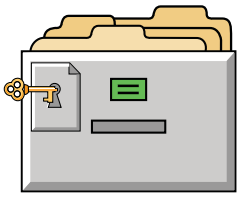
La INFORMACIÓN es PODER... sobre todo si está en una **base de datos**



Hugo César Coyote Estrada

Con la aparición de la computadora y su gran capacidad de almacenamiento, gobiernos, instituciones y empresas resguardan y organizan millones de datos, resultado de actividades cotidianas. Hoy en día las computadoras y redes, de gran capacidad y bajo costo, soportan mucha información y ofrecen magnífico desempeño.

El viejo adagio "la información es poder" destaca la necesidad de registrar, organizar y explotar los datos producto del acontecer de la vida y del comportamiento de la naturaleza. Por ello, la humanidad ha buscado preservar el conocimiento en forma de datos a través de medios que han perdurado más allá de la existencia misma de las culturas que los crearon. Los medios empleados para el almacenamiento han sido muy diversos: piedras, troncos y cortezas de árboles, huesos de animales, papel y, últimamente, medios magnéticos, semiconductores y ópticos. La naturaleza misma ha encontrado múltiples formas de transmitir información de padres a hijos, como lo atestigua el ácido desoxirribonucleico (ADN) y las miles de grandes moléculas que intervienen en ese maravilloso y único proceso de crear y preservar la vida.

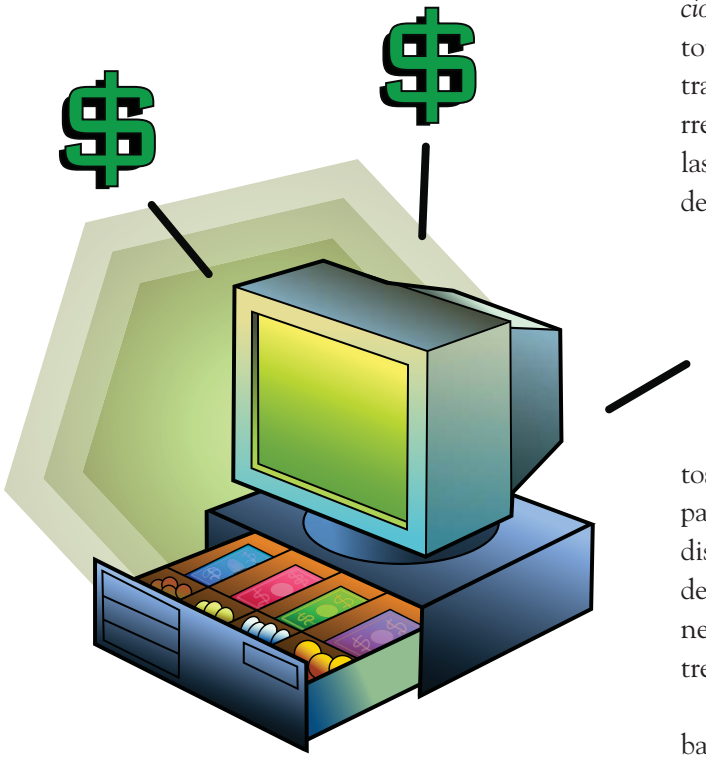
Con la aparición de las computadoras, con su enorme velocidad de cálculo y su gran capacidad de almacenamiento, empresas, gobiernos y universidades vieron en ellas el medio ideal para almacenar, organizar y explotar de manera intensiva los millones de datos producto de sus operaciones diarias, es decir, sus *bases de datos*, y su ulterior análisis en busca de mejoras continuas o para obtener mayor eficiencia.

En la actualidad prácticamente no existe una sola organización de un tamaño mediano o grande que no emplee de manera amplia bases de datos, y su uso se extiende cada día a un mayor número de sectores. Esta conclusión es confirmada

en el acontecer diario; tan es así que en México cada día son más las pequeñas y medianas empresas (PyMES) que almacenan toda su información en bases de datos. La inversión requerida para este tipo de empresas en realidad no es grande; con unos 6 mil pesos pueden adquirir una computadora que cuente con 250 *gigabytes* (un *gigabyte* equivale a mil millones de *bytes*) de disco duro, y para no incurrir en mayores gastos se sugiere emplear un sistema manejador de bases de datos (DBMS, por sus siglas en inglés) gratuito, como MySQL o PostGres, entre otros.

