actas II Panel de Discusión sobre tópicos de Computación 1975. Enero 1975.

[14] Álvarez, Juan. "Antecedentes, creación y primeros años del Departamento de Ciencias de la Computación de la Universidad de Chile". Revista Bits de Ciencia N°4. Primer semestre 2010.

[15] Salinas, Luis. "Desarrollo de la Computación en la UTFSM: una mirada retrospectiva muy personal". Revista Bits de Ciencia N°17. Primer semestre 2012.

[16] Eterovic, Yadran. "Treinta años del DCC de la PUC: una visión muy personal". Revista Bits de Ciencia N°8. Segundo semestre 2012.

[17] Farrán, Yussef; Durán, José. "Historia del desarrollo de la Computación en la Universidad de Concepción (1960-1980)". Revista Bits de Ciencia N°9. Segundo semestre 2013.

[18] Acuña, Gonzalo. "El DIINF de la USACH: mucha agua bajo los puentes". Revista Bits de Ciencia N°10. Primer semestre 2014.

[19] Planacap. "Seminario Internacional de Invierno sobre Desarrollo de Software Confiable". 24-28 julio 1978.

[20] Planacap. "II Seminario Internacional de Invierno sobre Ingeniería de Software". 23-26 agosto 1979.

[21] DCC-PUC. "Primer Seminario de Invierno en Ciencia de la Computación". 2-6 agosto 1982.

[22] UCH; UC. Actas "1ª Conferencia Nacional en Teoría de la Computación y desarrollo de software". 23-24 agosto 1979.

[23] UC; UCH. Actas "2ª Conferencia Nacional en Sistemas de Computación". 4-7 agosto 1980.

[24] UC; UCH. "Actas Primera Conferencia Internacional en Ciencia de la Computación". 24-27 agosto 1981.

[25] Pino, José. Editorial: "¿Asociaciones aportas?". Revista Informática. Vol.6 N°5. Julio 1984.

[26] SCCC. "Acta de Sesión N°1/84". Libro de Actas SCCC. Octubre 1984.

[27] Prenafeta, Sergio. "Pedro Hepp Kuschel: La nueva alianza para el progreso". Revista Informática. Vol.6 N°9. Noviembre 1984

[28] SCCC. "Boletín". Año 1 N°1. Noviembre 1984.

[29] SCCC. "Boletín de la SCCC". Año 2 N°1. Julio 1985.

[30] SCCC. Actas "5ª Conferencia Internacional en Ciencia de la Computación". 15-17 Julio 1985.

[31] DII-USACH. Actas "6ª Conferencia de la Sociedad Chilena Ciencia de la Computación". 28-30 Julio 1986.

[32] SCCC. "Acta de la Reunión de Directorio 86/1". Libro de Actas SCCC. 29 julio 1986.

[33] SCCC. "Boletín". Año 5 N°1. Junio 1988.

[34] SCCC. "Acta de la Sesión Extraordinaria de Socios del 7/88". Libro de Actas SCCC. 6 julio 1988.

[35] SCCC. Listado Socios. Abril 1990.

[36] SCCC. "Acta de la Reunión de Directorio del 7/88". Libro de Actas SCCC. 6 julio 1988.

[37] SCCC. Actas "IX Conferencia Internacional en Ciencia de la Computación y XV Conferencia Latinoamericana de Informática". Volumen I – Trabajos de Investigación. 1-14 julio 1989.

[38] SCCC. Actas "IX Conferencia Internacional en Ciencia de la Computación y XV Conferencia Latinoamericana de Informática". Volumen II – Computación Aplicada. 1-14 julio 1989.

[39] SCCC. "Boletín". Junio 1989.

[40] SCCC. Acta de Reunión de Directorio. 2 de mayo de 1988.

[41] SCCC. "Punto de vista de la Sociedad Chilena de Ciencia de la Computación sobre el desarrollo informático nacional". Abril 1990.

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

ESCUELA DE INGENIERÍA INFORMÁTICA. SU HISTORIA: GÉNESIS, DESARROLLO Y REALIDAD ACTUAL

La Universidad Católica de Valparaíso (UCV) inició sus actividades académicas el 25 de marzo de 1928, en el imponente y característico edificio de Avenida Brasil, impartiendo las carreras de Derecho, Ingeniería Química, Ingeniería Eléctrica e Ingeniería Mecánica; siendo la primera universidad de la Región de Valparaíso y la cuarta en el país. Fiel a los principios declarados en su misión y gracias a su incesante labor, al cumplir 75 años, en 2003, recibió un gran reconocimiento de la Santa Sede al conferírsele el título de Pontificia, distinción que sólo poseen 18 universidades en todo el mundo.

ALDOMIGLIARO

Aldo Migliaro participó en la creación del Centro de Ciencias de Computación e Información durante 1970. Dicho Centro de Estudios es ahora la Escuela de Ingeniería Informática de la PUCV. Tuvo a su cargo la organización de los Panel'74-79, hoy transformado en la Conferencia Latinoamericana de Informática de CLEI. Fue organizador del Centro Latinoamericano de Estudios en Informática, CLEI, el año 1979, siendo su primer Secretario Ejecutivo. Es miembro Honorario de CLEI desde 1992. Es además, socio creador de la empresa Informática Ingeniería de Software Ltda, ISL (1979). Ha sido Director y Profesor Titular de la Escuela de Ingeniería Informática, PUCV. Actualmente se desempeña como Gerente de A&F de ISL y como Profesor Extraordinario de la Escuela de Ingeniería Informática, PUCV.

amigliar@ingsoft.cl

Hoy la Pontificia Universidad Católica de Valparaíso (PUCV) tiene 9 Facultades, las que en total ofrecen más de 100 programas, considerando carreras de pregrado, postítulos, magísteres y doctorados, cubriendo casi la totalidad de las áreas del conocimiento; para ello se cuenta con16 sedes distribuidas en la Región de Valparaíso.

En 1970 en nuestra Universidad no existía una disciplina académica dedicada a la informática, pero la PUCV sí tenía en su área administrativa equipos mecánicos de registro, para el pago de las remuneraciones y los controles contables.

A continuación se presentará una síntesis de la historia de la informática en la Pontificia Universidad Católica de Valparaíso, incluyendo la idea inicial, su desarrollo, y finalizando con la consolidación y la concreción de las carreras y los programas de postgrado. Para comenzar esta historia es necesario retrotraerse a los últimos 40 años.

LA IDEA (1970) Y LA CREACIÓN

La idea se origina con dos profesores de nuestra Universidad, uno perteneciente a la Escuela de Ingeniería Química y el otro que pertenecía a la Oficina de Estudios y Planificación, de la administración central de la Universidad. Ambos impulsados por sus respectivos departamentos participaron en el primer curso de Análisis de Sistemas que impartía la Empresa Nacional de Computación - ECOM, durante el segundo semestre de 1970. Ésta fue la oportunidad en que los profesores de la PUCV, Aldo Migliaro y Amílcar Morales se conocieron, y supieron que venían de la misma casa de estudios.

Al finalizar el curso, ambos docentes visualizaron la idea de formar un ente al interior de la Universidad, primeramente se pensó en una unidad académica y luego en un Centro para la PUCV, orientado al estudio de la informática y la información, como resultado de su aplicación en las distintas áreas del conocimiento. La Universidad, a través de sus unidades académicas con las disciplinas impartidas en ellas, contaba con gran parte del conocimiento en la aplicación, además el Centro cumpliría con la característica de ser transversal a todas estas disciplinas.

→

Así se hacía necesario presentar un proyecto a las autoridades de la PUCV, cuyo Rector era el Profesor Raúl Allard, y posteriormente al Senado Académico, que en ese entonces equivalía al actual Consejo Superior de la Universidad, y estaba compuesto por los decanos y académicos elegidos, su presidente era el Profesor Reinhart Zorn.

Durante 1971 se iniciaron los estudios requeridos que necesitaba el proyecto para satisfacer la discusión al interior de la Universidad, y de esta forma definir la informática como una disciplina académica. Dado que la PUCV no contaba con ningún ente académico que se preocupara por esta disciplina, y culturalmente no estaba dentro de las inquietudes de la Universidad, el proyecto propuso que en el Centro se desarrollara la informática, como área académica al interior de la PUCV, a través de la formación, la investigación y la docencia hacia las otras unidades académicas de la Universidad.

Largas discusiones se desarrollaron en el Senado Académico en los años 1971/72, especialmente por el desconocimiento del tema en lo conceptual académico. Inicialmente dentro de esta discusión se habló que esto equivalía a una escuela de administrador de máquinas, e incluso a conformar un club de aficionados. Pero gracias a la discusión, se llegó al concepto de la información, el cual era la base central del proyecto.

Los autores del proyecto, se dieron cuenta que no era suficiente la definición operativa que argumentaba el proyecto, sino que en una institución como la Universidad, en ese entonces, se debía llegar al convencimiento teórico que esto era una disciplina del conocimiento. Se contaba con la aprobación del Rector, del presidente del Senado Académico, de la Escuela y el Departamento que lo auspiciaban, además de varios decanos que apoyaban la idea. A altas horas de la noche del 2 de febrero de 1972, se sometió a votación y se aprobó el proyecto, quedando por definir en la rectoría los procedimientos correspondientes.

Finalmente el proyecto fue aprobado por Decreto de Rectoría Nº 558/72 creando el **Centro de Ciencias de Computación e Información (CCCI)** de la Universidad Católica de Valparaíso, en el cual se debería desarrollar la investigación y la aplicación de esta nueva ciencia, con un quehacer que fuese transversal a las áreas del conocimiento. En la docencia, solamente se permitió la enseñanza de esta disciplina hacia las otras escuelas de la Universidad.

Para estos efectos se definió una planta de profesores y recursos, su dirección correspondió a una Comisión formada por un representante de la Escuela de Electrónica, del Instituto de Matemáticas y las dos personas que propusieron el proyecto. Su lugar de operación era un espacio pequeño de un edificio de bodegas, adquirido por la Universidad en esa fecha, que corresponde al Centro Universitario Rafael Ariztía (CURA), vecino a este edificio hoy se encuentra la Facultad de Ingeniería (FIN).

LA INVESTIGACIÓN Y DESARROLLO DE INFRAESTRUCTURA

Al inicio, sólo se contaba con el entusiasmo y esfuerzo de quienes pensaron la idea. Con posterioridad se contrató a otro profesor y una secretaria. También se hace necesario mencionar el esfuerzo entregado por los alumnos de los servicios académicos que prestaba el CCCI hacia las distintas carreras. Muchas tareas no hubieran sido posibles sin su entusiasmo.

Inicialmente no había computadores, pero se contaba con la colaboración de la Universidad Técnica Federico Santa María (UTFSM), en su equipo IBM 1620 (8 KB) se procesaban algunos programas. También se contaba con el equipamiento de ENAMI Ventanas, empresa que colaboraba facilitando el computador IBM 1130 en la noche, desde las 08:00 PM a las 08:00 AM. Adicionalmente, IBM instaló en las dependencias del CCCI un Terminal APL conectado a sus servidores en Santiago. Con estos recursos el Centro se dedicó a la investigación, a conocer las bases que sustentaban la Ciencia de la Informática, con proyectos que iban desde la enseñanza de la informática, hasta aplicaciones en sistemas de información.

Las ideas traídas por un profesor desde conferencias en Europa permitieron al CCCI iniciar el desarrollo y la aplicación de la Programación Estructurada, tanto en su uso, como en la enseñanza, permitiendo solucionar problemas que cada vez eran más grandes y complejos de resolver. Para aquellos lenguajes que no soportaban la programación estructurada, se hicieron preprocesadores como CVLOGO y CVFORTRAN que utilizaban dichos conceptos, y de esta forma se enseñaba lenguajes que sí la sustentaban. En este punto el CCCI fue pionero en Chile en el campo de la programación estructurada, muchas críticas se recibieron al respecto, pero finalmente primó esta nueva tecnología.

En paralelo se desarrollaban proyectos de búsqueda de información aplicada a bibliotecas y otras investigaciones sobre educación y lenguajes. En septiembre de 1973 se incorporó un profesor francés, Michel Rozay, que venía en calidad de profesor visitante a través de una Agencia de Ayuda, quien estuvo varios años en el Centro, siendo su principal interés de investigación los sistemas operativos.

En esos años, la situación de la Universidad fue compleja con el cambio de rectores civiles por miembros de la Armada de Chile. El Rector designado, eliminó la Comisión que administraba el CCCI y designó al profesor Migliaro como Director, y al profesor Morales como Jefe de Investigación.

En las investigaciones que se llevaban a cabo, ya se visualizaba el concepto de búsqueda de archivos a través de índices, pero no existían las tecnologías aplicadas orientadas a la base de datos. Uno de los proyectos de esa época fue METASYS, que era una base de datos en modelo de redes, investigación que se desarrollaba en el CCCI. Si bien se realizaba investigación, faltaban las instancias de interactuar con referentes

que emitiesen sus opiniones, siendo necesaria la participación y discusión de otros pares académicos y expertos. Esto fue un factor importante en lo que vendría después, referente a las Conferencias.

En los años siguientes el CCCI seguía con su desarrollo de investigaciones y difusión hacia las distintas unidades académicas de la PUCV, pero faltaba la disponibilidad de hardware adecuado. Es así como se creó el CIREC (Centro Interuniversitario Regional de Computación) entre las tres universidades regionales de esa época (PUCV, UTFSM y U. de Chile-Valparaíso), cuyo objetivo era adquirir un gran computador central de modo de utilizar terminales en las respectivas universidades. En este proyecto participó activamente el CCCI. Era un buen proyecto, dada la escasez de recursos de las universidades, y la necesidad de recursos computacionales de los académicos, pero lamentablemente éste fracasó.

Por esta situación, el CCCI en 1975 propuso a la Rectoría de la PUCV el arriendo de un IBM-370, con la opción de compra, esto para su rápida disposición y uso. De esta manera fue posible dar apoyo a las investigaciones del CCCI, a los alumnos de los servicios que se dictaban, a los profesores de todas las áreas de la PUCV y también como apoyo a la administración central de la Universidad.

Por otra parte, la PUCV creó un Servicio de Procesamiento de Datos, lo cual permitió que sus investigadores utilizaran esta herramienta en proyectos y docencia, además de su uso en la administración de la casa de estudios. Esto significó un gran paso para el CCCI, ya que de esta manera disponía de equipamiento en el cual se desarrollaban sus investigaciones. Se pudo concretar el proyecto METASYS, investigación orientada al desarrollo de un administrador de base de datos. Además, hubo nuevas contrataciones de profesores, se recibieron varios profesores visitantes (dados los contactos que se tenían a través de las Conferencias que se desarrollaron desde 1974), así como también la participación en cursos internacionales y el acceso a planes de postgrado en universidades de EE.UU.

LA CONFERENCIA PANEL Y CLEI

La necesidad de evaluar las investigaciones que se llevaban a cabo en el CCCI de la PUCV, implicaba que se requería la interacción con pares, referentes y expertos con el propósito de establecer una instancia de comunicación y debate para el desarrollo de esta tecnología. Éste fue el detonante que originó la idea de planificar una conferencia de informática en la PUCV. Es así que el Centro de Ciencias de Computación e Información de la PUCV, convoca a una conferencia de académicos y especialistas de empresas, especialmente de la Región, invitación extendida a las universidades nacionales.

La primera Conferencia se denominó Panel de Discusión de Tópicos de la Computación y se celebró en enero de 1974, siendo invitada el área de Computación de la UTFSM, y de otras universidades nacionales, además de expertos de las empresas locales. Dado el éxito que obtuvo esta primera Conferencia, el CCCI decidió convocar anualmente esta reunión, durante enero de cada año. El evento se denominó "**Panel de Tópicos de Computación**", y se conoció como PANEL (**Imagen 1**).

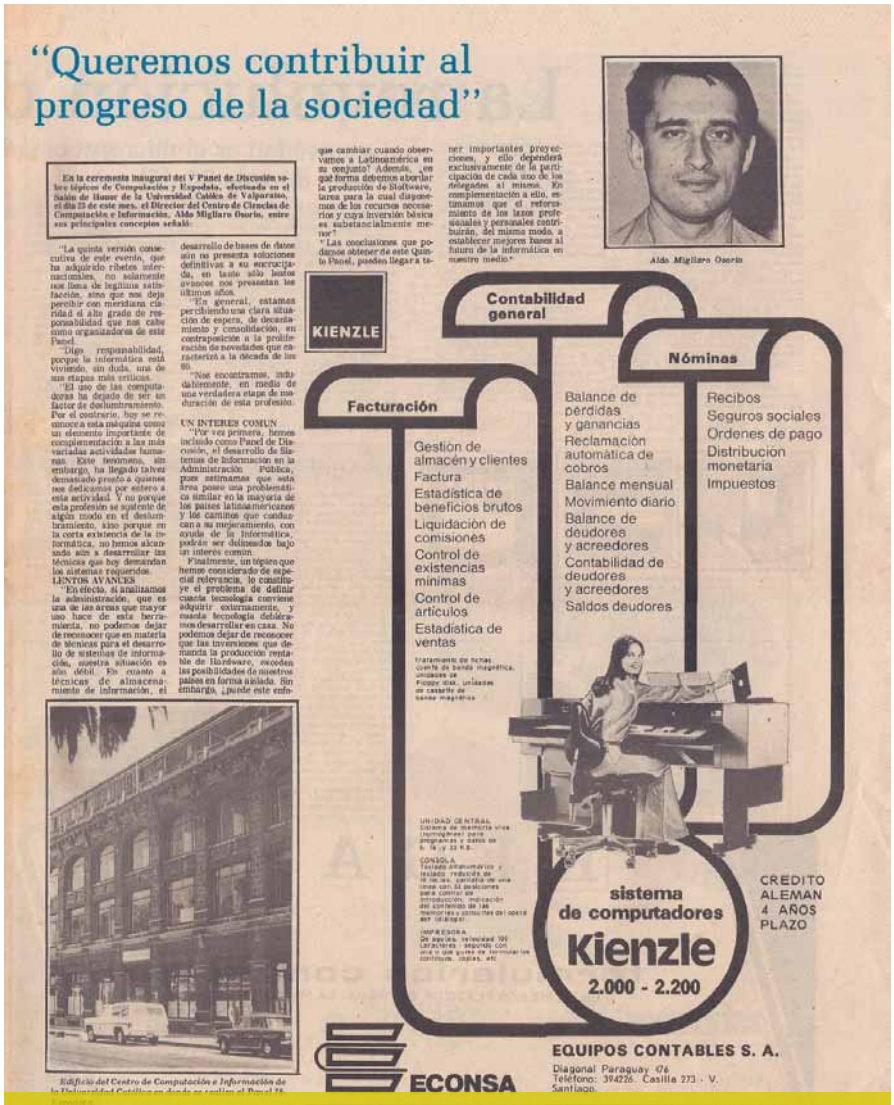


IMAGEN 1.
PERIODICO PANEL '78: PRESENTACIÓN DE UN DIARIO REGIONAL REFERENTE A LA EXPOSICIÓN DE LOS AÑOS '70-'80.

Por el alto costo que implicaba la organización de la Conferencia, se visualizó incrementar el contacto con las empresas de informática del país, y se recurrió a una fórmula de financiamiento conjunto para esta actividad. Así se decidió por la organización de una exposición de equipamiento computacional y software, denominándose la siguiente conferencia como PANEL-EXPO'75, denominación que se usó cada año venidero, hasta el PANEL-EXPO'79. Esta exposición implicó reacondicionar áreas de las antiguas bodegas del edificio, en las cuales las empresas hacían su exposición de hardware y software, quedando los espacios absolutamente habilitados para la academia, especialmente con salas de clases. Cada año, durante enero, se desarrollaba la Conferencia, y esta actividad crecía tanto en participantes como presentación de papers. También se agregaron discusiones sobre cada tema nuevo, venían participantes tanto de Latinoamérica como de EE.UU. y Europa; ya era una conferencia internacional.

En 1979 en que se celebró el PANEL-EXPO'79, y la participación extrajera duplicaba la participación nacional, fue el instante en que se acuerda la definición de las bases para la fundación del **Centro Latinoamericano de Estudios en Informática - CLEI**, creando este ente con una absoluta independencia de cualquier organización externa, que no fuesen las universidades latinoamericanas y asociaciones nacionales de informática de países latinoamericanos. CLEI se comprometía a hacer de PANEL una Conferencia Itinerante a través los países latinoamericanos, la cual se reconocería como **Conferencia Latinoamericana de Informática**, para el año siguiente sería CLEI'80 y así sucesivamente para los años venideros (**Imagen 2**).

En dicho momento la rectoría de la PUCV realizó un reconocimiento a los profesores del CCCI Amílcar Morales y Aldo Migliaro, así como la valiosa colaboración de los profesores Sr. Jorge Baralt de la Universidad Simón Bolívar (USB), Venezuela, y del Sr. Carlos José Pereira de Lucena de la Pontificia Universidad Católica de Río de Janeiro (PUC-RJ), Brasil. CLEI desde sus inicios utiliza como logo el del CCCI.

IMAGEN 2.
SALA DE LA ESCUELA DE INGENIERÍA INFORMÁTICA DE LA PUCV CON PÓSTERS DE LAS CONFERENCIAS ANTIGUAS.

El documento fundacional del CLEI, en 1979, en parte establece:

CONSIDERANDO:

Que el advenimiento de las computadoras y la sistematización del tratamiento de los datos han dado origen a la Ciencia de Computación e Informática.

Que los países latinoamericanos tienen ya instalados un importante parque computacional, contando con numerosas instituciones de enseñanza e investigación en informática y que han comenzado a producir sistemas de procesamiento de datos con tecnología propia.

Que el desarrollo de la enseñanza e investigación científica en el dominio de la informática constituye una base importante para el progreso económico y social de las naciones.

Que las computadoras han provocado un profundo efecto en la sociedad, con impacto en las costumbres y la cultura.

Que la Pontificia Universidad Católica de Valparaíso, ha venido realizando con éxito un congreso científico en los últimos seis años, al cual han concurrido docentes, investigadores, profesionales y estudiantes, nacionales y de países latinoamericanos en número cada vez mayor, y la necesidad de garantizar la continuidad de dicho certamen anual, así como promover acciones que permitan mejorar la comunicación entre dicho personal.

ACUERDAN:

Constituir el CENTRO LATINOAMERICANO DE ESTUDIOS EN INFORMÁTICA - CLEI, cuyo objetivo será promover el desarrollo de la informática en Latinoamérica a través del intercambio científico, técnico, y educacional entre los miembros participantes, así como estudiar los efectos sobre la sociedad.

En 1980 la conferencia CLEI'80 se desarrolló en Caracas, Venezuela, en la Universidad Simón Bolívar, posteriormente CLEI'81 en Buenos Aires, Argentina, y CLEI'82, en Lima, Perú. En 1983 se presentó un problema, lo que implicó que durante ese año no hubo Conferencia. CLEI aún no estaba consolidado, por lo que se corría el riesgo de perder todo el esfuerzo realizado hasta ese instante.

El hecho de que el profesor Migliaro se mantuviera como Secretario Ejecutivo de CLEI, y la disposición de la PUCV en cuanto a resolver esta situación, por el compromiso que había adquirido la Universidad con la informática latinoamericana, conllevó a otorgar su apoyo para que la Escuela de Ingeniería Informática de la PUCV llevara a cabo dicha convocatoria, realizándose la X Conferencia Latinoamericana de Informática durante abril de 1984, en la ciudad de Viña del Mar.

En CLEI'84 se contó con la asistencia de mil especialistas de las distintas universidades latinoamericanas, además con la presentación aproximadamente de 120 papers. Así mismo, varias mesas redondas, destacándose las de "La Informática un Factor de Desarrollo o de Dependencia para Latinoamérica: un enfoque prospectivo hacia el año 2000" y "Seminario de Informática Jurídica". A esta Conferencia vino como observador el profesor S. Narasimham, representante de IFIP - Federación Internacional de la Informática en Países de Desarrollo. Como conclusión después de ver los resultados de la Conferencia, propuso que la Conferencia de CLEI se incorporara como representante ante IFIP en su área de Países en Desarrollo.

La Conferencia Latinoamericana de CLEI, se ha potenciado, desde esa X Conferencia, de modo que se han desarrollado anualmente con un nivel que ha ido en aumento, tanto en cantidad como en calidad de las presentaciones de los temas. Ya prácticamente se ha desarrollado en casi todos los países latinoamericanos. Es así como el presente año se ha celebrado la XL Conferencia Latinoamericana de Informática CLEI'2014 en Montevideo, Uruguay.

CARRERA DE INGENIERÍA DE EJECUCIÓN INFORMÁTICA

En la década de los ochenta el gran objetivo del Centro de Ciencias de la Computación e Información - CCCI, era en ese entonces la formación de profesionales en el área. Un logro que había sido planteado desde sus inicios y no se había concretado por las funciones que se habían desarrollado hasta ese instante. Se mantenía una gran dedicación a la investigación y los servicios académicos hacia las otras unidades académicas, cubriendo prácticamente todas las carreras de la Universidad. Además la interacción con la comunidad universitaria tanto a nivel nacional como internacional, y asimismo las actividades del Secretario Ejecutivo del CLEI en cuanto a la participación en los encuentros anuales.

La misión era transformar al CCCI en una unidad académica dedicada a formar ingenieros en informática. Es así como primeramente la PUCV a solicitud del CCCI, por Decreto de Rectoría Orgánico Nº 134 del 2 de junio de 1981 incorpora al Centro de Ciencias de Computación e Información a la Facultad de Ingeniería de la PUCV. Luego el Acuerdo N° 12/82 del Consejo Superior de la Universidad, establece transformar el Centro de Ciencias de la Computación e Información en la Escuela de Ingeniería Informática; lo que se materializa con la promulgación del Decreto de Rectoría Orgánico Nº 160 del 30 de septiembre de 1982, que crea la unidad académica citada en la Facultad de Ingeniería, la cual debe asumir las funciones del Centro de Ciencias de Computación e Información CCCI, esto es principalmente investigación y docencia en servicio a otras carreras y, además, debe enfrentar la formación de pre y postgrado en el área.

Se hacen los estudios y las proposiciones correspondientes, analizando y aprobando el proyecto de formación por parte de las autoridades académicas, a nivel de Escuela, Facultad y Rectoría. Esta proposición correspondía a un Ingeniero Civil Informático de seis años, para el cual se definió su perfil, malla académica y programas, orientados a su formación. En aquella época las autoridades nacionales visaban cualquier nueva carrera que se abriera en el país por parte de las universidades, este proceso a nivel nacional se complementaba con la aprobación del programa propuesto por comisiones de pares de otras universidades que impartían la carrera correspondiente. Fue en ese instante que el Ministerio de Educación autorizó a la PUCV a dictar, no lo solicitado, sino que la carrera de Ingeniería de Ejecución en Informática con una duración de cuatro años; la decisión fue proseguir con el proyecto, readecuando la malla curricular al nuevo perfil.

Se crea la carrera de Ingeniería de Ejecución en Informática por Decreto de Rectoría Académico Nº 62/82 del 30 de septiembre de 1982. Se inició un plan para enfrentar los nuevos desafíos, lo cual consideraba la contratación de profesores, temas de espacios físicos e infraestructura. Con posterioridad la carrera de Ingeniería de Ejecución en Informática de la PUCV, recibe la aprobación requerida por la Universidades de Chile (UCH) y Pontificia Universidad Católica de Chile (PUC). La carrera se inició en marzo de 1985 con el ingreso de la primera cohorte de alumnos para obtener el título de Ingeniero de Ejecución en Informática.

El cuerpo de profesores debía enfrentar y resolver los requerimientos de una unidad académica como se definió a través de los decretos, esto es dedicados a la docencia, investigación y extensión. Lo anterior se fortalece con el contacto establecido a través de CLEI, lo cual permitió tener profesores visitantes, y así poder revisar los programas, organizar seminarios y otras actividades académicas.

Se disponía de equipamiento adecuado que estaba compuesto por un IBM de la serie 4300, equipos AIX RISC 6000, IBM S/34 (este equipo se consigue con IBM por la transformación de software de hospitales), IBM 5110, y el Zenith (computador personal). A través del equipo IBM S/34 se proporcionó, entre otros, las facilidades interactivas en la formación de nuestros ingenieros, tanto en su enseñanza, como para el desarrollo de sus proyectos, y las aplicaciones e implementación de investigaciones.

CARRERA DE INGENIERÍA CIVIL INFORMÁTICA

Ya en la década de los noventa, un objetivo que se había quedado en el camino, truncado por decisiones externas, era el de formar un Ingeniero Civil en Informática: situación adicionalmente afectada por la excesiva carga académica de los profesores, aquejados por el éxodo de profesores y la difícil contratación de sus reemplazos. Pero esta situación se logró revertir, y el año '96 se desarrolló un nuevo proyecto que presentaba la formación de dichos profesionales. Este proyecto no fue difícil de elaborar, ya que se tenía la experiencia del previo. Dicho proyecto define el perfil y el respectivo plan de estudios, éste se presenta a la Facultad y luego a Rectoría. Es así, que por Decreto de Rectoría Académico Nº 139/96 se aprueba la carrera conducente al título profesional de Ingeniero Civil Informático y grado de Licenciado en Ciencias de la Ingeniería, para alumnos ingresados a partir de marzo de 1997.

La situación del crecimiento del número de alumnos en la unidad académica y de los proyectos de títulos tanto de Ingeniería de Ejecución en Informática, como de los nuevos Ingenieros Civiles en Informática, debilitaron el área de investigación. En vista de lo anterior, hacia fines de la década y principio del año 2000, se orientó una parte de la investigación hacia el área de Comercio Electrónico; en aquella época recién se comenzaba a visualizar aplicaciones gracias a la disponibilidad de Internet. Para estos efectos se instaló un laboratorio de Comercio Electrónico, el cual estaba constituido por un equipo IBM i-SERIES, y además se contaba con la colaboración de una empresa de desarrollo de software - Ingeniería de Software Ltda. (ISL), que proporcionaba ideas, requerimientos y los contactó con empresas que potencialmente se podrían interesar en tales soluciones. Lo anterior se formalizó mediante un acuerdo entre PUCV, IBM e ISL, lo que daba la posibilidad de que los alumnos de la Escuela pudieran adentrarse en problemas reales y sus soluciones, en un área aún desconocida por la industria. Esto tuvo mucho éxito durante cuatro años, desarrollándose más de veinte proyectos de título, y facilitando la inserción laboral de los alumnos que trabajaron en estos proyectos.

