



La génesis de
GeneXus



Sobre los autores

Breogán Gonda **Presidente de Artech**

Ingeniero de Sistemas formado por la Facultad de Ingeniería de la Universidad de la República (UdelaR).

Ha sido profesor en la Facultad de Ingeniería de UdelaR, en la Pontificia Universidad Católica de Porto Alegre (Brasil) y en la Universidad Católica del Uruguay. También ha dictado cursos y seminarios en áreas de su especialidad, como profesor visitante, en carreras de postgrado en universidades de varios países latinoamericanos.

Desde 1976 a 1989 ha prestado asesoramiento en las áreas de proyecto de Base de Datos y desarrollo de aplicaciones a varias de las mayores empresas del Brasil y Uruguay. Ha dictado múltiples cursos en temas de su especialidad en Brasil desde 1976 a 1989.

Ha sido distinguido por la Academia Nacional de Ingeniería (Uruguay), junto con Nicolás Jodal, con el Premio Nacional de Ingeniería 1995 por el proyecto GeneXus. Ha sido reconocido por la Asociación de Ingenieros del Uruguay como "Ingeniero destacado del año 1996". En Julio de 1999 ha sido designado integrante de la Academia Nacional de Ingeniería (Uruguay)

Sus áreas de investigación son: Bases de Datos, Inteligencia Artificial, métodos de desarrollo automático de aplicaciones e interacción entre la informática y la empresa.

Es coautor del proyecto GeneXus. Es socio fundador y director de Artech y GeneXus Consulting en Uruguay y de GeneXus, Inc en EEUU, Artech de México, Artech do Brasil y GeneXus Japan Inc. en sus respectivos países.

Nicolás Jodal **Vicepresidente de Artech**

Ingeniero de Sistemas formado por la Facultad de Ingeniería de la Universidad de la República (UdelaR).

Ha sido profesor en la Universidad Católica del Uruguay Dámaso Antonio Larrañaga. Ha dictado múltiples cursos en temas de su especialidad en Brasil desde 1984 a 1989.

Desde 1984 a 1986 ha prestado asesoramiento en las áreas de proyecto de Base de Datos y desarrollo de aplicaciones a varias de las mayores empresas del Brasil y Uruguay.

Ha sido distinguido por la Academia Nacional de Ingeniería (Uruguay), junto con Breogán Gonda, presidente de Artech, con el Premio Nacional de Ingeniería 1995 por el proyecto GeneXus.

Sus áreas de investigación son: Bases de Datos, Inteligencia Artificial, métodos de desarrollo automático de aplicaciones e interacción entre la informática y la empresa.

Es coautor del proyecto GeneXus. Es socio fundador y director de Artech y GeneXus Consulting en Uruguay y de GeneXus, Inc en EEUU, Artech de México, Artech do Brasil y GeneXus Japan Inc. en sus respectivos países.

La génesis de GeneXus

Por Breogán Gonda y Juan Nicolás Jodal *

*En este trabajo utilizamos habitualmente la palabra “nosotros”. Con esta palabra nos referimos algunas veces a experiencias o realizaciones de alguno de los autores, o de ambos pero, en general, debe entenderse que nos referimos a experiencias o realizaciones de nuestro equipo. En ese equipo y en la labor generosa y muchas veces anónima de sus integrantes, radica buena parte de la fuerza de GeneXus.

1.

Introducción

Frecuentemente se nos ha pedido, dentro y fuera de la Comunidad GeneXus, que escribamos una historia de GeneXus, ¿cuál fue el origen de GeneXus?, ¿cómo hicimos GeneXus?

Pensamos que no tiene mayor interés contar “cómo hicimos GeneXus”, lo que encierra una fuerte determinación, mucha fe, múltiples actividades pequeñas, algunas grandes, múltiples descubrimientos pequeños y algunos grandes, logrados con nuestra investigación, sin prisa y sin pausa, en la que llevamos invertidos algunos cientos de años / persona.