Para darnos una idea de lo que representan 250 *gigabytes* de almacenamiento, basta decir que sería suficiente para almacenar una base de datos que contuviera la información de todos los mexicanos, bajo ciertas circunstancias. Según el censo de 2010, en el momento que se tomó esa "fotografía" habíamos 112 millones 322 mil 757 habitantes en el territorio nacional; si se dedicaran 2 mil caracteres (*bytes*) para guardar los datos personales de cada mexicano, el tamaño de esa base de datos requeriría poco menos de 225 *gigabytes*.

Las grandes organizaciones requieren equipo de cómputo y redes de mucha mayor capacidad, así como DBMS que soporten grandes cargas de trabajo y ofrezcan mayor desempeño. Además, también buscan tolerancia a fallas, seguridad y protección de los datos, y compañías que les den servicio técnico y las auxilien en momentos críticos. Dentro de ese sector de bases de datos "propietarias", las compañías dominantes son Oracle, con su DBMS del mismo nombre; IBM, con DB2, y Microsoft, con su SQL Server. Los DBMS del mundo del *software* libre juegan también un papel muy importante en este sector de compañías gigantes: Google, la compañía reconocida por tener la base de datos más grande del mundo, usa en sus aproximadamente medio millón de computadoras, localizadas en sus diversos centros de datos, una base de datos que ocupa del orden de 200 *petabytes* (millones de *gigabytes*) con el DBMS MySQL, con modificaciones hechas por la propia Google. En esta singular base de datos se resuelven todas las consultas que los usuarios lanzamos al buscador de Google; a su vez las "arañas" (programas buscadores de datos) de Google la actualizan constantemente con información extraída de las decenas de millones de servidores de red existentes alrededor del mundo.



Por la forma de uso de las bases de datos, éstas se dividen en dos grandes grupos: bases de datos empleadas para la operación de las organizaciones o *transaccionales*, y bases de datos *analíticas*, empleadas para la toma de decisiones. Normalmente las bases de datos transaccionales tienen que soportar el acceso concurrente de centenas o miles de usuarios, mientras que las bases de datos analíticas las usan un grupo reducido de directivos de las organizaciones (véase el artículo de Gilberto Martínez Luna en este mismo número de *Ciencia* sobre las bases de datos analíticas).

Por su ubicación geográfica, las bases de datos pueden ser *centralizadas* o *distribuidas*. En ocasiones, aunque las bases de datos se encuentren en un solo sitio, corren en máquinas paralelas para incrementar su desempeño; en otras se distribuyen lógica y físicamente con el mismo objetivo de incrementar su desempeño, y además por cuestiones de seguridad, como por ejemplo resistir a un desastre en el centro de datos de una gran organización.

A continuación analizaremos la evolución de las bases de datos, sus estándares, su estado actual, sus

principales retos y lo que se ha realizado en México en esa disciplina.

La evolución de las bases de datos

El concepto de bases de datos nació hace aproximadamente cincuenta años, como respuesta al desorden que se generaba en las empresas debido a que los programas manejaban sus propios archivos. Es fácil imaginar que un cambio en la estructura de un archivo significaba cambios en todos los programas que lo empleaban, reduciendo enormemente la productividad. Justamente la idea central que subyace a las bases de datos es la independencia de los datos, entendida como "la inmunidad de las aplicaciones [programas] a los cambios en la estructura de almacenamiento y en la estrategia de acceso a los datos".

Los primeros DBMS nacieron casi de forma simultánea en General Electric y en IBM, con sus productos de software IDS (*Information Data Storage*) e IMS (*Information Management System*), respectivamente. El modelo de datos elegido por esos sistemas fue *jerárquico*. Esta estructura es simple y fácil de entender; sin embargo, plantea serios inconvenientes debido a su incapacidad para modelar algunas relaciones existentes entre los datos. A finales de los sesenta se planteó un *modelo de red*, el cual proporcionaba mayor flexibilidad. Pero en ambos modelos los usuarios tenían que estar enterados de la organización física de sus datos, de apuntadores y otros detalles computacionales que hacían difícil lograr la independencia de los datos.

En 1970 nace el *modelo relacional*, propuesto por E. F. Codd (1970), de IBM, el cual sustituyó a los otros modelos, y que actualmente es el modelo de datos dominante. El modelo relacional se basa en el concepto de *relación matemática*, pero que en el mundo de las bases de datos se presenta como un modelo de tablas. Codd mismo introdujo la *notación tabular* para representar las relaciones, definiéndolas así:

> Una tabla la cual representa una relación n-aria R tiene las siguientes propiedades: 1) Cada fila representa una n-tupla de R; 2) El ordenamiento de filas es irrelevante; 3) Todas las filas son distintas; 4) El ordenamiento de columnas es significativo, el cual corresponde al ordenamiento $S_1, S_2, \ldots, S_n$ de los dominios sobre los cuales R está definida; y 5) El significado de cada columna es parcialmente expresado mediante el uso del nombre correspondiente del dominio.