POSTGRADO E INFRAESTRUCTURA

En la era del 2000, la Escuela visualizó que debía orientarse al postgrado, y para esto se necesitaba crecer en espacio físico, mejorar la infraestructura y crear nuevos laboratorios de especialidad, además de fortalecer su cuerpo académico con doctores en el área. A inicios del 2000, con un proyecto MECESUP de la Facultad de Ingeniería, se inició la construcción del nuevo Edificio Isabel Brown Caces (IBC) en la Avenida Brasil, contiguo a la Facultad de Ingeniería, en el cual la Escuela de Ingeniería Informática tendría aproximadamente el 50% de la nueva infraestructura, con lo cual se creció en casi cuatro veces el espacio que disponía en la anterior sede (CURA).

IMAGEN 3.
UNA DE LAS SALAS DE TRABAJO DEL PROGRAMA DE MAGÍSTER DE LA ESCUELA DE INGENIERÍA INFORMÁTICA DE LA PUCV.

En la actualidad la Escuela de Ingeniería Informática dispone de dos pisos en el edificio IBC, lo cual equivale aproximadamente a 2.000 m2; contando con salas de clases, un aula media, laboratorios, tanto de uso masivo como de especialidad, áreas administrativas, oficinas de profesores, sala de estudio de los alumnos, áreas de postgrado (**Imagen 3**), oficinas para proyectos especiales, y salas de reuniones.

Ya con el espacio disponible en el nuevo edificio, se continuó con el fortalecimiento del cuerpo académico, lo que fue reforzado con un nuevo proyecto MECESUP, el cual contemplaba la contratación de nuevos profesores con el grado de Doctor. Sumado a lo anterior y con la contribución de profesores visitantes, se estableció el diseño de los programas de postgrado. Primeramente el objetivo fue crear un programa conducente al grado de Magíster en Ingeniería Informática, y luego se consideró un proyecto para el programa conducente al grado de Doctor en Ingeniería Informática.

Con todo el avance hecho para alcanzar tales objetivos, la Escuela de Ingeniería Informática pone en marcha su Programa de Magíster. Por Decreto de Rectoría Nº 78/2005, se crea el grado académico de Magíster en Ingeniería Informática. Este programa se inició en agosto de 2006, con una muy buena participación, especialmente de la Región, de Santiago y de países latinoamericanos. Sus resultados han sido excelentes, a la fecha se cuenta ya con más de 80 graduados, y actualmente participan alrededor de 50 alumnos en el Programa.

Nuevamente un proyecto MECESUP permitió continuar el camino trazado, en esta oportunidad apoyando el estudio preliminar para el programa del Doctorado en Ingeniería Informática; lo cual se sustenta con las líneas de investigación de los académicos, así como con su productividad, y redes de contacto tanto a nivel nacional como internacional. Es así como se presentó el proyecto, primeramente a la Facultad de Ingeniería y luego a Rectoría de la PUCV, quien después del estudio y análisis, decide que por Decreto Académico Nº 40/2011, se cree el grado académico de Doctor en Ingeniería Informática. En agosto de 2011, se inicia el Programa. Éste comienza con cuatro alumnos, en la actualidad se cuenta con un total de ocho alumnos, de los cuales cinco son tesistas.

ACTUALIDAD

En síntesis, la Escuela de Ingeniería Informática cuenta la fecha con la siguiente realidad en el plano académico y de infraestructura:

- *Ingeniería de Ejecución en Informática: cuenta con un promedio de 300 alumnos, y con más 950 titulados que trabajan en el área. Carrera acreditada por 5 años.*
- *Ingeniería Civil en Informática: además otorga el Grado de Licenciado en Ciencias de la Ingeniería. Cuenta con un promedio de 340 alumnos, y con más de 200 titulados que trabajan en el área. Carrera acreditada por 5 años.*
- *Magíster en Ingeniería Informática: cuenta con aproximadamente 50 alumnos y 10 tesistas, y a la fecha tiene más de 80 graduados.*
- *Doctorado en Ingeniería Informática: cuenta con 8 estudiantes, siendo 5 de ellos tesistas candidatos a Doctor.*

La investigación al interior de la Escuela se enmarca principalmente en las siguientes líneas: Inteligencia Computacional e Ingeniería de Software. Ésta se encuentra respaldada por un importante número de publicaciones indexadas (ISI, Scopus y otras), alcanzando el segundo lugar al interior de la Facultad de Ingeniería de la PUCV. En la actualidad la Escuela cuenta con cuatro proyectos FONDECYT en desarrollo. En cuanto a infraestructura, además de los espacios detallados anteriormente, se dispone de los siguientes laboratorios de especialidad: Inteligencia Computacional y Optimización, Usabilidad y Robótica. ■

BIG DATA: UNA PEQUEÑA INTRODUCCIÓN

AIDAN HOGAN

Profesor Asistente Departamento de Ciencias de la Computación, Universidad de Chile. Ph.D. Computer Science, National University of Ireland, Galway; Licenciado en Ingeniería Electrónica, National University of Ireland, Galway. Líneas de Investigación: Web Semántica, Procesamiento de Datos a Gran Escala, Integración de Datos, Sistemas Distribuidos, Minería de Datos.

ahogan@dcc.uchile.cl

EL VALOR DE LOS DATOS

Soho, Londres, agosto 1854: siete años antes del descubrimiento de los gérmenes por Louis Pasteur, la gente moría por centenares de una enfermedad misteriosa llamada cólera. La sabiduría de la época decía que el cólera era causado por el miasma: algo malo en el aire, una niebla espesa que hacía que la enfermedad naturalmente se acumulara en zonas densamente pobladas. Pero a John Snow, un médico que trabajaba en Londres, no le parecía que esta teoría fuera creíble, por lo que se dispuso a encontrar una mejor.

Snow comenzó encuestando a los aquejados por el cólera en el área, marcando en un diario sus nombres, género, edad, ubicación, fecha de inicio de la enfermedad, fecha de muerte, los hábitos cotidianos, y así sucesivamente, aplicando diversas técnicas estadísticas y analíticas sobre los datos. A partir de su estudio, trazó todos los casos de cólera del Soho, como se muestra en la **Figura 1**. Cada rectángulo oscuro indica un caso de cólera en ese lugar y cada pila representa un hogar o un lugar de trabajo. Usando un diagrama de Voronoi, se hizo evidente que los casos de cólera se agrupaban alrededor de la bomba de agua en la intersección de Broad Street y Cambridge Street; las personas que vivían cerca de otra bomba no se hallaban afectadas. Snow convenció a las autoridades de quitar el mango de la bomba. Los nuevos casos de cólera cesaron.

A pesar de que los microscopios de la época no podían ver la causa física de la enfermedad nadando en el agua, 616 muertes y 8 días más tarde, los datos de Snow habían encontrado la causa: un pozo de agua abierto cavado cerca de un pozo ciego de aguas residuales. Éste fue un hallazgo revolucionario: el cólera no era algo en el aire, sino más bien algo en el agua.

Como científicos de la Computación, a veces olvidamos el valor de los datos. Al igual que un cerrajero podría considerar que las llaves son pedazos de metal que tienen que ser cortados y vendidos, tendemos a considerar los datos como algo que tiene que ser almacenado o analizado, algo que se mueve desde la entrada hasta la salida: compuesto de bytes en una máquina, o terminales en una gramática. Es sólo en el contexto de la importancia de los datos para otros campos –y la capacidad para producir conocimiento desde ellos– que el reciente e inusual ruido alrededor del término Big Data se puede entender.

John Snow comprendió el valor de los datos. Su trabajo en Soho, 1854, estableció el precedente para el campo de la epidemiología: el análisis de los datos para extraer patrones de causalidad sobre enfermedades en zonas pobladas, que abarca los controles de salud pública, los ensayos clínicos, la causa y la propagación de enfermedades infecciosas, la investigación y simulación de brotes, y así sucesivamente. Los casos de éxito de la epidemiología incluyen, entre otros, la erradicación de la viruela, el aislamiento de la poliomielitis a zonas localizadas, y una marcada reducción de los casos de malaria y cólera.

→

Por supuesto, el valor de los datos no se aprecia sólo en el campo de la epidemiología o la genética, o la medicina, o la astronomía observacional, o la física experimental, o la climatología, o la oceanografía, o la geología, o la ecología, o la sociología, o incluso la ciencia o la empresa. Los datos son los protagonistas en muchos de los aspectos científicos, comerciales y sociales de la vida. Al igual que en el trabajo de Snow, existen metodologías comunes a todos los campos que laboran con datos: la recolección de ellos, su curación, la generación de hipótesis, las pruebas estadísticas, la visualización, etc. A diferencia de los tiempos en que Snow trabajaba, los computadores permiten hoy en día recoger, gestionar y procesar los datos con niveles de escalabilidad y eficiencia que Snow no podría haber imaginado.

Pero aún así, parece que a medida que la capacidad de la sociedad para capturar más y más datos sobre el mundo que nos rodea continúa creciendo, las técnicas computacionales convencionales no son suficientes para darle sentido al resultado.

BIG DATA

Big Data, en su esencia, es una idea interdisciplinaria: tener más datos de lo que es posible para dar sentido. Big Data es un llamado a nosotros, los científicos de la Computación, para ofrecer una vez más aún mejores métodos para digerir datos aún más diversos, más complejos, más dinámicos, más granulares y más grandes.

No es difícil entender por qué tantos científicos de la Computación se sienten (en voz baja) desdeñosos del zumbido del término "Big Data". En nuestro campo, los temas fundamentales – como Bases de Datos, Lógica, la Web, Ingeniería de Software, *Machine Learning*, etc. – se fundan en tecnologías con un rico *pedigree*. Por el contrario, "Big Data" es una idea difícil de definir. Sin embargo, no es fácil permanecer totalmente indiferente una vez que uno ve titulares como los U$200 millones de inversión por parte del Gobierno de Estados Unidos en una serie de proyectos nacionales llamados "Big Data Initiative", o la inversión de 30 millones de libras –por parte del gobierno del Reino Unido y un filántropo privado– para crear in "Big Data Institute" en Oxford en temas de epidemiología y descubrimiento de drogas, u otras historias similares en las noticias.

Entonces, ¿qué es Big Data? La definición más canónica (pero aún bastante inescrutable) de Big Data hace referencia a cualquier escenario de uso intensivo de datos, donde el volumen, la velocidad y/o la variedad de los datos dificulta la utilización de "técnicas tradicionales de gestión". Este desafío es conocido como el de "las tres V's", y la mayor parte del énfasis hasta ahora se ha concentrado en el tema del volumen de los datos y (en menor medida) en su velocidad.

FIGURA 1.
PARTE DEL MAPA DE SNOW DE CASOS DE CÓLERA EN SOHO, 1854.

NUEVAS RAZAS DE BASES DE DATOS

La respuesta tradicional a trabajar con grandes cantidades de datos estructurados siempre ha sido simple: el uso de una base de datos relacional. Si el volumen o la velocidad de los datos impedían el uso de una base de datos relacional, entonces usted era (i) una empresa como Facebook, donde su equipo de altamente remunerados ingenieros le permitiría encontrar una solución personalizada, o (ii) un tipo con mala suerte. Si usted se enfrenta a un problema similar estos días, entonces tiene lo que se llama un problema de "Big Data".

La comprensión de que el sistema de bases de datos relacionales (RDBMS) no es –en palabras de Stonebraker [10]— "una solución de talla única", tomó muchos años, pero el espacio de las bases de datos ha sido ahora desmonopolizado bajo la bandera de "Big Data". La **Figura 2** proporciona una amplia perspectiva de este espacio, donde se ha hecho una selección de

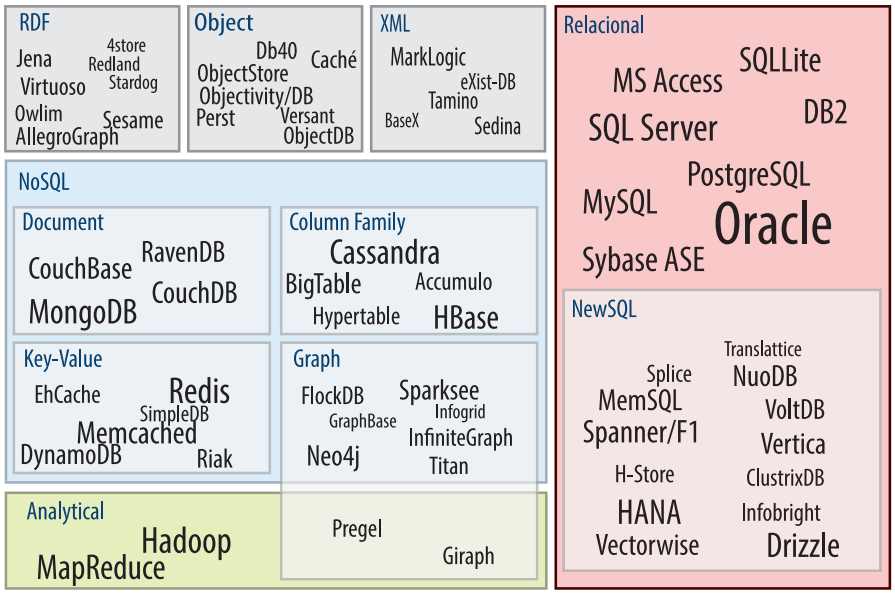


FIGURA 2.
PERSPECTIVA MODERNA DE BASES DE DATOS.

los sistemas principales de cada familia: los sistemas resaltados con letra de mayor tamaño son los que han sido mayormente adoptados[1].

En la esquina superior derecha, vemos a los sistemas de bases de datos relacionales tradicionales presentados en fuentes de gran tamaño. Estos siguen siendo la tecnología más prevalente, ofreciendo la seguridad de las garantías transaccionales (ACID) y un lenguaje de consulta poderoso (SQL). Sin embargo, estas características tienen un costo…

Tomando una base de datos moderna de código abierto en memoria (Shore) y un *benchmark* estándar (TPC-C), Harizopoulos [5] mostró que en su configuración original el sistema podía realizar 640 transacciones por segundo, pero que al apagar todas las funciones relativas a transacciones y persistencia como el *logging, latching, locking* y la gestión del *buffer*, el mismo sistema podía realizar 12.700 transacciones por segundo. Los autores estimaron que por lo tanto la base de datos sólo utilizaba un 6,8% del tiempo para realizar trabajo "útil" [5].

Por supuesto, este sistema ya no podía ofrecer garantías ACID, y el kilometraje para otros sistemas podría variar, pero en escenarios en los que tales requisitos pueden ser relajados, el mensaje central de este experimento anecdótico es que se pueden obtener enormes beneficios de rendimiento y escalabilidad reduciendo al mínimo el trabajo administrativo para las transacciones.

Siguiendo este razonamiento, y el antiguo refrán "La necesidad es la madre de todos los inventos", una nueva ola de bases de datos NoSQL (*Not Only SQL*) surgió de la era "Web 2.0", donde empresas como Facebook, Google, Amazon, Twitter, etc., se enfrentaron a un volumen y velocidad sin precedentes producidas por los datos aportados por los usuarios. Muchos de estos sistemas NoSQL se inspiraron en los Libros Blancos de compañías emergentes como Google y Amazon [1, 4] que describen las alternativas (a menudo ligeras) a las bases de datos relacionales que habían desarrollado para satisfacer estos nuevos desafíos.

NOSQL: NOT ONLY SQL

El objetivo de los repositorios NoSQL es permitir altos niveles de escalabilidad horizontal (a través de múltiples máquinas) y alto rendimiento al simplificar el modelo de base de datos, el lenguaje de consulta y las garantías de seguridad ofrecidas. Al relajar los requerimientos de ACID y SQL, y explotar la distribución, estos sistemas dicen permitir niveles de escalabilidad y rendimiento inalcanzables para las bases de datos relacionales tradicionales.

Con respecto a los modelos de datos, de acuerdo con la **Figura 2**, cuatro categorías principales de sistema NoSQL han aparecido:

- **Key-value:**
  Repositorios basados en un arreglo/map simple y asociativo. Tales sistemas permiten la búsqueda en una sola llave, devolviendo un valor que sigue un patrón. Los valores son a menudo objeto de versiones en vez de sobreescritura. Se pone énfasis en la distribución y replicación, normalmente siguiendo esquemas de hash compatibles [6]. Muchos de los sistemas en esta familia se hallan inspirados en los detalles del sistema *Dynamo* de Amazon [4], que se publicaron en 2007.

- **Document:**
  Repositorios basadas en un esquema *key-value*, pero donde los valores se refieren a "documentos" complejos sobre los cuales ciertas funciones *built-in* pueden ser ejecutadas. Una de los repositorios principales en esta familia es MongoDB, donde los valores son similares a documentos JSON.

1 Específicamente, para esto dependemos en gran medida en http://db-engines.com/en/ranking, para juzgar la popularidad de los sistemas, ranking que a su vez agrega menciones a los sistemas en páginas web, búsquedas en Google, perfiles de Linkedin, ofertas de trabajo en Google, sitios de preguntas y respuestas técnicas, Twitter, y así sucesivamente. En orden de mayor a menor, los tamaños de fuente en la Figura se refieren a las posiciones 1, 2-20, 21-50, 51-100, 101-150, 150+, resp. en la lista.

→

- **Column-family:**
  Repositorios que implementan una forma perezosa del modelo relacional a través de una abstracción *key-value*, donde las llaves son similares a las llaves primarias y los valores son multidimensionales y se ajustan a un esquema tabular flexible. *Key-values* con dimensiones similares se organizan en columna-*families*, similares a tablas relacionales. Las versiones se aplican típicamente a un nivel "celular". Así mismo las tablas están normalmente ordenadas por llave, permitiendo *range-queries* en los índices de prefijos. Estos repositorios están normalmente motivados por el diseño del sistema de BigTable de Google [1], cuyos detalles se publicaron en 2008.

- **Grafos:**
  Los repositorios que implementan adyacencias en sus índices de tal forma que atravesar de un dato/nodo a otro no requiere otra búsqueda en el índice, sino más bien un recorrido de los punteros. Lenguajes de consulta sobre caminos basados en expresiones regulares, y otros similares, permiten la navegación transitiva sobre los datos/nodos. Uno de los sistemas más importantes de esta familia es Neo4J.

Cada repositorio normalmente permite un lenguaje de consulta personalizado y ligero, que va desde búsquedas *key-version* para repositorios *key-value,* a las llaves con expresiones JSON/XML integradas para repositorios del tipo *document*, a una forma muy limitada y por tanto eficiente de SQL para los repositorios columna-*family*, a lenguajes con expresiones de camino para los *graphs*. Como un *trade-off* por la pérdida de SQL, los desarrolladores a menudo deben implementar *joins*, agregaciones y transacciones en el código de la aplicación.

En cuanto a las garantías que una base de datos distribuida puede ofrecer, el Teorema CAP establece que un sistema no puede garantizar la consistencia (acuerdo global sobre el estado/datos), disponibilidad (cada petición es atendida) y tolerancia a la partición (funcionalidad incluso si se pierden mensajes), todo al mismo tiempo. En un entorno NoSQL, la tolerancia a la partición es un objetivo fundamental ya que los datos pueden residir en cientos o miles de máquinas, aumentando la probabilidad de fallas.

A partir de entonces, los sistemas experimentan diferentes *trade-offs* entre la disponibilidad y la consistencia: una noción clave es consistencia eventual, en el cual las máquinas pueden converger hacia un acuerdo global solo después que una petición ha sido reconocida, traduciéndose en una mayor disponibilidad y mensajes reducidos, pero a costa de la consistencia. Por lo tanto si usted fuera Amazon, por ejemplo, en virtud de la consistencia eventual sus usuarios podrían ver los datos que tienen un día de antiguedad sobre las clasificaciones de productos, pero el sistema no va a rechazar nuevas calificaciones debido a mensajes de falla, y, eventualmente, todas las máquinas operativas verán estas nuevas clasificaciones. En el otro extremo del espectro, los protocolos de consenso —como los *commits* de dos o tres fases, o el algoritmo PAXOS de Lamport [9]— incurren en altos costos de comunicación, menos escalabilidad de escrituras a través de distintas máquinas, y pueden implicar tiempos de parada más frecuentes (menor disponibilidad), pero pueden garantizar nociones fuertes de consistencia.

El resultado de estos *trade-offs* que mejoran el rendimiento es el uso extendido de repositorios NoSQL en escenarios de uso intensivo de datos, de forma más prominente en empresas Web tales como Google, Facebook, Twitter, etc., pero también en muchas otras áreas: por ejemplo, repositorios centrados en documentos tales como MongoDB y CouchDB han sido utilizados en el CERN para manejar la gran cantidad de datos agregados producidos por los detectores de partículas en el Gran Colisionador de Hadrones [7].

NEWSQL: NO SOLO NOSQL

Los repositorios NoSQL han sido objeto de numerosas críticas por considerarse sobreutilizados y sobrepublicitados. Aunque este tipo de repositorios tienen, sin duda, casos de uso válidos, no todo el mundo se enfrenta a los mismos desafíos de uso intensivo de datos que Facebook o CERN. Su uso ingenuo puede implicar un riesgo innecesario de pérdida de datos o inconsistencia. La debilitación de las garantías de seguridad y los lenguajes de consulta de más bajo nivel incrementan la responsabilidad del desarrollador de la aplicación al tener que asegurarse que la base de datos se mantiene estable y que el rendimiento de las consultas más complejas es aceptable. Citando un reciente libro blanco de Google sobre la base de datos "Spanner":

> *"Creemos que es mejor que los programadores de aplicaciones se ocupen de los problemas de rendimiento debido al uso excesivo de las transacciones a medida que surgen cuellos de botella, en vez de estar siempre codificando en torno la falta de transacciones [2]".*

Aunque las características de las bases de datos relacionales de las cuales prescinden los repositorios NoSQL son computacionalmente caras, son importantes para muchas (aunque tal vez no todas) aplicaciones: fueron originalmente implementadas y añadidas a las bases de datos por una buena razón.

La familia de bases de datos NewSQL (representada en la **Figura 2**) ha surgido recientemente para lograr un compromiso entre las características de las bases de datos relacionales tradicionales y el desempeño de los repositorios NoSQL. Los sistemas NewSQL tienen como objetivo entregarle soporte a SQL y defender

ACID, pero a niveles mayores de rendimiento y escala que las bases de datos tradicionales (al menos en algunos escenarios fijos). El enfoque principal de NewSQL es diseñar bases de datos a partir de cero que exploten las arquitecturas de computadores modernas, haciendo use *cores* múltiples, y de grandes capacidades de memoria principal, GPU o clusters del tipo *shared-nothing*. Del mismo modo, muchos repositorios NewSQL optan por esquemas de indexación orientados a columnas que permiten una más eficiente agregación y filtrado sobre los valores de columnas completas que los repositorios tradicionales orientados por filas.

Sin embargo, las bases de datos no están diseñadas para el análisis de datos a gran escala, sino más bien para la ejecución de consultas en directo. Del mismo modo, muchas formas de *offline analytics* no requieren el gasto de construir y mantener los índices persistentes.

FRAMEWORKS ANALÍTICOS DISTRIBUIDOS

En la **Figura 2**, se incluye la categoría Analítica para *frameworks* de procesamiento distribuidos de datos: aunque no son, estrictamente hablando, bases de datos, ofrecen una alternativa a las bases de datos para realizar análisis.

Google propuso el *framework* MapReduce en 2004 [3] como una abstracción para la ejecución de tareas de procesamiento por lotes a gran escala sobre datos planos (no indexados/*raw* files) en un entorno distribuido. Muchos tipos diferentes de tareas de procesamiento distribuido (como las que utiliza Google, por ejemplo) tienen elementos en común: la partición de los datos a través de las máquinas, el apoyo a mecanismos de seguridad en caso de fallas de la máquina, ejecutar el procesamiento en cada máquina, mezclar los resultados de cada máquina en un resultado final, etc. Por lo tanto, el objetivo de MapReduce es ofrecer una interfaz que abstraiga de estos elementos comunes y permita el desarrollo a nivel superior de código de procesamiento distribuido.

Dado que MapReduce es considerado una de las principales tecnologías de Big Data, nos tomaremos un momento para obtener la esencia de cómo opera con el ejemplo ilustrado en la **Figura 3**. Los datos de entrada están ordenados por autor, e incluyen artículos y citaciones (note la repetición de los títulos de los artículos y las citas de cada autor). Podemos considerar esta tabla como una tabla plana (por ejemplo, un archivo CSV o TSV), además de ser muy grande: "frillones" de filas. Ahora nos gustaría saber quiénes son las parejas más productivas de coautores en nuestra tabla: queremos ver cuántas citas tiene cada par de coautores contando sólo aquellos documentos que han coescrito juntos. ¡Y tenemos un montón de máquinas para hacer esto!

MapReduce realiza agregación/*joins* en lotes mediante la ordenación de los datos. Tomando un ejemplo trivial, ya que los datos sin procesar de la entrada están ordenados por nombre de autor, podemos fácilmente obtener el recuento total de citas para cada autor, teniendo que almacenar en la memoria sólo al autor actual y su conteo actual, entregando este par de salida cuando el autor cambia. Si consideramos un entorno distribuido, aparte de la clasificación, tenemos que asegurarnos que todos los artículos de un autor terminan en la misma máquina.

Volviendo a nuestra no trivial tarea original de determinar los pares más productivos de coautores, MapReduce consta de dos fases principales: **Map** y **Reduce**. Lo primero que necesitamos figurarnos son los pares de coautores. Para hacer esto, tenemos que realizar un *join* distribuido sobre los artículos en términos de título/citas.

> Map_1: Para cada tupla A_i = ($Author_i$, $Title_i$, $Citation_i$) en la entrada, crear pares llave-valor (*key-value*) de la siguiente forma: ($Title_i$, <$Author_i$, $Citations_i$>), donde la llave es $Title_i$ y el valor es <$Author_i$, $Citations_i$>.

MapReduce asignará pares llave-valor con la misma llave a la misma máquina (por ejemplo, utilizando una función de hash en la llave). Esa máquina va a ordenar los pares de acuerdo a su llave utilizando la función `Map`. La partición y el ordenamiento son usualmente solucionados por el *framework*, lo que significa que el desarrollador no tiene que preocuparse acerca de qué datos van a cuál máquina física. Sin embargo, el valor por defecto de la partición y el ordenamiento pueden ser invalidados si es necesario.

Ahora, cada máquina tiene su propio *bag* de pares llave-valor ordenado por `Title`, con todos los autores de cada artículo disponible localmente. La siguiente es la fase de reducir:

> $Reduce_1$: En preparación para la reducción, la fase de ordenación agrupará los pares llave-valor (emitidos en Map_1) por título: ($Title_i$,{ <$Author_{i,1}$,$Citations_i$>; … ; <$Author_{i,n}$,$Citations_i$>}). Para cada uno de esos grupos, Reduce entregará como salida un conjunto de tuplas que represente cada par de coautores: {(<$Author_{i,j}$, $Author_{i,k}$>, $Citations_i$)| 1 ≤ j < k ≤ n}.[2]

El resultado de la fase `Reduce` es un *bag* de pares únicos de coautores y sus citas para cada artículo (ya no necesitamos el nombre del artículo). Nuestra tarea final es sumar las citas totales por cada par de coautores. En la etapa anterior, los datos fueron ordenados/*joined* con respecto al título del artículo, por lo que ahora tenemos que realizar otra fase de `Map/Reduce`.

2 Asumimos $Author_x < Author_y$ si y solo si x < y, etc., para mantener pares consistentes entre distintos artículos. →

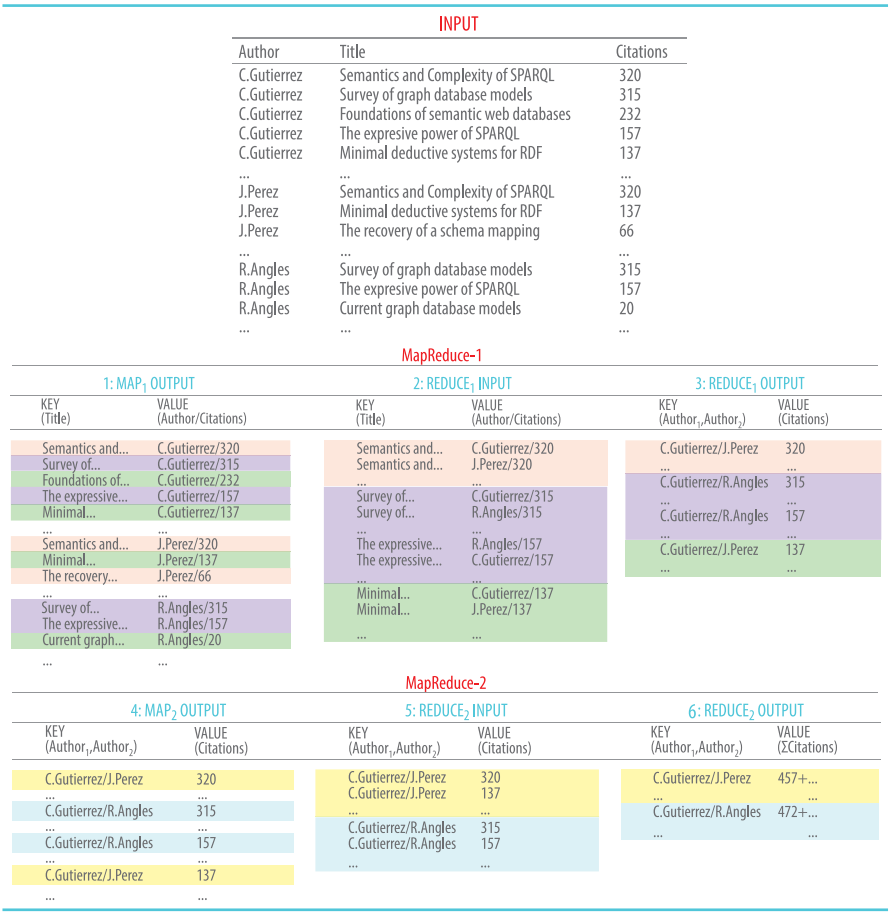


FIGURA 3.
UN EJEMPLO DE MAPREDUCE: ENCONTRAR LAS CITAS PARA CADA PAR DE COAUTORES (CUENTAN SOLO LOS ARTÍCULOS QUE HAN ESCRITO JUNTOS). LOS TONOS DIFERENTES DE LAS FILAS INDICAN UNA MÁQUINA DIFERENTE.

> Map_2: Por cada tupla (<$Author_x$, $Author_y$>, $Citations_z$) creada en el paso intermedio, mapee el mismo par llave-valor.

Este mapa no transforma los valores, sino que simplemente se asegura que los mismos pares de coautores terminan ordenados en la misma máquina para la fase Reduce final.

