

# Tendencias tecnológicas del software y de la computación

96

\* Dr. Adolfo Guzmán Arenas

## Resumen

Se trata de predecir hacia dónde irán la tecnología, los paquetes y programas de cómputo actuales, qué ocurrirá a principios del tercer milenio, qué tendencias se pueden observar. Estos pronósticos, aunque fundamentados o razonados tienen, como todos los augurios, un amplio margen de error.

La ingeniería de programas (ingeniería de software), tal como la conocemos, desaparecerá pronto: herramientas CASE, medios ambientes integrados, reuso de código, métricas. Todos son términos que han prometido mucho y entregado poco, quizá por lo difícil de su cometido. En su lugar han aparecido y continuarán apareciendo otras herramientas que ayudarán considerablemente a la construcción, implantación y adecuación de aplicaciones: programación visual, tecnología de agentes, minería de datos, servicios genéricos de transporte y

recolección, por nombrar algunos. En transmisión de información, reinará la transferencia en modo asíncrono (ATM).

Tres factores determinan este avance: el abaratamiento de la capacidad de cómputo (que se observa sobre todo en la familia de computadoras 80X86 y Pentium), el del disco duro, y la proliferación de redes locales y distantes para conectar las computadoras de una empresa. Estas redes de máquinas personales (PC), servidores pequeños y medianos, y estaciones de trabajo, están reemplazando (por su costo más bajo) a las computadoras grandes (llamadas mainframes), que poco a poco se van extinguiendo y reduciendo a nichos específicos.

---

\*Director del Centro Nacional de Cálculo, Instituto Politécnico Nacional.

Este trabajo fue presentado durante la XX Reunión Nacional del Comité de Informática de la Administración Pública Estatal y Municipal.



## I. Productos actuales, tendencias y extrapolaciones

### 1.1 Evolución de los lenguajes

X= fecha de defunción probable; L= desarrollo limitado

	1980	1990	1994	HOY	1998	2000	2004
		X Ensamblador.====					
				Difícil de utilizar, improductivo, fue sustituido por lenguajes de segunda generación (Fortran, Cobol, Lisp). Subsiste de manera muy limitada; por ejemplo, para escribir manejadores de dispositivos.			
X Basic		Lenguaje demasiado básico. Poco poderío.					
X Pascal====L====X		Lenguaje hecho "para enseñar" por un académico, y no para la vida real.					
X RPG		Lenguaje de poco poderío, excepto para manufactura de reportes.					
X PL/1		Lenguaje voluminoso, mezcla de Fortran y Cobol. IBM intentó impulsarlo como estándar.					
X APL		Lenguaje inventado por un académico. Requería caracteres especiales. IBM intentó impulsarlo.					
X Ada		Lenguaje inventado por un usuario (el depto. de Defensa de EEUU). Rígido, carece de objetos.					
		====LISP====L		Desarrollo limitado a áreas específicas (Intelig. Artificial).			
		====Prolog====L		De uso en trabajos académicos y de investigación.			
		====Cobol====L====X		De uso limitado, este viejo lenguaje va a pasar al próximo milenio.			
		====Fortran y C====L					
		====Objetos (como los conocemos hoy)====X Ver §1.8		El problema con los objetos como los trae C++ es que están diseñados para programación mediante el teclado (de 3a. generación), y las tendencias son hacia la programación visual (usando ratón y menús).			
			====Eiffel====L====X	Buen lenguaje, poco soporte comercial.			
		====SQL (para acceso a bases de datos)==== Seguirá usándose. No se estandarizará, sino que habrá (ya hay) convertidores.		Será reemplazado por (a) ODBC; (b) acceso visual; (c) acceso "intuitivo" a bases de datos poco familiares (Cf. §1.14.5).			
		====Programación en equipo (de hoy)====X Se necesitan buenas herramientas para programar en equipo.		Las herramientas actuales serán sustituidas por otras nuevas, basadas quizá en programación visual.			
		=Generación de aplicaciones=Tendrán bastante auge los sistemas que permiten un ajuste a la medida o parametrización por el usuario final=					
		=Programación visual (Access, Power Builder, FoxPro, Visual Basic, Delphi,...) Ver					