El modelo relacional es el primero que realmente se planteó lograr la independencia de los datos, tan es así que el lenguaje de consultas propuesto desde sus orígenes no es procedural sino descriptivo. En un lenguaje *procedural* los estatutos se expresan de una manera tal que el programador tenga que pensar en la representación física de los datos; es decir, señalar el camino a seguir a través de un árbol, una red o alguna otra estructura que modela los datos. En cambio, en un lenguaje *descriptivo*, simplemente se describe la condición que debe cumplir el conjunto resultado de la consulta sin pensar en la representación física de los datos (véase Recuadro 1).

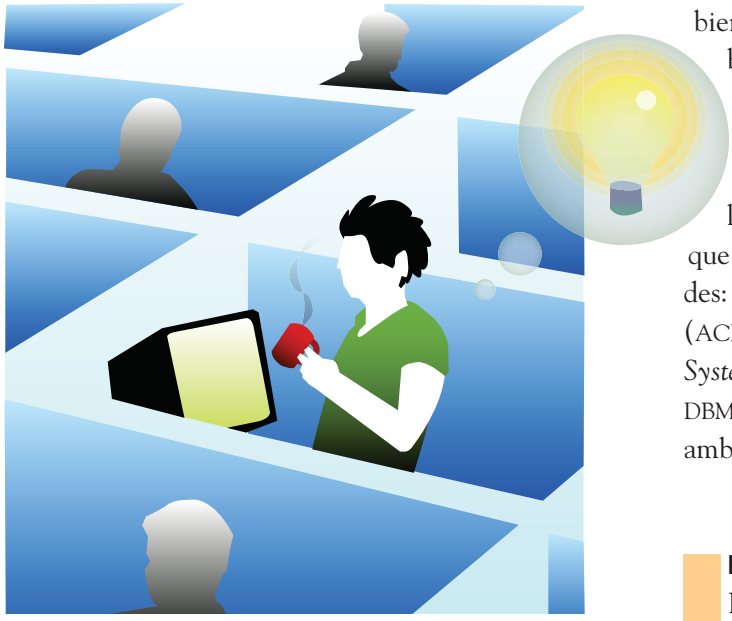
RECUADRO 1

Asumiendo que existe una base de datos cuyo esquema planteado de manera muy sintética es el siguiente:

1) Alumno (id#, nombre, escuela)
2) Profesor (id#, nombre, escuela, categoría)
3) Graduado (alumno, tutor, promedio, fecha)

Dar los nombres de los graduados cuyo promedio obtenido sea mayor o igual a 9, que hayan estudiado en la ESIME, se hayan graduado en el 2010 y cuyo director de tesis tenga la categoría de profesor titular B. La consulta en SQL queda como sigue:

```
SELECT alumno.nombre
FROM alumno, profesor, graduado
WHERE  alumno.escuela = 'ESIME' AND
       alumno.id = graduado.alumno AND
       graduado.promedio >= 9.0 AND
       YEAR(graduado.fecha) = 2010 AND
       graduado.tutor = profesor.id AND
       profesor.categoria = 'Titular B';
```



Para el estado de la tecnología de 1970, implantar en un DBMS el modelo relacional era considerado por algunos industriales e investigadores un reto imposible de lograr. Más que desanimar a los investigadores, los incitó a llevar a cabo investigaciones teórico-prácticas hasta lograr construir una teoría bien fundamentada, y DBMS relacionales que soportasen bases de datos con miles de usuarios concurrentes sin ocasionar pérdida de información. Las aportaciones fueron muchas, provenientes de la industria y la academia, pero de todas ellas destaca por su importancia el *System R*, desarrollado en IBM durante toda la década de los setenta. A partir del *System R* se produjeron aportaciones tan trascendentales que al paso de los años han dado como resultado el *procesamiento transaccional*, el lenguaje de consultas SQL (*Structured Query Language*), la ejecución eficiente de las consultas, reglas de normalización de bases de datos y el uso de estructuras de datos eficientes, entre otros avances.