> $Reduce_2$: Con grupos (<$Author_x$, $Author_y$>,{$Citations_1$,..., $Citations_n$}), sume las citas en cada bag y produzca un solo par de salida: (<$Author_x$, $Author_y$>, $\sum_{z=1}^{n} Citations_z$).

Ahora tenemos nuestro resultado final. Si quisiéramos extraer resultados del tipo top-k, podríamos pasar a MapReduce un orden descendente y mapear el total de citas como llave.

En cuanto al esfuerzo de codificación, el desarrollador debe implementar las etapas de mapeo y reducción. Por su parte, MapReduce se hará cargo de balancear la carga, la tolerancia a fallas, y así sucesivamente. Con un *cluster* de máquinas que ejecuten el marco MapReduce, las tareas pueden ser ejecutadas por múltiples usuarios en paralelo y el marco tratará de hacer el mejor uso de los recursos disponibles. Del mismo modo, las tareas MapReduce son portátiles: los trabajos se pueden ejecutar en cualquier *cluster* de máquinas, siempre que el marco esté instalado (y los datos disponibles).

MapReduce permite la escalabilidad, pero, por supuesto, no la garantiza: las tareas intratables lo seguirán siendo en varias máquinas. En nuestro ejemplo, `Reduce1` produce un número cuadrático de pares de coautores por artículo ($\frac{n(n-1)}{2}$) , pero como las listas de autores son generalmente cortas, podemos esperar que `Reduce1` opere casi linealmente. Bajo la suposición de que las tareas se pueden descomponer en fases del tipo `Map-Reduce`, y que estas fases no son computacionalmente costosas o aumentan demasiado el volumen de los datos, MapReduce es una abstracción conveniente y de gran alcance para los que deseen trabajar con grandes volúmenes de datos.

Desafortunadamente, el sistema MapReduce es en sí propietario y se mantiene cerrado por Google. Sin embargo, el proyecto de código abierto Apache Hadoop ofrece una aplicación madura del marco y es de uso generalizado.

En la **Figura 2**, se puede observar otros dos sistemas en la intersección de **Graph** y **Analytical**: Pregel y Giraph. Ambos son marcos de estilo MapReduce diseñados específicamente para el procesamiento de grafos. Pregel fue introducido por Google en 2009 [8] para realizar computación distribuida sobre grafos de gran tamaño, y Apache Giraph es una implementación de código abierto de las mismas ideas (construidas sobre Hadoop). En resumen, el núcleo del modelo computacional de Pregel/Giraph es un sistema de intercambio de mensajes entre los vértices de un grafo. La computación está organizada en iteraciones. Un vértice puede leer los mensajes que le fueron enviados en la iteración anterior, puede cambiar su estado (el que puede ser continuo), y puede reenviar mensajes a otros vértices para la siguiente iteración. Los mensajes pueden ser enviados a cualquier vértice con un id localmente conocido, pero por lo general estos son los vértices que están enlazados. Estos marcos computacionales ofrecen una abstracción intuitiva para la aplicación de análisis basado en grafos –para ejecutar tareas tales como caminos más cortos, componentes conexas, *clustering*, medidas de centralidad como *PageRank*, etc.– mientras transparentemente distribuyen la computación: manejo del balance de carga, eficiente loteo (*batching*) de mensajes y tolerancia a fallas.

OTROS MODELOS: OBJETOS, XML, RDF

Con respecto a la **Figura 2**, nos quedan por tanto tres familias (relativamente tradicionales) de bases de datos en la esquina superior izquierda: `Object`, `XML` y `RDF`.

Las bases de datos de objetos (`Object`) permiten almacenar la información en la forma de objetos (de los que ocurren en el software orientado a objetos). La motivación principal de estas bases de datos es ofrecer la opción de persistencia para los objetos del software en un formato nativo. Típicamente, las bases de datos de objeto están hechas a la medida de un conjunto fijo de lenguajes de programación orientado a objetos, ofreciendo la posibilidad para una estrecha integración entre el ambiente de ejecución y la base de datos. Estas bases de datos vienen equipadas normalmente con un lenguaje de consulta que permite buscar objetos con ciertos campos o valores; de la misma forma, utilizan punteros entre los objetos para evitar el uso de *joins*. Los sistemas podrían ofrecer variadas otras funcionalidades, como versionamiento, restricciones de integridad, *triggers*, etc.

Las bases de datos XML son típicamente nativas, donde los datos son almacenados en una estructura de datos del tipo XML. Dada la relativa popularidad de XML como modelo para el intercambio de información y su uso en la Web (e.g., XHTML, RSS y ATOM *feeds*, mensajes SOAP, etc.), las bases de datos XML almacenan tales datos en un formato nativo que les permite ser consultadas directamente a través de lenguajes como XQuery, XPath y XSLT.

Las bases de datos RDF se enfocan en proveer funcionalidad de consulta sobre datos representados en el modelo básico de datos de la Web semántica. Tales repositorios implementan típicamente esquemas optimizados de almacenamiento para RDF y ofrecen funcionalidad de consulta usando el lenguaje SPARQL recomendado por la W3C. Además, estos motores podrían implementar funcionalidades relacionadas al razonamiento, comúnmente con respecto a los estándares RDFS y/o OWL.

Aunque estas tres familias de bases de datos han recibido atención significativa a nivel de investigación en la década pasada, como puede observarse en la **Figura 2**, permanecen siendo mayormente una tecnología de nicho cuando se consideran en el contexto más amplio de las tecnologías de bases de datos.

¿VARIEDAD?

Hemos hablado mucho acerca del volumen, e.g., escalabilidad a través de múltiples máquinas e.g., aumento en el rendimiento de escritura mediante la relajación de las garantías de consistencia. Sin embargo, no hemos hablado mucho sobre el tercer reto del Big Data: variedad. Mientras los problemas de volumen y velocidad se enlazan con el rendimiento y pueden ser abordados desde una perspectiva ingenieril mediante la composición de técnicas existentes tales como la partición horizontal, la replicación, ordenamientos distribuidos, *hashing* consistentes, compresión, *batching*, filtros de Bloom, árboles de Merkle, indexamiento orientado a columnas, etc. en el diseño de un sistema maduro, el problema de la variedad plantea cuestiones de carácter más conceptual.

Podría decirse que la familia de bases de datos NoSQL ofrece algunas ventajas sobre las bases de datos relacionales tradicionales cuando es necesario procesar conjuntos de datos diversos: al relajar el modelo relacional, los datos pueden ser almacenados de una forma no-normalizada "rápida y sucia". Aunque esto podría ahorrar tiempo para el manejo de esquemas, y es más flexible para trabajar con datos incompletos o irregulares, emplear modelos más simples nuevamente sobrecarga la capa de aplicación, en la cual los desarrolladores deben asegurarse que los datos permanecen consistentes y que los índices se hallan presentes para cubrir eficientemente la carga de trabajo esperada de las consultas.

De la misma forma, la variedad puede ser parcialmente solucionada por el rango de bases de datos disponibles hoy. Por ejemplo, uno podría usar una base de datos XML para almacenar datos XML, una base de datos relacional para los datos relacional, una base de datos de grafos para los datos estructurados en forma de grafo, y así sucesivamente. Múltiples bases de datos podrían entonces ser compuestas para atacar el problema de la variedad: sin embargo, la carga nuevamente está puesta sobre los desarrolladores de la aplicación que deben conectar las bases de datos y asegurarse que los datos permanecen consistentes a través de los distintos sistemas.

En general, el problema de la variedad es mucho más profundo que la forma o la incompletitud de los datos. Además, la variedad está lejos de ser un problema nuevo: varias subáreas de la Ciencia de la Computación han enfrentado el problema de la "integración de datos", ya sea combinando múltiples bases de datos en una, o combinando documentos desde millones de fuentes en la Web, o combinando repositorios de código fuente desde varios proyectos, o …

LOS DATOS ESTÁN EVOLUCIONANDO

Pareciera que el único enfoque práctico para resolver el cuello de botella en Big Data es repensar lo que queremos decir por "datos". Desde los días de papel y tinta de Snow, los datos han evolucionado a través del código Morse, tarjetas agujereadas, formatos binarios, ASCII, anotaciones, y así sucesivamente. La evolución de los datos ha ido siempre en la dirección de mejorar la lectura de las máquinas. Así como podría ser considerado imposible contar todas las

→

ocurrencias de la palabra "brócoli" en los libros almacenados en una biblioteca bien equipada (y trivial de hacerse sobre un archivo de texto electrónico), existen muchas tareas intensivas en datos que podríamos considerar imposible de realizarse hoy en día simplemente porque nuestra noción de "datos" no lo permite.

Actualmente, tenemos una plétora de modelos de datos y sintaxis de datos disponibles para permitir que la información sea estructurada y analizada sintácticamente. Sin embargo, en su mayoría, las máquinas necesitan código especializado en datos para interpretar los datos antes mencionados. Por ejemplo, los buscadores han sido construidos específicamente para interpretar datos relacionados con HTML.

En términos de la evolución de los datos, parece que la única forma de avanzar es hacer explícita la semántica de los datos que están siendo entendidos, de tal forma que las máquinas puedan, con profundidad creciente, procesar el significado de los datos. Las máquinas aún no pueden aprender bien del lenguaje natural ni adaptarse bien a nuevos problemas: por eso la semántica necesita hacerse explícita como parte de los datos para ayudar a vencer la verdadera variedad en los datos estructurados; i.e., la capacidad de incorporar datos imprevistos.

En términos de hacer explícita la semántica, podemos comenzar simplemente por usar identificadores globales para los objetos descritos en los datos de tal forma que una máquina pueda conocer, por ejemplo, que el "Boston" descrito en un conjunto de datos es el mismo que el "Boston" descrito en otro conjunto de datos (e.g., utilizando URIs como identificadores); ahora la máquina puede correr *joins* sobre los dos conjuntos. Aún mejor, si la máquina puede saber que el "Boston" en cuestión es la banda de música, no la ciudad de Estados Unidos (e.g., usando un sistema de clasificación) ahora la máquina puede entregar resultados acerca de "Boston" para preguntas sobre bandas de música. Aún mejor incluso si la máquina puede saber que las bandas de música tienen miembros, y típicamente lanzan álbumes y… (e.g., a través de una ontología del dominio).

BIG DATA SEMÁNTICO

Muchos de estos principios han sido explorados por la comunidad de la Web Semántica; en la opinión sesgada de este autor, las técnicas de la Web Semántica podrían aún cumplir un rol importante en términos de Big Data, particularmente para hacer frente a la diversidad. La Web Semántica ha sido largamente mirada en menos por mucho como un ejercicio académico estéril, condenada al fracaso debido a una serie de preocupaciones. Y hay un grado de verdad en esas preocupaciones. Muchas de las técnicas propuestas en el área de las ontologías y métodos de razonamiento deductivo no son adecuadas en escenarios con montones de datos, y ciertamente no para escenarios con montones de datos (Web) desordenados. Sin embargo, los desafíos no son irremontables: un poco de semántica puede llevarnos lejos.

Recientemente, compañías como Google, Facebook, Yahoo!, Microsoft, etc., están comenzando a usar partes de las tecnologías de la Web Semántica para potenciar nuevas aplicaciones. Por ejemplo, en junio de 2011, Bing, Google y Yahoo! anunciaron la ontología *schema.org* para que los *webmasters* puedan dejar datos estructurados disponibles en su sitio. Google ha usado bases de conocimiento semánticas en la construcción de su aplicación *Knowledge-graph*. El protocolo *Open Graph* de Facebook –que trata de crear un red descentralizada de datos– utiliza RDFa: un standard de la Web Semántica. Más recientemente, Google anunció soporte para anotaciones semánticas en Gmail usando la sintaxis JSON-LD. En tales casos, estas grandes compañías han virado hacia las tecnologías de la Web Semántica para construir aplicaciones sobre gigantescos conjuntos de fuentes diversas provenientes de millones de contribuyentes arbitrarios. Y esta tendencia de utilizar la semántica para hacerle frente a la diversidad parece que va a continuar.

CONCLUSIONES

BIG DATA ES UN LLAMADO A LOS CIENTÍFICOS DE LA COMPUTACIÓN A INVESTIGAR MÉTODOS PARA TRATAR DATOS CON MÁS VOLUMEN, MÁS VELOCIDAD Y MÁS VARIEDAD. AUNQUE LAS BASES DE DATOS RELACIONALES HAN SERVIDO COMO CABALLO DE BATALLA SEGURO POR MUCHOS AÑOS, LA GENTE ESTÁ COMENZANDO A DARSE CUENTA QUE SE NECESITAN NUEVAS ALTERNATIVAS PARA ENFRENTAR LOS DESAFÍOS VENIDEROS PLANTEADOS POR EL EMERGENTE DILUVIO DE DATOS.

LAS PRINCIPALES TECNOLOGÍAS BIG DATA QUE HAN APARECIDO HASTA EL MOMENTO SE CENTRAN PRINCIPALMENTE EN LOS DESAFÍOS DE VOLUMEN Y VELOCIDAD. LOS REPOSITORIOS NOSQL OFRECEN NUEVOS NIVELES DE ESCALABILIDAD MEDIANTE LA DISTRIBUCIÓN DEL MANEJO DE LOS DATOS EN MÚLTIPLES MÁQUINAS, LA RELAJACIÓN LAS GARANTÍAS ACID Y UTILIZANDO LENGUAJES DE CONSULTA MÁS LIVIANOS QUE SQL. NEWSQL BUSCA ENCONTRAR UN BALANCE OFRECIENDO ACID/SQL COMO LAS BASES DE DATOS RELACIONALES TRADICIONALES, AL MISMO TIEMPO QUE PRETENDE COMPETIR CON EL RENDIMIENTO Y ESCALA DE LOS SISTEMAS NOSQL. ASÍ MISMO, MARCOS DE PROCESAMIENTO DISTRIBUIDO COMO MAPREDUCE OFRECEN UNA ABSTRACCIÓN CONVENIENTE PARA CLIENTES QUE DESEEN REALIZAR TAREAS ANALÍTICAS DE GRAN ESCALA.

POR OTRO LADO, LA VARIEDAD SE MANTIENE COMO PROBLEMA ABIERTO. EN PARTICULAR, EL OBJETIVO DE SER CAPAZ DE CONSTRUIR APLICACIONES QUE PUEDAN ROBUSTAMENTE DESCUBRIR E INCORPORAR NUEVAS FUENTES DE DATOS SIGUE ELUDIÉNDONOS. PARA SOLUCIONAR ESTE PROBLEMA, QUIZÁ NECESITAMOS REPENSAR NUESTRA CONCEPTUALIZACIÓN DE LOS DATOS. COMPARADO CON LOS TIEMPOS DE JOHN SNOW, HOY EN DÍA APRECIAMOS LA CONVENIENCIA CON DATOS SERIALIZADOS EN UN FORMATO ELECTRÓNICO QUE PUEDE SER LEÍDO POR UNA MÁQUINA. DE LA MISMA FORMA, TENER DATOS CON UNA SEMÁNTICA EXPLÍCITA PODRÍA CONVERTIRSE EN LA NORMA EN LAS SIGUIENTES DÉCADAS. ■

BIBLIOGRAFÍA

[1] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A Distributed Storage System for Structured Data. ACM Trans. Comput. Syst., 26(2), 2008.

[2] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. C. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, and D. Woodford. Spanner: Google's Globally-Distributed Database. In OSDI, pages 261-264, 2012.

[3] J. Dean and S. Ghemawat. MapReduce: Simplifed Data Processing on Large Clusters. In OSDI, pages 137-150, 2004.

[4] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: Amazon's highly available key-value store. In SOSP, pages 205-220, 2007.

[5] S. Harizopoulos, D. J. Abadi, S. Madden, and M. Stonebraker. OLTP through the looking glass, and what we found there. In SIGMOD Conference, pages 981-992, 2008.

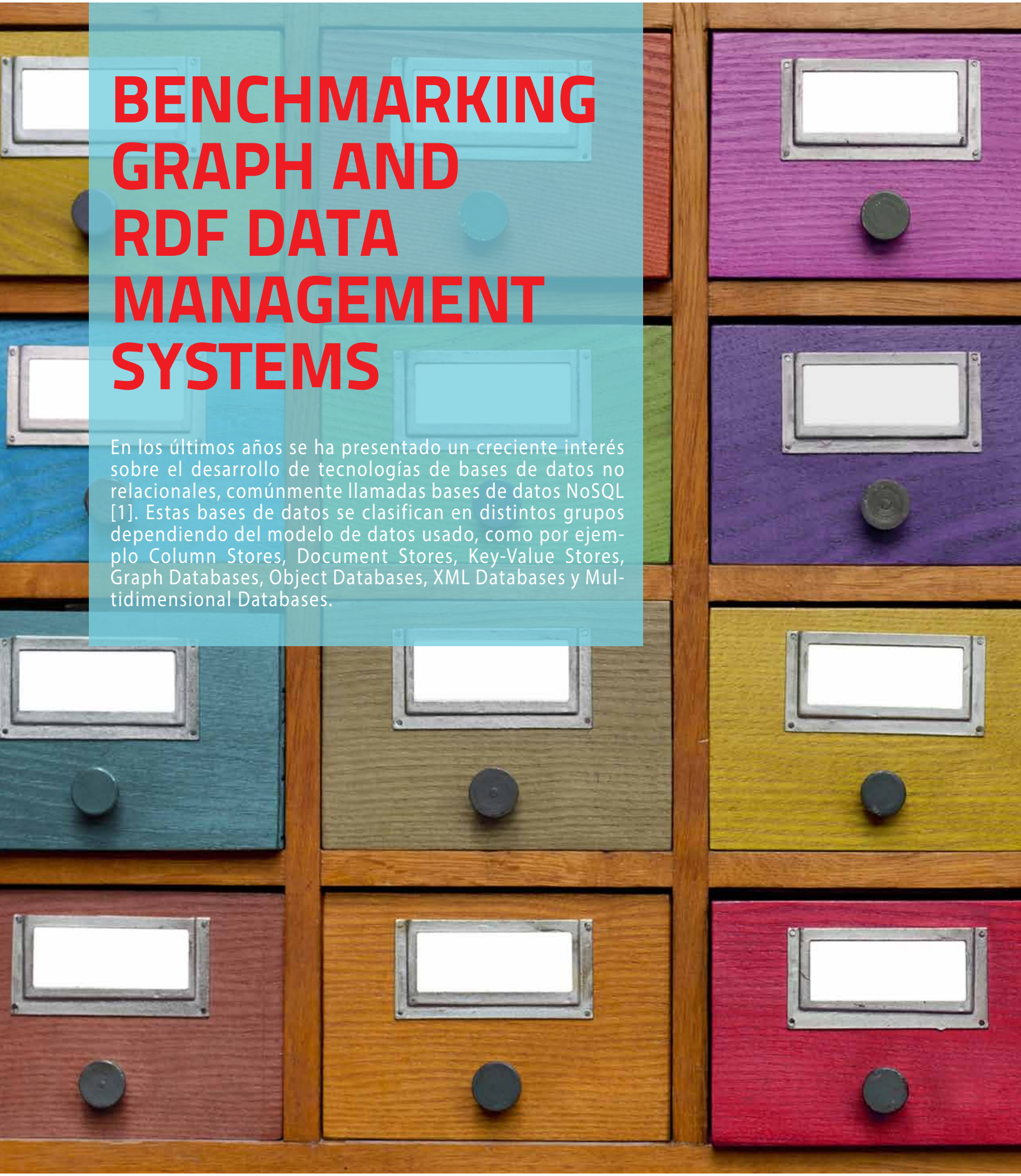
[6] D. R. Karger, E. Lehman, F. T. Leighton, R. Panigrahy, M. S. Levine, and D. Lewin. Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. In STOC, pages 654-663, 1997.

[7] V. Kuznetsov, D. Evans, and S. Metson. The CMS data aggregation system. In ICCS, number 1, pages 1535-1543, 2010.

[8] G. Malewicz, M. H. Austern, A. J. C. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: a system for large-scale graph processing. In SIGMOD Conference, pages 135-146, 2010.

[9] M. C. Pease, R. E. Shostak, and L. Lamport. Reaching agreement in the presence of faults. J. ACM, 27(2):228-234, 1980.

[10] M. Stonebraker. One size fits all: an idea whose time has come and gone. Commun. ACM, 51(12):76, 2008.



BENCHMARKING GRAPH AND RDF DATA MANAGEMENT SYSTEMS

En los últimos años se ha presentado un creciente interés sobre el desarrollo de tecnologías de bases de datos no relacionales, comúnmente llamadas bases de datos NoSQL [1]. Estas bases de datos se clasifican en distintos grupos dependiendo del modelo de datos usado, como por ejemplo Column Stores, Document Stores, Key-Value Stores, Graph Databases, Object Databases, XML Databases y Multidimensional Databases.

RENZOANGLES

Profesor Asistente, Departamento de Ciencias de la Computación, Universidad de Talca. Ingeniero de Sistemas, Universidad Católica de Santa María, Perú. Doctor en Ciencias mención Computación, Universidad de Chile. En 2013 realizó un postdoctorado en la VU University Amsterdam, participando en el proyecto "Linked Data Benchmark Council (LDBC)". Sus áreas de investigación son Bases de Datos de Grafos y Web Semántica, específicamente en Lenguajes de Consulta y Benchmarking de Bases de Datos para Grafos y RDF.

rangles@utalca.cl

BENCHMARKS PARA GRAPH/RDF DATABASES

La aparición del enfoque NoSQL se debió principalmente a las limitaciones que tienen las bases de datos tradicionales (aquellas basadas en el modelo relacional) para satisfacer los requisitos de gestión de datos en dominios de aplicación no tradicionales, los cuales requieren lidiar con grandes cantidades de datos de estructura compleja, como por ejemplo Big Data [2] o Linked Open Data [3]. En este sentido, las tecnologías NoSQL buscan mejorar la flexibilidad y el desempeño de los sistemas de gestión de datos basados en características como escalabilidad horizontal, independencia de esquema de datos, consistencia de datos parcial, replicación de datos y computación distribuida [4].

Las bases de datos orientadas a grafos (en inglés, *Graph Databases*) y las bases de datos RDF (en inglés, RDF *databases*, también llamadas *Triple Stores*) son dos enfoques NoSQL orientados a lidiar con datos no estructurados (heterogéneos, no relacionales) y altamente conectados. Las graph databases están diseñadas para almacenar datos con estructura de grafo y consultarlos a través de operaciones y/o lenguajes de consulta pensadas para explorar los grafos almacenados. Las RDF databases son bases de datos de grafo especialmente diseñadas para gestionar datos semiestructurados y metadatos creados en base al modelo de datos RDF, y permiten consultar dichos datos usando el lenguaje de patrones SPARQL, además de operadores especiales que permiten inferencia sobre los datos.

Actualmente existen diversas graph databases (Sparksee, Neo4j, AllegroGraph) y RDF databases (OpenLink Virtuoso, OWLIM, Sesame). Sin embargo, esta diversidad de sistemas genera las siguientes preguntas: ¿cuál es el desempeño de una graph/RDF database? ¿Cuál es la graph/RDF Database con mejor desempeño? Para poder responder a estas consultas, debemos evaluar y comparar los sistema de bases de datos usando herramientas denominadas benchmarks.

En el contexto general, un benchmark es una herramienta que permite comparar el desempeño de los sistemas. En el contexto de las bases de datos, un benchmark permite evaluar la capacidad de los sistemas de gestión de bases de datos, en particular su eficiencia para responder a las operaciones sobre los datos. De esta manera, los benchmarks muestran las fortalezas y debilidades de los sistemas.

Cabe destacar que los benchmarks no son pensados únicamente para evaluar los sistemas, más importante aún, estos buscan estimular el avance tecnológico a través de la identificación de posibles mejoras a nivel de desempeño y funcionalidad. En conclusión, los benchmarks ayudan a los usuarios finales de las bases de datos en la elección de productos de software competitivos y guían a la industria hacia el desarrollo de nuevas tecnologías.

La existencia de procesos estándar de benchmarking que definan la ejecución correcta de un benchmark sobre un sistema, son fundamentales para asegurar la confianza y aceptación de los benchmarks. Por ejemplo, el Transaction Processing Performance Council (TPC) [5] es un consorcio creado para supervisar el desarrollo y ejecución de benchmarks estándar para bases de datos relacionales; esto con la finalidad de asegurar resultados de benchmarking confiables para la industria y el mercado de usuarios finales.

En el contexto de las graph/RDF databases, no existen benchmarks estándar ni tampoco una autoridad independiente que se encargue de controlar los procesos de benchmarking. Sin bien existen algunos benchmarks provenientes del ámbito académico, estos no cumplen totalmente con las características deseadas en los benchmarks industriales, como por ejemplo alcance, relevancia, verificabilidad, en otras [6]. Además, los benchmarks académicos no modelan escenarios caracterizados por operaciones complejas sobre datos asimétricos y altamente correlacionados, como aquellos encontrados en casos de uso reales como Big Data o Linked Open Data [7].

THE LINKED DATA BENCHMARK COUNCIL

The Linked Data Benchmark Council (LDBC) [8] es un proyecto europeo que reúne una comunidad de académicos y expertos de la industria, cuyo objetivo común es el desarrollo de benchmarks estándar para la industria de graph/RDF databases.

El proyecto LDBC busca crear benchmarks siguiendo los principios de relevancia, simplicidad, confiabilidad y sostenibilidad. De manera especial, el LDBC busca el desarrollo de benchmarks que evalúen funcionalidades críticas de los sistemas, yendo más allá de los benchmarks creados en la academia. Con este fin, el LDBC entregará benchmarks de código abierto, desarrollados por grupos de trabajo integrados por expertos en arquitectura de bases de datos quienes conocen las funcionalidades críticas dentro de los motores de gestión de datos, y soportados por una comunidad de usuarios que entregan casos de uso y retroalimentación. Además, el LDBC espera incluir un mecanismo que asegure que los resultados de benchmarking sean revisados por una entidad independiente para verificar su conformidad.

A continuación describiremos brevemente los elementos que conforman un benchmark, según la guía de diseño elaborada al interior de LDBC, y luego describiremos brevemente dos benchmarks que se encuentran en pleno proceso de desarrollo: el Social Network benchmark y el Semantic Publishing benchmark.

DISEÑO DE BENCHMARKS EN LDBC

Un benchmark está compuesto generalmente de tres elementos: un generador de datos (*data generator*), el cual permite crear datos en base a un esquema de datos definido; un generador de carga de trabajo (*workload generator*), el cual define el conjunto de operaciones (workload) que el sistema bajo evaluación (system under test, SUT) tendrá que procesar; y un conductor de pruebas (test driver), el cual es usado para ejecutar el workload sobre el sistema bajo evaluación, siguiendo reglas de ejecución precisas y midiendo el desempeño según métricas bien definidas.

En la **Figura 1**, se muestra la estructura del Social Network benchmark tomando en cuenta los tres componentes descritos anteriormente. Observe que cada componente contiene subelementos con características específicas; por ejemplo, los datos generados serán no-estructurados e incluirán correlaciones y distribuciones no uniformes. Nótese además, que la **Figura 1** incluye características alrededor de los componentes del benchmark (ej. simplicidad), las cuales dirigen su diseño y construcción.

El desarrollo de un benchmark implica un número significante de detalles que deben tomarse en cuenta, en particular durante la etapa de diseño. Entre estos detalles podemos mencionar:

- *El caso de uso debe ser real, claro, comprensible y relevante para los usuarios y la comunidad.*
- *El workload debe ser representativo de las operaciones encontradas en el caso de uso seleccionado.*

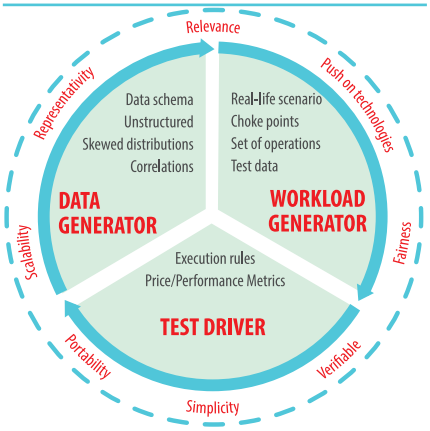


FIGURA 1.
ELEMENTOS Y CARACTERÍSTICAS DEL LDBC SOCIAL NETWORK BENCHMARK.

- *El workload debe ser diseñado cuidadosamente para asegurar que el benchmark incentive la innovación tecnológica. En este sentido, cada workload será definido en base a desafíos técnicos bien identificados denominados choke points. El objetivo de un diseño basado en choke points es asegurar que un workload presiona las funcionalidades técnicas más importantes de los sistemas actuales.*
- *Las reglas de ejecución y las métricas del benchmark deben ser definidas cuidadosamente para asegurar una evaluación confiable y una comparación justa de los sistemas.*

El desarrollo de benchmarks en el LDBC está a cargo de grupos de desarrollo denominados *task forces*. Una task force está formada por miembros del LDBC, incluyendo tanto usuarios finales como proveedores de tecnologías de base de datos. Se espera que los participantes de la industria sean personas expertas en los aspectos técnicos de los sistemas, de manera que faciliten la definición de choke points. En el caso de los usuarios finales, se espera recibir sus requisitos respecto a casos de uso relevantes así como la entrega de retroalimentación durante el desarrollo de un benchmark.

THE SOCIAL NETWORK BENCHMARK

El Social Network Benchmark (SNB) es un benchmark pensado para evaluar diversas funcionalidades de sistemas usados en la gestión de datos con estructura de grafo. El escenario de

este benchmark es una red social, y fue elegido con la siguientes metas en mente: deberá ser entendible para una gran audiencia, y esta audiencia deberá comprender la relevancia de gestionar dichos datos; deberá cubrir un conjunto de desafíos interesantes acorde con los alcances del benchmark; si bien los datos y el workload se crearán de manera sintética, estos deberán ser realistas en el sentido de reflejar características encontradas en la vida real.

El generador de datos del SNB está siendo diseñado para crear datos sintéticos con las siguientes características: el esquema de datos debe ser representativo de una red social; el método de generación debe considerar las propiedades existentes en redes sociales reales, incluyendo correlaciones entre los datos y distribuciones estadísticas; las herramientas de software generadas deben ser fáciles de usar, configurables y escalables.

El esquema de datos del SNB modela una red social con perfiles de usuario enriquecidos con intereses, etiquetas (tags), mensajes (posts) y comentarios. Adicionalmente, los datos generados exhiben correlaciones reales entre valores (ej., los nombres de las personas son creados de acuerdo con su nacionalidad), correlaciones de estructura (ej., dos personas amigas en su mayoría viven en lugares cercanos geográficamente), y distribuciones estadísticas (ej., la relación de amistad sigue una distribución de ley de potencias). El generador de datos está implementado para ejecutarse en Hadoop, lo cual permite una generación rápida y escalable de archivos de datos de gran tamaño.

