

Desarrollando y manteniendo grandes aplicaciones corporativas con *GENEXUS*.

Una experiencia de seis años y más de 1300 clientes.

Copyright © 1996 Breogán Gonda, Juan Nicolás Jodal, Karina Santo,
todos los derechos reservados.

¿Qué tipo de aplicaciones son desarrolladas y mantenidas con *GENEXUS*?. Esta es una pregunta que nos es formulada con mucha frecuencia. En rigor, detrás de esta pregunta, está una duda hasta cierto punto razonable: los que recién conocen *GENEXUS* se entusiasman con sus características únicas y, al mismo tiempo, dudan de ellas. Saben que los viejos métodos no resuelven sus problemas pero, paralelamente, tienden a ver a *GENEXUS* como algo mágico, y piensan que algo mágico no puede resolver grandes problemas.

Obviamente, *GENEXUS* no tiene nada de mágico: es producto de la tecnología más avanzada que existe actualmente, y no hay nada más lejano a la magia que la tecnología.

El proceso de investigación y desarrollo que condujo a *GENEXUS*, fue motivado por nuestra percepción de los problemas que, hace algo más de diez años, enfrentábamos todos aquellos que tratábamos de implementar sistemas sobre bases de datos corporativas. Otra verdad es que, en general, y más allá de las herramientas que diferentes fabricantes han lanzado al mercado en los últimos años, dichos problemas subsisten, y las herramientas basadas en métodos convencionales han traído muchas ayudas para documentar el modelo de datos, pero poco o nada han avanzado en la resolución de los verdaderos problemas.

El problema fundamental no está constituido por las herramientas, sino por los viejos métodos que aún son bastante utilizados. Las herramientas nos ayudan a documentar, pero seguimos haciendo manualmente cosas tediosas y rutinarias que, como la normalización o la generación inteligente (no basada en esqueletos) de código podrían, y deberían, hacerse automáticamente..

En realidad las herramientas generalmente utilizadas realizan (o nos ayudan a hacerlo) las tareas de la misma manera, y con los mismos métodos, que hubiéramos empleado manualmente. La mayoría de los fabricantes no ha aprovechado que, al disponerse de enormes capacidades de almacenamiento y procesamiento automático, se pueden hacer cosas mucho más interesantes.

Es necesario asumir un cambio de paradigma, y esto siempre es dificultoso para los humanos.

Veamos los problemas de los viejos métodos, y como solucionarlos.

Existe un problema teórico básico, que es común a todas las metodologías tradicionales, y que está detrás de casi todas las dificultades: Se asume como premisa fundamental **“podemos obtener un modelo estable de nuestra organización”**. Si esta premisa fuera cierta, podríamos describir nuestros procesos en función de dicho modelo.

Lo que se hace en las metodologías tradicionales es, mediante un proceso de Análisis de Datos obtener el Modelo Corporativo de la organización (generalmente expresado en términos del modelo Entity-Relationship; E-R, de Peter Chen) y, luego, mediante múltiples procesos de Análisis Funcional obtener las descripciones de los procesos.

Si la premisa enunciada fuera cierta, esta manera de proceder sería adecuada. Lamentablemente, es falsa: **no existe un modelo estable para una determinada organización**. El modelo, por bueno que sea, evolucionará constantemente por detección de errores, y por modificaciones en la vida misma de la empresa. Esto fue siempre cierto pero hoy, ante la necesidad cada vez más crítica que tienen todas las empresas de ser competitivas y de, en buena parte por ello, adicionar permanentemente nuevas tecnologías en su actividad fin, es aún más importante.

¿Cuales son las consecuencias de este problema?.

En el desarrollo de aplicaciones pequeñas, no se deben esperar grandes problemas.

Cuando las aplicaciones crecen todo es bastante diferente: la inexistencia de modelos de datos estables hace que los costos aumenten en forma exponencial respecto al tamaño de la aplicación, e inciden en la baja calidad de dichas aplicaciones, (y aún de los fracasos totales que a veces obligan a abandonar proyectos a mitad de camino, luego de incurrir en grandes costos).

En el mantenimiento el problema es permanente. En EE UU, por ejemplo, las estadísticas dicen que de los recursos teóricamente destinados al desarrollo, sólo el 20% se utilizan en esa función, mientras que el 80% son utilizados realmente para mantenimiento. Obviamente nada autoriza a pensar que en los países donde no existen estadísticas la situación sea mejor.