Procesamiento transaccional

Es el responsable directo de que los DBMS puedan soportar miles de usuarios y garanticen la consistencia de la base de datos. Se basa en un principio muy simple y universalmente aceptado de una transacción comercial: "si acordamos que yo te compro un bien, yo te pago completamente y tú me entregas el bien". Es decir, la transacción debe llevarse a cabo completamente o nada de ella –es *atómica*–; debe ser *duradera*, estar *aislada* de otras transacciones comerciales y debe ser *consistente* en cuanto a que la persona que compra debe tener el dinero y el que vende debe tener el bien. Estas cuatro propiedades: *atomicidad*, *consistencia*, *aislamiento* y *durabilidad* (ACID), propuestas por Jim Gray e implantadas en el *System R*, son las que verdaderamente hicieron de los DBMS relacionales productos de *software* útiles en ambientes críticos.

El lenguaje de consultas SQL

En el *System R* se propuso el antecesor directo del actual SQL, denominado ESQL (E de *english*), creado por D. Chamberlin y R. Boyce. SQL es un lenguaje descriptivo, sumamente amplio y en constante evolución. No es un lenguaje de propósito general, por lo que se auxilia de funciones de biblioteca o preconstruidas; sin embargo, para el mundo de bases de datos es sumamente expresivo y muy fácil de usar (véase Recuadro 1).

SQL ha sido estandarizado en diversas ocasiones y se le puede considerar un lenguaje en constante evolución. Así, podemos citar las siguientes estandarizaciones: en 1986 el Instituto Nacional Estadounidense de Estándares (ANSI, por sus siglas en inglés) lo formalizó por primera ocasión; posteriormente en 1989 sufrió algunas ligeras modificaciones; en 1992 requirió de una revisión mayor, que generó el estándar ISO 9072, conocido como SQL 2. En 1999, dada la creciente influencia del paradigma de programación orientada a objetos, se generó el estándar SQL 3, en el cual se agregaron aspectos relacionados con ese paradigma. Esta técnica de programación facilita enormemente el manejo de objetos complejos como audio, imágenes, video, hojas de cálculo y otros más. A partir de aquí el modelo relacional tomó una gran influencia, lo que orilló a que el modelo actualmente imperante sea el relacional-objeto soportado por los DBMS más modernos.

Otra tecnología muy relacionada con el mundo moderno, y particularmente en la red, es la XML (*eXtensible Markup Language*, lenguaje de marcas extensible), la

cual se ha impuesto como el estándar *de facto* para el intercambio de archivos de información en la red. Es tal su influencia que los manejadores de bases de datos actuales usan internamente archivos XML y por supuesto existe un lenguaje para operar sobre ellos, denominado Xquery. En 2003, 2006 y 2009 el SQL ha sufrido revisiones para incorporar aspectos justamente asociados a XML y Xquery.

Optimización de consultas

Desde sus orígenes, se contempló que como SQL es un lenguaje descriptivo, resulta imprescindible que en la evaluación de las consultas se lleve a cabo una optimización para evitar el crecimiento exponencial de las tablas temporales (básicamente, las tablas involucradas en la cláusula FROM se operan entre sí mediante un producto cartesiano, lo que puede conducir a tablas excesivamente grandes). Las técnicas son muchas, y existe una abundante investigación para hacer de los DBMS productos de mayor desempeño (véase Recuadro 2).

Reglas de normalización

Desde la propuesta original del modelo relacional se contempló, como parte del diseño de bases de datos, el empleo de reglas de normalización de relaciones que ayudaran a evitar anomalías en las mismas. Codd propuso la primera, junto con un procedimiento para transformar la relación universal en primera forma normal. Boyce, coautor del ESQL, junto con Codd, propuso una forma normal que lleva sus nombres (BCNF). En este momento existen seis formas normales bien entendidas, y aún se investiga para encontrar otras.

Estructuras de datos eficientes

Sin la existencia de estructuras de datos eficientes, las bases de datos serían sumamente lentas y por tanto inservibles. Por ejemplo, mediante el empleo de los árboles B^+, que permiten implantar índices sobre una o más tablas, se puede localizar a un mexicano en la base de datos que contiene más de 112 millones de registros con tan sólo 10 accesos al disco. Existe una gran variedad de estructuras de datos eficientes, y es un tema de investigación abierto.