Con el objetivo de cubrir los requisitos más relevantes de las aplicaciones que gestionan datos sobre redes sociales, el SNB entrega tres workloads distintos (de cierto modo el SNB es tres *benchmarks* en uno): un *interactive workload*, orientado a evaluar consultas relativamente simples y operaciones de actualización concurrentes; un *business intelligence workload*, compuesto de consultas complejas que simulan un análisis en línea del comportamiento de los usuarios, esto con el propósito de realizar marketing; y un *graph analytics workload*, pensado para evaluar la funcionalidad y escalabilidad de los sistemas para el análisis de grafos a través de operaciones complejas, las cuales usualmente no pueden ser expresadas usando un lenguaje de consulta.

Adicionalmente, cada workload incluirá una o más métricas para medir el desempeño de los sistemas, como por ejemplo el tiempo de respuesta o el throughput (métrica que mide el número de operaciones por unidad de tiempo, por ejemplo, transacciones por minuto).

THE SEMANTIC PUBLISHING BENCHMARK

El Semantic Publishing Benchmark (SPB) está diseñado para simular la gestión y consumo de metadatos RDF sobre contenido multimedia. El escenario específico se basa en una organización de noticias, la cual mantiene descripciones RDF de su catálogo de noticias además de los trabajos creativos. El SPB simula un workload donde un gran número de agentes consultan el catálogo de artículos noticiosos y al mismo tiempo se tienen operaciones de edición y descripción del contenido multimedia.

Para la generación de datos, el SPB emplea una ontología que define numerosas propiedades para el contenido, por ejemplo fecha de creación, resumen, descripción, entre otros. Además, una ontología de etiquetas es usada para clasificar los trabajos creativos en diversas categorías como deportes, geografía o información política.

El SPB incluye dos workloads que demandan alto desempeño para la ejecución de consultas (que pueden calcularse de manera paralela), así como para operaciones de actualización continuas y concurrentes. El *Editorial* Workload simula la creación, actualización y borrado de metadatos sobre trabajos creativos. Este workload se basa en que las compañías de medios usan procesos manuales y semiautomáticos para gestionar descripciones de trabajos y clasificarlas de acuerdo a categorías definidas en ciertas ontologías, además de incluir referencias a otras fuentes de datos. El *Aggregation workload* simula la agregación dinámica de contenido para su inmediato consumo (ej., a través de un sitio web). La acción de publicar contenido es considerada dinámica ya que el contenido no se selecciona ni arregla manualmente, en lugar de esto se usan plantillas que entregan formato al contenido, el cual es seleccionado cuando el usuario lo accede. En este workload se usan consultas SPARQL para encontrar contenido relevante.

De manera preliminar, el SPB considera el throughput como métrica para medir las operaciones de actualización y consulta que ejecutan los agentes editores y consumidores de contenido durante una cantidad definida de tiempo.

NOTAS FINALES

El software y los documentos asociados al SNB y al SPB pueden ser descargados desde GitHub [9] y la información sobre su desarrollo se encuentra disponible en el Wiki [10] administrado por las Task Force. Se invita al lector a unirse a la comunidad del LDBC para colaborar e influenciar en el desarrollo de benchmarks para graph/RDF databases. ■

REFERENCIAS

[1] NoSQL Databases. http://nosql-database.org
[2] C. Snijders, U. Matzat and U. Reips. Big Data: Big Gaps of Knowledge in the Field of Internet Science. International Journal of Internet Science, Vol. 7, Nº 3, 2012.
[3] T. Heath and C. Bizer. Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology, Vol. 1, Nº 1, 2011.
[4] M. Stonebraker. SQL Databases V. NoSQL Databases. Communications of the ACM, Vol. 53, Nº 4, 2010.
[5] Transaction Processing Performance Council (TPC). http://www.tpc.org/
[6] K. Huppler: "The Art of Building a Good Benchmark", in: TPCTC, 2009.
[7] S. Duan, A. Kementsietsidis, K. Srinivas and O. Udrea. Apples and Oranges: A Comparison of RDF Benchmarks and Real RDF Datasets. Proc. of the International Conference on Management of Data (SIGMOD), 2011.
[8] Linked Data Benchmark Council (LDBC). http://www.ldbc.eu
[9] LDBC Software and Documentation. https://github.com/ldbc/
[10] LDBC Technical User Community Wiki. http://138.232.65.142:8090/display/TUC/

ANDES WALL SIZED DISPLAY:
VISUALIZACIÓN DE BIG DATA EN ALTA RESOLUCIÓN A DISPOSICIÓN DE LA COMUNIDAD CIENTÍFICA Y LA INDUSTRIA CHILENA

Profesionales de diversas áreas requieren visualizar e interactuar con grandes volúmenes de datos de forma eficiente y sencilla. En astronomía, los observatorios capturan una gran suma de imágenes de alta resolución. Se requieren técnicas de visualización que permitan visualizarlas simultáneamente y complementarlas con información existente. Una sala de control recibe grandes volúmenes de datos provenientes de diferentes fuentes en tiempo real por lo que los equipos de trabajo podrían beneficiarse de visualizaciones comunes que favorezcan la conciencia situacional [1].

IMAGEN PROVENIENTE DEL TELESCOPIO SPITZER EN ANDES (396032*27040 PÍXELES).

CLAUDEPUECH

Profesor de Ciencia de la Computación en la Universidad Paris-Sud, Orsay, Francia. Se encuentra actualmente en misión para Inria en Chile, desempeñándose como Director Ejecutivo de la Fundación Inria Chile. Recibió en 2004 el primer Eurographics "Distinguished Career Award" por sus contribuciones fundamentales en temas de Computer Graphics y de Computational Geometry, además de su continuo rol, a lo largo de los años, en la capacitación de jóvenes investigadores. Muchos de sus antiguos estudiantes se perfilan hoy como algunos de los investigadores más exitosos en estos campos.

claude.puech@inria.cl

EMMANUELPIETRIGA

Phd en Ciencias de la Computación del INPG, Francia (2002). Trabajó para INRIA y el Xerox Research Centre Europe, y realizó un postdoctorado en el equipo W3C del MIT. Actualmente es investigador científico en Inria en Chile y en Francia. Sus intereses de investigación se centran en técnicas de interacción para interfaces de usuario multiescala, muros de pantallas, y técnicas de visualización para datos masivos.

emmanuel.pietriga@inria.cl

MARÍAJESÚSLOBO

Magíster en Ciencias de la Ingeniería e Ingeniera Civil Industrial con mención en Tecnologías de la Información de la Universidad Católica. Sus intereses de investigación se centran en informática educativa y técnicas de interacción y visualización para sistemas geográficos. Actualmente es parte del equipo Datos Masivos en Inria Chile.

maria.lobo@inria.cl

Usuarios de sistemas de información geográficos necesitan visualizar grandes mapas y acceder tanto al detalle como al contexto. Los muros de pantallas presentan una alternativa para este tipo de situaciones. Permiten la visualización en detalle de cientos de megapíxeles mediante un conjunto de pantallas de alta resolución. Éstas se disponen en forma matricial, formando así un gran panel, y permiten presentar numerosos datos de diferentes tipos. La navegación física permite, gracias a la alta resolución, percibir el detalle al acercarse a las pantallas y el contexto al alejarse. ANDES es el primer dispositivo de este tipo en Chile y se basa en la experiencia de WILD (Wall-sized interaction with large datasets) en Inria, Francia. Está compuesto de 24 paneles LED de bisel estrecho en una matriz de 6x4, piloteada por un cluster de 12+1 computadores de alto rendimiento y posee interfaz táctil. Con aproximadamente seis meses de funcionamiento ANDES se encuentra a disposición de la comunidad científica y es un soporte útil y flexible para el análisis visual y el manejo de datos masivos.

MOTIVACIÓN

La implementación de un muro de pantallas trae consigo desafíos inherentes al dispositivo. Por un lado, cómo diseñar técnicas de interacción y visualización que utilicen al máximo las capacidades del muro y que permitan la interacción mediante diferentes aparatos de entrada. Los tradicionales como el mouse y el teclado no son adecuados, por lo que se prefiere la interacción mediante dispositivos novedosos como tablets y smartphones. Por otro lado, cómo componer la representación gráfica utilizando de forma eficiente la potencia del cluster, tanto para la distribución de los datos como para la sincronización del render. Cada terminal se puede utilizar para mostrar el mismo contenido o una parte de éste. En este último caso se requiere alto rendimiento para dar la impresión de una imagen única y continua durante la navegación. Los usuarios, además, utilizan diferentes tipos de contenido: algunos pasivos como archivos pdf e imágenes, y otros activos como páginas web. Esto hace que se requiera integrar estas fuentes de datos heterogéneas en un entorno homogéneo. Finalmente, se necesita facilitar la tarea de los desarrolladores, agilizando el desarrollo, testeo y la depuración de prototipos. En ANDES y WILD se considera una arquitectura modular que desacopla la interacción, el render gráfico y el manejo de contenido mediante librerías desarrolladas especialmente.

→

Dado que algunos sets de datos son demasiado grandes incluso para su visualización completa en ANDES, se han diseñado técnicas especiales de navegación para su visualización. OpenStreetMap, por ejemplo, en su máximo nivel de detalle para todo el mundo abarcaría 18 * 10^15 píxeles. Se cuenta también con imágenes artísticas gigantescas, como el panorama de París de 26 giga píxeles o el de Shangai de más de 200 giga píxeles. En estos casos, se utiliza el conjunto de monitores como una gran pantalla donde es posible mostrar y navegar sobre estos grandes objetos [2]. Ya existen aplicaciones que presentan imágenes astronómicas de billones de píxeles (de Spitzer, ESO), donde el detalle es accesible mediante zoom. Otra técnica para acceder a la información detallada es mediante la analogía de las lupas, donde se magnifica una región elegida y se presenta como parte del contexto [3].

Alternativamente, es posible manejar cada pantalla de forma individual con el objetivo de comparar representaciones relacionadas. En estudios sobre WILD en Francia, por ejemplo, se utilizaron para presentar 64 diferentes cerebros en 3D para identificar los que padecían de una patología específica. Una maqueta física de un cerebro se ideó como forma de interacción con el sistema para controlar la orientación de todas las representaciones expuestas en el muro. La comparación podría ser útil igualmente para simulaciones de modelación matemática, al comparar resultados con variaciones en los parámetros. Por ejemplo, en Inria Chile, se trabaja en modelos que simulan el viento para aplicaciones de energía eólica, que dependen de parámetros como el número de molinos. ANDES podría ser útil para comparar de forma sencilla diferentes simulaciones, al mostrarlas en paralelo en las pantallas.

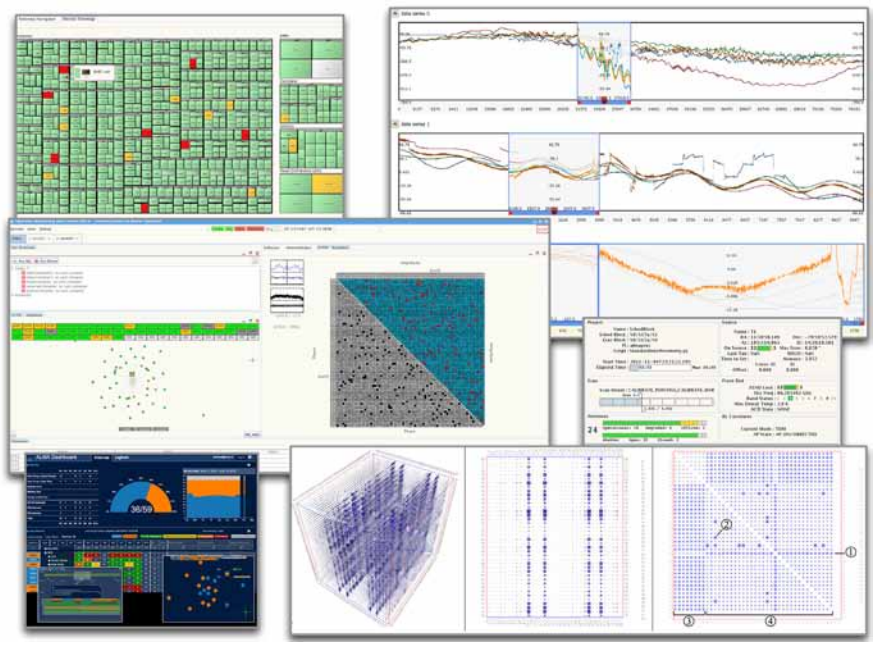


FIGURA 1.
EJEMPLOS DE VISUALIZACIONES INTERACTIVAS DISEÑADAS POR INRIA CHILE PARA LA SALA DE CONTROL DE ALMA.

FUNCIONAMIENTO E IMPLEMENTACIÓN

El desarrollo de aplicaciones para el muro de pantallas se basa principalmente en dos librerías open source: ZVTM [4] (http://zvtm.sourceforge.net) y jBricks [5]. ZVTM permite la implementación de interfaces de usuarios multiescala para trabajar en espacios complejos. Se basa en la existencia de múltiples espacios infinitos dispuestos en capas, donde cada uno puede poseer numerosos objetos gráficos. Las capas son visibles a través de cámaras que observan una región definida.

La librería jBricks funciona en complemento a ZVTM y permite separar el render gráfico en los diferentes computadores del cluster de la aplicación, ocultando así la complejidad asociada. Permite la reutilización de código; mediante cambios en pocas líneas de código una aplicación desarrollada con ZVTM para un computador de escritorio puede ser llevada al muro. Cada computador posee una réplica de la aplicación o un cliente que realiza el render de una parte de la escena. Además, cada uno sabe qué parte debe representar y se le asigna una cámara para mostrar solo el área que corresponde. Una aplicación máster sincroniza los cambios en la escena y en las cámaras. Además de imágenes y gráficos vectoriales, ZVTM posibilita la utilización de varios tipos de datos como documentos PDF. Esta librería ha demostrado ser útil en múltiples escenarios, se utiliza por ejemplo en el desarrollo de visualizaciones interactivas avanzadas para el telescopio ALMA (**Figura 1**).

El manejo de input mediante jBricks es genérico, permitiendo así diversos dispositivos de entrada y la comunicación con aplicaciones distintas. Las formas de interacción pueden modificarse sin cambios en el código de la aplicación; mediante el protocolo, cada instrumento sabe con qué objetos puede interactuar. Para la creación y edición de tipos de entrada jBricks soporta protocolos como USB-HID, OSC, TUIO y VRPN, y variados dispositivos: mouse, teclado, tablets, controles de Nintendo Wii, dispositivos de rastreo de movimiento, etc.

Desde hace poco, además, es posible utilizar Smarties (http://smarties.lri.fr) [6]. Smarties es una librería que permite la interacción de aplica-

ciones creadas para el muro con múltiples dispositivos móviles Android simultáneamente. Smarties combina una aplicación de entrada para los dispositivos móviles, un protocolo de comunicación entre la aplicación para el muro (servidor) y los dispositivos (clientes) y una librería que esconde lo anterior al desarrollador. De esta forma, el servidor recibe los eventos de los dispositivos y puede actuar de acuerdo a ellos agregando pocas líneas de código en la aplicación del muro. La aplicación android reconoce los gestos táctiles en las tablets y es posible agregarle widgets (botones, checkbox, etc) para la interacción con el usuario. Los usuarios intervienen en el muro mediante los llamados pucks. Estos objetos circulares se comportan como una extensión del mouse y permiten manipular los objetos presentes en el muro. Igualmente, se pueden compartir entre los diferentes usuarios creando así un ambiente colaborativo.

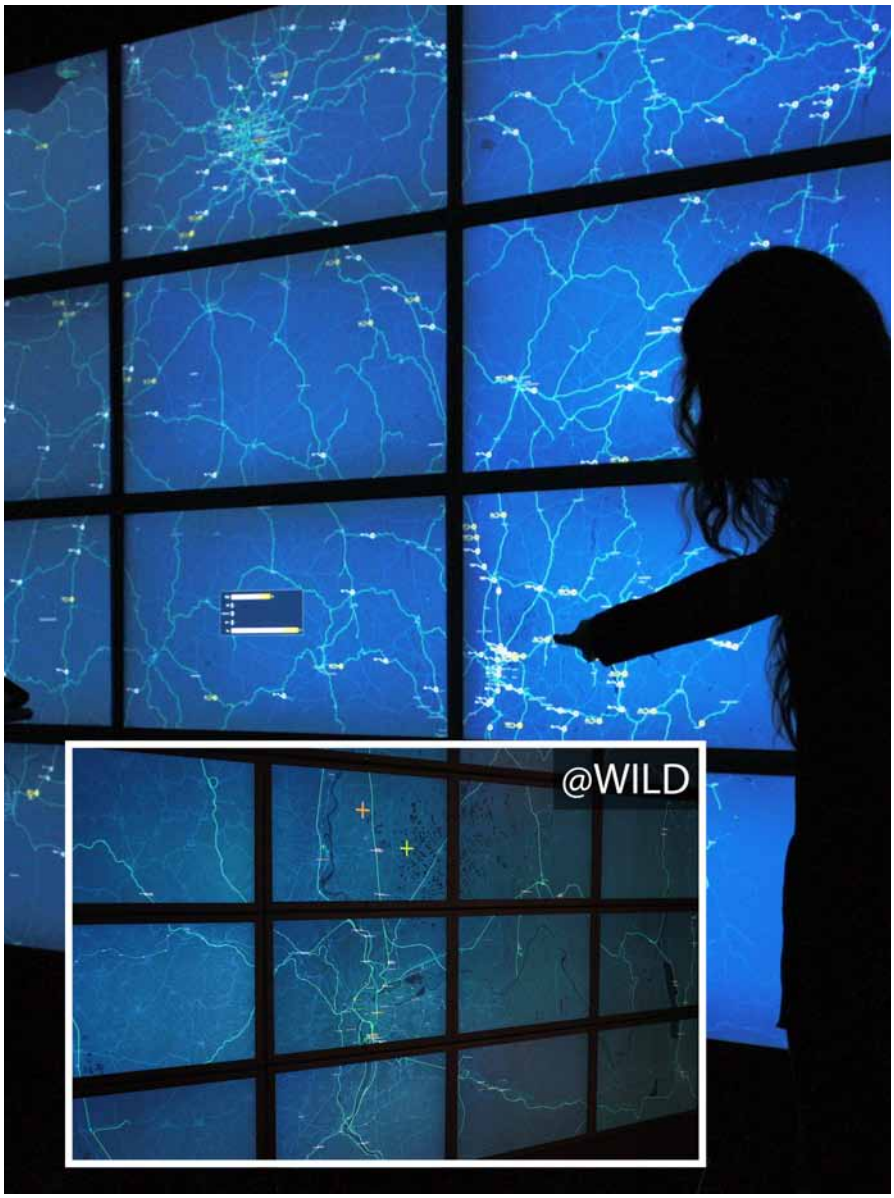


FIGURA 2.
APLICACIÓN QUE PRESENTA EL ESTADO DE LOS TRENES EN TIEMPO REAL EN FRANCIA, CON DATOS DE WWW.RAILDAR.FR LA INTERACCIÓN SE REALIZA MEDIANTE UNA TABLET UTILIZANDO SMARTIES.

ANDES:

DETALLES TÉCNICOS

- ANDES está compuesto por 24 pantallas FullHD de bisel estrecho para una resolución total de 11520 x 4320 píxeles.

- Posee una interfaz táctil mediante sensores infrarrojos.

- Utiliza 24 servidores nVidia Quadro 2000 en 12 64-bit Dell Precision R5500, ejecutándose con Linux y cada uno equipado con:

  - UCP: 2x Intel Xeon E5606.
  - Memoria: 12GB DDR3 RAM.
  - Capacidad de almacenamiento basada en SSD.

ALMA

Situado a 5,000 metros de altura en el desierto de Atacama, Alma es el radiotelescopio submilimétrico más grande del mundo. Posee una resolución espacial y espectral superior por dos órdenes de magnitud a los radiotelescopios existentes y permite acceder a imágenes de formaciones de planetas y estrellas nunca antes vistas. Las interfaces gráficas de ALMA se deben adecuar a la complejidad del sistema y a la gran cantidad de datos que los operadores y los astrónomos deben manejar. Para resolver este problema, se ideó un diseño centrado en el usuario usando técnicas propias del área de estudio Interacción Humano-Computador. Luego de entrevistas a operadores y astrónomos, prototipos rápidos y workshops participativos de diseño se llegó a una solución que incorpora técnicas novedosas de interacción y visualización. Con éstas se construyen visualizaciones escalables del estado del sistema y vistas que relacionan los diferentes componentes mediante interfaces con diferentes niveles de detalle, consultas dinámicas, múltiples vistas coordinadas, y herramientas adecuadas al tipo de datos, por ejemplo visualizaciones avanzadas para series de tiempo de monitoreo. Todo esto en base a la librería ZVTM. Alma utiliza la red REUNA, de la cual Inria forma parte desde el año pasado y que permite una conectividad rápida y confiable.

CONCLUSIÓN

ANDES PRESENTA VARIAS VENTAJAS PRODUCTO DE SUS CARACTERÍSTICAS TÉCNICAS Y DE LAS LIBRERÍAS QUE SE HAN DESARROLLADO PARA SU UTILIZACIÓN. POR UN LADO, DEBIDO A SU TAMAÑO Y A LA INTERACTIVIDAD DISPONIBLE, ES UNA PLATAFORMA QUE SE PRESTA PARA FACILITAR EL TRABAJO COLABORATIVO. PARA ESTO ES NECESARIO IDEAR HERRAMIENTAS DIFERENTES A LAS TRADICIONALES, COMO EL TECLADO O EL MOUSE. SMARTIES, A TRAVÉS DE DISPOSITIVOS ANDROID, PERMITE POR EJEMPLO QUE VARIAS PERSONAS PUEDAN MANTENER UN ESPACIO DE TRABAJO Y DE VISUALIZACIÓN COMÚN Y TAMBIÉN INTERACTUAR DE FORMA INDIVIDUAL. POR OTRO LADO, PERMITE COMBINAR Y VISUALIZAR DIFERENTES TIPOS DE DATOS DE FORMA EFICIENTE, POR EJEMPLO VÍDEOS, MODELOS EN TRES DIMENSIONES Y ARCHIVOS EN PDF. EL ORIGEN DE LOS DATOS ES TAMBIÉN FLEXIBLE; NO ES NECESARIO TENERLOS DE FORMA LOCAL PARA MOSTRARLOS. DE HECHO, RECIENTEMENTE SE DESARROLLÓ EN INRIA CHILE UNA APLICACIÓN PARA VISUALIZAR EL ESTADO DE LOS TRENES EN FRANCIA CON LOS DATOS DEL SITIO WWW.RAILDAR.FR **(FIGURA 2)**. LA INFORMACIÓN SE RECOLECTA EN TIEMPO REAL MEDIANTE CONSULTAS A UN SERVICIO WEB Y SE PROCESA DENTRO DE UNA APLICACIÓN CONSTRUIDA EN BASE A ZVTM QUE DESPLIEGA LA INFORMACIÓN EN EL MURO.

EN CONCLUSIÓN, ANDES ES UNA PLATAFORMA ÚNICA PRODUCTO DE LA VARIEDAD DE USOS Y LA POTENCIA DE LAS CAPACIDADES DE EXPLORACIÓN QUE OFRECE. PERMITE ESPECIALMENTE:

- VISUALIZAR E INTERACTUAR CON GRANDES IMÁGENES IMPOSIBLES DE PRESENTAR EN OTROS DISPOSITIVOS, COMO LAS ASTRONÓMICAS. EN ESTE CASO, ADEMÁS, ES POSIBLE COMPARAR DE FORMA INTERACTIVA IMÁGENES CORRESPONDIENTES A DIFERENTES LONGITUDES DE ONDA CONSERVANDO UNA PERSPECTIVA GENERAL.

- VISUALIZAR Y EXPLORAR OBJETOS COMPLEJOS EN LOS QUE SE REQUIERE PRESENTAR TANTO EL DETALLE COMO EL CONTEXTO, COMO SISTEMAS DE INFORMACIÓN GEOGRÁFICOS.

- VISUALIZAR Y MANIPULAR REPRESENTACIONES ABSTRACTAS DE OBJETOS COMPLEJOS, COMO PODRÍA SER EL ESQUEMA DE FUNCIONAMIENTO DE UNA FÁBRICA.

- VISUALIZAR SETS DE ELEMENTOS RELACIONADOS, COMO DIFERENTES RESULTADOS DE SIMULACIONES MATEMÁTICAS, Y ASÍ FACILITAR LA COMPARACIÓN Y EL CONTRASTE ENTRE ELLOS.

ANDES SE ENCUENTRA ACTUALMENTE EN LAS OFICINAS DE INRIA CHILE EN SANTIAGO. INRIA CHILE ES UNA FUNDACIÓN CREADA POR INRIA (INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE), INSTITUCIÓN PÚBLICA FRANCESA DEDICADA A LA INVESTIGACIÓN Y EL DESARROLLO DE TECNOLOGÍAS DIGITALES DEL FUTURO DESDE LAS CIENCIAS DE LA COMPUTACIÓN Y LAS MATEMÁTICAS APLICADAS, Y FORMA PARTE DE LA PRIMERA GENERACIÓN DE CENTROS DE EXCELENCIA INTERNACIONAL SELECCIONADOS POR CORFO. SU OBJETIVO PRINCIPAL ES REALIZAR TRANSFERENCIA TECNOLÓGICA ENTRE LA INVESTIGACIÓN CIENTÍFICA Y LAS EMPRESAS CHILENAS. INRIA CHILE PONE A DISPOSICIÓN ANDES PARA LA INNOVACIÓN EN EMPRESAS Y EN LA COMUNIDAD CIENTÍFICA. SE ENCUENTRA ACTUALMENTE DESARROLLANDO APLICACIONES TANTO PARA LA ASTRONOMÍA, CON EL PROPÓSITO DE MOSTRAR GRANDES COLECCIONES DE IMÁGENES FITS Y SU INFORMACIÓN ADICIONAL PROVENIENTE DE CATÁLOGOS ASTRONÓMICOS EN LÍNEA, COMO PARA VISUALIZAR INFORMACIÓN DE SISTEMAS DE INFORMACIÓN GEOGRÁFICOS. ESPERAMOS QUE SE TRANSFORME EN UNA PLATAFORMA VALIOSA A TRAVÉS DE LAS HERRAMIENTAS NOVEDOSAS QUE OFRECE PARA LA INTERACCIÓN CON GRANDES VOLÚMENES DE DATOS EN LA INDUSTRIA CHILENA. ■

REFERENCIAS

[1] M. R. Endsley and D. G. Jones, editors. Designing for Situation Awareness: an Approach to User-Centered Design. CRC Press, Taylor & Francis, 2012.

[2] Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., & Mackay, W. (2011, May). Mid-air pan-and-zoom on wall-sized displays. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 177-186). ACM.

[3] C. Pindat, E. Pietriga, O. Chapuis, and C. Puech. Jellylens: content-aware adaptive lenses. In Proceedings of the 25th annual ACM symposium on User interface software and technology, UIST '12, pages 261–270. ACM, 2012.

[4] E. Pietriga. A toolkit for addressing HCI issues in visual language environments. In Proc. VL/HCC'05, 145–152. IEEE, 2005.

[5] Pietriga E., Huot S., Nancel M. & Primet R. Rapid development of user interfaces on cluster-driven wall displays with jbricks. EICS '11, pages 185-190. ACM, 2011.

[6] O. Chapuis, A. Bezarianos, and S. Frantzeskakis. Smarties: An input system for wall display development. In Proceedings of the 32nd international conference on Human factors in computing systems, CHI'14, pages 2763-2772 (10 pages). ACM, 2014.

EL MODELO DETRÁS DE MAPREDUCE

MapReduce es un modelo de programación orientado al procesamiento de grandes volúmenes de datos, desarrollado específicamente para sacar ventaja del poder de procesamiento paralelo que ofrecen los clusters de computadores. Desde que Google lo introdujera públicamente en 2004 [1], hemos visto una proliferación de aplicaciones que usan MapReduce para realizar cómputos en paralelo. Google afirma realizar varios miles de procesos día a día usando MapReduce, al igual que otros de sus usuarios más famosos como Yahoo!, Facebook y Twitter. Existen también aplicaciones especializadas en ciertos dominios basadas en MapReduce, como por ejemplo Pig, un sistema de bases de datos relacionales con un lenguaje similar a SQL; Apache Hive, una herramienta para Business Intelligence; y ZooKeeper, que provee servicios para la configuración de sistemas distribuidos.

JUAN REUTTER

Profesor Asistente del Departamento de Ciencia de la Computación de la Pontificia Universidad Católica de Chile. Recibió su Doctorado en la Universidad de Edimburgo en mayo de 2013. Sus intereses se enmarcan en el área de Manejo de Datos, incluyendo Sistemas de Bases de Datos, Lenguajes de Consulta, Web Semántica y Lenguajes Formales.

jreutter@ing.puc.cl

MapReduce ha sido llamado un "cambio de paradigma" en Computación [2]. Sus adeptos no dudan en asegurar que el uso de MapReduce es capaz de acelerar los cómputos de cualquier aplicación. Algunos incluso han llegado a afirmar que MapReduce va a dejar obsoletos a los sistemas de bases de datos relacionales. Y las cifras parecen avalar este cambio: la firma Cloudera afirma que al año 2013 la mitad de las compañías en el listado Fortune 50 (un listado de las compañías más importantes de los Estados Unidos) usa MapReduce en alguno de sus procesos [3].

A pesar de esta aparente colonización de MapReduce en aplicaciones que lidian con grandes volúmenes de datos, también hay quienes ponen en duda su eficacia. En efecto, los detractores de MapReduce terminan siendo igual de duros que sus adeptos, y han llegado a afirmar que "MapReduce es un gran paso hacia atrás (para la comunidad de manejo de datos)". Es más, estudios de la Universidad de Cornell afirman que la solución ofrecida por MapReduce es generalmente menos eficiente que la obtenida al usar otros modelos más clásicos de computación paralela [4]. Académicos de la Universidad de Brown también han llegado a una conclusión similar: comparando sistemas de bases de datos tradicionales con sistemas basados en MapReduce, concluyeron que los sistemas tradicionales consiguen un mejor rendimiento en casi todas las tareas, con excepción de una familia particular de tareas llamadas Extraer, Transformar y Cargar (ETL por sus siglas en inglés) [5].