## 1.2 Panorama global de la informática

		HOY					
1980	1990	1994		1998	2000		2004
X repositorio de		Un diccionario global de datos de toda la empresa es difícil de hacer y mantener.					
X Redes neurales====		X Redes neurales=====		El campo ya había muerto y volvió a revivir. Estos perceptrones tienen limitaciones bien conocidas y han tenido pocas aplicaciones prácticas. Están ahora de moda (otra vez).			
=====Archivos indexados, con llaves; acceso directo=====							
Para muchas aplicaciones donde se accede a un solo registro (por número de cuenta, número de boleta, número de cuenta de cheques), son superiores a las bases de datos.							
=====Bases de datos relacionales=====							
Con tecnología madura, la ventaja de una base de datos es que ofrece respuestas a casi cualquier pregunta formulada en SQL, un lenguaje propio para formularlas.							
Reuso=X Ver §1.9		Escribir código reutilizable es buena idea, el problema es cómo. Mientras no sepamos cómo, la "rama" o "ciencia" del Reuso no deja de ser el cascabel del gato.					
=Herramientas CASE (de hoy)=X Ver §1.9		Las herramientas CASE nacieron con muchas promesas pero pocos resultados. Además, tienen la desventaja de que se diseñaron para lenguajes de teclado. Los nuevos lenguajes visuales simplifican el hacer programas.					
==Sistemas (empresas) que miden el grado de madurez del proceso de manufactura de software=====X		Está muy bien medir qué tan efectivo es el "proceso" de fabricación de software. Empero, lo que se mide hoy es muy subjetivo y sólo representa una aproximación burda al proceso completo. Los métodos actuales que miden esta madurez tienen aspectos de "rollos" sofisticados. Deben verse con escepticismo. <i>Idem:</i> Norma ISO					
==Sistemas expertos==X		Se difundieron comercialmente, pero poseen dos desventajas fuertes: su base de conocimientos es lenta de construir, y después de un cierto número de reglas, empieza a haber interacciones indeseables entre ellas. Serán reemplazados por clasificadores supervisados, capaces de trabajar con datos numéricos y simbólicos.					
		==Reingeniería====X Ver §1.9		La Reingeniería de aplicaciones es temporal, y terminará cuando no haya más aplicaciones que convertir de máquinas grandes a máquinas chicas.			
=====Generación de aplicaciones=====							
Es decir, programas que permitan que usuarios comunes generen aplicaciones o hagan algo útil de modo general, "programando" algo, ajustado a la medida, cambiando parámetros.							
=====Computación distribuida, procesos débilmente acoplados=====							
Al estar baratas las máquinas personales, se usarán muchas de ellas, conectadas con canales "angostos" (acoplamiento débil), bajo el régimen de intercambio de mensajes. Ver §1.6							
=====Algoritmos genéticos=====X		Su aplicación se ve limitada por el "razonamiento" que se usa para crear nuevos algoritmos "mezclando" pedazos de algoritmos viejos pero "prometedores".					

		HOY					
1980	1990	1994		1998	2000		2004
		<p>==Tecnología cliente-servidor==X Ver §1.7</p>		<p>La arquitectura actual supone un cliente flaco y un servidor gordo: al engordar los clientes, se simplificará la aplicación, que volverá a ser (casi) monolítica otra vez.</p>			
		<p>====Agentes (procesos coordinados dispersos)==== Se generalizará la programación con "programas padres" que engendran agentes (programas hijos) que son sembrados, vía la red, en máquinas remotas, para trabajar para uno, bajo control nuestro. Son <i>nuestros</i> agentes. Ver §1.15.2.</p>					
		<p>===Bodega de datos (de hoy)===X</p>		<p>No tiene caso juntar <i>todos</i> los datos de una empresa en una bodega de datos. Será sustituida por el almacenamiento <i>selectivo</i> de resúmenes de información.</p>			
<p>=====Sistemas propietarios=====X</p>				<p>=====Sistemas abiertos===== La madurez de la industria del software implica que sólo haya software "abierto", estándar. Compare con la industria de tornillos</p>			
		<p>=====Servicios genéricos de transporte de datos===== Replicación, sincronización de archivos, sumariación y transmisión, serán servicios "normales" para redes. Ejemplo: Sumarizador/Transmisor de AnaSin. Ejemplo: Lotus Notes.</p>					
<p>=Correo electrónico= Seguirá fuerte su utilización como medio rápido de comunicar pocos bits.</p>							
<p>El manejo de información por papel, fax, copiadora, se reemplazará por manejo electrónico.</p>		<p>=====Flujo electrónico de documentos===== Será (dentro de diez años, digamos) reemplazando a su vez por manejo de información principalmente de máquina a máquina (en vez de tener como destinatario principal un ser humano).</p>					
<p>==Máquina de bases de datos=====X</p>		<p>Tener una máquina que ella sola sea capaz de muchas transacciones por segundo es factible, pero es mucho más económico y confiable utilizar redes de PC u otras máquinas pequeñas para ello.</p>					
		<p>Edificios inteligentes=L Dotar a los edificios de sensores, chapas electrónicas, etc., es bueno, pero los hace "inteligentes".</p>		<p>El desarrollo será limitado: se volverá algo rutinario instalar este tipo de electrónica de baja velocidad, sobre todo si los precios bajan y se implantan estándares (por ejemplo, para "cableado inalámbrico" vía el éter (radio).</p>			
<p>====Computadoras altamente paralelas====L (Supercomputadoras) tendrán un nicho limitado a cálculos muy extensos.</p>		<p>Por su precio, serán desplazadas casi totalmente por redes de computadoras pequeñas y rápidas enlazadas con fibra óptica.</p>					
		<p>===Realidad virtual (de hoy)===L</p>		<p>Pocas aplicaciones. Irán aumentando conforme el costo de computar tantas imágenes por segundo se justifique (o disminuya simplemente por avance tecnológico).</p>			