El futuro de las bases de datos

Los DBMS constituyen una tecnología madura, un mercado sano y en constante crecimiento, pero se siguen construyendo al viejo estilo del *System R* (Agrawal, 2009). Además, existen otros factores que obligan a plantearse nuevos retos y por tanto a llevar a cabo investigaciones novedosas. Entre éstos se pueden citar los siguientes:

RECUADRO 2

En nuestro ejemplo del Recuadro 1, en la cláusula FROM aparecen tres tablas entre las cuales, teóricamente, se lleva a cabo un producto cartesiano. Así, si hubiera 100 mil estudiantes, 50 mil graduados y 2 mil profesores, la tabla resultante del producto cartesiano de las tres tendría 10^{13} registros. Si el procesamiento de cada registro tomara 0.1 milisegundos, se requerirían 10^9 segundos, o sea 31.7 años.

Sin embargo, cualquier DBMS puede entregar ese resultado en cuestión de sólo unos pocos segundos. Para ello, evalúan de manera inteligente la consulta; por ejemplo, tal vez sólo 5 por ciento se graduó en 2010, y de ellos tan sólo 4 por ciento obtuvo un promedio mayor o igual a 9. Por tanto, de los 50 mil alumnos graduados, primero se seleccionan esos escasos 100 que cumplen con esas condiciones. Posteriormente, se busca quiénes de esos 100 alumnos estudiaron en la ESIME, para obtener su tutor y a continuación buscar que el tutor tenga la categoría deseada. De esta forma, no hay necesidad de crear esa tabla temporal gigante, sino más bien varias tablas temporales mucho más pequeñas. Además, esto contribuye a que el usuario no tenga que adquirir un disco duro de capacidad innecesaria.

- *La evolución tecnológica de los procesadores*: ahora cuentan con varios núcleos (*cores*), y cada núcleo tiene la capacidad para llevar a cabo varios hilos de ejecución (*hyperthreads*) en forma simultánea. Los DBMS prácticamente no aprovechan estas características.
- *La proliferación de datos y fuentes de datos*: inmensas cantidades de datos que se encuentran en la red en forma estructurada (formas), semiestructurada (correos electrónicos, *blogs*, etcétera), o sin ninguna estructura (texto). Los DBMS han integrado esta funcionalidad pero en forma muy limitada (véase el artículo de Alma Delia Cuevas en este mismo número de *Ciencia*).
- *Nuevas formas de hacer cómputo*: por ejemplo, el cómputo en nube (*cloud computing*), donde cientos de miles o millones de computadoras son rentadas a los usuarios en las condiciones tecnológicas, el número de procesadores y el tiempo que los usuarios requieran. Los DBMS que corren en la nube tienen aún una funcionalidad muy limitada.
- *Nuevas formas de construir y vender el* software: una tendencia es integrar el *software* mediante servicios que se adicionan de acuerdo con las necesidades de los clientes (*software as a service*). Los DBMS se tendrán que adaptar a esta novedosa forma de comprar el *software*.
- *Nuevos paradigmas para resolver problemas*: por ejemplo, tal como lo propone *MapReduce* de Google. Aquí los datos de entrada son distribuidos a cientos de miles de computadoras, donde el proceso *Map* realiza un procesamiento a los datos y sus resultados fluyen al proceso *Reduce*, que se ejecuta igualmente en ese inmenso número de computadoras y produce los resultados finales. La optimización de la evaluación de consultas puede aprovechar esta nueva forma de procesar datos.
- *Aparición de nuevas fuentes de datos*: existe una variedad enorme de fuentes de datos que de manera continua e ininterrumpida generan grandes volúmenes que no pueden ser almacenados y exigen un tratamiento en línea. Por ejemplo, reconocer la cara de las personas que entran a un banco para descubrir a un ladrón. Los DBMS deben saber tratar este tipo de consultas.
- *Nuevas formas de interactuar con los manejadores de bases de datos*: es necesario facilitar aún más el uso de los DBMS para que el gran público pueda construir y explotar sus propias bases de datos.

Sin duda alguna en los próximos años veremos propuestas muy notables en el campo de las bases de datos, y tal vez esta década se parezca a la fructífera década de los setenta.

Manejadores de bases de datos hechos en México

En México ha habido logros importantes en el desarrollo de DBMS relacionales, entre los que se pueden contar al menos tres:

- SiMBaD, hecho en el Centro Nacional de Investigación y Desarrollo Tecnológico (Cenidet), bajo la dirección de Rodolfo Pazos Rangel;
- SQLmx, hecho en el Centro de Investigación en Computación del Instituto Politécnico Nacional (IPN), bajo la dirección de Hugo Coyote Estrada y Gilberto Martínez Luna, y
- El manejador paralelo de bases de datos realizado por Jaime Aguilera como tesis de doctorado en el Centro de Investigación en Computación (CIC) del IPN.