Esta controversia se genera porque nadie conoce con certeza en qué casos MapReduce ofrece una aceleración significativa a la hora de realizar cómputos. Por lo mismo, es difícil saber cuánto de la popularidad de MapReduce se debe a las ventajas computacionales de su arquitectura, cuánto se debe a su simpleza y cuánto se debe al bombardeo publicitario que sus desarrolladores continúan llevando a cabo.

En el pasado la Ciencia de la Computación logró resolver problemas similares en otros contextos, mediante el desarrollo de teorías capaces de entender y modelar el comportamiento de distintos paradigmas de computación. De la misma forma, para poder hacer un análisis más preciso de las ventajas y desventajas de esta arquitectura y el tipo de problemas para los que es beneficioso usar MapReduce, necesitamos un modelo formal que sea capaz de predecir el comportamiento de las aplicaciones MapReduce en la práctica. En este artículo discutiremos las características mínimas que debe tener este modelo y hacia dónde tiene que avanzar la comunidad científica para lograrlo.

EL MODELO DE MAPREDUCE

El modelo de MapReduce es simple. Existen dos funciones, llamadas Map y Reduce, dividiendo la computación en dos etapas: primero se ejecuta un número de llamadas a Map (en forma paralela); tras lo cual los datos entregados por estas funciones son recolectados y ordenados,

→

y luego con estos datos se ejecuta un número de llamadas a Reduce.

Más específicamente, y siguiendo el modelo de Rajaraman y Ullman [6], podemos resumir una ejecución de MapReduce de la siguiente forma:

1. El sistema solicita un número de funciones Map y entrega a cada una de éstas un pedazo de los datos a procesar (que generalmente provienen de un sistema de manejo de datos distribuidos). Cada función Map transforma estos datos en una secuencia de pares (llave,valor).

2. Un controlador maestro recolecta todos los pares (llave,valor) de todas las funciones Map y los ordena de acuerdo a sus llaves. Luego distribuye todas las llaves sobre varias funciones Reduce, de forma que todos los pares (llave,valor) que tienen la misma llave van a parar a una misma copia de Reduce.

3. Cada Reduce toma todos los valores que recibe para una misma llave y realiza algún tipo de cómputo con los valores que recibe.

4. El resultado final de la operación es la combinación de los resultados de cada una de las funciones Reduce.

A modo de ejemplo, imaginemos que tenemos un archivo de texto y queremos contar cuántas veces se repite cada palabra en este documento. La función Map debería recibir un pedazo de este documento, separarlo en palabras distintas, y emitir, para cada ocurrencia de una palabra *w* en el texto, el par (*w*,1), donde *w* es la llave, y el valor corresponde a 1. La función Reduce entonces recibirá una secuencia de pares (llave,valor) idénticos (*w*,1),...,(*w*,1), y debe contar el número de ocurrencias de este par que recibe, lo que corresponde exactamente a la cantidad de veces que aparece la palabra *w* en el documento. Como el controlador llama a una función Reduce por cada palabra distinta en el documento, una vez finalizado el proceso de cómputo cada una de las funciones Reduce nos entregará el número de ocurrencias de una palabra en particular.

MAPREDUCE PARA TRABAJAR GRANDES VOLÚMENES DE DATOS

Como ya hemos mencionado, nos interesa construir un modelo teórico que sea capaz de clasificar a los problemas de acuerdo a la eficacia con que pueden ser resueltos usando MapReduce, dividiéndolos en dos grupos: aquellos para los que es evidentemente beneficioso usar MapReduce, y aquellos para los que no es así.

Para comenzar es necesario mirar dónde están los cuellos de botella de MapReduce. Volvamos a nuestro problema de contar las palabras del documento. En este caso, cada mapper procesará solo un pequeño pedazo del documento, contará las ocurrencias de las palabras de su pedazo y las enviará a los *reducers* para que agrupen la información. Si asumimos que estamos procesando un texto en lenguaje natural, la comunicación estará bien distribuida, ya que vamos a invocar a una función Reduce distinta por cada una de las palabras del documento.

Otro ejemplo típico de la utilidad de MapReduce es el *join* de dos relaciones en una base de datos relacional. Imaginemos que tenemos una relación *Teléfono*, con atributos *nombreUsuario* y *numTelefono*, que almacena nombres de personas y sus números de teléfono; y otra relación Dirección, con atributos *numTelefono* y *direccionTelefono*, que almacena la dirección asociada a cada número telefónico. Si queremos computar los nombres de las personas junto a sus direcciones, tenemos que hacer un *join* entre las tablas Teléfono y Dirección, uniendo los registros asociados al mismo número telefónico. Podemos resolver este problema fácilmente en MapReduce: la función map identifica como llave a los números telefónicos en las tablas Teléfono y Dirección, de forma que la función Reduce reciba los nombres y las direcciones de todos los registros asociados a un mismo teléfono y compute la respuesta de manera local. Nuevamente, la comunicación enviada por cada *mapper* estará dividida en varios *reducers* (uno por cada teléfono) y los *reducers* recibirán pedazos pequeños de la base de datos.

Los dos problemas que hemos visto tienen un punto en común: en ambos casos el cómputo puede ser efectivamente dividido en múltiples pedazos, y luego fácilmente vuelto a reunir. Pero, ¿qué pasa cuándo no tenemos esa garantía, cuando el problema no se puede dividir?

Consideremos por ejemplo el problema de conectividad de un grafo: se tiene un gran número de nodos, los que están conectados entre sí mediante aristas. El problema de conectividad busca saber si un determinado nodo n_1 está conectado a otro nodo n_2. Este problema es importante, por ejemplo, en redes sociales: los nodos representan personas, las aristas son las relaciones entre estas personas, e interesa saber si una persona está "conectada" a otra.

¿Cómo podemos usar MapReduce para resolver el problema de conectividad? Este problema ha sido estudiado con bastante entusiasmo y aún no hay una respuesta clara. El consenso es que, en general, utilizar MapReduce para resolver conectividad no es una buena alternativa. De hecho, si nos ponemos en el peor de los casos, puede que el camino entre ambos nodos sea extremadamente largo, y los nodos y relaciones que participan en el camino entre n_1 y n_2 estén todos divididos en *mappers* distintos. La única solución parece ser juntar todos los pedazos en un mismo *reducer*, pues necesitamos de todo el grafo para resolver el problema. Si nuestro grafo es una gran red social, ¡esto es tremendamente ineficiente!

A través de estos ejemplos hemos observado dos conclusiones clave para nuestro análisis: por un lado, los problemas en que podemos dividir el cómputo en pedazos pequeños y luego unirlos directamente parecen darse bien en MapReduce. Pero además nos interesa restringir la comunicación entre cada *mapper* o cada *reducer*, de forma que todos los datos no vayan a parar a un número muy pequeño de *reducers*.

De esta forma, nuestro modelo teórico debe tomar en cuenta estas conclusiones a la hora de clasificar los problemas. Pero lamentablemente las teorías actuales que intentan discernir cuando un problema es o no paralelizable no son satisfactorias en nuestro contexto. Para empezar, los modelos teóricos actuales de computación paralela se basan casi siempre en una arquitectura en la que todos los nodos tienen acceso gratis al conjunto de los datos (llamadas *share everything*, o compartirlo todo). En cambio MapReduce es una arquitectura en la que los *mappers* no tienen más acceso que a su pedazo de input y los *reducers* tienen cada uno la información asociada a una llave (es prácticamente una arquitectura tipo *shared nothing,* o compartir nada). En complejidad computacional, por otro lado, se asocia la característica de "poder ser computados en paralelo" a una clase muy simple de problemas que excluye a muchísimos ejemplos que usan MapReduce para buenos resultados en la práctica (ver p.ej. [7]).

UN MODELO TEÓRICO PARA MAPREDUCE

Teóricamente, podemos modelar todo algoritmo de MapReduce con dos funciones, *M* y *R*, por Map y Reduce. Tal como ocurre en la práctica, la función *M* toma como input una secuencia de pares llave-valor y retorna una secuencia de pares llave-valor; y la función *R* toma como input una llave junto a una secuencia de valores y genera nuevamente una secuencia de pares (llave,valor). Dado un problema arbitrario a ser resuelto con MapReduce, asumimos que el input a ese problema está dividido en pedazos $p_1,\ldots, p_n$. La función *M* (*Map*) toma un pedazo de input p_i y lo transforma en una secuencia de pares *(llave,valor)* $M(p_i) = \{(k_1, v_1),\ldots,(k_m, v_m)\}$. Posteriormente se toma la unión de todos los pares (llave,valor) de todas las llamadas a *M*; junta todos los valores asociados a una misma llave, genera una copia de la función *R* por cada llave, y le entrega a *R* esta llave junto con todos sus valores asociados.

De esta forma, para funciones *M* y *R* dadas, el resultado de aplicar MapReduce sobre un input $(1, p_1),\ldots,(n, p_n)$ se define de la siguiente forma. Primero, el resultado de agrupar el output de todas las funciones *M* corresponde a

$$Map((1, p_1),\ldots,(n, p_n)) = \bigcup_{i=1}^{i=n} M(p_i)$$

Este conjunto corresponde a una secuencia de pares (llave,valor). Posteriormente, para cada llave *k* que sea parte de algún par en $Map((1,p_1),\ldots,(n,p_n))$, definimos el conjunto de los pares (llave,valor) asociados a *k* como $pares(k)=\{(k,v) \in Map((1,p_1),\ldots,(n,p_n))\}$. El resultado de MapReduce sobre $(1, p_1),\ldots,(n, p_n)$ se define como

$$MR((1, p_1),\ldots,(n, p_n)) = \bigcup_k R(pares(k)) \quad (1)$$

Es decir, el resultado de MapReduce es el resultado de aplicar *R* sobre cada una de las llaves, junto a sus valores asociados, que pertenecen a la unión de las secuencias llave-valor entregadas por cada llamada a *M* para cada par (i, p_i) del input.

Por ahora no hemos hecho nada más que representar las funciones de MapReduce de forma matemática, pero tenemos que avanzar mucho más si queremos un modelo que cumpla nuestra meta. Para empezar, no hemos especificado ninguna condición sobre la forma en que dividimos nuestro input, por lo que, hasta ahora, podemos simular cualquier algoritmo *A* que computa cierto problema de forma serial (es decir, no en paralelo) de la siguiente forma: para cada input *p* para *A*, generamos una función *M* que reciba *p* y retorne (1, *p*), y definimos a *R* de forma que reciba (1, *p*) y ejecute *A* sobre *p*. En otras palabras, *M* no hace más que generar una llave arbitraria para *p*, y *R* es una copia de *A*. El resultado, obviamente, es el mismo que en el caso serial, aunque no nos estamos aprovechando en absoluto de la arquitectura paralela de MapReduce, porque siempre ejecutaremos una sola función *M* y una sola función *R*. Y no es realista pensar que el input va a venir bien formado en un solo pedazo; por el contrario, como el input es generalmente muy grande, lo más lógico sería pensar que está almacenado en un gran número de pedazos en algún sistema de almacenamiento distribuido de datos.

Para agregar esta restricción a nuestro modelo, fijamos con anterioridad el número de *mappers* (o llamadas a la función *M*) a usar. Para un número *n* arbitrario y una secuencia de *n* pares (llave,valor) $S = \{(1, p_1),\ldots,(n, p_n)\}$, hablaremos de la ecuación (*1*) como el resultado de aplicar MapReduce "usando *n* mappers" sobre la secuencia *S*. Si *n*=1, entonces claramente estamos hablando de un algoritmo que no es paralelo. Pero si asumimos que *n* es relativamente grande, volvemos a nuestra pregunta original, esta vez más refinada: ¿Qué problemas podemos implementar usando MapReduce con *n* mappers?

Con esto podemos modelar el hecho de que nuestro input esta bien distribuido, pero nuevamente podemos simular cualquier algoritmo *A*, incluso si asumimos que *n* es grande. Imaginemos que dividimos arbitrariamente el input *p* de *A* en una secuencia de pares (llave,valor) $S = \{(1, p_1),\ldots,(n, p_n)\}$. Definamos la siguiente función *M*: toma un par (i, p_i) y entrega el par $(1, p_i)$. El resultado de aplicar *n mappers* sobre *S es* $\{(1, p_1),\ldots,(1,p_n)\}$. Ahora podemos definir *R* como la función que une nuevamente todos los pedazos y luego llama a *A*. Si bien esta vez comenzamos con un sistema de datos distribuidos, lo único que hace nuestro algoritmo MapReduce es juntar todos esos pedazos en uno y pasárselo todo a un *reducer*; por lo que nuevamente ejecutamos *A* en un solo nodo de nuestra arquitectura.

→

¿Cómo podemos entonces modelar todas las ejecuciones de MapReduce en las que aprovechamos, en cierta forma, la arquitectura paralela? Una alternativa es limitar la comunicación entre los *mappers* y los *reducers*. Para explicar esto, volvamos a nuestra algoritmo *A* que recibe input *p*, y supongamos que el tamaño de nuestro input es *t* (podemos pensar en *t* como el número de caracteres o de bits que tiene *p*). Si usamos *n mappers*, y distribuimos el input de forma equitativa sobre cada *mapper*, éstos van a recibir un input de tamaño *t/n*. Vamos a obligar a que el output de la función *M*, para cada llave distinta, sea siempre menor que *t/n*. Para ser más formales, digamos que el output de *M*, para cada llave distinta que genere, no puede ser mayor al logaritmo *log(t)* de *t*: sabemos que, para valores grandes de *t*, *t/n* siempre va a ser mayor a *log(t)*.

Hemos llegado a nuestro modelo definitivo. Decimos que un algoritmo *A* puede ser simulado con MapReduce usando *n mappers* si existen funciones *M* y *R*, tal que:

(c.1) Para todo input *p* de tamaño *t* y toda división de un input *p* para *A* en una secuencia $S = \{(1, p_1),\ldots,(n, p_n)\}$, el resultado de la ecuación (1) es equivalente al resultado de aplicar *A* a *p*.

(c.2) La suma del tamaño de los valores de $M(p_i)$ asociados a cada llave no puede superar *log(t)*.

Finalmente, como nos interesan los casos en los que MapReduce consigue computar más rápidamente el algoritmo *A* que el resultado de ejecutar *A* en forma serial, sin paralelismo, agregaremos una tercera condición:

(c.3) El tiempo que demora el mejor algoritmo en serie para computar *A* es siempre mayor o igual al tiempo total que demora computar *A* usando MapReduce, es decir, el máximo entre el tiempo que demora cada *M* sumado al máximo del tiempo que demora cada *R*.

Podemos usar este modelo de la siguiente forma: si podemos simular un problema de acuerdo a estas características, entonces MapReduce es un buen candidato para resolver este problema. Por ejemplo, es evidente que los problemas de contar ocurrencias de palabras y de join podrán ser simulados con nuestro modelo, siempre y cuando el input nos asegure una buena distribución. Es razonable pensar que el problema de conectividad no puede ser resuelto de esta forma, pero, ¿podemos demostrar formalmente este hecho para éste u otro problema? Más importante aún, ¿podemos afirmar que los problemas que no podemos simular con este modelo no se dan bien en MapReduce? La respuesta a estas preguntas tendría una implicancia inmediata en la práctica, al momento de pensar si instalar o no MapReduce para resolver un problema.

Otra dirección importante es la aplicación de rondas sucesivas de MapReduce: hemos señalado que el resultado de MapReduce puede ser la entrada de otro algoritmo de MapReduce distinto. Entonces es natural preguntarnos: ¿existen problemas que no puedan ser simulados usando una ronda de MapReduce, pero que sí lo sean si usamos dos rondas? Y, ¿qué hay de un número mayor de rondas? ¿Es cierto que el problema de conectividad puede ser simulado si usamos un número de rondas igual al tamaño del grafo original, o igual al logaritmo del tamaño del grafo original?

Finalmente, existen otras arquitecturas paralelas bastante simples que se usan hoy en la práctica, siendo quizás los ejemplos más importantes GraphLab o Pregel. Es necesario trabajar en una definición formal similar a lo hecho con MapReduce, para determinar el alcance de éstas otras arquitecturas. Esta tarea quizá podría llevarnos a descubrir una teoría mucho más general de computación paralela. ■

BIBLIOGRAFÍA

[1] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113.

[2] Patterson, D. A. (2008). The data center is the computer. Communications of the ACM, 51(1), 105-105.

[3]Prn news blog, http://www.prnewswire.com/news-releases/altiors-altrastar---hadoop-storage-accelerator-and-optimizer-now-certified-on-cdh4-clouderas-distribution-including-apache-hadoop-version-4-183906141.html

[4] Stampede Visual Workshop, https://www.cac.cornell.edu/Stampede/default.aspx

[5] Pavlo, A., Paulson, E., Rasin, A., Abadi, D. J., DeWitt, D. J., Madden, S., & Stonebraker, M. (2009, June). A comparison of approaches to large-scale data analysis. In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data (pp. 165-178). ACM.

[6] Rajaraman, A., & Ullman, J. D. (2012). Mining of massive datasets. Cambridge University Press.

[7] Lynch, N. A. (1996). Distributed algorithms. Morgan Kaufmann.

BIG DATA CHILE

TRANSANTIAGO COMO FUENTE DE DATOS

LOS DATOS PASIVOS PUEDEN AYUDARNOS A HACER UNA MEJOR GESTIÓN DE LA CIUDAD

MARCELA MUNIZAGA

Profesora Asociada y Subdirectora del Departamento de Ingeniería Civil Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile. Ingeniero Civil (UCH) y Doctora en Ciencias de la Ingeniería (PUC). Directora Proyecto FONDEF Herramientas avanzadas para la ciudad del futuro. Áreas de investigación: modelamiento del comportamiento de usuarios, recolección y procesamiento de datos.

mamuniza@ing.uchile.cl

La llegada de Transantiago como sistema integrado de transporte público de Santiago, Chile, a partir de febrero de 2007 fue polémica, algo traumática… en fin, podríamos decir mucho, y ya se ha dicho mucho sobre el tema, pero hay algo sobre lo que no se ha hablado tanto, que es el beneficio colateral de la generación constante de enormes cantidades de datos.

En el Departamento de Ingeniería Civil (DIC) de la Universidad de Chile vimos esa oportunidad y en 2008 comenzamos a trabajar con los datos en un proyecto PBCT (Programa Bicentenario de Ciencia y Tecnología). El primer desafío fue limpiar y procesar datos de posicionamiento de los más de 6.000 buses del sistema, que cuentan con dispositivos GPS que emiten una señal de posición cada 30 segundos, generando del orden de 80 millones de registros a la semana. Estos permiten observar el movimiento de los buses con un nivel de cobertura y precisión que nunca antes había estado disponible, y generar perfiles de velocidad (**Figura 1**) [Cortés et al., 2011].

Por otra parte están los datos de transacciones bip!, del orden de 35 millones a la semana, que de por sí contienen información valiosa, al mostrar la distribución temporal de la demanda, pero que además al cruzarlos con la base de datos de posicionamiento hace posible asignar posición a los registros de subida (**Figura 2**). Esto hace posible obtener la distribución espacio-temporal de la demanda. Mediante una metodología desarrollada en el DIC, que se basa en observar la secuencia de transacciones de una misma tarjeta [Munizaga y Palma, 2012], se estima el paradero o estación de bajada como aquel más conveniente para acceder a la posición de la siguiente subida, dentro de un radio de 500m. Esto se logra para sobre el 80% de las transacciones, abriendo la puerta a una variada gama de análisis, que incluyen construir matrices origen-destino de viajes en transporte público, construir perfiles de carga de buses y Metro (Gschwender et al., 2012), construir indicadores de calidad de servicio, etc. Todas éstas son valiosas herramientas para quienes están encargados de realizar la planificación y gestión de nuestro sistema de transporte público, que si bien se pueden obtener mediante mediciones, su costo es muy elevado, y su nivel de cobertura espacio-temporal muchísimo menor, dado que existen recursos limitados para realizar este tipo de análisis. Por ejemplo, las matrices origen-destino de viajes tradicionalmente se obtienen de encuestas origen-destino que se realizan a una muestra de la población. En Santiago se realiza una cada diez años, la última se realizó en 2012-2013, con una muestra de alrededor de 18.000 hogares y un costo aproximado de 700 millones de pesos, incluyendo la realización de las encuestas y mediciones complementarias. Los resultados aún no están disponibles, pues con posterioridad al proceso de recolección de datos se requiere un detallado postproceso para filtrar errores y realizar la expansión a la población. Con los datos de transacciones bip! se obtiene información detallada del 80% de los viajes en transporte público, que corresponden a más cuatro millones de viajes diarios de más de dos millones de usuarios (tarjetas bip!), logrando una cobertura espacio-temporal que es imposible alcanzar con datos de encuestas. Mediante postprocesamiento de los datos de transacciones se distingue los transbordos de los destinos de viaje donde los usuarios realizan actividades [Devillaine et al., 2012], y para los usuarios frecuentes (aquellos que viajan al menos cuatro veces a la semana) se identifica la zona de residencia, observando la posición de la primera transacción del día en todos

los días en que la tarjeta es observada. Si éstas tienen coincidencia espacial, se estima que esa zona corresponde al lugar de residencia del usuario (tarjeta). Asimismo, hay otro tipo de análisis que sería interesante realizar, como por ejemplo observar los patrones de viaje y completar información faltante.

Una segunda etapa en el desarrollo de este proyecto es la validación, porque si bien se logra realizar estimaciones de paradero de bajada para sobre el 80% de las transacciones, identificar destinos de actividades para esos viajes, e incluso estimar zona de residencia para los usuarios frecuentes del sistema, se requiere información exógena para comprobar que esas estimaciones sean correctas. Para realizar la validación hasta ahora se ha contado con una pequeña muestra de validación que proviene de la encuesta origen destino de viajes en Metro, realizada en 2010, en que a una muestra de usuarios se les registra el número identificador de la tarjeta bip!, además de aplicárseles la encuesta origen-destino. Los resultados son positivos, pero insuficientes, debido al reducido tamaño muestral (882 encuestas). Sin embargo, próximamente contaremos con los resultados de la EOD 2012-2013, que es una muestra representativa de la población de Santiago, que podrá ser usada para validación, dado que en ésta también se incluyó una pregunta que pide registrar el número identificador de la o las tarjetas que el encuestado utiliza para viajar.

FIGURA 1.
TRAYECTORIA ESPACIO-TIEMPO DE LOS BUSES DE UN SERVICIO.

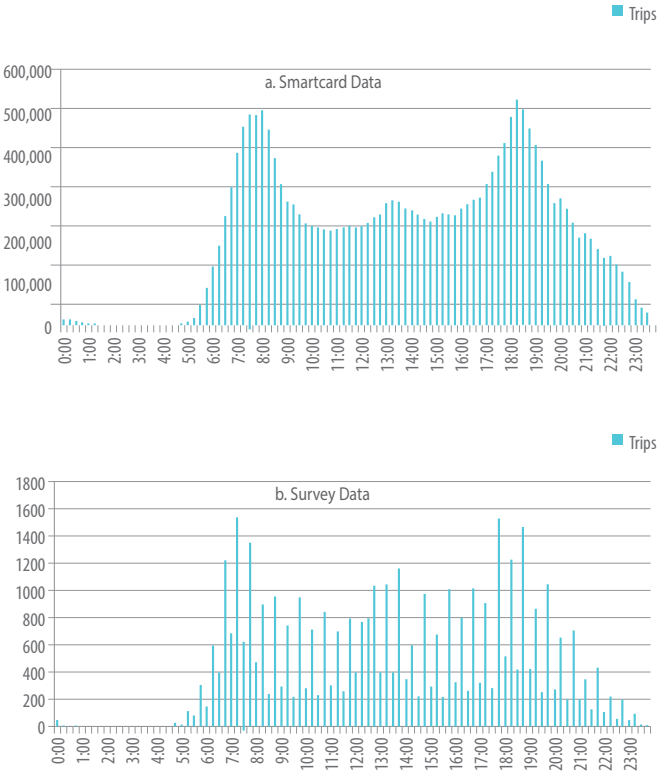


FIGURA 2.
COMPARACIÓN DEL NIVEL DE PRECISIÓN OBTENIDO CON DATOS DE TRANSACCIONES BIP! Y CON DATOS DE ENCUESTA.

Otra línea de desarrollo interesante a partir de este proyecto es la modelación. Lo que se obtiene de las transacciones y GPS de los buses es una observación de la realidad actual, algo así como una foto de alta definición, pero para poder aportar al proceso de toma de decisiones, requerimos más que una foto, necesitamos elaborar modelos de comportamiento que nos permitan predecir qué va a suceder con el sistema si aplicamos cambios en él. Por ejemplo, ¿qué va pasar con las velocidades de los buses y los tiempos de viaje de los usuarios si construimos corredores segregados para los buses? ¿Qué efecto tendrá en la demanda mejorar la regularidad de los intervalos entre pasadas de buses de un mismo servicio? Dado que ya contamos con información de casi ocho años consecutivos, con amplia cobertura espacio-temporal, es posible observar esos cambios y deducir leyes de comportamiento. Ésta es una etapa fascinante en una disciplina que tradicionalmente se ha enfrentado a la escasez de datos. Hay factores relevantes para las decisiones de los usuarios como por ejemplo la variabilidad del tiempo de viaje o el tiempo de espera, que difícilmente han sido analizados con propiedad debido a la escasez de datos. Ahora cambiamos de paradigma, pasamos de la escasez de datos a la abundancia abrumadora de ellos, y el desafío es utilizarlos adecuadamente en beneficio de la sociedad. El gran desafío es utilizar los datos que se generan automáticamente mediante la operación del sistema, para contribuir a mejorar la gestión de la ciudad, haciendo que todos sus sistemas funcionen de forma más eficiente, amable y sustentable. La invitación está abierta a quienes quieran realizar investigación en esta área, porque aún hay mucho por hacer, y las más diversas disciplinas pueden contribuir a ello, incluyendo por cierto a la Ciencia de la Computación.

PROYECTOS QUE HAN CONTRIBUIDO AL FINANCIAMIENTO DE ESTA INVESTIGACIÓN:

- "Herramientas avanzadas para la ciudad del futuro". Proyecto FONDEF D10I1002 2012-2014.

IMAGEN 1.
DE IZQUIERDA A DERECHA: RICHARD IBARRA*, ANTONIO GSCHWENDER, CRISTIÁN HERRERA, MARCELA MUNIZAGA, MAURICIO ZÚÑIGA*, RAMÓN CRUZAT*, CÉSAR NÚÑEZ, NÉSTOR GALLEGOS (*GRADUADOS DEL DCC U. DE CHILE).

- "Modeling public transport demand and operation using detailed information from smart-card and AVL data". FONDECYT Regular 1120288. 2012-2014.

- "Transport demand modelling considering quality of service and its impact on demand stability". Proyecto FONDECYT Regular 1090204. 2009-2011.

- "Redes Urbanas" Proyecto Bicentenario de Ciencia y Tecnología PBCT R-19.

- "Instituto Sistemas Complejos de Ingeniería" ISCI (ICM P-05-004-F, CONICYT FBO16).

ALUMNOS GRADUADOS EN EL MARCO DE ESTA INVESTIGACIÓN:

- Cristián Herrera. Memoria: Desarrollo de un modelo de elección de ruta en Metro. Ingeniero Civil, mención Transporte. Universidad de Chile (2014).

- Margarita Amaya. Tesis: Análisis de patrones de viaje utilizando datos masivos de transporte público. Ingeniero Civil y Magíster en Ciencias de la Ingeniería, mención Transporte. Universidad de Chile (2013).

- Flavio Devillaine. Tesis: Estimación de viajes y actividades en base a sistemas tecnológicos de transporte público. Ingeniero Civil y Magíster en Ciencias de la Ingeniería, mención Transporte. Universidad de Chile (2012).

- Diego Silva. Memoria: Validación exógena de la estimación de paradero de bajada y destino de actividades de usuarios de Transantiago. Ingeniero Civil, mención Transporte. Universidad de Chile (2012).

- Claudio Navarrete. Memoria: Análisis de patrones anómalos en la secuencia de transacciones de pago en el sistema de transporte público de Santiago. Ingeniero Civil, mención Transporte. Universidad de Chile (2012).

- Richard Ibarra. Memoria: Diseño e implementación de un software de cálculo y visualización de perfiles de carga para buses de Transantiago. Ingeniero Civil en Computación. Universidad de Chile (2012).

- Daniel Fischer. Memoria: Obtención de una matriz origen-destino de viajes en transporte público a partir de datos pasivos. Ingeniero Civil Industrial. Universidad de Chile (2010).

- Mauricio Zúñiga. Memoria: Metodología y software para estimar velocidades con datos de Transantiago. Ingeniero Civil en Computación. Universidad de Chile (2010).

- Pamela Mora. Memoria: Generación de datos de patrones de viaje a partir de transacciones BIP. Ingeniero Civil, mención Transporte. Universidad de Chile (2010). ■

REFERENCIAS

Cortés, C., Gibson, J., Gschwender, A., Munizaga, M.A. y Zúñiga, M. (2011) Commercial bus speed diagnosis based on GPS-monitored data. Transportation Research C 19(4), 695-707.

Devillaine, F., Munizaga, M.A. y Trepanier, M. (2012) Detection activities of public transport users by analyzing smart card data. Transportation Research Record 2276, 48-55.

Gschwender, A., Ibarra, R., Munizaga, M. y Palma, C. (2012) Monitoring Transantiago through enriched load profiles obtained from GPS and smartcard data. CASPT Santiago, Chile 23-29 julio.

Munizaga, M.A. y Palma, C. (2012) Estimation of a disaggregate multimodal public transport origin-destination matrix from passive Smart card data from Santiago, Chile. Transportation Research 24C(12), 9-18.

BIG DATA: ¿LA MISMA CERVEZA PERO CON OTRO ENVASE?

JUAN VELÁSQUEZ

Profesor Asociado del Departamento de Ingeniería Industrial (DII), de la U. de Chile. Dr. en Ingeniería de la Información de la Universidad de Tokio, Japón.

jvelasqu@dii.uchile.cl

Hace unos 20 años, cuando comenzábamos a trabajar con algoritmos de procesamiento masivo de datos, específicamente utilizando Redes Neuronales Artificiales y Algoritmos Genéticos, nos maravillábamos con la posibilidad de analizar estos grandes volúmenes de información en sólo unas cuantas horas usando súper computadores. La idea central era encontrar patrones en los datos que nos permitieran crear un modelo predictivo del fenómeno en estudio, algo así como una bola de cristal que "adivinaba el futuro".