Cuando en este trabajo nos referimos a “costo” lo estamos haciendo al complejo de recursos utilizados (dinero, tiempo, etc.). Todas sus componentes son importantes pero, muy frecuentemente, la componente tiempo es la preponderante: ¿cuál es el costo de no disponer oportunamente de la información necesaria para la toma de decisiones importantes?.

Si hilamos más fino, podemos ver que parte importante del problema de los altos costos de los grandes proyectos está dado por el “mantenimiento no explícito” que existe dentro de esos proyectos durante el propio proceso de desarrollo: las especificaciones fatalmente van cambiando (o se demuestran erróneas), y se comienzan a pagar costos de mantenimiento importantes antes de poner el proyecto en funcionamiento.

¿Como resolver este problema?. Esta es la pregunta básica que nos planteábamos en 1984, ante un megaproyecto en una gran empresa industrial brasileña, que nos llevó a una base de datos corporativa de gran porte, con las dificultades consiguientes. Para responderla hemos hecho una gran inversión en investigación y desarrollo. Como fruto de esa tarea se obtuvo la tecnología única en la que se basa **GENEXUS**.

En rigor lo que se hizo fue transformar el problema de desarrollo y mantenimiento de sistemas en un problema matemático. Una vez lograda una sólida fundación teórica, existen herramientas calificadas de la matemática que pueden ayudar. Obviamente, todo esto debe quedar encapsulado en **GENEXUS**, siendo totalmente transparente al Analista y, fundamentalmente, al Usuario.

Parece claro que resolver un problema pequeño es mucho más fácil que resolver uno grande.

Algo que podemos ensayar es tratar de olvidarnos del modelo corporativo, y del resto de la organización y, en vez de ello, concentrarnos en una aplicación o, aún, si esta aplicación superara cierto tamaño, en una parte de ella. Probablemente el lector entienda que esta es una hipótesis simplificadora demasiado fuerte. Permítasenos utilizarla, por lo menos transitoriamente.

En estas condiciones (sistema pequeño), podemos suponer que con cualquier software y cualquier metodología que utilizemos, tenemos una buena probabilidad de implementar en tiempo breve un sistema de buena calidad.

Si utilizamos un lenguaje de tercera generación, la productividad del desarrollo inicial será muy baja pero, lo que es mucho más importante, cualquier cambio deberá hacerse en el bajo nivel, y los costos de hacer un desarrollo incremental y, de esa manera, llegar a aplicaciones corporativas, hacen inviable este abordaje.

Si utilizamos un buen lenguaje de 4a. generación, en la etapa de desarrollo tendremos importantes disminuciones de costo sobre el caso anterior. Sin embargo cuando, luego, queremos seguir desarrollando en forma incremental, los costos serán muy importantes. ¿Por qué? Porque estos lenguajes, más allá del alto nivel de muchas de sus características, se refieren en sus especificaciones a tablas de la base de datos. Cuando la composición de esas tablas se modifique sustancialmente (por ejemplo, cuando un atributo de una tabla pasa a otra, cuando se introducen o eliminan redundancias, etc.), esas especificaciones se tornarán inválidas, y deberán ser modificadas manualmente.

En los últimos años se ha progresado mucho en dos sentidos: la aparición de los generadores automáticos de código, y de los lenguajes orientados a objetos. Ambos tipos de herramientas permiten una elevada productividad en el desarrollo, pero su independencia de datos está limitada a la que puedan ofrecer los sistemas de gerencia de base de datos, que es muy baja en todos los casos, por lo que el problema de mantenimiento enunciado para los lenguajes de 4a. generación, se reproduce aquí.

Alguien podrá pensar que esto es fatalmente así: que el problema no ha sido resuelto porque no tiene solución. Nada más lejos de la verdad. Para demostrar esta aseveración, nada mejor que mostrar un ejemplo que resuelve el problema.

¿Como haríamos con **GENEXUS**?