SiMBaD fue desarrollado en Pascal en la década de los ochenta, y SQLmx se realizó primordialmente en el lenguaje de programación "C" y algunos módulos en JAVA en la década de los noventa. En ambos se realizaron compiladores para SQL y *drivers* (manejadores) JDBC, de tal forma que se podía insertar sentencias SQL dentro del código en JAVA; también ambos implantaron árboles B+ para crear índices en tablas o en grupos de ellas. Además, en SQLmx también se realizó un driver ODBC con el fin de emplear SQL dentro de código en "C", y se incorporó el manejo de XML en el DBMS. En los dos proyectos se llevaron a cabo diversas tesis de maestría y se contó con el apoyo económico del Consejo Nacional de Ciencia y Tecnología (Conacyt).

El DBMS paralelo de Jaime Aguilera fue realizado en la computadora paralela SP2 de IBM, con la cual contaba en esa época el CIC-IPN; la implantación se realizó en "C" empleando el ambiente MPI que permite paralelizar programas bajo el paradigma de *múltiples-datos un simple-programa*. Este DBMS implantó SQL e hizo gran énfasis en la optimización de consultas mediante el empleo del paralelismo. El acceso a las bases de datos se llevaba a cabo desde un lenguaje de cuarta generación propio, en el cual se insertaban estatutos SQL.

Existen otros trabajos mexicanos de gran relevancia en bases de datos; entre ellos es importante citar el trabajo realizado por el grupo dirigido por Renato Barrera Rivera en el Centro de Instrumentación de la Universidad Nacional Autónoma de México (UNAM). En esta institución se ha llevado a cabo recientemente la construcción de un núcleo mediador o extractor de bases de datos heterogéneas implantadas en diversos manejadores de bases de datos, denominado SIBA (*Sistema de Informática para la Biodiversidad y el Ambiente*). ¿Cuál es la razón de crear un núcleo mediador entre bases de datos si ya existe un estándar? La razón es muy simple: a pesar de ese estándar, los DBMS presentan muchas particularidades que afloran justamente cuando se quiere hacer interactuar bases de datos de distintos DBMS. Un mediador o núcleo extractor recibe una consulta en SQL, la analiza de acuerdo con un esquema global y la redirige a cada uno de los DBMS integrados, con el fin de que cada quien realice la parte de la consulta que le corresponde. Una vez recibida la respuesta de todos los DBMS, el mediador integra la respuesta de forma unificada y la proporciona al usuario. Este tipo de problema es muy común en las grandes instituciones en nuestro medio. Un mediador es la solución transparente cuando los datos se encuentran en bases de datos administradas por diversos DBMS.

Conclusiones

Las bases de datos son una tecnología que llegó para quedarse y nadie duda que en el futuro van a ser aún de mayor importancia. El modelo actualmente dominante es el objeto-relacional. Los DBMS actuales manejan una gran cantidad de tipos de datos simples y compuestos, pero a pesar de ello no manejan todos; día a día aparecen nuevos tipos de datos y nuevas fuentes de datos, de forma que es muy difícil prever el tratamiento que vayan a requerir. Se prevé que en unos cuantos años, todas las PyMES en nuestro país empleen bases de datos para su operación diaria.

Hugo César Coyote Estrada es doctor en computación por la Universidad de París VI. Sus áreas de interés son las bases de datos, los sistemas distribuidos y los sistemas operativos. Actualmente trabaja en el Centro de Investigación en Computación del Instituto Politécnico Nacional y es miembro de la Academia de Ingeniería de México. Ha sido profesor visitante de la Universidad de California en Irvine, e ingeniero de sistemas en Chorus Systèmes, Francia, y en Intel México. Ha colaborado en el Instituto de Investigaciones Eléctricas y en Telmex, así como en diversas universidades mexicanas, como el IPN, la Universidad Autónoma Metropolitana, la Universidad del Valle de México, el Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM), la Universidad Iberoamericana (UIA), el Instituto Tecnológico Autónomo de México (ITAM) y la Universidad Anáhuac.
Hugo.C.Coyote@gmail.com

Bibliografía

Agrawal, R. y colaboradores (2009), "The Claremont report on database research", *Comm.* ACM; 52(6): 56-65.
Codd, E. F. (1970), "A relational model of data for large shared data banks", *Comm.* ACM; 13(6): 377-387.