Datos, información y conocimiento. ¿Cuáles son las distinciones fundamentales? Los datos son sólo un registro de un evento, por ejemplo la temperatura de un objeto. La información nos permite tomar decisiones. Por ejemplo, si un indicador de gestión nos dice que estamos mal en las ventas de un producto, hay que tomar acciones al respecto. El conocimiento es algo un poco más elaborado, y del punto de vista tecnológico tiene que ver con "patrones y reglas de uso". Por ejemplo, si el paciente manifiesta X,Y y Z síntomas, entonces debemos aplicar el procedimiento P.

Pongamos las cosas en un contexto simple. Con un bit se puede almacenar un estado de "verdadero/falso". Un byte (8 bits) almacena un carácter. 1024 bytes corresponden a 1Kilobyte, lo cual permite almacenar una oración. 1 Mega byte, o 1024 Kilobytes, sirven para almacenar una novela corta, y en 1 Gigaybte, o 1024 Megabytes, podemos guardar una película. En un disco de 1TeraByte se pueden almacenar los documentos/libros de una biblioteca grande, como la del Congreso Nacional. Las próximas escalas ya comienzan a ser cifras tan enormes que con la tecnología actual se necesitarían edificios llenos de discos duros de 1Terabyte para poder almacenar todos esos datos.

El concepto Big Data nos propone trabajar con Terabytes de datos, y en algunos casos con PetaBytes o más (es decir, millones de Gigabytes), en una enorme gama de disciplinas con un sueño común: encontrar patrones que permitan descubrir un nuevo conocimiento desde las cordilleras de datos. Es importante notar que en este caso los datos no están estructurados utilizando tan solo las bases de datos relacionales tradicionales, sino que se va mucho más allá. Son las bases de datos no estructuradas, donde todos los tipos de datos de la historia están presentes, las que más concitan la atención de quienes desarrollamos algoritmos y herramientas Big Data.

PROBLEMAS Y DESAFÍOS

Son los de antaño, los de siempre, es decir, cómo capturar, almacenar, buscar, comparar, analizar y visualizar grandes volúmenes de datos. Pero con la sutileza de que ahora estamos muy lejos de contar con una solución tecnológica para la creación de discos que sean capaces de almacenar millones de TeraBytes, y con un acceso lo suficientemente rápido como para ser transmitidos casi sin retardos por una red de alta velocidad. Hasta el momento la solución más práctica ha consistido en colocar sendos arreglos de discos duros para almacenar los datos en forma distribuida, para luego hacer uso de estos a través de computación paralela. Pero todo tiene su costo, y el principal es la energía utilizada para mantener esta infraestructura.

Revisitemos nuevamente un viejo problema: ¿cómo obtener información valiosa a partir de los datos que nos permita tomar la mejor decisión táctica/estratégica para una situación en particular? Más aún, ¿qué nuevo conocimiento puedo extraer a partir de los datos que le permitan a mi institución lograr una ventaja competitiva frente a su competencia? Una alternativa muy recurrida cuando por problemas tecnológicos no se puede procesar toda la base de datos, es tomar una muestra representativa de ésta y alimentar algún algoritmo extractor de patrones que nos permita crear un modelo

→

predictivo. Lo anterior, obviamente corre el riesgo de dejar pasar datos que pueden ser claves a la hora de descubrir un nuevo conocimiento, pero al menos nos da una buena aproximación del fenómeno en estudio. La otra alternativa es derechamente preprocesar toda la base de datos, aplicar algoritmos de extracción de patrones y armarse de paciencia, suponiendo que tenemos un computador de amplias capacidades. Los resultados pueden ser realmente sorprendentes, sobre todo cuando se cruzan datos provenientes de fuentes diversas y que complementan el análisis.

IMPLICANCIAS DE BIG DATA

Privacidad: imaginemos una situación donde la aplicación de Big Data nos va proponiendo compras en distintas etapas de nuestras vidas y luego al detectar que fuimos a una clínica, nos propone un seguro de vida. ¿Qué pasa con la privacidad de nuestros datos personales? [2].

Backtracking de decisiones: ¿cómo se tomarían las decisiones si pudiésemos analizar todas las alternativas posibles, con solo cambiar las entradas del modelo que se originó a partir de los datos? [4].

Personalización de la oferta: un viejo sueño del *marketing one to one* se puede hacer real: conocer los gustos, preferencias, necesidades etc. de los clientes en forma individual, y orientar el mensaje ya no a la masa, sino a cada persona, logrando una efectividad sin precedentes.

Nuevos negocios basados en datos: ya hay empresas que venden el servicio de almacenamiento de datos. Otras que se dedican a su procesamiento. ¿Qué tal algunas cuyo rubro sea la limpieza de datos? U otras que solo se dediquen a la generación de reportes, etc. [3].

ALGUNAS APLICACIONES

Health Informatics: con el avance de la ciencia médica, la esperanza de vida ha aumentado considerablemente. La medicina preventiva está siendo cada vez más recurrida para asegurar la detección temprana de enfermedades. Para lograrlo, se debe mantener un registro histórico de todos los exámenes que se le ha realizado a un paciente. Nuevamente la cantidad de datos es enorme, lo que se traduce en un tremendo desafío.

Brain Informatics and neuromarketing: ¿cuál es la estructura y contenido correcto para que un sitio web atraiga y retenga a sus visitantes? Utilizando técnicas de la neurociencia, como los electroencefalogramas y dispositivos *eye tracking,* podemos conocer la respuesta de los usuarios web a estímulos visuales, tales como imágenes, colores, vídeo etc. presentes en una página web. Con los sistemas de *eye tracking* se realiza un seguimiento del movimiento ocular y análisis de dilatación pupilar, la cual está directamente relacionada con la aceptación/rechazo del estímulo por parte del usuario. Adicionalmente, los datos generados en el electroencefalograma nos permiten clasificar la respuesta emocional del usuario frente al estímulo visual [3]. Una sesión de 30 minutos utilizando estas técnicas puede generar varios terabytes de datos crudos (más detalles ver www.akoriproject.com). Estas técnicas también son utilizadas para conocer las preferencias de los consumidores en el retail, analizando cuáles son los niveles de emoción, atención y memoria, configurando el área del marketing conocida como *neuromarketing*.

Web opinión mining: a partir de la extracción de información desde las redes sociales se pueden

IMAGEN 1.
EQUIPO DE TRABAJO DEL PROFESOR JUAN VELÁSQUEZ.

IMAGEN 2.
DE IZQUIERDA A DERECHA: JUAN VELÁSQUEZ, YERKO COVACEVICH Y FRANCISCO MOLINA.

lograr sorprendentes análisis respecto de las percepciones, sentimientos, opiniones que tienen los internautas sobre un producto o servicio en tiempo real. El problema es que solo en Twitter se están generando más de 40.000 tweets por segundo, cifra que irá en aumento en los próximos años [1].

REFLEXIÓN FINAL

Hay algo muy interesante en casi todos los problemas relacionados a Big Data: los algoritmos y técnicas que se desarrollan no son para un tipo de dato en específico. Dicho de otra manera, si desarrollamos un algoritmo de análisis de serie de tiempo para datos generados en un radiotelescopio, y luego lo aplicamos a los generados por un electroencefalograma, con algunos ajustes claro está, podremos extraer patrones a partir de las ondas cerebrales de un paciente y quién sabe, detectar en forma temprana una anomalía. Como en todo nuevo concepto, hay mucho de mito y poco de realidad. Aparecen miles de expertos, gurúes del área, pero que están igual que todos nosotros: somos testigos del nacimiento de algo grande que comienza recién a dar sus primeros pasos en ciencia, en tecnología y en los negocios.

Big data: ¿la misma cerveza pero con otro envase? Aún no lo tenemos claro, pero conviene que esta vez se la tome con calma y muy helada, sino puede causarle una big indigestion. ■

REFERENCIAS

[1] "Detecting Trends on the Web: A Multidisciplinary Approach", Dueñas Fernández R., Velásquez, Juan D. and L'Huillier, Gastón. Information Fusion, 20:129-135, 2014.

[2] "Web Mining and privacy concerns: Some important legal issues to be consider before applying any data and information extraction technique in Web-based environments", Velásquez, Juan D., Expert Systems with Applications, 40 (1): 5228-5239, 2013.

[3] "A neurology-inspired model of web usage", Pablo E. Román and Juan D. Velásquez, Neurocomputing, 131: 300-311 ,2014.

[4] "Are you ready for the era of 'big data'", B. Brown, M. Chui J. Manyika, McKinsey Quarterly, 4:24-35, 2011.

[5] "Big data: the management revolution", A. McAfee and E. Brynjolfsson, Harvard Business Review, 90(10):60-68, 2012.

LA NUEVA ERA DE DATOS EN ASTRONOMÍA

FAVIOLA MOLINA

Investigadora postdoctoral en el Departamento de Ciencias de la Computación, Universidad de Chile, donde trabaja con el profesor Alexandre Bergel. Doctora en Ciencia de la Universität Heidelberg y fellow del Instituto Max Planck para Astronomía en Heidelberg, Alemania (2013). Astrónoma de soporte en el Observatorio Europeo Austral (2008-2009). Magíster en Astronomía y Astrofísica de la Pontificia Universidad Católica de Chile (2008). Licenciada en Física de la Universidad de Los Andes, Venezuela (2004). Áreas de interés: Astro-computación, análisis estadístico del medio interestelar y formación de estrellas, poblaciones estelares y recientemente formación de discos planetarios y de transición.

fmolina@dcc.uchile.cl

El desarrollo de cualquier disciplina científica involucra el manejo de datos. En el caso particular de la Astronomía, el incremento de la cantidad y tamaño de los archivos de datos ha ido creciendo con el paso de los años, considerando así a ésta como una ciencia de datos intensivos [Hassan and Fluke, 2011].

Hasta mediados del siglo XX, en específico en la Astronomía óptica, los detectores eran placas fotográficas. La exposición de las mismas podía tomar horas de acuerdo a la intensidad del objeto que se quería observar. Más tarde, se dio paso a los fotómetros fotoeléctricos que ofrecían mayor sensibilidad, precisión, linealidad y un mayor rango dinámico para el análisis que las placas fotográficas[1].

1 http://star-www.rl.ac.uk/docs/sc5.htx/node7.html

El curso de los datos astronómicos cambió dramáticamente cuando en 1975, dadas las mejoras tecnológicas en los métodos de recolección de imágenes, se comenzó a proponer la idea de implementar los dispositivos de carga acoplada (CCD por su siglas en inglés[2], Charge Coupled Device) en la obtención de datos [Samuelsson, 1975; McCord and Bosel, 1975]. Como resultado de estas propuestas, en 1976, los CCDs revolucionaron la astronomía cuando J. Janesick y B. Smith adosaron un CCD a un telescopio de 155 cm. de diámetro (localizado en el Monte Bigelow, Arizona) para obtener imágenes de Júpiter, Saturno y Urano (Parimucha and Vanko, 2005). La ganancia en sensibilidad fotométrica y cobertura espectral a partir de esa época ha ido aumentando drásticamente.

Hoy en día, la cantidad de datos astronómicos (tanto observacionales como teóricos) en archivos sobrepasan los Petabytes[3] [Hassan and Fluke, 2011]. El término "big-data" se ha implementado para describir sets de datos que son muy grandes para ser manejados con las herramientas de procesamiento, análisis, y visualización, existentes a la fecha. Con el paso del tiempo, la cantidad de observatorios obteniendo datos experimentales ha crecido, así como los centros de investigación teórica que producen grandes cantidades de datos simulados. Con la nueva generación de telescopios e instrumentos, la resolución espacial y espectral de las imágenes ha aumentado de una manera sin precedentes. Junto con esto, el incremento en la capacidad de cómputo y almacenamiento ha provocado que el tamaño de cada archivo de datos crezca significativamente. Por ejemplo, cuando el Atacama Large Millimeter/sub-millimeter Array (ALMA) se encuentre en completa operación, generará más de 750 Gb de datos por día[4], que se traduce en unos 250 Tb por año[5]. Otro ejemplo son las simulaciones cosmológicas con las cuales se resuelven numéricamente las ecuaciones que rigen la dinámica del Universo. Estos códigos usan del orden de 10^{10} partículas y/o grandes mallas adaptativas tridimensionales en el computo que producen archivos de datos de varios Terabytes, incluso pudiendo llegar al orden de los Petabytes [6] [Springel et al., 2005].

Por otro lado, los datos astronómicos cada vez están más interconectados. Dada la cantidad de observaciones realizadas durante las últimas décadas, y la recopilación y digitalización de datos históricos (en distintas bandas del espectro electromagnético), ha nacido el Observatorio Virtual (The Virtual Observatory, VO: http://www.ivoa.net/). De este gran repositorio es posible obtener datos que posibilitan estudios sistemáticos, pancromáticos y estadísticamente significativos de la evolución de sistemas astronómicos [Brunner et al., 2002]. El VO alberga datos provenientes de archivos de una gran cantidad de observatorios terrestres y espaciales, entre ellos están el Telescopio Espacial Hubble (HST), el Observatorio de Rayos X Chandra, El Sondeo Digital Sloan del Cielo (SDSS), el Sondeo de Todo el Cielo en Dos Micrones (2MASS), el Sondeo del Cielo Digitalizado del Observatorio Palomar (DPOSS), el Observatorio Europeo Austral (ESO), el Telescopio para el Sondeo de Astronomía Visible e Infrarroja (VISTA), y más recientemente ALMA, además de muchos otros. Esta nueva manera de hacer astronomía permite el uso de cualquier tipo de datos a científicos y estudiantes desde cualquier parte del mundo, y que antiguamente no tenían acceso a grandes observatorios.

La cantidad de datos promete continuar incrementándose estrepitosamente con el paso del tiempo. En el futuro cercano, nuevos telescopios tales como el Telescopio Sinóptico Grande para Sondeos (LSST)[7], el Telescopio Magallanes Gigante (GMT)[8], el Telescopio de Treinta Metros (TMT)[9], el Square Kilometer Array (SKA)[10], el Telescopio Europeo Extremadamente Grande (E-ELT), entre otros; abrirán nuevas posibilidades en la Minería de Datos.

En esta nueva era astronómica es importante notar que, no solo es necesario contar con grandes infraestructuras de almacenamiento y altas velocidades de transferencia, sino también que es prioritario desarrollar herramientas de visualización que posibiliten realizar análisis de manera ágil y rápida. Actualmente, en conjunto con el Profesor del Departamento de Ciencias de la Computación (DCC) de la Universidad de Chile, Alexandre Bergel, nos encontramos desarrollando una plataforma dinámica y diligente para la visualización y análisis de datos astronómicos: AstroCloud (http://astrocloudy.wordpress.com). Esta herramienta está enfocada en el análisis específico de un problema físico determinado, pero con la flexibilidad de agregar fácilmente nuevos módulos que incluyan distintos tipos de análisis. Por otro lado, la nueva estructura de datos astronómicos requiere de una adecuada visualización y análisis tridimensional[11]. AstroCloud será una herramienta que facilitará y reducirá el tiempo empleado en el análisis de datos masivos en dos y tres dimensiones. Ésta es justamente una de las necesidades principales de observatorios como ALMA. Actualmente contamos con la colaboración de la Profesora Nancy Hitschfeld (DCC U. de Chile), el Profesor Lucas Cieza (Núcleo de Astronomía, Facultad de Ingeniería, Universidad Diego Portales), la Dra. Paola Pinilla (Observatorio de Leiden, Holanda), así como con la ayuda de Juan Cortés (ALMA) quien ayudará a probar la versión beta de la herramienta. Este proyecto multidisciplinario está abierto a estudiantes entusiastas que deseen participar en el desarrollo de esta dinámica herramienta. ■

2 En lo sucesivo, todas las siglas y acrónimos refieren a las siglas en inglés correspondientes a la expresión.
3 1015 bytes.
4 El tamaño de cada imagen depende del modo de observación.
5http://www.almaobservatory.org/en/press-room/announcements-events/542-virtual-observatories-chilean-developmentof-astronomical-computing-for-alma
6 e.g., http://www.cfa.harvard.edu/itc/research/movingmeshcosmology/
7 http://www.lsst.org/lsst/
8 http://www.gmto.org
9 http://www.tmt.org
10 http://www.ska.ac.za/about/project.php
11 En Astronomía, las tres dimensiones son dos en posición-posición que corresponde al plano del cielo, y la tercera (profundidad) corresponde al espacio de frecuencias y/o velocidades.

REFERENCIAS

Brunner, R. J., Djorgovski, S. G., Prince, T. A., and Szalay, A. S.: 2002, in J. S. Mulchaey and J. T. Stocke (eds.), Extragalactic Gas at Low Redshift, Vol. 254 of Astronomical Society of the Pacific Conference Series, p. 383.

Hassan, A. and Fluke, C. J.: 2011, Publications of the Astronomical Society of Australia. 28, 150.

McCord, T. B. and Bosel, J. P.: 1975, in Charge-Coupled Device Technology for Scientific Imaging Applications, pp 65-69.

Parimucha, S. and Vanko, M.: 2005, Contributions of the Astronomical Observatory Skalnate Pleso 35, 35.

Samuelsson, H.: 1975, ESA Scientific Technical Review 1, 219.

Springel, V., White, S. D. M., Jenkins, A., Frenk, C. S., Yoshida, N., Gao, L., Navarro, J., Thacker, R., Croton, D., Helly, J., Peacock, J. A., Cole, S., Thomas, P., Couchman, H., Evrard, A., Colberg, J., and Pearce, F.: 2005, Nature 435, 629.

MANEJO DE DATOS MASIVOS EN BIOMEDICINA COMPUTACIONAL

VÍCTOR CASTAÑEDA

Ingeniero eléctrico, Universidad de Chile (2005) y Doctor en Ciencias de la Computación, TUM-Alemania (2012), especializado en el procesamiento de imágenes médicas. Su Tesis Doctoral estuvo enfocada en el procesamiento de imágenes endoscópicas y sensores 3D tales como cámaras Time-Of-Flight y Kinect. Actualmente hace su postdoctorado (Proyecto FONDECYT 3140444) en el seguimiento de núcleos y segmentación de membranas celulares provenientes de microscopía Light Sheet, en SCIAN Lab del Prof. Härtel, ICBM, Facultad de Medicina, U. de Chile.

vcastaneda@med.uchile.cl

A nivel mundial, la investigación de excelencia en el ámbito biológico, clínico, médico y biomédico depende en forma crucial de la capacidad de análisis de los datos recolectados por experimentos que generan crecientes volúmenes de datos. Como ejemplo, la microscopía confocal *in vivo* es capaz de generar cientos de gigabytes (GB) de imágenes tridimensionales en una sola captura. En estos casos los investigadores de BioMedicina Computacional recurren a información cuantitativa, mediante modelamiento matemático y computacional para entender y predecir procesos biológicos con relevancia en medicina y ciencia básica.

Las aplicaciones involucradas entrelazan a los mundos de la Salud Pública (bases de datos), Clínico/Hospitalario (sistemas de información), (Neuro) Ciencias Biomédicas (imágenes, bioinformática y biología computacional), Ciencias de la Computación/Ingeniería (algoritmos), Física y Matemática (herramientas y modelos). Se reconoce que la creación del campo de la BioMedicina Computacional requiere un esfuerzo mayor a través de los años para generar equipos multidisciplinarios y una nueva cultura de trabajo desde las ciencias básicas hasta la investigación clínica, salud pública, y la introducción de nuevos servicios en sistemas de salud. Hasta la fecha, Chile y la mayoría de los países latinoamericanos no cuentan con respuestas adecuadas en este tema, principalmente por la falta inversión de fondos estratégicos con visión de mediano y largo plazo. La obtención de esta capacidad conlleva a la creación de alianzas estratégicas entre las disciplinas involucradas para desarrollar y acceder a nuevas tecnologías necesarias para el análisis de datos masivos. Instituciones emblemáticas en todo el mundo han respondido a este desafío a través de la creación de centros o institutos que persiguen misiones afines: (i) Johns Hopkins University [1], (ii) U-Michigan [2], (iii) U-Cincinnati [3], (iv) Janelia Farm [HHMI] [4], (v) BioQuant, DKFZ, Uni-Heidelberg [5], y (vi) Mt. Sinai [6], por nombrar solo algunos.

El análisis de datos en BioMedicina Computacional requiere de una

→

alta capacidad de procesamiento, siendo necesaria la interacción con arquitecturas de computación de alto rendimiento (high performance computing, HPC). Como ejemplo, un equipo para la digitalización de muestras de tejido (como biopsias) puede escanear una serie de portaobjetos completos, generando varios GB de imágenes en una captura (aproximadamente 10 terapíxeles). Esta imagen de gran tamaño debe ser procesada para extraer características que permitan reconocer distintos tipos de células características de diversas patologías como el cáncer. Su procesamiento requiere de muchos procesadores (como cluster de CPU o GPU, comunes en HPC) para disminuir los tiempos de respuesta a niveles que permitan el diagnóstico y tratamiento oportuno. Otro ejemplo está en el ámbito de la genómica, donde un gran número de algoritmos matemáticos son aplicados para extraer características de los genomas recolectados provenientes de distintos individuos, generando varios GB de datos que deben ser almacenados en una memoria principal para su procesamiento, lo que también genera la necesidad de acceder a arquitecturas HPC.

El manejo de datos masivos en el ámbito de la BioMedicina Computacional genera grandes desafíos, desde luego en el análisis pero también problemas de almacenamiento, manipulación, confidencialidad, seguridad y transmisión de datos. En este ámbito reviste especial importancia la seguridad en el almacenamiento y transmisión de datos relacionados a pacientes, que deben estar salvaguardados ante cualquier evento indeseado. Es por esto que se requieren diseños para sistemas con altos niveles de seguridad y confidencialidad, que garanticen su correcto uso y manipulación en caso de ser utilizados en escenarios de investigación, planificación o reportes. Se deriva también un desafío importante para generar herramientas de cómputo con HPC accesibles a usuarios sin conocimientos relacionados con Ciencia de la Computación, como por ejemplo investigadores del área de la Biología.

BIOMED-HPC: UNA INICIATIVA PIONERA EN SUDAMÉRICA

Dentro de los ejemplos regionales está la reciente creación del Insti-

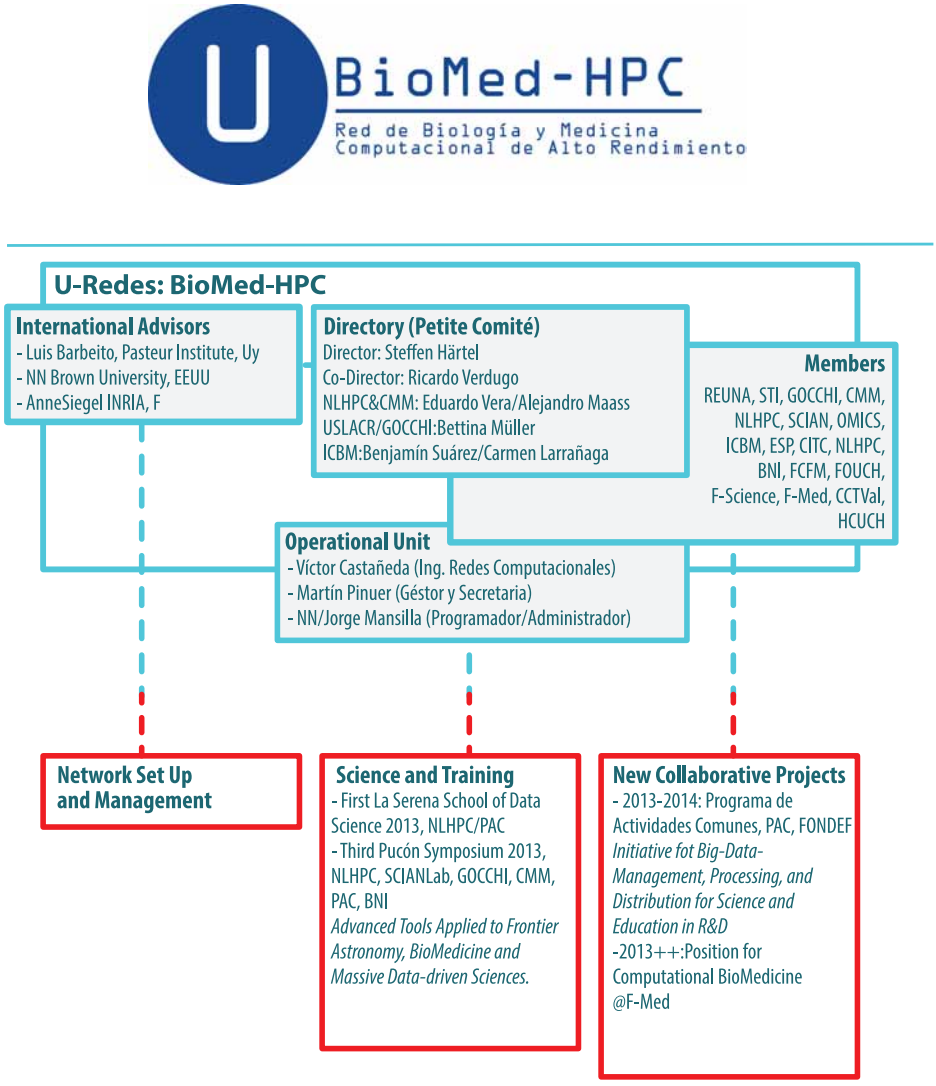


FIGURA 1.
ORGANIGRAMA DE U-REDES.

tuto de Investigación en Biomedicina de Buenos Aires CONICET, relacionado con la Sociedad Max Planck en Buenos Aires, dedicado a temas actuales de las biociencias, especialmente investigación en biomedicina [7], y la reciente creación de la red interna BioMed-HPC en la Universidad de Chile, que interconecta las facultades de Medicina, Odontología, Química y Farmacia, Ciencias y Ciencias Físicas y Matemáticas, en una amplia e interdisciplinaria red de colaboración (**Figura 1**).

El proyecto BioMed-HPC está dentro del contexto de los proyectos internos de la Universidad de Chile, llamados U-Redes Domeyko II. BioMed-HPC busca fortalecer el desarrollo de investigación básica y aplicada en todas las áreas abarcadas por los miembros de la red, desde la física a estudios en el cáncer y los sistemas de información hospitalarios. Para alcanzar el objetivo, la estrategia de la Red es: (i) facilitar la gestión de proyectos colaborativos, sobre todo en aspectos financieros y organizacionales, y (ii) asegurar el financiamiento necesario para mejorar la infraestructura de Intranet en la U. de Chile que permita el flujo de grandes cantidades de datos entre unidades académicas ubicadas en distintas facultades. Los tres pilares que sustentan esta Red son una unidad de operaciones, la organización de eventos formativos y la implementación de infraestructura. El primer pilar es una unidad de operaciones, la cual es financiada por U-Redes y contribuciones del Instituto Ciencias Biomedicas - ICBM, National Laboratory for High Performance Computing - NLHPC, Red Universitaria Nacional - REUNA, Dirección de Servicios de Tecnologías de Información y Comunicaciones - STI U-Chile, United States-Latin America Cancer Research Network - USLACRN. El segundo pilar es la formación avanzada de científicos mediante la organización de simposios y escuelas de verano, en que especialistas internacionales del amplio campo de la informática médica y HPC compartirán eventos con los integrantes de BioMed-HPC. Por último, el tercer pilar es la creación de una nueva red física de investigación de alta velocidad (10 Gbps) que permitirá: (i) transmisión de los datos entre el Campus Norte (en un comienzo la Facultad de Medicina) y el clúster HPC del proyecto NLHPC en el CMM, en la Facultad de Ciencias Físicas y Matemáticas, (ii) transmisión segura de datos masivos, (iii) análisis de los datos en el clúster antes mencionado, (iv) acceso a almacenamiento seguro y confidencial, y (v) creación de herramientas remotas para la utilización de HPC para usuarios no especializados en supercómputo. Esta Red se habilitará a mediados de 2014, comenzando con la marcha blanca de la red física exclusiva de investigación. ■

REFERENCIAS

[1] Development of quantitative approaches for understanding the mechanisms and treatment of human disease through applications of mathematics, engineering and computational science (R. Winslow, Dir. of the Institute for Computational Medicine, J Hopkins University, www.icm.jhu.edu).

[2] Creation of novel and impactful informatics and computationally-based methods, tools, algorithms, and resources to extend basic and clinical research capabilities and results (Department of Computational Medicine and Bioinformatics, U-Michigan).

[3] Usage of data and computational systems to make disease more preventable, illness more predictive and treatment more personalized (Computational Medicine Center, U-Cincinnati).

[4] Understanding the core principles that give rise to the developmental building plans of animals from single-cell to whole-organism level, Janelia Farm, Howard Hughes Medical Institute (HHMI), (http://janelia.org/lab/keller-lab).

[5] The Center for "Quantitative Analysis of Molecular and Cellular Biosystems", which functions as a platform for the development and constant refinement of mathematical models of complex biological systems as well as the swift validation of scientific hypotheses via experimental data, BioQuant, Heidelberg University (http://www.bioquant.uni-heidelberg.de/).

[6] Icahn School of Medicine at Mount Sinai - New York City. http://www.mssm.edu/

[7] Polo Científico Tecnológico, Agencia Nacional de Promoción Científica Tecnológica y CONICET, Argentina. http://www.mincyt.gob.ar/polo

NAVEGANDO A TRAVÉS DEL DILUVIO DE DATOS ASTRONÓMICOS

La Astronomía se ha transformado rápidamente en una ciencia de grandes volúmenes de datos. Giga, Tera y próximamente Petabytes deben ser procesados en tiempo real para aprovechar el potencial de los telescopios de nueva generación. Esto pone nuevos desafíos informáticos sobre la mesa: distribución y descripción de datos, taxonomías, semántica, minería de datos, visualización y procesamiento estadístico, dentro de otros. Para tener éxito al abordar estos problemas es indispensable la colaboración entre equipos interdisciplinarios de astrónomos, estadísticos, ingenieros y científicos del área de Computación.

GUILLERMOCABRERA

Candidato a Doctor en Ciencias de la Computación, Magíster en Ciencias de la Computación, Ingeniero Civil en Computación y Licenciado en Astronomía, Universidad de Chile. Fundador del Laboratorio de Astroinformática del Centro de Modelamiento Matemático de la U. de Chile.

gcabrera@dim.uchile.cl

Una gran revolución científica se vivió cuando los primeros telescopios se inventaron a principios del siglo XVII. La humanidad comenzó a ver cosas que antes no podía, tales como cráteres en la luna, cometas, los anillos de Saturno e incluso sus lunas. Una nueva revolución se vivió con la creación de placas fotográficas a finales del siglo XIX con las cuales los científicos podían almacenar las observaciones para su futuro análisis. Luego, siguió la revolución digital, con la creación de los llamados Charged-Coupled Devices (CCDs) con los cuales se crearon las primeras cámaras digitales. El astrónomo entonces podía analizar computacionalmente sus imágenes de manera automática (o semi). Hoy en día, vivimos una nueva revolución: la revolución de los datos. Los telescopios están creando cada año más y más datos siguiendo un crecimiento exponencial. Este inmenso volumen de datos está empujando a diversas disciplinas hacia la frontera de sus conocimientos, incluyendo la astronomía, la ingeniería, las estadísticas y la Ciencias de la Computación, entre otras.