Mucho más importante es decir que GeneXus es el producto de un magnífico equipo humano, con una alta calificación científico / tecnológica, trabajando con singular generosidad, entusiasmo, dedicación y fe, disfrutando siempre de lo que hace, más allá de que, a veces, pueda ser muy árido.

Pero también debemos decir que el GeneXus de hoy y, sobre todo, el de mañana, no sería posible sin la interacción cada vez mayor con la Comunidad GeneXus, con sus 70.000 desarrolladores que, hoy, en todo el mundo, llevan a cabo su actividad profesional alrededor de GeneXus.

Pensamos que son mucho más importantes las preguntas que las respuestas

Pensamos que son mucho más importantes las preguntas que las respuestas. Que en el mundo de hoy, cuando tenemos una pregunta bien formulada (rigurosamente formulada) existe y está disponible un enorme arsenal de herramientas para ayudarnos a responderla.

¡Qué solos estamos a la hora de formular las preguntas que nos llevan a innovar!

Sin embargo, ¡qué solos estamos a la hora de formular las preguntas que nos llevan a innovar!

GeneXus es obra de su equipo y de su Comunidad; ambos asumieron siempre la necesidad de la innovación y han tenido un buen nivel de acierto en la formulación de las cuestiones importantes para lograrla.

Por ello creemos que la mejor manera de contar la historia de GeneXus es recrear algunos eventos y algunas preguntas que han sido esenciales en su construcción y que han ido surgiendo a través del trabajo y del tiempo.

2.

¿Qué hicimos antes de GeneXus?

Diversas cosas, pero lo que viene al caso es una fuerte actividad de consultoría y docencia en el área de bases de datos.

Nuestra actividad de consultoría parece haber sido satisfactoria para nuestros clientes, pero no así para nosotros.

Generalmente no se nos llamaba en el momento del proyecto, sino en el de los problemas (pérdidas de integridad de la base de datos, tiempos de respuesta demasiado grandes, etc.).

Cada cliente implementaba varias bases de datos, cada una de ellas para tratar un tipo de problema particular. No existían bases de datos corporativas.

Esas bases de datos eran grandes, si las me-

díamos en millones de registros, pero muy pequeñas desde el punto de vista de la completitud de la información y, en particular, de su utilidad para apoyar a la empresa en la toma de decisiones.

Las diversas bases de datos de cada cliente eran fuertemente redundantes y, en consecuencia, inconsistentes. Esa inconsistencia hacía que no fuera una buena idea combinar datos de las diversas bases de datos, aunque pertenecieran a la misma empresa.

En estas circunstancias comenzaron a surgir algunas preguntas: **¿son de utilidad las bases de datos?, ¿es útil nuestra función de consultores?**

3. Un gran proyecto en 1984

Un hecho incidental, que nos llevó a nuestras investigaciones que condujeron a GeneXus, fue un importante proyecto de consultoría que desarrollamos a partir de mediados de 1984.

Un hecho incidental nos llevó a nuestras investigaciones que condujeron a GeneXus

El cliente era una gran empresa brasileña, con casa central en la ciudad de São Paulo, que actuaba fundamentalmente en las áreas de vestimenta y calzado deportivos. Era una empresa muy grande y nuestro interlocutor no era un colega, cosa habitual por aquellos tiempos, sino el Director General. Su pensamiento era muy claro:

“en esta empresa cualquier funcionario de nivel medio, que toma sólo decisiones bastante

irrelevantes, siempre está muy bien apoyado por información, pero yo, y la alta gerencia en general, jamás disponemos de la información adecuada para apoyarnos en nuestras decisiones, las que pueden conducir a la empresa al éxito o al fracaso”;

“es inútil me pidan que determine a priori qué información necesitaré para tomar mis decisiones; cada caso es diferente y sólo se sabe en el momento qué información se precisa: es esencial la capacidad de definirla nosotros mismos y obtenerla de inmediato”;

“me he convencido de que necesitamos que todos nuestros sistemas utilicen una única base de datos corporativa, que nos permita en cualquier momento obtener de ella la información que necesitamos”;

“les ofrecemos a Uds. la tarea de hacer una reingeniería total de nuestra informática para lograrlo, usando como fuerza de trabajo básica a nuestro personal técnico actual, aunque estamos abiertos a algunas contrataciones adicionales, en el caso de que sean necesarias habilidades o perfiles que nuestros técnicos actuales no posean”;

“tendrán a su disposición la tecnología más avanzada”;

“pero no creo en proyectos de larga duración y, además, tenemos mucha prisa: todo debe estar pronto en un año”.

El desafío era enorme, la oportunidad también

El desafío era enorme, la oportunidad también: en ese momento mucho se hablaba, en los países más desarrollados, de bases de datos corporativas e información corporativa, pero nada o casi nada se hacía.

Era la gran oportunidad, la oportunidad soñada, nos sentíamos capaces de aprovecharla y la enfrentamos decididamente.

4.

La dimensión del problema

Cuando se enfrenta un problema real de gran tamaño como éste, sobre el cual no existe experiencia, los riesgos son grandes.

Cuando se enfrenta un problema nuevo de gran tamaño los riesgos son grandes

Si, además, se verifica que toda la bibliografía se refiere a casos hipotéticos teóricos y que nadie de los que escriben se ha embarrado los zapatos resolviendo algún problema real comparable, aumentan los riesgos y la incertidumbre.

En nuestro caso también aumentó el entusiasmo y nuestra evaluación del tamaño de la oportunidad.

En el análisis de datos inicial, hecho en la primera semana de trabajo para tener una visión general del problema, se identificaron unas 100 entidades y se pudo estimar que el modelo completo llegaría a más de 500 tablas (el modelo real tuvo 750 tablas), muy diferente de los modelos con que se trabajaba en la época, que jamás pasaban de las 40 tablas.

Primer problema: ¿Cómo lograr que los usuarios finales fueran capaces de formular las consultas que necesitaban?

Suponiendo que consiguiéramos construir la base de datos y los programas necesarios por la aplicación, ¿cómo lograr que el usuario final o su asistente pueda formular las consultas necesarias, en cada momento en que se presente esa necesidad?

Éste era un problema nuevo: lo ideal sería que se pudiera resolver con SQL pero rápidamente verificamos que eso no era realista. El SQL es un lenguaje de nivel demasiado bajo desde el punto de vista de su usabilidad, porque requiere un muy buen conocimiento de la base de datos (qué tablas contienen los elementos necesarios y cómo navegar entre ellas).

El SQL es un lenguaje de nivel demasiado bajo desde el punto de vista de su usabilidad

Identificamos, entonces, el primer problema no tradicional a resolver: ¿cómo interrogar la base de datos en cualquier momento con consultas no previsible, definidas por personal no técnico?

El SQL no era la solución, los “lenguajes orientados a usuarios” disponibles en la época tampoco. Éstos últimos eran amigables pero no ayudaban nada a resolver el problema de los grandes modelos.

Nuestro primer tema de investigación, era obtener un lenguaje donde el sistema se responsabilizara por escoger las tablas necesarias y formular la navegación entre ellas, todo automáticamente.

Buscábamos que el sistema se responsabilizara de la navegación en la base de datos automáticamente

Análisis de datos. Nos dispusimos a efectuar el acostumbrado análisis de datos detallado. Hasta entonces trabajábamos con el modelo E-R (Entity Relationship) introducido en la década de los '60 por Charles Bachman y luego popularizado por Peter Chen: buscábamos en la organización los objetos relevantes al problema y sus relaciones, y los representábamos en el modelo E-R. Pero aquí, en principio, todo era relevante ya que se buscaba una base de datos corporativa.

Rápidamente nos fue claro que esta forma de proceder, que funciona bastante bien en pequeños modelos, como los que se usaban hasta entonces, presentaba muchas dificultades en los grandes modelos corporativos.

También percibimos que sería muy útil visualizar partes del modelo con un gráfico E-R. O sea: el modelo E-R parecía ser un output deseable, pero un input inútil, en los modelos corporativos reales.

Segundo problema: ¿Cómo podemos construir y administrar grandes modelos?

Se trataba de modelos de datos mucho más grandes de los acostumbrados, ¡ciertamente cometeríamos muchos errores técnicos asociados a ese mayor tamaño!

Nos enfrentamos a modelos de datos mucho más grandes de los acostumbrados

Para ayudarnos a lidiar con este problema fuimos implementando pequeñas herramientas. Paralelamente buscamos internacionalmente herramientas que nos pudieran ayudar. Lamentablemente no existían.

Otros problemas. Las dificultades emanadas del tamaño del modelo, rápidamente se nos mostraron como sólo la “punta del iceberg”.

Tercer problema: ¿Dónde está el conocimiento?

¿Quién, en la organización, conoce los datos con el necesario nivel de objetividad y detalle? La respuesta fue categórica: ¡NADIE!

Entonces ¿qué hacer?, ¿podemos resolver un problema sin tener el conocimiento adecuado?: **NO.**

¿Hay paliativos por la vía de entrenar a los usuarios o diseminar en la empresa “administradores de datos departamentales”, etc.?

Paliativos siempre se pueden encontrar pero, cuando se pasa de repente a problemas por lo menos 10 veces más grandes de lo habitual,

difícilmente esos paliativos sean suficientes: parece el momento de repensar todo con la mayor libertad, de asumir que muy poco o nada nos pueden ayudar los antecedentes y la bibliografía.

¿Dónde está el conocimiento válido?, ¿podemos sustituir el conocimiento de los datos, que hemos comprobado no existe, por otro objetivo y con el detalle necesario para permitirnos inferir de él el modelo de datos?



5. El isomorfismo con la perspectiva

Si damos una simple mirada en la historia del dibujo y la pintura, vemos que al principio había grandes diferencias con lo actual.

¿Cómo se dibujaba al principio?, ¿cómo dibujaríamos intuitivamente?, ¿cómo dibujan los niños?: tratando de conocer bien el objeto a dibujar, si es posible tocándolo, conociendo su naturaleza y todos los detalles posibles. Luego lo dibujaríamos **“como sabemos que es”**.

¿Cómo son esos dibujos primitivos?: fuertemente deformados.

Un día apareció la **“perspectiva”**: en el Renacimiento algunos pintores y algunos arquitectos, que tenían problemas de dibujo mucho mayores que los pintores, comenzaron a pensar que debían cambiar el paradigma de **“dibujamos como sabemos que es”** a **“dibujamos como lo vemos”**.

En 1417, en Florencia, Filippo Brunelleschi, artista y arquitecto italiano, para poder representar los edificios en perspectiva, formalizó un conjunto de reglas (principios de una geometría descriptiva) que se utilizan hasta hoy.

Como todo nuevo paradigma, la perspectiva

fue muy resistida (sus detractores afirmaban que “se pretende sustituir el arte de dibujar y pintar por una técnica pedestre y nada creativa”) pero, cuando se impuso, desplazó totalmente al anterior.

Como todo nuevo paradigma, la perspectiva fue muy resistida

Pero **¿qué tiene que ver la perspectiva con el análisis de datos?**

Cuando se ha decidido repensar todo, cuando se piensa con total libertad, fuera de cualquier contexto preestablecido, nada debe descartarse a priori. La historia de la perspectiva constituyó una buena fuente de inspiración.

La historia de la perspectiva constituyó una buena fuente de inspiración

Lo más importante es el principio básico de su cambio de paradigma: se pasó de un abordaje complejo, confuso y subjetivo a un abordaje descriptivo, simple y objetivo. ¡Si pudiéramos hacer algo parecido con los datos, habríamos recorrido buena parte del camino!

Visto así el problema, volvamos a la pregunta: “¿Quién en la organización conoce los datos con el necesario nivel de objetividad y detalle?” y tratemos de sustituirla por otra que pueda responderse con un Sí y que nos ayude en la construcción del modelo de datos que queremos.

¿Sobre qué existe conocimiento objetivo y suficientemente detallado?

En nuestra búsqueda encontramos que, de la misma manera que no hay buen conocimiento sobre los datos, cada usuario tiene un muy buen conocimiento de las visiones de esos datos que él utiliza.

Volvamos a la perspectiva: tanto en la pers-

pectiva como en el análisis de datos, “describimos visiones”.

Tanto en la perspectiva como en el análisis de datos “describimos visiones”

Parece auspicioso pero en la perspectiva, lo que dibujamos es directamente la visión, aquí la visión parecía importante, pero lo que nosotros buscábamos era el modelo de datos.

La pregunta obvia es: **dado un conjunto de visiones de usuarios ¿podemos inferir de ellas un modelo de datos que las satisfaga?**

La pregunta es muy buena porque lleva el problema al mundo de la matemática:

Primero, debemos definir un marco de referencia:

Nuestros elementos de datos (atributos) se identificarán por su nombre y cumplirán, básicamente, reglas sencillas:

Un atributo se identificará siempre con el mismo nombre, con independencia de donde aparezca en el modelo.

No existirán dos atributos diferentes que respondan al mismo nombre.

Atribuiremos los nombres de manera que representen lo mejor posible el significado de cada atributo.

Segundo, debemos representar la estructura de las visiones:

Cada visión involucrará uno o varios atributos, organizados según una determinada estructura.

Pero nunca es bueno reinventar la rueda: *A esta altura nos encontramos con que, fundamentalmente Jean Dominique Warnier y nuestro amigo Ken Orr y, por otro lado también Michael Jackson, habían avanzado mucho en la descripción de las estructuras de datos.*

Ni Warnier-Orr ni Jackson pretendían con sus estructuras de datos definir la base de datos, sino la estructura de los programas y la mayor parte de la bibliografía de que disponíamos se refería a programas “batch”. Pero aquellos trabajos fueron un apoyo invaluable para la representación de nuestras visiones de datos.

Los trabajos de Warnier-Orr y de Jackson fueron un apoyo invaluable

Tercero, necesitamos un procedimiento para pasar de un conjunto de visiones de datos al modelo correspondiente.

Estamos ahora en el mundo de la matemática y la pregunta es: **dado un conjunto de visiones de datos ¿existe un modelo relacional mínimo que las satisface?**

Por último, ahora ya tenemos todas las preguntas relativas a este asunto o, dicho de otro modo, hemos formulado rigurosamente el problema.

En este estado existe una multitud de herramientas que nos pueden ayudar a resolverlo.

En el caso concreto hemos utilizado, además de las herramientas informáticas usuales, técnicas y herramientas de la matemática, la lógica y la inteligencia artificial.

Los trabajos realizados nos condujeron a la resolución del problema. Un subproducto importante fue introducirnos en un mundo nuevo y promisorio: el de la prototipación rápida.

Hemos utilizado mucho la prototipación en nuestras investigaciones

Hemos utilizado mucho la prototipación en nuestras investigaciones y, luego, hemos viabilizado su uso a nuestros clientes con GeneXus.

6.

¿Podemos generar automáticamente los programas que necesitamos?, ¿algunos?, ¿todos?, ¿cuáles?, ¿qué ventajas tendrían estos programas sobre los escritos a mano?

En el comienzo no pensábamos en la generación de programas, pero el fracaso que tuvo siempre la informática con las llamadas “bases de datos estables” nos fue convenciendo de que debíamos, en algún momento, encarar este problema, ya que de otra manera nuestros clientes vivirían siempre presionados por grandes costos de mantenimiento.

No teníamos experiencia en la generación automática de programas, pero el asunto no era nuevo:

Desde casi sus comienzos, la informática ha encarado la generación automática de programas.

Durante mucho tiempo los generadores de programas fueron bastante primitivos y, sobre todo, orientados a la generación de reportes simples a partir de archivos planos.

Durante mucho tiempo los generadores de programas fueron bastante primitivos

En la segunda mitad de la década de los 80 se avanzó bastante y estos generadores comenzaron a generar tanto programas “batch” como transaccionales y a interactuar con bases de datos.

Estos generadores se basaban en “plantas” o “esqueletos” que comenzando con un abordaje del tipo “fill on the blanks” y, luego,

adquiriendo más y más sofisticación, llegaban a resolver una parte razonable de las necesidades de una instalación.

Nos fue claro que la generación automática de programas se tornaría un asunto esencial

Nos fue claro que la generación automática de programas, en algún momento se tornaría para nosotros un asunto esencial.

7.

¿Existen las bases de datos estables?

Las “bases de datos estables” constituyen un asunto recurrente de la informática. La idea era la siguiente:

Si conseguimos “la base de datos correcta” para una determinada empresa, esa base de datos se mantendrá estable en el futuro. Como consecuencia, nos limitaremos a través del tiempo, a escribir programas que la utilicen.

Si esto no es posible, significa que no hemos conseguido definir “la base de datos correcta”.

Sobre este tema se ha escrito mucho.

Pero la tesis es falsa: ¡sólo si una organización está anquilosada o muerta puede tener un modelo estable!

Entonces es bueno no utilizar esfuerzos en la búsqueda de dichos modelos estables, sino prepararse para trabajar con los modelos posibles, reales, inestables.

8.

¿Cómo trabajar bien con bases de datos inestables?

El punto anterior nos sugiere esta pregunta, pero, estudiándola acabamos descomponiéndola en las siguientes:

- **¿Cómo reorganizar la base de datos cuando se producen cambios estructurales en la misma?**
- **¿Cómo modificar los programas para que puedan funcionar bien con la nueva base de datos?**

La primera de estas preguntas nos lleva a un profundo estudio de cómo convertir el contenido corriente de una base de datos con la vieja estructura al nuevo contenido, con la nueva estructura.

El problema teórico inicial es: **¿puede hacerse esta conversión sin perder datos?**

Si la respuesta es sí, aparecen otras nuevas: **¿qué debemos hacer para efectuar la conversión sin perder datos? y ¿podemos generar automáticamente los programas que realicen dicha conversión?**

La segunda pregunta tiene una respuesta obvia: si somos capaces de generar los programas ¡generemos nuevamente todos los programas!

El uso de la fuerza bruta casi siempre nos da una primera solución

El uso de la fuerza bruta casi siempre nos da una primera solución, pero cuando tenemos miles de programas, no parece una buena

solución, aún teniendo en cuenta el grande y constante aumento de la potencia del hardware.

Ciertamente, esta repuesta no nos convence mucho y nos sugiere otra pregunta: ante cambios en la base de datos **¿podemos determinar qué programas se ven afectados para regenerarlos?**

Si logramos lo anterior será muy bueno pero, sin embargo la situación nos sugiere una nueva pregunta: **¿existen programas generados para la vieja estructura que funcionan correctamente con la nueva y que, sin embargo, ahora podrían ser sustituidos por otros, más eficientes?**

9. Los dioses también juegan: las “tablas extendidas”

Un domingo de agosto de 1986, en New York, cuando aún no pensábamos hacer una empresa y un producto, sino empaquetar un conjunto de descubrimientos científico / tecnológicos y licenciarlos a terceros, descubrimos algo estéticamente formidable: las “**tablas extendidas**”.

Un domingo de agosto de 1986 descubrimos las tablas extendidas

No vamos a dedicar mayor espacio a este asunto, que es bastante conocido en la Comunidad GeneXus, ya que este trabajo pretende poner el énfasis en las preguntas y no en las

respuestas, pero digamos lo siguiente, que contiene algunas simplificaciones no esenciales:

¿Qué es una “tabla extendida”? Para cada registro de una determinada tabla, existe un registro virtual formado por la concatenación del registro original y todos los de otras tablas de la base de datos que queden directa o indirectamente determinados por él.

Al conjunto de estos registros le llamaremos “tabla extendida” de la tabla original y a aquella, “tabla base” asociada a la tabla extendida.

¿En qué consiste la importancia de las tablas extendidas?

Las descripciones expresadas en términos de tablas extendidas, se mantienen vigentes a través de los cambios estructurales de la base de datos.

Como consecuencia, por más que algunos programas dejen de ser correctos u óptimos por cambios en la base de datos, las descripciones de las visiones de usuarios de GeneXus se mantienen válidas y, entonces, es posible propagar automáticamente los cambios, identificando los programas inválidos y generándolos nuevamente a partir de las descripciones originales de sus visiones.

Nunca tuvimos dudas de que este descubrimiento era estéticamente formidable

Nunca tuvimos dudas de que este descubrimiento era estéticamente formidable. Con el tiempo, la realidad nos ha mostrado que es mucho más que eso.

10.

GeneXus

Nuestra intención original era licenciar la tecnología obtenida a grandes jugadores. La terca realidad nos probó que era un propósito ingenuo: **¿qué credibilidad pueden tener ideas tan avanzadas, ante grandes jugadores tecnológicos de los países más desarrollados, cuando vienen de un país que no tiene una fuerte tradición de productor de tecnología?**

¿Qué credibilidad pueden tener ideas tan avanzadas cuando vienen de un país que no tiene tradición tecnológica?

La disyuntiva era entre publicar los descubrimientos realizados y dar por terminada la investigación o hacer una empresa, un producto y tratar de comercializarlo, comenzando por Uruguay y los países vecinos para seguir luego con el mundo todo.

Escogimos la segunda opción. Entendimos que, a través del tiempo, si nuestro producto es realmente útil a nuestros clientes, si la cantidad de clientes crece constantemente aunque no sea a gran velocidad, si conseguimos soportar oportunamente las nuevas tecnologías que vayan apareciendo y si mantenemos un comportamiento empresarial espartano, obtendremos el éxito.

Asumimos este objetivo con el mayor compromiso y, a fines de 1988, fundamos Artech, dimos el nombre de GeneXus a nuestro producto y nos dispusimos a liberar en el segundo semestre de 1989 su primera versión.

Algunas preguntas que se nos presentaron entonces:

¿Para qué plataforma debe generar aplicaciones GeneXus en los primeros tiempos?

Lo realista era escoger inicialmente una única plataforma y especializarnos en ella hasta que consiguiéramos un volumen empresarial mínimo que nos permitiera encarar otras.

Escogimos el IBM AS/400.

¿En qué plataforma debe funcionar GeneXus en los primeros tiempos?

Se necesitaba una plataforma que no requiriera grandes inversiones, que fuera eficiente y que asegurara un desarrollo permanente.

También era deseable la mayor independencia posible de la plataforma para la que generaríamos las aplicaciones, para luego viabilizar la generación para otras plataformas.

Era deseable la mayor independencia posible de la plataforma para la que generaríamos las aplicaciones

Escogimos PC con sistema operativo DOS.

¿Qué problemas debe resolver GeneXus de inmediato y cuáles serán postergados para el futuro?

Nos era claro que, al principio, teníamos aún carencias teóricas e inconvenientes prácticos que nos impedían generar aplicaciones completas.

Un objetivo era generar automáticamente todo lo posible.

Un segundo objetivo era que todo lo que generáramos

Un segundo objetivo era que todo lo que generáramos pudiéramos mantenerlo automáticamente

mos pudiéramos mantenerlo automáticamente.

Por ello descartamos de plano generar partes de programas y dejar al desarrollador la tarea de completarlos, dado que no podríamos mantener automáticamente lo que el desarrollador escribía a mano.

No sabíamos cómo describir programas procedurales como, por ejemplo, procesos “batch” o rutinas casuísticas que no se pudieran derivar de las transacciones.

Podríamos introducir un lenguaje de 4ª generación, que resolvería los problemas anteriores, pero ello significaría renunciar al mantenimiento automático de estos programas. Resolvimos no hacerlo.

Entonces resolvimos implementar aquello que podíamos generar y mantener automáticamente sin restricciones: transacciones, consultas y reportes simples.

Estimamos que el primer GeneXus generaba y mantenía automáticamente el 70% de los programas

Estimamos que el primer GeneXus generaba y mantenía automáticamente el 70% de los programas de una instalación y se hacía cargo del diseño, generación y mantenimiento automáticos de la base de datos. El resto requería fuerte investigación adicional.

¿Podríamos generar y mantener automáticamente el 100% de las aplicaciones?

Era nuestro objetivo y estábamos trabajando para ello, pero pensábamos que teníamos

Pensábamos que los clientes contrataban GeneXus sólo por su fuerte aumento de productividad en el desarrollo

bastante tiempo para lograrlo.

Generar y mantener automáticamente el 70% de los programas parecía un gran éxito: sólo algunos generadores podían generar cosas parecidas, pero ninguno podía ofrecer mantenimiento automático.

Nos era claro que el mantenimiento automático era una característica muy importante y única de GeneXus, pero pensábamos que los clientes contrataban GeneXus por su fuerte aumento de productividad en el desarrollo, mientras que el mantenimiento automático era una característica nueva, inesperada y mirada con cierto escepticismo fuera de Artech.

La realidad nos sorprendió cuando varios clientes nos presentaron su pensamiento:

“Generar automáticamente el 70% de los programas nos ayuda mucho y así lo valoramos.”

“Tener que escribir a mano el 30% de los programas es una restricción que aceptamos confiando que Uds. la levantarán en el futuro.”

“Sin embargo, lo que se nos hace inaceptable, es tener que mantener manualmente aquellos programas que GeneXus no genera.”

¿Cómo satisfacer a nuestros clientes?, ¿cómo podemos estar seguros de poder generar aplicaciones completas?: con un lenguaje procedural.

Nuestra pregunta fue **¿podemos construir un lenguaje procedural cuyas descripciones (programas fuentes en ese lenguaje) no se tornen inválidas ante modificaciones estructurales de la base de datos?**

La respuesta fue Sí y nos permitió liberar el primer GeneXus completo.

Luego, las preguntas no pararon de aparecer...

¿Podemos generar aplicaciones para otras plataformas y para otras arquitecturas?

¿Podemos identificar patrones en nuestras

descripciones y, a partir de ellos, generar automáticamente los objetos GeneXus que responden a ellas?

¿Podemos extender GeneXus fuera de Artech?, ¿podemos viabilizar a la Comunidad GeneXus que lo haga?

...y muchas más...

Formularnos nuevas preguntas es una tarea permanente y cada vez más gente participa en ella

Formularnos nuevas preguntas es una tarea permanente y cada vez más gente participa en ella: el equipo de investigación y desarrollo, toda Artech, las casas de software, los clientes en general, la Comunidad GeneXus toda.

11.

Respuestas

En todos estos años de trabajo han aparecido muchas preguntas, que en general generaron respuestas, muchas respuestas. Sin embargo podemos sintetizarlas todas en un principio:

¡Es posible “describir” en vez de “programar”!

y una convicción:

¡Nunca debemos perder nuestra libertad de pensar!

12.

Agradecimientos

Debemos agradecer muy especialmente a muchas personas y empresas: a nuestro equipo, a nuestros clientes, a la Comunidad GeneXus toda.

Pero también, con toda humildad, queremos agradecer a aquellos que, sin ser clientes y sin integrar nuestra Comunidad y, a veces, aún sin creer en GeneXus, nos presentan permanentemente nuevas dificultades y desafíos, porque siempre aprendemos de esas dificultades y desafíos.

¡El mundo es maravilloso y todo aquello que aumente nuestra capacidad de conocerlo constituye un privilegio y una gran oportunidad!