Mediante **GENEXUS** describimos las diferentes visiones de los usuarios. Existen los siguientes objetos **GENEXUS** para describir visiones:

Transacciones: visiones que tienen asociado un diálogo, y que pueden modificar el contenido de la base de datos;

Reports: visiones que no tienen asociado un diálogo, que permiten extraer información de la base de datos, y que no pueden modificar el contenido de la misma;

Procedures idem que Reports, pero que pueden modificar el contenido de la base de datos;

Work Panels: consultas interactivas que permiten al usuario obtener información en forma dinámica, orientando la búsqueda en tiempo de ejecución. Indirectamente, llamando Procedures adecuados, pueden determinar cualquier modificación en la base de datos; y

Web Panels: idem que Work Panels, pero aquí el diálogo es asíncrono, vía Internet o Intranet. Permiten crear páginas Web dinámicas con las que se implementan los diálogos necesarios y, llamando procedures, modificaciones en la base de datos.

La experiencia ha demostrado que estos objetos son suficientes para describir cualquier aplicación, sin necesidad de codificación manual alguna.

GENEXUS captura el conocimiento existente en estas visiones, y lo sistematiza en una Base de Conocimiento activa que, permanentemente, representa el problema que se está resolviendo.

En cualquier momento, **GENEXUS** es capaz de:

Diseñar la base de datos mínima, en tercera forma normal, capaz de satisfacer a todas las visiones del problema.

Permitir al Analista definir redundancias (generalmente por razones de performance), en la base de datos. Luego de ello, **GENEXUS** se encarga del mantenimiento automático de dichas redundancias.

Generar la base de datos y todos los programas necesarios para recrear las visiones introducidas (todos los programas necesarios para el pleno funcionamiento de la aplicación).

Ayudar al analista a definir, en cualquier momento, en forma muy simple, y sin necesidad de conocimiento de la estructura de la base de datos, nuevos Reports, Procedures, Work Panels y Web Panels.

En caso de modificaciones en las visiones, determinar el impacto de dichas modificaciones sobre datos y procesos. Una vez que dicho impacto haya sido aceptado por el Analista, (y/o el Administrador de Datos y/o Bases de Datos, en su caso), propagarlo automáticamente sobre datos y procesos:

- Modificar automáticamente la base de datos (estructura y contenido), y
- Escoger automáticamente que programas deberán ser generados o regenerados:

los que corresponden a objetos nuevos, o

a objetos existentes cuya descripción se haya modificado, o bien

a objetos existentes cuya descripción no se haya modificado, pero cuyos programas asociados no funcionarían más por los cambios que ha sufrido la base de datos y, aún, aquellos que no se han modificado, y cuyos programas asociados funcionarían perfectamente, pero serían menos eficientes que otros que, ahora, podríamos generar

y generarlos automáticamente.

¿Que se habrá logrado de esta manera (con **GENEXUS**), en el pequeño proyecto, de que estábamos hablando?

A menos que el proyecto fuera extremadamente pequeño, cierto aumento de productividad.

A medida que los proyectos sean mayores, podremos abordarlos con equipos menos numerosos que con métodos y herramientas tradicionales y, en estas condiciones, los aumentos de productividad serán cada vez mayores.

Calidad estándar garantizada: no existen “remiendos” en ningún caso. La estructura de la base de datos, y las de los programas, son siempre las óptimas.

Supongamos, por un momento, que tenemos más de una aplicación desarrolladas por separado con **GENEXUS**. ¿Qué necesitamos para poder integrarlas?

La integración es inmediata si las aplicaciones a integrar están inscriptas en un marco de referencia común. En **GENEXUS** los nombres de los diferentes elementos y, en particular, los de los atributos (columnas de las tablas) tienen especial importancia, porque en ellos reposa la semántica de las aplicaciones.

GENEXUS implementa con precisión su concepto de dominio: dado un determinado elemento, nos referiremos a él siempre por el mismo nombre; dos elementos diferentes deberán tener nombres diferentes; si un determinado atributo desempeña diferentes roles y, en particular, puede aparecer más de una instancia del mismo en otro objeto, debemos registrar este hecho por la vía de señalar los diferentes roles con diferentes nombres, indicando que estos son sub-tipos del original.

Debemos utilizar una nomenclatura que respete estas reglas. En particular, se sugiere utilizar el estándar GIK (**GENEXUS** Incremental KnowledgeBase).

Si tenemos dos bases de conocimiento de aplicaciones diferentes, con un marco de referencia común, su integración es automática. Existe un componente único de **GENEXUS** especialmente sofisticado (el Knowledge Manager) que permite estudiar el impacto de la integración e informar de él al Analista, (y/o al Administrador de Datos y/o Bases de Datos, en su caso). Una vez aceptado dicho impacto, el Knowledge Manager consolida automáticamente el conocimiento de ambas aplicaciones, y propaga los cambios necesarios.

Si las bases de datos no tienen un marco de referencia común como, por ejemplo, cuando se compra conocimiento descrito con una nomenclatura diferente a la que hemos adoptado (en otro idioma, etc.), debe hacerse la transformación de los nombres con antelación a la consolidación (**GENEXUS** ofrece facilidades para ello).

La forma normal de trabajar requerirá un estudio previo para determinar los principales objetos de interés corporativo, y establecer con la mayor precisión posible el alcance de cada proyecto.

En el desarrollo se tendrá la precaución de que, cuando un objeto es de interés para más de una aplicación, colocarlo en un modelo separado, asociado a esas aplicaciones, que será el único lugar donde luego se lo mantendrá, exportándolo luego a todos los modelos de las aplicaciones donde sea necesario.

Este procedimiento facilitará mucho la consolidación definitiva.

Una duda adicional que probablemente tenga el lector es: muy bien, esto funciona perfectamente cuando se parte de cero. ¿Qué ocurre cuando las aplicaciones que se desarrollan y administran con **GENEXUS** deben convivir con una base de datos preexistente? (cosa muy habitual en la práctica). Para resolver este problema, **GENEXUS** dispone de una componente de Ingeniería Inversa que permite, considerando la base de datos existente, inferir la base de conocimiento **GENEXUS** correspondiente. A partir de ahí, tanto las partes de la base de datos que sean administradas totalmente por **GENEXUS** como las preexistentes, serán tratadas de la misma manera para la generación y mantenimiento automático de los programas.

Esta situación es muy habitual: casi siempre las aplicaciones **GENEXUS** deben convivir por un tiempo con las viejas aplicaciones, que serán sustituidas en cierto tiempo pero, otras veces, la convivencia debe ser permanente (por ejemplo cuando el Cliente ha comprado paquetes especializados para resolver problemas puntuales). Se ha dado, incluso, el caso en una gran empresa norteamericana donde, en el momento de contratar **GENEXUS**, el cliente estaba utilizando desde hacía ya tres años una herramienta competidora, utilizada por un numeroso equipo humano, y no estaba dispuesto a desarrollar nuevamente sistemas que eran bastante nuevos: se resolvió que todos los grandes desarrollos nuevos se harían con **GENEXUS**, y se organizó una coexistencia pacífica que ha funcionado bien.

En resumen: **GENEXUS** ayuda en el desarrollo, diseñando y generando automáticamente la base de datos y los programas pero, además, mantiene en forma 100% automática tanto dicha base de datos como dichos programas, y es el único producto que lo hace, a nivel mundial. Esta es la gran diferencia.

Seguidas las precauciones vistas, es sencillo trabajar con grandes aplicaciones corporativas.

Para una mayor ilustración sobre el tema, se recomienda leer: **GENEXUS** Methodology, ARTech 1988-1995.

Esta manera de trabajar tiene otras consecuencias muy favorables: luego de un estudio inicial, se determinan cuales serán los objetos de interés general, y cuál es el alcance de cada una de las aplicaciones a abordar, evitando los estudios detallados a priori, siempre llenos de errores.

En el relevamiento y análisis detallados se usa **GENEXUS**. Las diferentes visiones de los usuarios son descritas utilizándolo y, con la frecuencia que se quiera, se muestran al usuario (y se ponen a su disposición para que los pruebe), prototipos. Esto no evitará muchos de los errores de interpretación que se provocan en todo diálogo humano, pero permitirá encontrarlos oportunamente, cuando su corrección es fácil y no tiene grandes costos.

El usuario no entiende (ni lo pretende) especificaciones abstractas, detalladas y plagadas de términos técnicos y de errores. En cambio es el único capaz, al ejecutar el prototipo del nuevo sistema (funcionalmente idéntico al definitivo), de encontrarle errores, carencias o inadaptaciones. Esta crítica oportuna es invaluable para mejorar la calidad y disminuir el costo del sistema que se está desarrollando.

Lo anterior es muy importante. Sin embargo existe una consecuencia que puede serlo aún más. Los usuarios de todos los niveles, normalmente, han perdido su confianza en la informática. La relación de ellos con los técnicos está muy desgastada, porque sus expectativas se han visto frustradas muchas veces. Pero, con el uso de **GENEXUS** y la metodología incremental, participan (y se sienten participantes), entonces adoptan una actitud mucho más activa, se tornan más colaboradores, y mucho más exigentes. Se sienten protagonistas del proyecto, y lo son, (son los protagonistas más importantes, los únicos imprescindibles), suben al mismo bote y reman para el mismo lado, y todos tenemos mucho a ganar con este cambio de actitud.

Otra consecuencia, derivada de lo anterior es que, en virtud del alto grado de prototipación y participación efectiva de los usuarios, las aplicaciones desarrolladas con **GENEXUS**, en el momento de ponerse en funcionamiento, tienen un grado de madurez mucho mayor que el habitual cuando se emplean otros métodos y herramientas de desarrollo.

Por otra parte es bueno ver que las aplicaciones administradas por **GENEXUS** son independientes de las arquitecturas, y de las plataformas de Hardware y Software (Centralizadas, Cliente / Servidor de Archivos, Cliente / Servidor de Base de Datos, Cliente / Servidor de Web), y que el Usuario puede llevarlas, de unas a otras de las arquitecturas y plataformas soportadas, sin cambio alguno de sus especificaciones, (obviamente los programas que se generan, que necesitan estar optimizados para cada plataforma de hardware, sistema operativo, sistema de gerencia de base de datos, y lenguaje son muy diferentes, pero eso es totalmente transparente).

En general se admite que los costos de los sistemas aumentan en forma exponencial con el tamaño de los mismos. En los métodos de desarrollo tradicionales, las causas son bastante claras, y se ha escrito mucho sobre ellas.

¿Evitaremos totalmente esto con **GENEXUS**?, ¿es indiferente desarrollar sistemas pequeños o grandes?. En realidad no es indiferente, pero es siempre posible desarrollar exitosamente un sistema con **GENEXUS**, con independencia de su tamaño y de su contexto. Es más: es la única tecnología que, en su esencia, permite llegar a grandes sistemas totalmente integrados y con un alto grado de calidad en el diseño.

Desde un punto de vista teórico puro, los costos siguen siendo exponenciales, pero hemos conseguido “achatar” la curva de tal manera que esos costos, en términos prácticos, tienen un comportamiento sensiblemente lineal.

O sea que se consigue una verdadera “escalabilidad” del desarrollo y, un elemento adicional, que va mucho más allá de la teoría: utilizando correctamente **GENEXUS** y su metodología, al final todo funciona perfectamente, como se puede ver por la experiencia, y el alto nivel de satisfacción, de los Usuarios.

Este proceso ha comenzado a fines de 1989, con nuestros primeros clientes, han transcurrido poco más de seis años y hoy tenemos la experiencia de servir a más de 1300 clientes en todo el mundo. ¿Que tipo de empresas utilizan **GENEXUS**?: todo tipo de empresas, desde pequeñas empresas uruguayas a grandes corporaciones internacionales como Laboratorios Roche en la mayor parte de los países de América Latina, como Saint Gobain, líder mundial de la industria del vidrio en su filial brasileña, como Gillette do Brasil, las embotelladoras de Coca Cola de São Paulo y Ciudad de México (por citar sólo las de dos de las mayores ciudades del mundo), y muchas otras de diferentes países, Cervecería Quilmes, algunas grandes empresas norteamericanas, múltiples bancos, empresas financieras, empresas comerciales, empresas industriales, empresas gubernamentales, empresas de software, etc..

Casi todos los proyectos, sin embargo, tienen algo en común: casi siempre se implementan todas las aplicaciones sobre un gran modelo corporativo, con cientos de tablas. Casi siempre se trata de aplicaciones de misión crítica. Casi siempre el usuario utiliza **GENEXUS** como única herramienta, y no escribe una sola línea de código fuera de él.

Para tener una idea de algunas aplicaciones desarrolladas con **GENEXUS**, podemos citar:

Empresa	Aplicación	Tablas	Programas
ACODIKE (Empresa dedicada al negocio del gas licuado de petróleo, Uruguay)	Sistema integral de gestión	320	1400
BANCO ITAMARATÍ (São Paulo, Brasil)	Sistema general bancario, desarrollado directamente para sustituir 2 mainframes	800	5000
BROWSE (Casa de Software, Chile)	LISA sistema general de gestión de empresas	300	4000
	Sistemas Gubernamentales (Contabilidad, Control Presupuestario, etc.)	200	1200

CENTRAL LANERA (Empresa dedicada a la comercialización de lana, en todas sus variantes, Uruguay)	Sistemas de Productores y de Stock	300	1200
COIMBRA (empresa de gran porte, del grupo francés Dreyffus, dedicada al negocio de “commodities” como café, cacao, soja, naranjas, etc., São Paulo, Brasil)	Sistema general de gestión del negocio utilizando bases de datos distribuidas en diversas localidades del Brasil	500	3000
COOPERATIVA BANCARIA (Cooperativa de consumo, Uruguay)	Sistema general de gestión	290	1400
CORPORACIÓN INFORMÁTICA (Casa de Software, Uruguay)	TRUCK - gestión de ventas y expedición en negocios de distribución	200	1300
DE LARROBLA & ASOCIADOS (Casa de Software, Uruguay)	BANTOTAL - sistema genérico que resuelve totalmente la operativa bancaria y que está instalado en 9 bancos del Uruguay, 6 del Paraguay, 1 de Argentina	300	3000
EQUITAL (Empresa de TV para abonados, Uruguay)	Sistema integral de gestión de TV para abonados	500	5000
GENEXUS CONSULTING (Casa de Software, Uruguay)	Sistema genérico de gestión integral de tarjetas de crédito	300	1300
INTERFASE (Casa de software, Uruguay)	Sistema integral genérico de gestión de personal	420	2000
MACRO MERCADO MAYORISTA (Uruguay)	Sistemas generales de gestión de un supermercado al por mayor.	300	1500
ROCHE ARGENTINA	Costos, Consultas médicas (e-mail, estadísticas de consultas, etc.). Aplicación corporativa que es utilizada por múltiples filiales de varios países).	100	400
REAL SEGUROS	Sistema de gestión de seguros de vida	140	650
VIDRARIA SANTA MARINA (Integrante del grupo Saint Gobain de Francia, líder mundial de la industria del vidrio, São Paulo, Brasil)	Sistemas generales. En un plan de tres años están sustituyendo el procesamiento centralizado realizado en tres mainframes de gran porte con arquitectura centralizada, por Cliente / Servidor con múltiples servidores y más de 5000 puestos de trabajo. El desarrollo está hecho 100% en GENEXUS y los números siguientes corresponden al primer año, ya realizado	700	8000

<p>Gran empresa textil, fabricante y distribuidor, con fábricas en América Central y Extremo Oriente, y distribución en todo el territorio de los EE UU</p>	<p>Sistemas generales de gestión de las actividades comerciales e industriales.</p> <p>En este caso, en el momento de contratar GENEXUS, el cliente ya llevaba tres años trabajando con un producto competidor y un numeroso equipo humano.</p> <p>A partir de allí, el desarrollo se ha hecho básicamente con GENEXUS, conviviendo las aplicaciones sobre una base de datos común.</p> <p>Los números aquí expresados corresponden a la parte totalmente administrada por GENEXUS.</p>	<p>400</p>	<p>3000</p>
--	--	------------	-------------

Sobre los autores:

Breogán Gonda y Juan Nicolás Jodal son Ingenieros de Sistemas, formados por la Facultad de Ingeniería de la Universidad de la República (Uruguay), y han desarrollado una vasta actividad docente y profesional, en el Uruguay, y en el exterior.

Sus áreas principales de interés son: Bases de Datos Relacionales, Inteligencia Artificial y Desarrollo Automático de Aplicaciones.

Son socios y directores de ARTech y **GENEXUS** Consulting, en el Uruguay, y de **GENEXUS**, Inc. en los EE. UU.

Karina Santo es Ingeniera de Sistemas, formada por la Facultad de Ingeniería de la Universidad de la República (Uruguay), y ha desarrollado múltiples trabajos de consultoría especialmente en Uruguay, Argentina, Chile, Brasil y EE UU.

Es la responsable de consultoría y soporte a cliente **GENEXUS**, en todo el mundo.

Es socia y directora ejecutiva de **GENEXUS** Consulting, y directora de ARTech.