Jim Gray vio venir este diluvio años atrás, y lo llamo "el cuarto paradigma" [Tony Hey, 2009]. Según Gray, históricamente existen tres paradigmas científicos clásicos: ciencia empírica (descripción de fenómenos naturales), ciencia teórica (modelos y generalizaciones), y ciencia computacional (complejas simulaciones computacionales). Hoy nos vemos enfrentados a un nuevo paradigma: la ciencia de la exploración de datos (eScience), donde se unifica la teoría con los experimentos y simulaciones en torno al análisis masivo de datos. Esto ha ocurrido en diversas áreas de investigación, donde uno de los ejemplos más claros puede ser encontrado en la bioinformática (por ejemplo, en genómica) en la cual se producen y analizan terabytes de datos. La Astronomía ha llegado a estos niveles de producción de datos en los últimos diez años, llegándose a hablar de la necesidad de una nueva disciplina: la llamada Astroinformática [Borne, et al., 2009].

GRANDES TELESCOPIOS Y GRANDES VOLÚMENES DE DATOS

Existen distintos telescopios para distintos objetivos científicos. Una primera distinción se puede hacer en función del tipo de radiación electromagnética que se desea observar. Podemos definir esta radiación por el tamaño de su longitud de onda (o, equivalentemente, su frecuencia). De esta forma tenemos telescopios como el Very Large Array (VLA), el Atacama Large Millimeter/submillimeter Array (ALMA)(**Imagen 1**) o el Wilkinson Microwave Anisotropy Probe (WMAP) en el radio (longitud de onda mayor que 10^{-3} m.), el Spitzer Space Telescope (SST) y el James Webb Space Telescope (JWST) en el infrarrojo (10^{-3}–10^{-6}m.), el Hubble Space Telescope (HST), el Sloan Digital Sky Survey (SDSS), el Large Synoptic Survey Telescope (LSST)(**Imagen 2**) y GAIA en el visible (lo que podemos ver con nuestros ojos, →

IMAGEN 1.
EL ATACAMA LARGE MILIMITER/SUBMILIMITER ARRAY (ALMA). CADA PAR DE ANTENAS FORMAN UNA LINEA DE BASE, I.E. UN PUNTO EN EL PLANO DE FOURIER. WWW.ALMAOBSERVATORY.ORG CRÉDITO: EFE / ARIEL MARINKOVIC.

IMAGEN 2.
FOTOGRAFÍA DEL SITIO E IMAGEN GENERADA POR COMPUTADOR DEL LSST. EL LSST TRABAJARÁ SOBRE VARIABLES ESPACIALES Y TEMPORALES Y PRODUCIRÁ APROXIMADAMENTE 30 TB DE DATOS POR NOCHE A PARTIR DEL 2022. FUENTE: WWW.LSST.ORG

aproximadamente entre 400 y 700 nm.), el Galaxy Evolution Explorer (GALEX) y Hisaki en el ultravioleta (~10^{-6} – 10^{-8} m.), el Nuclear Spectroscopic Telescope Array (NuSTAR) para rayos-x (~10^{-8} – 10^{-11} m.), y el Fermi Gamma-ray Space Telescope para rayos gamma (longitud de onda menor que < 10^{-11} m.) dentro de muchos otros. Aún cuando para observar el universo en distintas longitudes de onda se requiere de distintos telescopios, todos los nuevos grandes instrumentos tienen un problema en común: la gran cantidad de datos a analizar. Por ejemplo, ALMA produce aproximadamente 250 TB por año, el LSST producirá 30 TB por noche (a partir del 2022) y el Square Kilometer Array (**Imagen 3**) producirá más de 100 GB por segundo (2030). En este último caso, la cantidad de datos crudos será tan grande, que no será posible almacenarla, por lo que se deberá procesar mediante hardware y guardar solamente los datos procesados.

Con este gran volumen de datos vienen nuevos problemas, los cuales están delineados en [Borne, et al., 2009]: organización (cómo y dónde), descripción (metadatos), taxonomías o reglas, definición de conceptos y relaciones (semántica), minería de datos y aprendizaje de máquinas, visualización y astroestadística. Todos estos aspectos son los que aborda el área de la Astroinformática.

PROCESANDO GRANDES VOLÚMENES DE DATOS ASTRONÓMICOS

PRODUCIENDO IMÁGENES

Uno puede pensar en un telescopio digital clásico de forma similar que en la cámara digital de nuestro celular: un montón de píxeles. Una diferencia importante está en las condiciones de lo que queremos fotografiar. Al tomar una fotografía con nuestro celular a un paisaje o a nuestro rostro favorito, con una iluminación adecuada no tendremos problemas de ruido o de resolución más allá del tamaño de los píxeles. En las imágenes astronómicas la cosa se complica un poco. Los objetos están realmente lejos y antes de ser detectada en el CCD la luz sufre aberraciones al pasar por el telescopio o por la atmósfera (en el caso de telescopios terrestres). Debido a que los CCDs "cuentan" fotones las imágenes contienen ruido Poissoniano además de ruido térmico de distribución Gaussiana. Al mismo tiempo, si queremos llegar realmente profundo debemos tomar imágenes con un tiempo de exposición prolongado, pero lo más corto posible de tal forma de poder observar mayor cantidad de zonas del cielo. Además, al fotografiar (u observar) grandes campos del cielo, hay que tener cuidado con objetos muy brillantes, porque los CCDs se saturan comportándose no lineales (i.e. las "cuentas" no van linealmente con el número de fotones). Por otro lado, se debe también considerar la luz del cielo debido a ciudades u objetos brillantes cercanos (background), los electrones introducidos por segundo por píxel durante la exposición de los CCDs (corriente obscura) y la variación en la eficiencia de cada píxel (flats).

UTILIZANDO VARIOS TELESCOPIOS COMO UN SOLO GRAN INSTRUMENTO

La resolución de un telescopio se define como la distancia angular mínima a la cual es posible distinguir dos fuentes puntuales como independientes. Esta resolución es directamente proporcional al tamaño del telescopio, e inver-

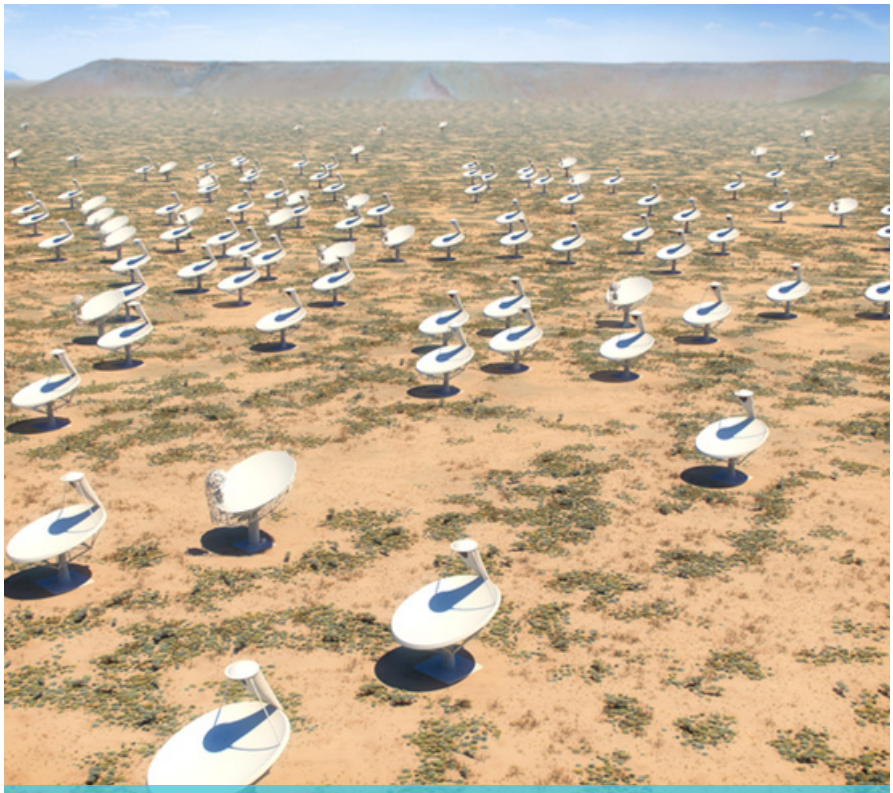


IMAGEN 3.
IMAGEN GENERADA POR COMPUTADOR DEL SKA. EL SKA ABARCARÁ UN DIÁMETRO DE APROXIMADAMENTE 3.000 KM Y PRODUCIRÁ APROXIMADAMENTE 100 GB POR SEGUNDO A PARTIR DEL AÑO 2030. FUENTE: WWW.SKATELESCOPE.ORG.

samente proporcional a la longitud de onda de la radiación que se quiere observar. En ese sentido, mientras más grande el lente de un telescopio, mejor resolución tendrá. Al mismo tiempo, mientras más grande sea la longitud de onda de la luz a observar, peor será la resolución. Por supuesto, existen límites físicos para la creación de lentes o espejos (sería un gran desafío crear un lente de 1km. de diámetro, por ejemplo), por lo que la observación de grandes longitudes de onda (como el radio) a alta resolución no es un tema trivial. Para resolver este problema existe la *interferometría,* técnica a través de la cual se puede obtener una imagen de alta resolución mediante la utilización de muchos pequeños telescopios. En ese sentido, al ubicar dos telescopios a una distancia *D* obtenemos una resolución similar a la de un telescopio con un lente o espejo de diámetro *D*. Ahora sí podemos tener nuestro telescopio de 1 km. de diámetro, pero esto supone nuevos problemas.

En el caso de la interferometría, la señal de cada par de telescopio (o antenas) se correlaciona obteniendo un punto en el plano de Fourier de la imagen, llamado *visibilidad*. En particular, ALMA utiliza un supercomputador diseñado especialmente para realizar las correlaciones entre antenas [Escoffier, et al., 2007]. A medida que se van agregando más antenas, el plano de Fourier se va poblando (para N antenas tenemos N*(N-1) líneas de base considerando las dos orientaciones de cada par). La configuración de antenas puede ir cambiando gracias a la rotación de la tierra o al movimiento físico de ellas, pero los puntos en el plano de Fourier están distribuidos de manera no uniforme. Debido a esto, la reconstrucción de imágenes a partir de las llamadas visibilidades (puntos en el plano de Fourier) no es tan fácil como simplemente realizar una transformada inversa. Para esto existen diversos algoritmos tales como CLEAN [Högbom, 1974], MEM [Cornwell & Evans, 1985]y métodos Bayesianos [Pina & Puetter, 1993][Sutton & Wandelt, 2006][Cabrera, Casassus & Hitschfeld, 2008]. CLEAN consiste en modelar iterativamente la señal como fuentes puntuales, las cuales son ubicadas en los píxeles de mayor brillo. Para cada una de estas fuentes puntuales se determinan sus visibilidades modeladas y se restan a las observadas, una a una. El Maximum Entropy Method (MEM) es un método variacional donde se optimiza la verosimilitud más un término denominado *entropía*, el cual se determina mediante prueba y error. Los métodos Bayesianos extienden este enfoque reemplazando la entropía por un prior calculado probabilísticamente. Curiosamente, para el caso particular de ALMA y el VLT el paquete oficial de reducción de datos, llamado CASA (Common Astronomy Software Applications) [McMullin, Waters, Schiebel, Young, & Golap, 2007] solamente tiene implementado el algoritmo CLEAN, el cual data de 1974. Esto se debe a razones históricas: en general si un algoritmo es bien comprendido por la comunidad astronómica y funciona relativamente bien, pocos esfuerzos se hacen por mejorarlo. Esto por lo menos hasta hace algunos años, cuando comenzó la real revolución de datos.

ENFOQUE CLÁSICO VERSUS ASTRONOMÍA DE SURVEYS

El paradigma clásico de observación astronómica es el siguiente: el astrónomo pide tiempo de observación en un telescopio presentando un proyecto. Este proyecto es evaluado por el Time Allocation Comitee (TAC), quienes, en caso de aceptar, agendan sus observaciones para alguna fecha particular. Entonces el astrónomo debe ir al telescopio (o envía a alguien) y observa durante una o varias noches. Luego se lleva los datos a la oficina y puede pasar meses (o años) analizando los datos para luego obtener alguna conclusión científica.

Con la creación de los primeros surveys este paradigma ha estado cambiando durante los últimos años. Un survey es un conjunto de datos creados con un telescopio dedicado a observar varias regiones del cielo (o varios objetos) durante un largo período de tiempo. Luego, estos datos se hacen públicos (a veces solo dentro de la colaboración) en forma de imágenes, espectros y sobre todo catálogos a través de grandes bases de datos. En la **Imagen 4**, por ejemplo, podemos ver una imagen obtenida del SDSS. El astrónomo entonces ya no debe ir a observar: puede buscar sus objetos de interés dentro de estas grandes bases de datos, e incluso utilizar todos estos datos para obtener conclusiones científicas a través de herramientas estadísticas.

IMAGEN 4.
TIPO DE IMÁGENES A SER ANALIZADAS POR LOS SURVEYS. SOBRE ESTAS IMAGENES ES NECESARIO REALIZAR DETECCION, ASTROMETRÍA, FOTOMETRÍA Y CLASIFICACIÓN AUTOMÁTICAMENTE.
FUENTE: SLOAN DIGITAL SKY SURVEY. WWW.SDSS.ORG

Aún cuando los surveys existen desde la época de las placas fotográficas (por ejemplo, POSS [Minkowski & Abell, 1963]), la real revolución se dio con los primeros surveys digitales. El Sloan Digital Sky Survey (SDSS) es un ejemplo de esto. El último Data Release del SDSS contiene aproximadamente 70 TB de datos, donde la mitad son datos crudos y la otra mitad son catálogos [SDSS Collaboration, 2013]. Al igual que los telescopios, existen distintos surveys para distintos objetivos científicos. La **Tabla 1** muestra distintos surveys y su volumen de datos. Durante los últimos años se han creado proyectos que permitirán incluir también la variable de tiempo a las observaciones. Dos ejemplos de esto son el Dark Energy Survey (DES), el cual comenzó a operar en 2013, y el Large Synoptic Survey Telescope (LSST) [Ivezic, et al., 2008], el cual se espera que comience a operar en 2022. Este último, tomará una imagen completa del cielo Sur cada tres noches. En otras palabras, tendremos una película de todo el cielo con un *timeframe* de tres noches. El LSST producirá diariamente 30 TB de datos (15 TB crudos más 15 TB procesados), los cuales deben ser procesados completamente en menos de un día, ya que a la siguiente noche tendremos nuevamente 30 TB más.

Además de la reducción inicial de las imágenes descrita anteriormente (también llamada primera capa), los surveys deben realizar un conjunto de tareas propias de sus tipos de datos incluyendo: detección automática de objetos (estrellas, galaxias, asteroides, etc.) [Stetson, 1987] [Bijaoui & Rué, 1995] [Bertin & Arnouts, 1996] [Miller, et al., 2001], astrometría (calzar distintas imágenes en un mismo sistema de coordenadas)[Valdés, Campusano, Velásquez, & Stetson, 1995] [Lang, Hogg, Mierle, Blanton, & Roweis, 2010], calce de resolución entre imágenes (PSF matching) [Bertin, 2011], fotometría (calcular exactamente cuanta luz emite cada objeto) [Bertin & Arnouts, 1996] y catalogar las distintas fuentes (ajustes de modelos, clasificación, series de tiempo, etc., ver por ejemplo [Ivezic, et al., 2008]). A todo esto se le suele llamar segunda capa y los encargados de estas tareas son generalmente los mismos observatorios.

La tercera y última capa incluye el análisis astroestadístico de los catálogos y la obtención de conclusiones científicas. Esto no necesariamente lo realizan los mismos observatorios, sino que generalmente es aquí donde el astrónomo de la universidad se conecta a la base de datos para acceder ya sea a catálogos o imágenes y procesarlos. Es aquí además donde se requiere de experiencia en el área de Minería de Datos y Estadística, por lo que los equipos interdisciplinarios juegan un rol fundamental.

EQUIPOS INTERDISCIPLINARIOS PARA PROBLEMAS INTERDISCIPLINARIOS

Ya lo decía Jim Gray antes de desaparecer misteriosamente: los problemas científicos de grandes volúmenes de datos requieren de equipos interdisciplinarios. En nuestro caso, es impensable que un equipo compuesto 100% por astrónomos sea capaz de enfrentar todos los problemas anteriormente descritos. Gray decía que para que un proyecto en eScience sea exitoso se requiere de cuatro piezas fundamentales: los científicos, quienes proporcionan las preguntas a ser respondidas, los plomeros quienes son los encargados de diseñar y mantener las bases de datos, los mineros quienes desarrollan los algoritmos de minería de datos, y los desarrolladores de las herramientas de visualización de preguntas y respuestas. La comunicación entre integrantes de estos equipos debe ser fluida, por lo que,

Survey	Año de inicio	Año de fin	Longitud de Onda	Ubicación	Tamaño total de datos
POSS	1949	1958	visible	California, USA	3TB
2MASS	1997	2001	infrarojo	Arizona, USA+Chile	10TB
GALEX	2003	2013	ultravioleta	Espacio	30TB
SDSS	2000	2020	visible	New Mexico, USA	70TB
GAIA	2013	2020	visible	Espacio	1PB
DES	2013	2018	visible	Chile	1-5PB
PanSTARRS	2008	-	visible	Hawaii	40PB
LSST	2022	2032	visible	Chile	75PB

TABLA 1.
SURVEYS ASTRONÓMICOS.

aún cuando no es necesario que todos sean expertos en todas las áreas, si es necesario que conozcan al menos superficialmente un poco de cada tema. De esta forma, los expertos en computación y estadística deben saber algo de astronomía y los astrónomos deben entender al menos los aspectos básicos de computación y estadística.

Durante los últimos años se han creado varios de estos grupos interdisciplinarios, los cuales han sido exitosos en el desarrollo de nuevos métodos para abordar problemas astronómicos a través de grandes volúmenes de datos. Algunos ejemplos son: detección de transientes [Bailey, Aragon, Romano, Thomas, Weaver, & Wong, 2007] [Brink, et al., 2013] , clasificación morfológica de galaxias [Ball, et al., 2004] [Huertas-Company, Aguerri, Bernardi, Mei, & Sánchez Almeida, 2011] [Lintott, et al., 2011] clasificación de espectros [Daniel, Connolly, Schneider, Vanderplas, & Xiong, 2011], obtención de períodos en curvas de luz [Huijse, Estévez, Zegers, Principe, & Protopapas, 2011] [Graham, et al., 2013], y descubrimiento de relaciones entre variables astronómicas [Graham, Djorgovski, Mahabal, Donalek, & Drake, 2013] entre otros.

Otro aspecto interesante de estas colaboraciones, es cómo al enfrentar problemas astronómicos se descubren nuevos problemas computacionales y se crean nuevos algoritmos, los cuales incluso pueden ser aplicados a otras áreas. Algunos ejemplos de esto han ocurrido al indexar y calzar series de tiempo [Keogh, Wei, Xi, Vlachos, Lee, & Protopapas, 2009], en clasificación con datos incompletos [Pichara & Protopapas, 2013] y detección y corrección de etiquetas sesgadas en aprendizaje supervisado [Cabrera, Miller & Schneider, 2014].

En el caso particular de Chile, en el Centro de Modelamiento Matemático de la Universidad de Chile llevamos más de cinco años trabajando en estos temas. Hemos creado un laboratorio en Astroinformática (AstroLab) formado por un gran equipo interdisciplinario que incluye científicos del área de astronomía, modelamiento matemático, estadística, machine learning, HPC, procesamiento de imágenes, ingeniería eléctrica, etc. Algunos de los proyectos que se trabajan actualmente en el AstroLab incluyen detección de transientes en tiempo real, reconstrucción de imágenes interferométricas, clasificación morfológica de galaxias, reglas de asociación para líneas moleculares, registro y superresolución de imágenes astronómicas, entre otros.

CONCLUSIONES

CON LOS NUEVOS INSTRUMENTOS, DURANTE LOS PRÓXIMOS AÑOS SE NOS VIENEN ENCIMA GIGA, TERA Y PETABYTES DE DATOS ASTRONÓMICOS LOS CUALES DEBEN SER PROCESADOS EN TIEMPO REAL. ESTOS DATOS SERÁN COMPLETAMENTE NUEVOS Y LO QUE ELLOS CONTENGAN NO LO SABEMOS AÚN. ESPERAMOS ENCONTRAR NUEVOS TIPOS DE OBJETOS Y NUEVA ASTROFÍSICA, ASÍ COMO TAMBIÉN EXPLICAR LOS MISTERIOS QUE AÚN NO HEMOS PODIDO RESOLVER. PARA TENER ÉXITO EN ESTE DESAFÍO ES DE SUMA IMPORTANCIA LA CREACIÓN DE EQUIPOS INTERDISCIPLINARIOS DE INVESTIGADORES PARA ASÍ PODER ATACAR EL PROBLEMA CON TODO TIPO DE HERRAMIENTAS, TANTO MATEMÁTICAS Y ESTADÍSTICAS COMO COMPUTACIONALES. ■

AGRADECIMIENTOS

Muchas gracias a Sebastián Pérez por ayudar a recopilar algunos de los datos de este artículo. Este trabajo ha sido parcialmente financiado por el Centro de Modelamiento Matemático de la Universidad de Chile, por el proyecto Fondef D11I1060, y por la Beca de Doctorado Nacional de CONICYT.

BIBLIOGRAFÍA

Cabrera, G. F., Casassus, S., & Hitschfeld, N. (2008). Bayesian Image Reconstruction Based on Voronoi Diagrams. The Astrophysical Journal, 672, 1272-1285.

Cabrera, G. F., Miller, C. J., & Schneider, J. (2014). Systematic Labeling Bias: De-Biasing Where Everyone Is Wrong. 22nd International Conference on Pattern Recognition. Stockholm.

Lang, D., Hogg, D. W., Mierle, K., Blanton, M., & Roweis, S. (2010). Astrometry.net: Blind Astrometric Calibration of Arbitrary Astronomical Images. The Astronomical Journal , 139, 1782-1800.

Lintott, C., Schawinski, K., Bamford, S., Slosar, A., Land, K., Thomas, D., et al. (2011). Galaxy Zoo 1: data release of morphological classifications for nearly 900 000 galaxies. Monthly Notices of the Royal Astronomical Society , 410, 166-178.

Cornwell, T. J., & Evans, K. F. (1985). A simple maximum entropy deconvolution algorithm. Astronomy and Astrophysics , 143, 77-83.

Ball, N. M., Loveday, J., Fukugita, M., Nakamura, O., Okamura, S., Brinkmann, J., et al. (2004). Galaxy types in the Sloan Digital Sky Survey using supervised artificial neural networks. Monthly Notices of the Royal Astronomical Society , 348, 1038-1046.

Bailey, S., Aragon, C., Romano, R., Thomas, R. C., Weaver, B. A., & Wong, D. (2007). How to Find More Supernovae with Less Work: Object Classification Techniques for Difference Imaging. The Astrophysical Journal , 665, 1246-1253.

Bertin, E. (2011). Automated Morphometry with SExtractor and PSFEx. Astronomical Data Analysis Software and Systems XX (p. 435). Boston: Evans, I. N.;Accomazzi, A;Mink D.J.;Rots A.H.

Bertin, E., & Arnouts, S. (1996). SExtractor: Software for source extraction. Astronomy and Astrophysics Supplement, 117, 393-404.

Bijaoui, A., & Rué, F. (1995). A multiscale vision model adapted to the astronomical images. Signal processing , 46 (3), 345-362.

Borne, K., Accomazzi, A., Bloom, J., Brunner, R., Burke, D., Butler, N., et al. (2009). Astroinformatics: A 21st Century Approach to Astronomy. Astro2010: The Astronomy and Astrophysics Decadal Survey, Position Papers, no. 6.

Brink, H., Richards, J. W., Poznanski, D., Bloom, J. S., Rice, J., Negahban, S., et al. (2013). Using machine learning for discovery in synoptic survey imaging data. Monthly Notices of the Royal Astronomical Society, 435, 1047-1060.

Daniel, S. F., Connolly, A., Schneider, J., Vanderplas, J., & Xiong, L. (2011). Classification of Stellar Spectra with Local Linear Embedding. The Astronomical Journal , 142, 203.

Escoffier, R. P., Comoretto, G., Webber, J. C., Baudry, A., Broadwell, C. M., Greenberg, J. H., et al. (2007). The ALMA correlator. Astronomy and Astrophysics , 462 (2), 801-810.

Graham, M. J., Djorgovski, S. G., Mahabal, A. A., Donalek, C., & Drake, A. J. (2013). Machine-assisted discovery of relationships in astronomy. Monthly Notices of the Royal Astronomical Society , 431, 2371-2384.

Graham, M. J., Drake, A. J., Djorgovski, S. G., Mahabal, A. A., Donalek, C., Duan, V., et al. (2013). A comparison of period finding algorithms. Monthly Notices of the Royal Astronomical Society , 434, 3423-344.

Högbom, J. A. (1974). Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines. Astronomy & Astrophysics Supplement Series, 15, 417 - 426.

Huertas-Company, M., Aguerri, J. A., Bernardi, M., Mei, S., & Sánchez Almeida, J. (2011). Revisiting the Hubble sequence in the SDSS DR7 spectroscopic sample: a publicly available Bayesian automated classification. Astronomy and Astrophysics, 525, A157.

Huijse, P., Estévez, P. A., Zegers, P., Príncipe, J. C., & Protopapas, P. (2011). Period Estimation in Astronomical Time Series Using Slotted Correntropy. IEEE Signal Processing Letters , 18, 371-374.

Ivezic, Z., Tyson, J. A., Abel, B., Acosta, E., Allsman, R., AlSayyad, Y., et al. (May de 2008). LSST: from Science Drivers to Reference Design and Anticipated Data Products. arxiv.

Keogh, E., Wei, L., Xi, X., Vlachos, M., Lee, S.-H., & Protopapas, P. (2009). Supporting exact indexing of arbitrarily rotated shapes and periodic time series under Euclidean and warping distance measures. The VLDB Journal , 18, 611-630.

McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. (2007). CASA Architecture and Applications. In R. A. Shaw, F. Hill, & D. J. Bell (Ed.), Astronomical Data Analysis Software and Systems XVI, 376, p. 127. Tucson.

Miller, C. J., Genovese, C., Nichol, R. C., Wasserman, L., Connolly, A., Reichart, D., et al. (2001). Controlling the False-Discovery Rate in Astrophysical Data Analysis. The Astronomical Journal , 122, 3492-3505.

Minkowski, R. L., & Abell, G. O. (1963). The National Geographic Society-Palomar Observatory Sky Survey. In K. A. Strand, Basic Astronomical Data: Stars and stellar systems (p. 481). Chicago: University of Chicago Press.

Pichara, K., & Protopapas, P. (2013). Automatic Classification of Variable Stars in Catalogs with Missing Data. The Astrophysical Journal , 777, 83.

Pina, R. K., & Puetter, R. C. (1993). Bayesian image reconstruction - The pixon and optimal image modeling. Astronomical Society of the Pacific, Publications , 105, 630-637.

SDSS Collaboration. (2013). Data Volume Table. Retrieved 09 de 2014 from SDSS III: https://www.sdss3.org/dr10/data_access/volume.php

Stetson, P. B. (1987). DAOPHOT - A computer program for crowded-field stellar photometry. Astronomical Society of the Pacific, Publications, 99, 191-222.

Sutton, E. C., & Wandelt, B. D. (2006). Optimal Image Reconstruction in Radio Interferometry. The Astrophysical Journal Supplement Series, 162, 401-416.

Tony Hey, S. T. (2009). The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research.

Valdés, F. G., Campusano, L. E., Velásquez, J. D., & Stetson, P. B. (1995). FOCAS Automatic Catalog Matching Algorithms. Publications of the Astronomical Society of the Pacific , 107, 1119.

ENTREVISTA A RICARDO ZILLERUELO

| Por Pablo Barceló

RICARDOZILLERUELO

Mi nombre es Ricardo, me apasiona y obsesiona construir algoritmos, modelos matemáticos y escribir programas computacionales. Esto me ha llevado, hasta la fecha, a ser miembro de tres emprendimientos, entre ellos Zappedy. Éste último adquirido por Groupon, empresa para la que trabajo actualmente en California, Estados Unidos.

¿Qué es lo que más valoras de la formación del DCC y cómo ha influenciado tu carrera?

Diría que los puntos más importantes son sus sólidas bases en matemáticas, la diversidad de las personas en la Facultad, el énfasis de la carrera en los fundamentos de la Computación más que en las tecnologías de Computación y los espacios de investigación.

El desarrollo de sólidas bases matemáticas, sin duda me ha dado ventajas profesionales en Chile y Estados Unidos. Con la penetración de la Computación en diversas áreas, en un mundo donde los sistemas computacionales dan soporte a todas las áreas productivas y a cada una de nuestras vidas de manera cotidiana, sumado a las enormes cantidades de datos que estas interacciones producen, se han abierto miles de nuevas posibilidades para investigar, descubrir el mundo y resolver problemas a través de la Computación. Saber matemáticas me ha sido muy útil para modelar este mundo complejo y descubrir soluciones y maneras de hacer cosas de manera simple y efectiva, las cuales no serían obvias de otra manera.

La diversidad de la Universidad, en cuanto a líneas de pensamiento, orígenes económicos de sus alumnos y la posibilidad de compartir con alumnos y profesores de otras carreras. El primer emprendimiento en el que estuve involucrado jamás hubiera sido posible si no hubiera participado en una organización del Departamento de Ingeniería Industrial llamada Despertar, que buscaba incentivar la innovación en los alumnos. Allí conocí a un grupo de alumnos de Ingeniería Industrial con los cuales hicimos nuestra primera empresa llamada Metrik y que luego pasó a llamarse Suipit.

A su vez, aprecio la formación general a la que accedemos en la Universidad. Por ejemplo, tuve la oportunidad de tomar cursos de filosofía, sociología y biología del conocimiento, los cuales ampliaron mi manera de pensar.