### 1.3 Sistemas operativos

		HOY					
1980	1990	1994		1998	2000		2004
Sistemas propietarios: VMS, Non-Stop,...=X							
		=====Unix=====X Continuará fuerte, pero será desplazado por un sistema operativo que ofrezca multiprogramación pero con sencillez y facilidad de aprender.					
		=====DOS=====X Debería morir antes, pero la gran cantidad de ejemplares instalados hará gradual su extinción. Principal problema: monoprogramación.					
		=====MacIntosh=====X Las computadoras Apple aún conservan nichos importantes de aplicación.					
		=====OS/2=====? Tendrá un número importante de seguidores, pero al fin, creo que no podrá consolidarse.					
		=====Variantes de Unix. Ejemplo; Sun Solaris=====					
		=====NextStep=====X Creo que no podrá consolidarse, por falta de seguidores.					

### 1.4 Bases de datos

Las bases de datos se dividen en "de escritorio": DBase, Clipper, FoxPro (llamadas colectivamente XBase); Paradox; y "servidores relacionales", que corren en plataformas más grandes: Informix, Unix, Progress, Sybase, Oracle. Un poco intermedia está SQL Base (servidor) + SQL Windows (para desarrollar clientes) de Gupta, y SQL Server de MicroSoft. Se generalizará el acceso vía ODBC a los servidores de bases de datos, para permitir desarrollar

aplicaciones desde Windows, utilizando herramientas de desarrollo visual (PowerBuilder, Visual Basic, Delphi,...). Para accesos desde Unix, habrá desarrolladores de la parte cliente que utilicen herramientas CASE adaptadas a las bases de datos relacionales, y capaces de generar, por ejemplo, las definiciones de las tablas y vistas a partir de diagramas entidad-relación-atributo.



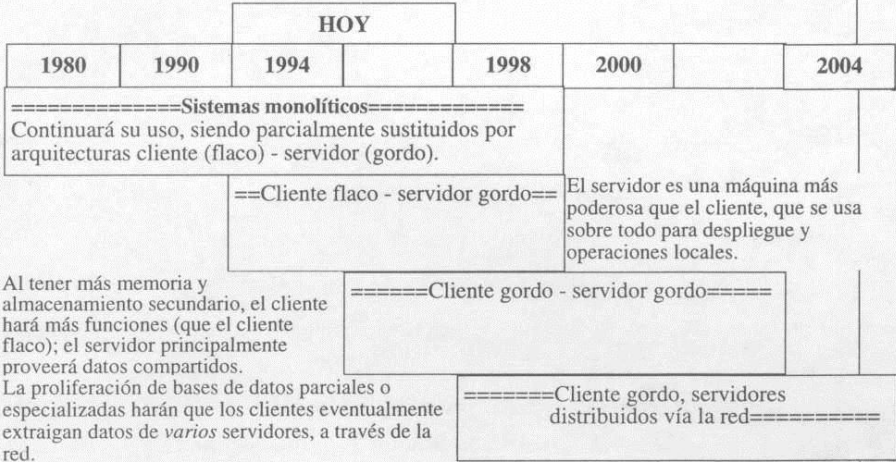
### 1.5 Ambientes visuales (ambientes gráficos)

		HOY					
1980	1990	1994		1998	2000		2004
=====MacIntosh=====							
=====Windows=====?							
Microsoft producirá variantes de Windows, pero probablemente otros ambientes visuales emerjan que ofrezcan ventajas en nichos especiales y, eventualmente, uno de ellos proliferará (siguiendo la misma historia de Unix, de incubación lenta).							
=====XWindows=====?							
====Presentation manager (OS/2)=====?							
IBM hará un esfuerzo para consolidar este producto.							
=====Windows NT=====?							

### 1.6 Redes locales

		HOY					
1980	1990	1994		1998	2000		2004
=====Redes locales (vía par trenzado, cable o fibra óptica)=====						Continuarán en amplio uso. Habrá (ya hay) redes interconectadas entre sí.	
El uso de Internet (como interconector público de redes) irá en aumento.							
(Inversión del uso del éter) habrá una pugna por arrebatárselo a las estaciones de televisión el uso del éter, para emplearlo para computación móvil, y relegar la transmisión de señal de TV a cable. La batalla será dura.						====Redes locales vía el éter====	
						Tal vez ganen las estaciones de radiodifusión. Pero la introducción de tecnologías híbridas (Frame Relay) e isócronas (ATM) podría unificar los dos frentes.	
=====Hipermedios=====L						Transmisión de voz, video, imágenes, texto (datos) y programas por un medio unificado. Crecimiento limitado, una vez que las principales aplicaciones se encuentren emplazadas.	
==Redes locales tipo Ethernet y Token Ring=						Este popular modo de enlazar PC será gradualmente desplazado por conmutadores centrales más rápidos.	
Redes Ethernet con un switch (conmutador) central. Red local virtual. Proporciona más capacidad de comunicación (Mbits/seg) que una red local.							
Proporciona más velocidad y ventajas que las redes tipo Ethernet.						=====Frame Relay=====	
Proveen un servicio unificado de transmisión de voz, video, datos, etc. Garantizan la entrega de información en un tiempo límite (característica necesaria para transmitir video en tiempo real, por ejemplo).						==Redes ATM (transferencia en modo asíncrono)== Ver §1.11. Todos están de acuerdo en que ATM es "lo mejor"; su implantación será lenta, hasta que los proveedores principales hagan las inversiones	

1.7 Tecnología cliente-servidor



1.8 Objetos

Los objetos nacieron con Small Talk, y se propagaron con Objective C y con C++ (el más popular).  
 Otros lenguajes de objetos: Eiffel, Linda. El problema actual de los lenguajes de programación con objetos es que son de teclado, o sea se programan con caracteres, por lo que el programador no es productivo. Por tal motivo, creo que los lenguajes actuales de objetos cederán paso a otros más visuales, que no requieran que el programador sepa dónde poner

el punto y coma, o los tipos de argumentos que requiere tal o cual función (una función se llama método cuando se trabaja con objetos. Pudieran ser Visual C++.

1.9 Evolución de la Ingeniería de Programas. Herramientas CASE

		HOY					
1980	1990	1994		1998	2000		2004
	====Herramientas CASE====X (Herramientas de ingeniería de programas, o de ingeniería de software)						Las herramientas CASE <i>tal como se conocen hoy</i> , se enfocan a programación con el teclado, y no a programación visual. Surgirá un nuevo conjunto de herramientas, fabricadas por los mismos fabricantes de lenguajes visuales.
	===Métricas====X						Importante medir el grado de complejidad, el porcentaje de errores, etc., de un programa. Pero, ¿cómo?, las soluciones actuales son empíricas y subjetivas.
		Medición del <b>grado de madurez de un proceso</b> de generación de software ==X					Importante medirlo. Pero, ¿cómo?, las soluciones actuales son empíricas y subjetivas.
	=Reuso== ==X						Bueno es reutilizar software, pero ¿cómo construir software reusable? ¿Cómo buscar el software que voy a reutilizar, en una gran biblioteca? ¿Cómo adaptarlo al nuevo uso? La "ciencia" del reuso ofrece ahora más preguntas que respuestas.
	====Traducción de especificaciones a código====X						La traducción directa de especificaciones a código enfrenta graves problemas, pues el diseño o solución a un problema requiere experiencia, conocimientos, cosas que los traductores (compiladores, transformadores) no poseen en general. En estos momentos resulta ingenuo suponer que, de manera mecánica, podemos pasar de una especificación (descripción del problema a resolver) a una solución (código que lo resuelve). Con ayuda de una base de conocimientos, aún es campo de investigación.
		===Validación de especificaciones haciendo rápidamente prototipos===?					La técnica tiene validez y uso.
			==Refinamiento progresivo de los				Tiene validez y uso.
	====Verificación de la completéz de especificaciones====X						No es posible, en general, por métodos mecánicos saber si a una especificación le falta algo o ya está completa. Se requeriría leer nuestra mente, o saber de antemano que el problema a resolver cae dentro de tal o cual taxonomía o apartado.
		====Ingeniería a la inversa, Reingeniería (de software)====X					Aunque es un campo válido, se acabará cuando ya no haya aplicaciones que convertir a redes locales, ya sea porque ya fueron convertidas, o porque se abandonaron las no convertibles.
	Verificar que una especificación esté correcta==X						No es posible, en general (es un problema recursivamente indisoluble), saber si una especificación es inconsistente. En casos sencillos la inconsistencia puede determinarse por métodos sintácticos. Poca ayuda ofrecen los verificadores actuales.
	=Programación usando diseño estructurado (métodos de Yourdon, Booch, Warnier, etc.)=====L o X						Estos métodos, enfocados a lenguajes de teclado, irán desapareciendo conforme la popularidad de los lenguajes visuales crezca.
		====Herramientas para hacer pruebas repetitivas=====L					Seguirá existiendo la necesidad de probar nuestros desarrollos. Cuando la interacción es principalmente con el ratón y el menú, estas herramientas ahorran incontables horas de prueba manual.



1980	1990	1994	1998	2000	2004
<p>=====<b>Diagramas entidad-relación-atributo</b>=====X                      Válidos ahora, irán dando paso a métodos más gráficos de representar relaciones, y a otros basados en extracción de definiciones (de campos y atributos) de tablas similares a las que se desea construir.</p>					
<p>=====<b>Generación automática de diagramas ERA</b>=====X                      Útiles ahora, correrán la misma suerte de los diagramas entidad-relación-atributo.</p>					
<p>=====<b>Programación visual, sin uso extenso del teclado</b>=====                      Tendrá gran aceptación, ya que aumenta la productividad del programador, imponiendo sobre el editor gráfico la responsabilidad de generar lenguaje (de teclado) sintácticamente correcto.</p>					
<p>=====<b>Diseño automatizado de interfaces con el usuario</b> (captura, etc.)====                      Tendrán gran aceptación, ya que aumentarán la productividad del programador, utilizando unos cuantos casos "típicos" o machotes para generar interfaces, formas de captura, reportes, etc.</p>					
<p>Encontrarán utilidad (como la hoja de cálculo) en campos limitados pero importantes.</p>					
<p>=====<b>Programación con íconos y metáforas</b>====</p>					
<p>=====<b>Uso de agentes que colaboran entre sí</b>====                      Tendrá sentido utilizar una gran cantidad de programas distribuidos para realizar un cómputo o proceso. §1.15.2</p>					

1.10 Evolución de los sistemas de programas (es decir, de las aplicaciones)

		HOY				
1980	1990	1994	1998	2000	2004	
===== <b>Sistemas propietarios</b> ====X Sus diferentes partes están hechas por el mismo fabricante y son incompatibles con las de otro fabricante.		Tienen su apogeo al comienzo de la computación, en su fase de inmadurez. Poco a poco se consolidan estándares para permitir que cualquier martillo le pegue a cualquier clavo, o cualquier rosca le venga a cualquier tornillo, independientemente del fabricante.				
== <b>Sistemas abiertos</b> =Poseen partes intercambiables con las de otro proveedor==						
===== <b>Arquitecturas monolíticas</b> =====L Programa donde todo corre en una sola máquina.		No serán 100% reemplazadas por arquitecturas cliente-servidor. Ver §1.7				
===== <b>Arquitectura cliente-servidor</b> =====L Programa dividido en dos partes, el lado cliente donde la máquina interacciona con el usuario, con el ratón, etc., y el lado servidor que proporciona, sobre todo, datos.		Proliferará, pero disminuirá conforme el cliente se vuelve "gordo" y lleno de recursos (memoria secundaria, velocidad de acceso a través de la red, etc.) Ver §1.7				
===== <b>Servicios genéricos de transporte</b> ===== La red pasará a tomar una parte activa en las aplicaciones, haciendo resúmenes, sincronizando archivos, ofreciendo respaldos, etc.						
===== <b>Acceso homogéneo a bases de datos heterogéneas</b> , localizadas en toda la empresa== Aun cuando una empresa tenga varias marcas de bases de datos, hay productos (como Sybase) que funcionan como puentes y permiten accesos heterogéneos.						
===== <b>Bases de datos distribuidas</b> =====? Esta tecnología parece no ofrecer limitaciones, pero requiere de una conectividad alta, de 24 horas.						

### 1.11 Redes ATM

La progresión de redes locales es, primero, hacia una red local virtual, donde en vez de red se tiene un poderoso (rápido) conmutador electrónico que hace la intercomunicación entre las máquinas de la red local. Luego, vienen servicios más rápidos y generales, como los ofrecidos por la estructura de red de Frame Relay. Finalmente, las redes ATM (transferencia en modo asíncrono) ofrecen entregas de varios tipos: entrega isócrona (se garantiza que los paquetes lleguen a su destino antes de un cierto tiempo x); entrega a velocidad constante,

entrega a velocidad variable, entrega a la capacidad disponible (o sea, servicios donde se utiliza la capacidad cuando ésta existe). El resultado es que una misma red puede proporcionar servicios de transmisión de voz (telefonía), video (televisión, movimiento), imágenes, datos, y servicios de cómputo.

Aunque las empresas proveedoras están convencidas de que las redes ATM son "lo mejor" del momento (y por buen tiempo, quizá) sólo de manera paulatina se irán ofreciendo los equipos para trabajar bajo ATM. Esto es por lo oneroso de la inversión inicial.

### 1.12 Evolución de las computadoras

		HOY					
1980	1990	1994		1998	2000		2004
=====Minis=====							
====Máquinas grandes (mainframes)====X				La relación costo/beneficio y la distribución geográfica de la capacidad de cómputo y del almacenamiento de la información, claramente favorecen a las máquinas pequeñas conectadas en redes locales y semilocales.			
=====Servidores=====?							
Serán ampliamente usados. Su uso decaerá conforme se abarate el costo de almacenamiento local y/o el costo de transmisión de información (o sea, cuando los clientes sean "gordos" y puedan ir por mucha información a un repositorio central).							
====Computadoras personales y, más específicamente, Pc's (80x86, P5, P6,...)====							
=====Estaciones de trabajo=====							
Serán ampliamente usadas, confundiéndose gradualmente con Pc's robustas.							
Máquinas para bases de datos====X		Fueron enterradas por las redes locales, que ofrecen la misma cantidad de transacciones por segundo a un costo menor.					
====Computadoras altamente paralelas====L				Estas supercomputadoras, de muchos MIPS y acoplamiento fuerte, serán reemplazadas en su mayoría por redes con fibra óptica de computadoras RISC tipo SUN o IBM RS 6000, o por computadoras SIMD tipo Connection			
=====Procesamiento distribuido (redes), débilmente acoplados=====							

### 1.13. Situación de la Inteligencia Artificial en 1995

#### 1.13.1. Campos con pocos resultados:

Redes neurales X  
Algoritmos genéticos X

#### 1.13.2. Campos con poco progreso

Reconocimiento de caracteres  
Aprendizaje  
    Juegos (no me refiero a los juegos de Nintendo, etc.)  
Traducción de lenguaje natural

#### 1.13.3. Campos con progreso limitado

Sistemas expertos L

Lo que ocurre es que su utilidad se ve limitada cuando el número de reglas rebasa un umbral (digamos 300 reglas) y empieza a haber interacciones indeseables entre ellas.

Visión

Después de un periodo relativamente fértil (máquinas para procesamiento de imágenes; dilaciones, erosiones; foco de atención; amalgama de sensores de varias modalidades), entra la Visión por Computadoras a un periodo caracterizado por (a) ir directamente a modelar en tres dimensiones, utilizando pares estereoscópicos o sensores de profundidad; (b) utilizar una base de imágenes (3-d) grande, y (c) técnicas de indexación para rápidamente reconocer qué objetos (de la base de datos) están presentes en la imagen tridimensional analizada.

Reconocimiento de patrones

El reconocimiento de patrones sobre objetos que tienen rasgos con valores numéricos (edad; sueldo) está relativamente estático. Tiene más dinamismo la clasificación (supervisada) utilizando valores no numéricos (profesión, partido político, colonia donde vive) y la teoría de Testores.

Minería de datos (campo promisorio)

Es la búsqueda sistemática (y exhaustiva, generalmente) de condiciones "interesantes", anomalías, desviaciones y tendencias, en un mar de datos. El usuario da una idea general (o modifica las "ideas" iniciales) de lo que es para él interesante, y los mineros hacen el resto: aplican el predicado y escudriñan la

base de datos (numérica) buscando lugares (ventas, productos, vendedores,...) que cumplen con lo señalado. Desde el punto de vista técnico, un minero es un "ajustador de curvas" glorificado, que aprovecha la existencia de muchas señales correlacionadas para aumentar la relación señal/ruido y así descubrir tendencias o desviaciones ligeras, cuando es fácil tomar acción o corrección.

#### 1.13.4 Campos promisorios

Sistemas basados en el conocimiento, basados en el sentido común.

Una gran cantidad de conceptos propiamente ligados (zapato es el papá o superconjunto de huarache, sandalia, mocasín, bota, alpargata. Austin, San Antonio, Dallas, Houston son hijos de Texas. Un watt es un voltio por un amperio) harán que el manejo de datos no numéricos tenga cierta "inteligencia" o "iniciativa". Por ejemplo, si guardo cierto dato en "San Antonio" y después lo busco en "Texas", será factible encontrarlo. Se define un árbol del conocimiento común, donde los nodos son conceptos y las ligas o lados de la red son relaciones, generalmente subconjunto, parte de, contenido en, etc.

Bases de datos deductivas

Puedo pedir los datos en dólares, y si están en nuevos pesos, la base de datos hace una conversión. Puedo pedir los resultados de Jalisco, y la base de datos se da cuenta que no los tiene, pero que sí tiene los de Guadalajara, Chapala, Ciudad Guzmán, Atotonilco, por lo que procede a hacer sumas para proporcionarme lo pedido.

Acceso a bases de datos poco familiares

Una empresa distribuida (que lleva a cabo operaciones en varias localidades) tiene por lo general varias bases de datos, con cierto grado de especialización. Es común, entonces que los usuarios que no están familiarizados con una cierta base de datos, tengan dificultad en explotarla, en entender sus valores, sus unidades, lo que representa cada campo, cada renglón, aun cuando puedan accederla físicamente (vía una red y remote login, por ejemplo). Utilizando el árbol del sentido común, es posible "entender" una base exógena (poco familiar), mediante un mapeo de los conceptos específicos de la base, a conceptos más familiares precisamente, del sentido común.

Sistemas evolutivos

Utilizando perceptrones (lógica sumatoria y de umbral), es posible en muchos casos obtener versiones simplificadas pero satisfactorias de algunos de los problemas que se resuelven con mucho conocimiento.

#### Visión en tiempo real

El procesamiento de imágenes (en movimiento) en tiempo real. Utilidad: para reaccionar a un medio ambiente cambiante. Por ejemplo, para que un robot pueda conducir a un vehículo.

Visión integrada (integración de diversos sensores). Ver §1.13.3.

### 1.14. Herramientas emergentes promisorias

#### 1.14.1 Programación visual

##### Editor que conoce la sintaxis

La programación visual (PowerBuilder, Access, FoxPro, Visual Basic, SQL Windows, Delphi) deja en el editor la responsabilidad de la sintaxis del programa que se está construyendo. Quiere decir, por ejemplo, que si tengo una variable ALFA de tipo texto o carácter, las operaciones posibles que me ofrece el editor son UNION, CONCATENACION, SUBCADENA, y otras que tienen sentido para variables de texto, y no me ofrece SUMA, MULTIPLICACION, RAIZ CUADRADA, porque no tienen sentido para ALFA. Esto hace que de súbito la cantidad de información preliminar que el programador debe aprender (del manual) sobre el lenguaje sea mucho menor, y que pueda ser productivo más rápidamente. Como además el editor trabaja con elementos gráficos como menús, selecciones, acciones del ratón, la productividad es grande.

Otro ejemplo: en vez de pedirme que teclee yo una variable o un campo, me ofrece una lista de los que ya hay. Esto elimina los errores sintácticos (variable indefinida) aunque los errores semánticos (me equivoqué y seleccioné ingreso neto cuando debería haber escogido ingreso bruto) aún son responsabilidad del programador.

##### Generación de código distribuido

#### 1.14.2 Tecnología de agentes

Conforme aumenta el número de procesadores a mi disposición (debido a su reducción en costo), es posible ahora definir algoritmos (programas, aplicaciones, procesos) que se efectúen en varias máquinas a la vez. Una forma un tanto sencilla de programación distribuida es el uso de agentes, que son procesos (programas) que se instalan en máquinas remotas, y que poseen un modo de sincronización o cooperación para efectuar un trabajo común. Por ejemplo, para hallar cuál de mis almacenes tiene más de 100 cajas de jabón Zote, envío (o coloco) agentes en cada (computadora de) almacén, y les pido

que aquellos que detectan más de 100 cajas, me envíen su identificación (o el número de cajas de jabón). Un programa central puede entonces simplemente listar, sumar, o de otra forma manipular la información proporcionada por los agentes.

*1.14.3 Redes y servicios de transmisión ATM (transferencia en modo asíncrono)*  
Ver §1.11.

*1.14.4 Servicio genérico de transporte*

Desvanecimiento de la frontera entre el almacenamiento y la transmisión de la información.

Conforme el ancho de banda aumenta (redes más rápidas), ya no distingue el usuario si la información es local (proviene de un archivo local) o remota (proviene de un archivo remoto vía una red rápida) o de un cálculo "al vuelo" (se está computando o calculando la información solicitada, por ejemplo, haciendo sumas parciales con algunos sumandos provenientes tal vez de archivos remotos).

Se pueden tener servicios de transporte que simulen tener gran ancho de banda, pero que en realidad trabajan con redes modestas (líneas telefónicas, por ejemplo). Supongamos que desde el edificio central del Autotransporte Urbano deseo consultar en tiempo real las licencias de conducir (que en cada delegación se expiden). En vez de tener una red de gran ancho de banda para acceso en tiempo real, yo puedo cada noche vaciar las licencias expedidas durante el día en un repositorio central, para su consulta. Entonces puedo consultar este repositorio en vez de enlazarme en tiempo real a la Delegación Contreras. Sólo en caso de que se trate de una licencia de Contreras expedida hoy, necesito el enlace en tiempo real. Es claro que esto disminuye en mucho mis necesidades de comunicación, a expensas de replicar la información. Es la técnica de guardar y retransmitir («store and forward»).

Ejemplo de herramienta: Sumarizador/Transmisor de AnaSin.

Ejemplo de herramienta: Lotus Notes

*1.14.5. Acceso a bases de datos poco familiares.* Ver §1.13.4.

*1.14.6 Minería de datos.* Ver §1.13.3

*1.14.7. Redes de sistemas ejecutivos*

Las herramientas como el Sumarizador Transmisor (Cf. §1.2), Lotus Notes y otros sistemas genéricos de transporte (Cf. §1.14.4).

### **1.15. Algunos problemas en los que la informática incursionará**

#### *1.15.1. Manejo de texto*

Identificar los temas a los que un artículo pertenece.

Recuperación inteligente de documentos. Por conceptos que contiene, no por palabras clave. Clasificación de documentos utilizando el árbol del sentido común, de suerte que si lo guardé en "Austin", y después lo busco en "Texas", lo encontraré. Si lo clasifico en "DDT" y lo busco después en "pesticidas", lo hallaré.

Resúmenes por computadora. Extracción de las ideas principales de un documento.

Entender un escrito: \*Hacer un resumen. \*Contestar preguntas no triviales sobre su contenido. Hallar tendencias o cambios comparando dos o más documentos.

#### *1.15.2. Agentes*

Generación y distribución automática de código que colabora y se relocaliza remotamente. Ver §1.14.2 y §II.

## II. Tendencias en la organización de la información

### 2.1 Evolución de almacenamientos de información

		HOY				
1980	1990	1994		1998	2000	2004
<p><b>=Archivos planos, archivos indexados, llaves, llaves alternas, llaves duplicadas==</b>                      Estas técnicas, aunque viejas, continuarán siendo útiles para sistemas operacionales, donde las operaciones o transacciones son sobre un cliente, estudiante, cuenta, etc., identificado por un número.</p>						
<p><b>Bases de datos jerárquicas y de redes=====X</b></p>		<p>Han sido sustituidas por bases de datos que siguen el modelo relacional, el que proporciona ventajas teóricas y prácticas (operaciones útiles bien definidas y generales).</p>				
<p><b>Abaratamiento de las CPUs (80X86)=====</b></p>		<p>La disminución en costo es responsable de que se utilicen más computadoras, las que son sincronizadas ahora vía una red y después vía agentes (procesos asíncronos dispersos coordinados).</p>				
<p><b>Abaratamiento del disco duro de cabeza móvii (Winchester)=====</b></p>		<p>Permite almacenar datos "puntuales" (de cada cliente, de cada compra) y efímeros. También favorece el almacenamiento <i>local</i> (los datos se guardan donde se usan).</p>				
<p><b>Proliferación de redes locales=====</b>                      Ver en §1.6 una gráfica más completa</p>		<p>Motivada por el abaratamiento de la capacidad de proceso (y después, por el de la comunicación).</p>				
<p><b>=====Bases de datos relacionales. Ver §1.4=====</b>                      Continuarán en amplio uso. Habrá (ya hay) replicación, sincronización, replicación diferida, multibases, bases de datos confederadas.</p>						
<p><b>=====Bases de objetos=====L</b>                      Bases de datos donde se pueden guardar otros objetos: imágenes, sonido, video.</p>		<p>Su uso estará eventualmente limitado por la complejidad de los métodos que se requieran para manejar objetos "peculiares".</p>				
<p><b>=====Bases de datos distribuidas=====?</b></p>						
<p><b>=====Bodega de datos=====X</b></p>		<p>Replicar los datos sin hacer resúmenes, o con sólo tener duplicados, puede no ser tan buena idea.</p>				
<p><b>=====Sumarizador/transmisor=====</b>                      =                      Este tipo de software que ofrece servicios genéricos de "replicación selectiva" (por ejemplo, hace resúmenes y los envía a otras</p>						