Respecto a la formación de la carrera de Computación, creo que su enfoque en los fundamentos teóricos más que en la tecnología ha sido clave para mí, ya que nos permite conocer los elementos comunes entre diversas tecnologías y potencia la capacidad de aprenderlas e incorporarlas como herramientas. Además, permite reconocer nuevas aplicaciones de la tecnología existente y predecir más o menos la evolución de la industria.

Los espacios de investigación, también me aportaron profundamente. En cuarto año de mi carrera entré a trabajar al Laboratorio de Inteligencia Computacional del Departamento de Ingeniería Eléctrica. Comencé ahí porque estaba interesado en el área de Aprendizaje de Máquinas, la cual hoy es muy demandada en la industria (sobre todo en el área de procesamientos de datos masivos). Tuve una experiencia increíble y aprendí en profundidad sobre los algoritmos y técnicas actuales del área. En conjunto con Pablo Estévez, el profesor que dirigía ese laboratorio, y otros alumnos, desarrollé nuevos algoritmos y pude publicar en revistas internacionales un par de veces. Esto me entregó elementos útiles para mi carrera, además del dominio profundo de un área demandada por la industria. Me enseñó a leer publicaciones, cómo distinguir buenas de malas y cómo aplicarlas. Esto ha sido muy importante para mi carrera profesional, dado que en las publicaciones he encontrado conocimiento útil que he aplicado en mi trabajo y proyectos.

A la vez, trabajar en el laboratorio me sirvió para subir mi autoestima profesional y demostrarme que podía ser altamente competitivo a nivel internacional.

Más adelante, durante mi postgrado en Ciencias de la Computación en la Universidad de Chile, entré a trabajar en el centro de investigación que Yahoo! tiene en el Departamento de Ciencias de la Computación. Allí tuve la oportunidad de trabajar con investigadores como Georges Dupret (mi profesor guía) y Carlos Castillo. En el laboratorio, mi investigación estuvo centrada en el área de Recuperación de la Información, y estudié técnicas para el procesamiento de datos masivos, muy populares en estos últimos años. Las publicaciones que hice trabajando para estos laboratorios han sido excelentes credenciales profesionales en el extranjero, probablemente más que mi título profesional.

Por el contrario, ¿qué le agregarías a la formación del DCC, que pueda ayudar a formar Ingenieros en Computación mejor preparados para los desafíos actuales?

Una crítica personal que hago a la Facultad en general, es el alto nivel de exigencia a la que somete a sus alumnos. Constantemente los alumnos sienten que no merecen estar estudiando ahí, producto de la gran carga académica y lo exigente de las pruebas que buscan no sólo que los alumnos sepan la materia, si no aún más, que sean capaces de aplicarla de manera creativa e ingeniosa. Esto en sí no es malo, pero en el contexto de una prueba de tres horas o ¡incluso ocho!, es muy estresante. En mi opinión esto mina la confianza personal de los alumnos, limitando su capacidad de creer que pueden hacer cosas y agregar valor al mundo. Esto contrasta con otras universidades donde uno claramente ve una mayor tendencia de sus alumnos a crear empresas o a involucrarse en proyectos de innovación.

→

Pienso que la creatividad debería incentivarse en otro tipo de contextos, tales como proyectos de investigación independientes de los cursos y semestres. Creo que en la Facultad ya existen espacios para esto en los distintos laboratorios y proyectos, tales como el laboratorio de Inteligencia Computacional o el proyecto Eolian en el Departamento de Ingeniería Eléctrica, el CIWS, PLEIAD, NIC Labs y Yahoo! Labs en el DCC. Pero aún no existen suficientes incentivos para que los alumnos se involucren en este tipo de proyectos.

Tu carrera ha estado directamente ligada a temas de innovación. ¿Le recomendarías a alguien que esté interesado en estos temas seguir una carrera de Computación?

Primero me gustaría destacar que en mi opinión, el interés en la innovación es una consecuencia más que un fin en sí. Por esta razón, no recomiendo a ningún joven que se interese en la innovación. Por el contrario, recomiendo que se dediquen a áreas que los apasionen por descubrir alguna dimensión del mundo y resolver problemas que entreguen valor a las personas. Que busquen ser excelentes en esas áreas y trabajadores incansables.

Personalmente la carrera de Computación me ha traído muchas satisfacciones. Creo que es una de las pocas carreras profesionales que permiten agregar valor a muchas personas con pocos recursos. Un pequeño equipo en una pieza, con una buena conexión a Internet y sus computadores, es capaz de generar grandes innovaciones. Esto es ideal en el escenario chileno donde no es muy fácil encontrar financiamiento para proyectos riesgosos, lo que hace difícil innovar en áreas que requieren de mayor infraestructura y/o número de personas.

Los problemas en los que he podido trabajar, las personas que he podido conocer y los lugares donde me ha llevado, me ha llenado de satisfacciones. Ver cómo el código que uno escribe es usado por millones de personas, o es capaz de encontrar respuestas entre los teras y teras de data, anteriormente inalcanzables sin el poder computacional de las máquinas, me ha dado una fuerte sensación de logro personal.

Hoy los sistemas computacionales están presentes en prácticamente todas las dimensiones de nuestras vidas y no creo que esa tendencia cambie en el futuro. Por el contrario es muy probable que nuestra relación con los sistemas computacionales se vuelva más y más interconectada en los años venideros. Y sin duda, saber programar será un requisito tan fundamental como saber leer. Sin importar a qué área uno se dedique, tener buenos fundamentos en Computación es y será una gran ventaja profesional.

Desde antes de terminar tu Magíster comenzaste a desarrollar diferentes emprendimientos. ¿Puedes contarnos un poco de qué se trataban y tu experiencia en ellos?

A finales de mi segundo año de universidad me involucré en mi primer emprendimiento llamado Metrik, el cual posteriormente pasó a llamarse Suipit. La visión de la compañía era hacer las conversaciones entre personas más efectivas. Para ello desarrollamos una serie de herramientas que registraban e indexaban todas las conversaciones escritas como emails, actas de reuniones y documentos en general. Nuestras herramientas organizaban ese contenido para facilitar su acceso y entregaba trazabilidad a cómo esas conversaciones se transformaban en acuerdos y tareas. Además, generaba métricas que facilitaban a nuestros usuarios concientizarse sobre las conversaciones que se daban en su organización. Hay diversas razones por las que creo que este proyecto no fue exitoso. Personalmente, creo que aún era muy inmaduro profesionalmente y mi foco estaba en los estudios más que en el proyecto. Finalmente mi falta de foco me alejó del equipo, al punto en que fue mejor desligarme del proyecto.

Una vez terminada mi carrera de Ingeniería Civil en Computación y mientras empezaba a trabajar en mi Tesis de Magíster en Ciencias de la Computación, me involucré en otro emprendimiento llamado Expenews. La idea fue registrar y difundir todas las expediciones extremas del mundo. Personalmente disfruté mucho trabajando para ese proyecto. Por un lado las personas involucradas eran muy interesantes y técnicamente el proyecto tenía desafíos ya que era necesario que los expedicionarios pudieran enviar sus actualizaciones desde cualquier lugar del planeta, donde la única forma de comunicación disponible era a través de satélites. Después de dedicarnos por alrededor de un año al proyecto, vimos que a pesar de resolver un problema real y que a los expedicionarios les gustaba usar nuestro producto, el mercado era muy pequeño y no daba abasto para financiar a todos los miembros del equipo.

Por esto, finalmente decidimos enfocarnos en otros proyectos y delegar la mantención del servicio a otra persona, que hasta el día de hoy lo mantiene andando y sirviendo a nuevas expediciones.

Durante este tiempo de transición, junto a otro miembro del equipo de Expenews nos involucramos en Zappedy, siendo los primeros desarrolladores full time del emprendimiento. Nos fuimos a Palo Alto, California, a trabajar en el garaje de la casa del fundador del proyecto por tres meses, construyendo la tecnología y probando diversas alternativas del producto. La visión original de la compañía era conectar el mundo de la publicidad online con el mundo offline de las transacciones de las tarjetas de créditos en tiendas. Esto con el objetivo de

hacer calzar avisos publicitarios hechos en la Web con transacciones de tarjetas de créditos hechas en las tiendas, y de esta manera permitir que comerciantes sin presencia online pudieran hacer campañas publicitarias en la Web y medir el impacto de éstas en su negocio. Con el tiempo nuestro producto cambió y terminamos construyendo un sistema que permitía a cada comerciante en Estados Unidos implementar programas de fidelización de clientes. Para ello no necesitaban hacer ningún cambio: Los clientes podían seguir utilizando sus tarjetas de crédito y los terminales de venta no requerían ninguna modificación. Nuestro producto "mágicamente" recolectaba la información necesaria para implementar los programas de fidelización y entregar la información al comerciante.

En ese entonces comenzamos a tener clientes, y el equipo conformado por siete personas decidió levantar capital de riesgo. Tratamos inicialmente en Estados Unidos, donde encontramos fondos interesados, pero solo dispuestos a financiarnos si otro fondo también invertía en nosotros. A partir de lo cual decidimos buscar fondos en Chile. Después de tocar diversas puertas, dos fondos estuvieron dispuestos a invertir en nosotros, después de lo cual uno de los fondos en Estados Unidos decidió invertir en nosotros. Con 1M de dólares nos pusimos las pilas como equipo y trabajamos como nunca para sacar el proyecto adelante. Abrimos oficinas sencillas en Estados Unidos y Chile. A lo largo del tiempo el equipo fue creciendo y llegó a estar conformado por once chilenos y dos estadounidenses, cada uno de una calidad personal y profesional de excelencia. Haber sido parte de ese equipo ha sido uno de los mayores honores y mejores experiencias profesionales de mi vida.

Alrededor de un año después empezamos a trabajar con varios sitios de cupones online, muy de moda en esa época. Nuestro interés en trabajar con ellos radicaba en el hecho de que estos sitios tenían varias tiendas como clientes, y para nosotros se transformaron en excelentes canales de distribución, dado que al momento de cerrar un trato con uno, teníamos acceso a miles de tiendas. Uno de estos sitios era Groupon, el cuál después de un tiempo decidió comprar nuestra empresa e importar el equipo completo a sus oficinas en Palo Alto.

Dada tu experiencia personal, ¿existen en Chile las condiciones para que los jóvenes interesados en la innovación puedan desarrollar emprendimientos tecnológicos de calidad?

Creo que Chile tiene algunas de las condiciones necesarias para la innovación, pero carece de otras, lo cual hace que ésta sea poco común en contraste con otros países.

En Chile hay muy buenos profesionales, totalmente comparables a profesionales de excelencia de otros países tales como Estados Unidos, Alemania, India, China, etc. Existen muy buenas universidades que forman excelentes profesionales y desarrollan investigación de nivel mundial. Además, siento que en Chile hay interés en el desarrollo de proyectos de innovación. Esto es visible en la cobertura en los medios de comunicación, los programas de apoyo del Estado y el interés de las personas en desarrollar proyectos personales. Hoy, además, es mucho más aceptado socialmente dedicarse al emprendimiento de lo que era hace una década atrás, y se entiende que ésta es el desarrollo de la inspiración y las herramientas para descubrir e idear soluciones que agreguen valor a otros.

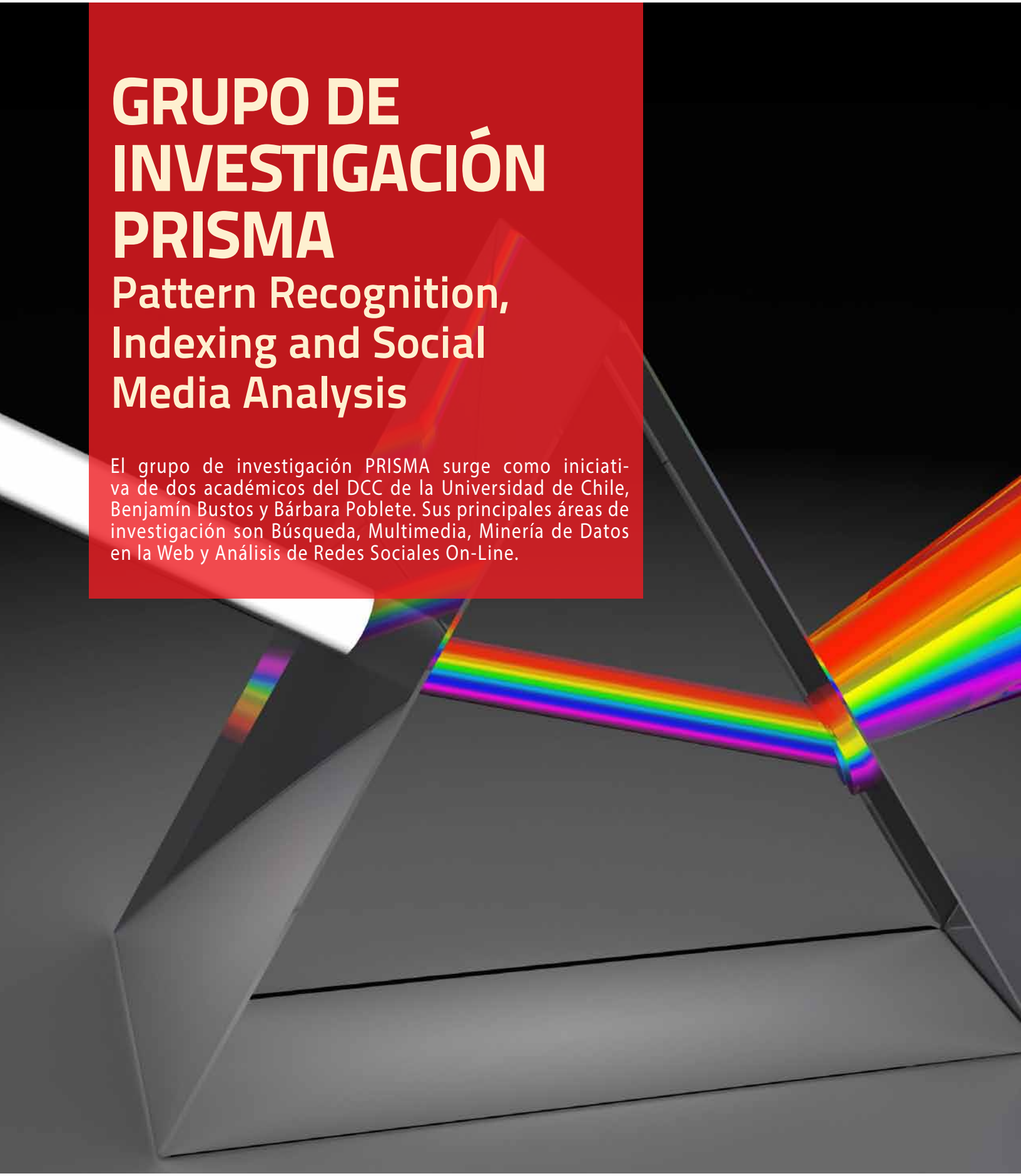
Respecto a las carencias, creo que la más importante es el tamaño de nuestra población. Somos un país muy pequeño y el ingreso promedio es tres veces menor que el de países con mejores condiciones para la innovación. La principal consecuencia de esto es que el tamaño del mercado interno es muy pequeño. Esto dificulta a las personas trabajando en nuevos proyectos encontrar la sustentabilidad necesaria –ya sea mediante financiamientos o venta– para no preocuparse de sus gastos de vida, delegar a terceros aquellas cosas que no son centrales de su negocio, y la adquisición del talento e infraestructura necesaria para desarrollar la tecnología. Para contrarrestar esta variable, es común escuchar que los emprendedores chilenos debieran enfocarse en el mundo y no en Chile. Sin embargo, en mi opinión esta visión no es realista, dado que no creo que exista "Chile y el resto del mundo". El resto del mundo está constituido por varios países independientes con mercados diversos, sujetos a distintas regulaciones, diferentes culturas y otras necesidades. En consecuencia, ahora la empresa en vez de dedicarse a resolver los problemas de un lugar, tiene que dedicarse a resolver estos problemas en todos los países en los que se enfoque. Empresas como Apple, Google, Microsoft, Facebook, Yahoo!, Groupon, etc. partieron totalmente enfocadas en el mercado local estadounidense, y una vez que fueron exitosas en ese mercado local, gracias al tamaño de éste, tuvieron los recursos suficientes para "comprar" su globalización por medio de la adquisición de empresas en otros países, la contratación de servicios de consultoras que los apoyaran en ámbitos legales y culturales, la posibilidad de abrir sucursales y contratar personas en los países donde se expandieron, etc.

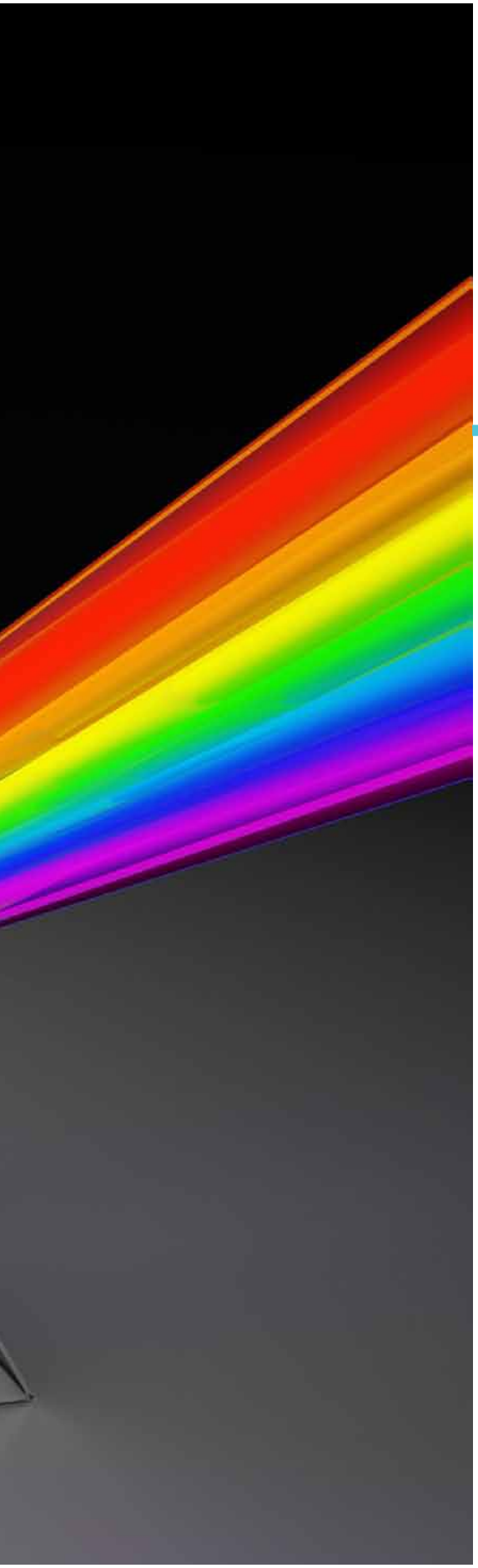
Creo que los emprendimientos nacionales debieran enfocarse en un mercado que tenga el tamaño para asegurar la globalización de la compañía. Para esto me parece que es importante que el núcleo de la empresa esté físicamente presente en aquel mercado y que algunos miembros de su equipo sean originarios del país donde están insertos. Es imprescindible que las personas que diseñan el producto estén día a día cerca de sus clientes y entiendan la cultura. ■

GRUPO DE INVESTIGACIÓN PRISMA

Pattern Recognition, Indexing and Social Media Analysis

El grupo de investigación PRISMA surge como iniciativa de dos académicos del DCC de la Universidad de Chile, Benjamín Bustos y Bárbara Poblete. Sus principales áreas de investigación son Búsqueda, Multimedia, Minería de Datos en la Web y Análisis de Redes Sociales On-Line.





BÁRBARAPOBLETE

Profesora Asistente del Departamento de Ciencias de la Computación de la Universidad de Chile. PhD en Computación, Universitat Pompeu Fabra (2009); Magíster en Ciencias mención Computación, Universidad de Chile (2004); Ingeniero Civil en Computación, Universidad de Chile (2004). Líneas de Especialización: Minería de grandes volúmenes de datos; Minería de Logs de Buscadores; Privacidad de Datos en la Web; Análisis de Redes Sociales en línea.

bpoblete@dcc.uchile.cl

En el área de Búsqueda, la investigación actual del grupo se centra en búsqueda por similitud de objetos 3D, búsqueda de anomalías en series temporales multivariantes y búsqueda en la Web. Para este último tema, búsqueda en la Web, la investigación desarrollada por el grupo tiene como finalidad mejorar la tecnología utilizada en los motores de búsqueda, tanto para texto como para multimedios. Para esto el grupo colabora activamente con Yahoo! Labs Santiago. Adicionalmente, para los temas de Minería de Datos en la Web y Análisis de Redes Sociales, el grupo trabaja con grandes volúmenes de datos extraídos de fuentes públicas de información como Twitter y otras plataformas sociales en línea. El objetivo principal de este tipo de investigación es la extracción de conocimiento útil en forma automática, por ejemplo en temas de comprensión de eventos a nivel mundial y noticias.

En el área de Multimedia, el grupo PRISMA ha desarrollado investigaciones en las áreas de búsqueda por contenido en vídeos, detección de copias, y búsqueda en colecciones de imágenes basada en *sketches* (o bosquejos).

En detalle, la investigación del grupo se compone de los siguientes temas:

- Búsqueda de multimedia en la Web: trabajo centrado principalmente en el estudio de características visuales y metadatos de objetos multimedia para mejorar las técnicas actuales de recuperación de la información en esta área. Esto contempla, entre otros, el etiquetado automático de imágenes.

- Análisis de sentimiento y opinión en redes sociales en línea: este trabajo tiene por objetivo estudiar diferentes perfiles emocionales de los usuarios de Twitter, además del análisis de técnicas de predicción de emociones y opiniones en plataformas sociales. Esto incluye el estudio de sentimientos en torno a eventos como elecciones presidenciales.

- Q&A en Twitter: este trabajo analiza la dinámica de preguntas y respuestas en redes sociales y microblogs, con el objetivo de detectar patrones interesantes.

- Visualización de eventos: esta línea de investigación busca dar sentido de forma visual al acontecer mundial reflejado en las redes sociales. El objetivo es permitir al usuario detectar información valiosa e

→

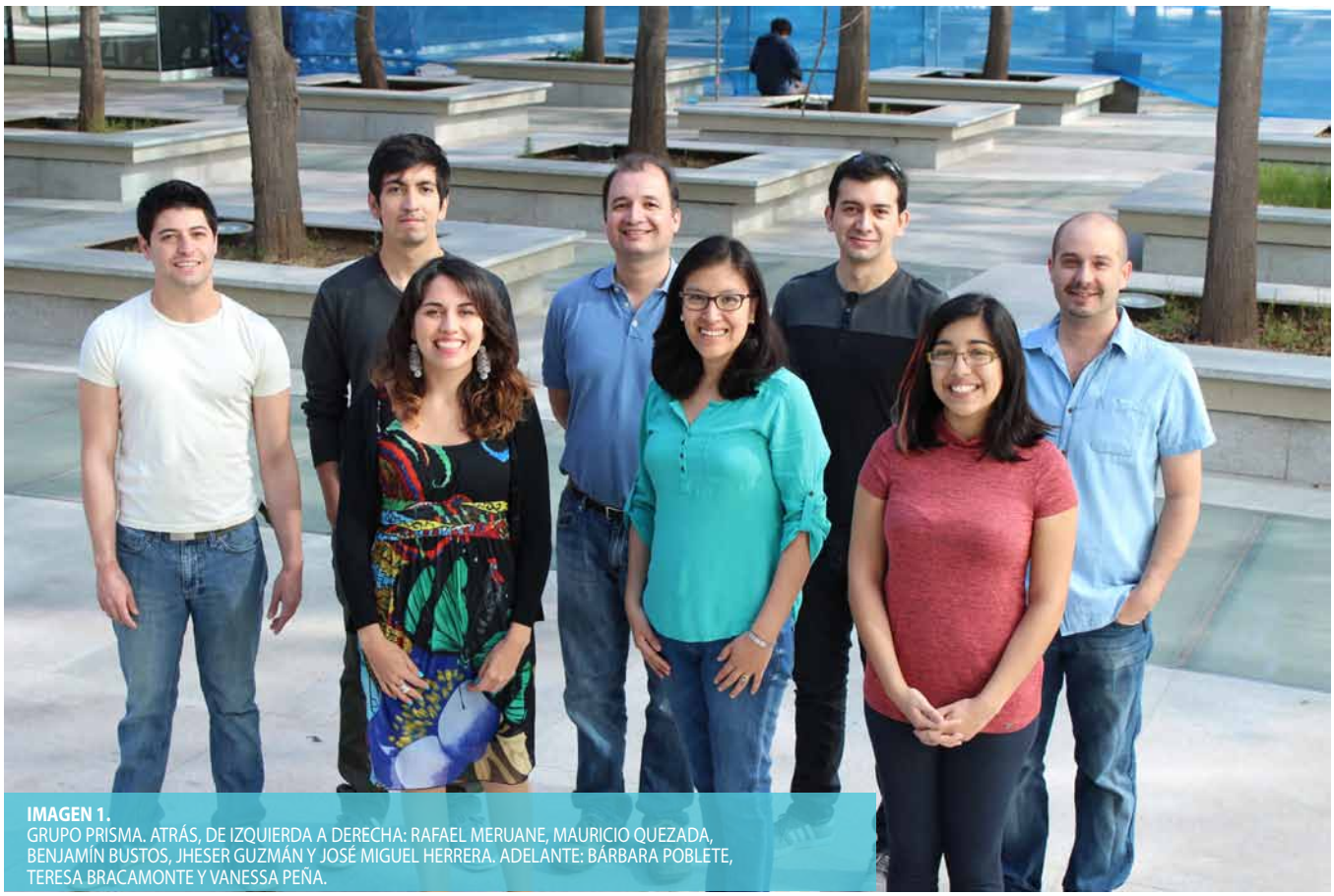


IMAGEN 1.
GRUPO PRISMA. ATRÁS, DE IZQUIERDA A DERECHA: RAFAEL MERUANE, MAURICIO QUEZADA, BENJAMÍN BUSTOS, JHESER GUZMÁN Y JOSÉ MIGUEL HERRERA. ADELANTE: BÁRBARA POBLETE, TERESA BRACAMONTE Y VANESSA PEÑA.

interesante que no se puede percibir trivialmente por otros medios.

- Estudio de noticias: este trabajo consiste en caracterizar y analizar el ciclo de vida de las noticias que se comentan en las redes sociales. El objetivo es entender diferentes tipos de eventos para luego reconocerlos en forma automática.

- Detección de eventos en tiempo real: esta investigación tiene por finalidad la detección de eventos emergentes en redes sociales en tiempo real. Los principales desafíos que se enfrentan en este tema son la volatilidad y gran volumen de los datos.

- Búsqueda de anomalías en series temporales: en esta investigación se desarrollan algoritmos para encontrar anomalías (patrones fuera de lo común) en series temporales multivariantes. Esto tiene aplicaciones, por ejemplo, en el análisis de datos científicos.

- Matching de formas 3D: en esta investigación se están estudiando y desarrollando algoritmos de búsqueda y matching en colecciones de objetos 3D, basados en el cálculo de características locales de los objetos. Éstos tienen aplicaciones muy diversas, como por ejemplo en el tema de Patrimonio Cultural, donde uno de los problemas es la reconstrucción de artefactos arqueológicos con apoyo computacional.

En la actualidad PRISMA está compuesto por seis alumnos de Doctorado y dos alumnos de Magíster.

El grupo ha contado con financiamiento de diferentes fuentes, entre ellas CONICYT por medio de dos proyectos FONDECYT y un proyecto FONDEF, además de fondos de investigación provenientes de Yahoo! Inc. por medio de su "Faculty Engagement Program" y el proyecto U-Inicia de la Universidad de Chile.

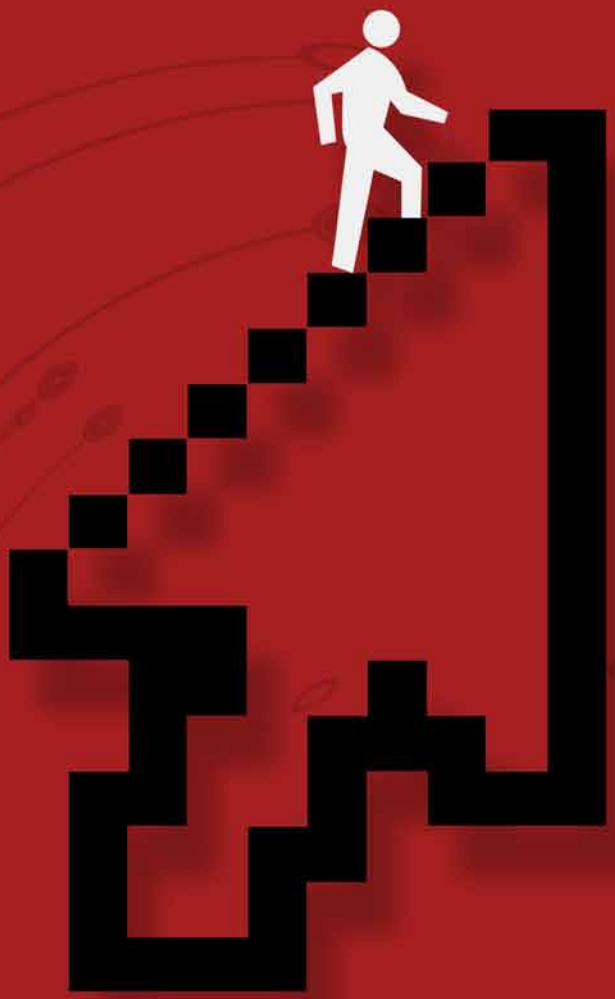
Las principales conferencias y revistas en que participa y publica el grupo son: SIGIR, WWW, KDD, MM, CIKM, ECIR, ICCV. ■

MAGÍSTER

- Tecnologías de la Información (vespertino)

DIPLOMAS DE POSTÍTULO

- Calidad de Software
- Ciencia e Ingeniería de Datos
- Gestión Informática
- Ingeniería de Software
- Ingeniería y Calidad de Software
- Seguridad Computacional
- Tecnologías de Información



PROGRAMA DE EDUCACIóN CONTINUA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

Sigue avanzando

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

facebook.com/pec.dcc

capacita@dcc.uchile.cl

www.dcc.uchile.cl/pec

2 2978 4965

Bits DE CIENCIA

www.dcc.uchile.cl
revista@dcc.uchile.cl

fcfm
Ciencias de la Computación
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE