

A gênese do *GeneXus*



Sobre os autores

Breogán Gonda

Presidente da Artech

Engenheiro de Sistemas formado pela Faculdade de Engenharia da Universidad de la República (UdelaR).

Foi professor na Faculdade de Engenharia da UdelaR, na Pontifícia Universidade Católica de Porto Alegre (Brasil) e na Universidad Católica del Uruguay. Também ministrou cursos e seminários em áreas de sua especialidade, como professor visitante, em cursos de pós-graduação em universidades de vários países latino-americanos.

De 1976 a 1989 prestou assessoramento nas áreas de projeto de Base de Dados e desenvolvimento de aplicações para várias das maiores empresas do Brasil e Uruguai. Ministrou múltiplos cursos em temas de sua especialidade no Brasil de 1976 a 1989.

Foi distinguido pela Academia Nacional de Engenharia (Uruguai), junto com Nicolás Jodal com o Prêmio Nacional de Engenharia 1995 pelo projeto GeneXus. Foi reconhecido pela Associação de Engenheiros do Uruguai como "Engenheiro destacado do ano 1996". Em Julho de 1999 foi designado integrante da Academia Nacional de Engenharia (Uruguai).

Suas áreas de pesquisa são: Bases de dados, Inteligência artificial, Métodos de desenvolvimento automático de aplicações e interação entre a informática e a empresa. É coautor do projeto GeneXus. Sócio fundador e diretor da Artech e GeneXus Consulting no Uruguai, GeneXus, Inc., Artech de México, Artech do Brasil e GeneXus Japan Inc. em seus respectivos países.

Nicolás Jodal

Vice-presidente da Artech

Engenheiro de Sistemas formado pela Faculdade de Engenharia da Universidad de la República de Uruguay (Udelar).

Foi professor na Universidad Católica del Uruguay. Ministrou múltiplos cursos em temas de sua especialidade no Brasil de 1984 a 1989.

De 1984 a 1986 prestou assessoramento nas áreas de projeto de Base de Dados e desenvolvimento de aplicações para várias das maiores empresas do Brasil e Uruguai.

Foi distinguido pela Academia Nacional de Engenharia (Uruguai) junto com Breogán Gonda, presidente da Artech, com o Prêmio Nacional de Engenharia 1995 pelo projeto GeneXus.

Suas áreas de pesquisa são: Bases de dados, Inteligência artificial, Métodos de desenvolvimento automático de aplicações e interação entre a informática e a empresa. É coautor do projeto GeneXus. Sócio fundador e diretor da Artech e GeneXus Consulting no Uruguai, GeneXus, Inc., Artech de México, Artech do Brasil e GeneXus Japan Inc. em seus respectivos países.

A gênese do GeneXus

por Breogán Gonda e Juan Nicolás Jodal *

*Neste trabalho normalmente utilizamos o pronome “nós”. Com esse pronome nos referimos algumas vezes a experiências ou realizações de algum dos autores, ou de ambos, mas em geral deve-se entender que nos referimos a experiências ou realizações de nossa equipe. Nessa equipe e no trabalho generoso e muitas vezes anônimo de seus integrantes está boa parte da força do GeneXus.

1.

Introdução

Frequentemente nos pedem, de dentro e de fora da comunidade GeneXus, para escrevermos uma história do GeneXus. Qual foi a origem do GeneXus? Como fizemos o GeneXus?

Acreditamos que não é tão interessante contar “como fizemos o GeneXus”, o que envolve uma forte determinação, muita fé, diversas atividades pequenas, algumas grandes, diversas descobertas pequenas, algumas grandes, alcançadas com nossas pesquisas, devagar e sempre, nas quais já investimos algumas centenas de anos/pessoa.

Muito mais importante é dizer que o GeneXus é o produto de uma excelente equipe, com uma alta qualificação científica e tecnológica, trabalhando com inigualável generosidade, entusiasmo, dedicação e confiança, sempre com paixão pelo que faz, independente de que às vezes possa ser um campo bastante árido.

Mas também devemos dizer que o GeneXus de hoje e, acima de tudo, o de amanhã, não seria possível sem a interação cada vez maior com a comunidade GeneXus, com seus 70.000 desenvolvedores que em todo o mundo atualmente desempenham suas atividades profissionais ao redor do GeneXus.

Acreditamos que as perguntas são bem mais importantes que as respostas

Acreditamos que as perguntas são muito mais importantes que as respostas. Acreditamos que, no mundo de hoje, quando temos uma pergunta bem formulada (rigorosamente formulada), existe e está disponível um arsenal

Quão solitários estamos na hora de formular as perguntas que nos levam a inovar!

enorme de ferramentas para nos ajudar a respondê-la.

Mas, como estamos sozinhos na hora de formular as perguntas que nos levam a inovar!

O GeneXus é obra de sua equipe e de sua comunidade; ambas sempre reconheceram a necessidade de inovação e tiveram um bom nível de acerto na formulação das questões importantes para chegar a isso.

Por isso, acreditamos que a melhor forma de contar a história do GeneXus é recriando alguns eventos e algumas perguntas que foram fundamentais para a sua construção e que foram surgindo ao longo do trabalho e do tempo.

2.

O que fizemos antes do GeneXus?

Diversas coisas, mas o que vem ao caso é uma forte atividade de consultoria e docência na área de bases de dados.

Nossa atividade de consultoria parece ter sido satisfatória para nossos clientes, mas para nós não era.

Geralmente não nos chamavam no momento do projeto, e sim na hora dos problemas (perda de integridade da base de dados, tempos de resposta longos demais, etc.).

Cada cliente implementava várias bases de

dados, cada uma delas para tratar de um tipo específico de problema. Não existiam bases de dados corporativas.

Essas bases de dados eram grandes, se fôssemos medi-las em milhões de registros, mas muito pequenas do ponto de vista de quão completa era a informação e, especialmente, de sua utilidade para dar apoio à empresa na hora da tomada de decisões.

As diversas bases de dados de cada cliente eram altamente redundantes e, consequentemente, inconsistentes. Essa inconsistência fazia com que não fosse uma boa ideia combinar dados das diversas bases de dados, mesmo que pertencessem à mesma empresa.

Nestas circunstâncias, começaram a surgir algumas perguntas: as bases de dados são úteis? Nossa função de consultores é útil?

3. Um grande projeto em 1984

Um acontecimento fortuito, que nos levou às pesquisas que nos conduziram ao GeneXus, foi um grande projeto de consultoria que desenvolvemos a partir de meados de 1984.

Um fato incidental levou-nos às pesquisas que nos conduziram ao GeneXus

O cliente era uma grande empresa brasileira, com sede na cidade de São Paulo e que estava principalmente no ramo de roupas e calçados esportivos. Era uma empresa muito grande e nosso interlocutor não era um colega, algo comum naquela época, mas o diretor geral da empresa. Seu pensamento era muito claro:

“Nesta empresa, qualquer funcionário de nível

médio, que toma decisões bastante irrelevantes, sempre está muito bem apoiado pela informação, mas eu e a alta gerência, em geral, nunca dispomos da informação adequada na qual nos apoiarmos em nossas decisões, decisões que podem levar a empresa ao sucesso ou ao fracasso.”

“É inútil me pedirem para determinar a priori que informação vou precisar para tomar minhas decisões; cada caso é diferente e só no momento é que sabemos que informação é necessária: é essencial que tenhamos a capacidade de defini-la nós mesmos e de obtê-la de imediato.”

“Estou convencido de que precisamos que todos os nossos sistemas utilizem uma única base de dados corporativa, que nos permita, a qualquer momento, obter a informação de que precisamos.”

“Oferecemos a vocês a tarefa de fazer uma total reengenharia de nossa informática para conseguir isso, usando como força de trabalho básica o nosso pessoal técnico atual, embora possamos considerar algumas contratações adicionais, caso sejam necessários habilidades ou perfis que nossos técnicos atuais não possuam.”

“Vocês terão à sua disposição a tecnologia mais avançada.”

“Mas não acredito em projetos de longa duração e, além disso, temos pressa: tudo deverá estar pronto em um ano.”

O desafio era enorme, a oportunidade também: naquele momento, falava-se muito, nos países desenvolvidos, de bases de dados corporativas e de informação corporativa, mas não se fazia nada ou quase nada.

O desafio era enorme, a oportunidade também

Era a grande oportunidade, a oportunidade sonhada. Sentíamos que seríamos capazes de aproveitá-la e a encaramos decididamente.

4.

A dimensão do problema

Quando se enfrenta um problema real de grande tamanho como este, sobre o qual não existe experiência, os riscos são grandes.

Quando se enfrenta um problema novo de grande porte, os riscos são grandes

Se, além disso, verifica-se que toda a bibliografia se refere a casos hipotéticos teóricos e que nenhuma das pessoas que escreve arregaçou as mangas resolvendo algum problema real comparável, aumentam os riscos de incerteza.

No nosso caso, também aumentou o entusiasmo e nossa avaliação do tamanho da oportunidade.

Na análise de dados inicial, feita na primeira semana de trabalho, para se ter uma visão geral do problema, foram identificadas umas 100 entidades e foi possível estimar que o modelo completo chegaria a mais de 500 tabelas (o modelo real acabou tendo 750 tabelas), muito diferente dos modelos com os quais se trabalhava na época, que jamais passavam de 40 tabelas.

Primeiro problema: Como fazer com que os usuários finais fossem capazes de formular as consultas que necessitavam?

Supondo que conseguíssemos construir a base de dados e os programas necessários para a aplicação, como fazer com que o usuário final, ou seu assistente, pudesse formular as consultas necessárias sempre que surgisse essa necessidade?

Esse era um problema novo. O ideal seria que pudesse ser resolvido com SQL, mas logo percebemos que isso não era realista. O SQL é uma linguagem de nível muito baixo do ponto de vista de sua usabilidade, porque requer um conhecimento muito bom da base de dados (saber que tabelas contêm os elementos necessários e saber navegar entre elas).

Identificamos, então, o primeiro problema não tradicional a ser resolvido: **como interrogar a base de dados a qualquer momento, com consultas não previsíveis, definidas por pessoal não técnico?**

O SQL não era a solução, nem as “linguagens orientadas a usuários” disponíveis na época. Elas eram amigáveis, mas não ajudavam em nada a resolver o problema dos grandes modelos.

O SQL é uma linguagem de muito baixo nível do ponto de vista de sua usabilidade

Nosso primeiro tema de investigação seria obter uma linguagem na qual o sistema se responsabilizasse por escolher as tabelas necessárias e formular a navegação entre elas, tudo automaticamente.

Procurávamos que o sistema se responsabilizasse pela navegação na base de dados automaticamente

Análise de dados. Dispusemos-nos a fazer a análise de dados detalhada. Até então, trabalhávamos com o modelo E-R (Entity Relationship), introduzido na década de 60 por Charles Bachman e mais tarde popularizado por Peter Chen: na organização, buscávamos os objetos relevantes ao problema e suas relações e os representávamos no modelo E-R. Mas aqui, no início, tudo era relevante, já que se buscava uma base de dados corporativa.

Rapidamente ficou claro para nós que esta

forma de proceder, que funciona muito bem em pequenos modelos, como os que eram usados até então, apresentava muitas dificuldades nos grandes modelos corporativos.

Também percebemos que seria muito útil visualizar partes do modelo com um gráfico E-R. Ou seja, o modelo E-R parecia ser um output desejável, mas um input inútil, nos modelos corporativos reais.

Segundo problema: Como podemos construir e administrar grandes modelos?

Tratava-se de modelos de dados muito maiores do que os de costume. Certamente cometéria-mos muitos erros devido a esse maior tamanho!

Enfrentamos modelos de dados bem maiores que os costumeiros

Para nos ajudar a lidar com este problema, fomos implementando pequenas ferramentas. Paralelamente, buscamos internacionalmente ferramentas que pudessem nos ajudar. Infelizmente não existiam.

Outros problemas. As dificuldades emanadas do tamanho do modelo rapidamente mostraram ser somente a ponta do iceberg.

Terceiro problema: Onde está o conhecimento?

Quem, na organização, conhece os dados com o nível necessário de objetividade e detalhe? A resposta foi categórica: **NINGUÉM!**

Então, o que fazer? Podemos resolver um problema sem ter o conhecimento adequado? **NÃO.**

Há caminhos paliativos através do treinamento dos usuários ou disseminação na empresa de “administradores de dados por departamento”, etc.?

Paliativos sempre existem, mas quando se passa de repente a problemas no mínimo 10 vezes maiores do que o habitual, dificilmente esses caminhos paliativos sejam suficientes.

Parece o momento de repensar tudo com a maior liberdade, de reconhecer que os antecedentes e a bibliografia podem nos ajudar muito pouco, ou nada.

Onde está o conhecimento válido? Podemos substituir o conhecimento dos dados, que comprovamos que não existe, por outro objetivo, e com o detalhe necessário para nos permitir inferir dele o modelo de dados?

5.

O isomorfismo com a perspectiva

Se dermos uma olhada na história do desenho e da pintura, veremos que no princípio eram muito diferentes da atualidade.

Como se desenhava a princípio? Como desenhariamos intuitivamente? Como as crianças desenham? Procurando conhecer bem o objeto a ser desenhado, se possível tocando-o, conhecendo sua natureza e todos os detalhes possíveis. Então desenhariamos “**como sabemos que é**”.

Como são esses desenhos primitivos? Altamente deformados.

Até que um dia apareceu a **perspectiva**. No Renascimento, alguns pintores e arquitetos, que tinham problemas de desenho muitos maiores que os pintores, começaram a pensar que deveriam mudar o paradigma de “**desenhar como sabemos que é**” para “**desenhar como vemos**”.

Em 1417, em Florença, Filippo Brunelleschi, artista e arquiteto italiano, formalizou um conjunto de regras (princípios de uma geometria descritiva) para poder representar os edifícios em perspectiva, e essas regras são usadas até hoje.

Como todo novo paradigma, houve muita resistência quanto à perspectiva (seus detratores afirmavam que “pretendia substituir a arte de desenhar e pintar por uma técnica banal e nada criativa”), mas quando esta se impôs, substituiu totalmente o paradigma anterior.

Como todo novo paradigma, a perspectiva foi muito resistida

Mas o que a perspectiva tem a ver com a análise de dados?

Quando se decide repensar tudo, quando se pensa com total liberdade, fora de qualquer contexto pré-estabelecido, nada deve ser descartado a priori. A história da perspectiva foi uma boa fonte de inspiração.

A história da perspectiva constituiu uma boa fonte de inspiração

O mais importante é o princípio básico da mudança de paradigma: passou-se de uma abordagem complexa, confusa e subjetiva, a uma abordagem descritiva, simples e objetiva. Se pudéssemos fazer algo parecido com os dados, teríamos percorrido boa parte do caminho!

Visto assim o problema, voltemos à pergunta: “Quem, na organização, conhece os dados com o nível necessário de objetividade e detalhe?” e tratemos de substituí-la por outra, que possa ser respondida com um SIM e que nos ajude na construção do modelo de dados que queremos.

Sobre o que existe conhecimento objetivo e suficientemente detalhado?

Em nossa busca, encontramos que, da mesma forma que não há bom conhecimento sobre os dados, cada usuário tem um conhecimento muito bom das visões desses dados que utiliza.

Tanto na perspectiva como na análise de dados “descrevemos visões”

Voltemos à perspectiva: tanto na perspectiva como na análise de dados, “descrevemos visões”.

Parece prometedor, mas na perspectiva o que desenhamos é diretamente a visão. Aqui, a visão parecia importante, mas o que estávamos buscando era o modelo de dados.

A pergunta óbvia é: **dado um conjunto de visões de usuários, podemos inferir delas um modelo de dados que as satisfaça?**

A pergunta é muito boa, porque leva o problema para o mundo da matemática:

Primeiro: temos que definir um marco de referência.

Nossos elementos de dados (atributos) serão identificados por seu nome e cumprirão, basicamente, com regras simples:

Um atributo sempre será identificado pelo mesmo nome, independente de onde apareça no modelo.

Não haverá dois atributos diferentes com o mesmo nome.

Atribuiremos os nomes de modo a representarem da melhor forma possível o significado de cada atributo.

Segundo: devemos representar a estrutura das visões:

Cada visão envolverá um ou vários atributos, organizados de acordo com uma determinada estrutura.

Mas nunca se deve reinventar a roda.

A esta altura percebemos que, fundamentalmente, Jean Dominique Warnier e nosso amigo Ken Orr, além de Michael Jackson, tinham

avançado muito na descrição das estruturas de dados.

Nem Warnier-Orr nem Jackson pretendiam, com suas estruturas de dados, definirem a base de dados, mas sim a estrutura dos programas, e a maior parte da bibliografia da qual dispúnhamos se referia a programas “batch”. Mas aqueles trabalhos foram um apoio extraordinário para a representação de nossas visões de dados.

Os trabalhos de Warnier-Orr e de Jackson foram um apoio invaluable

Terceiro: precisamos de um procedimento para passar de um conjunto de visões de dados ao modelo em questão.

Agora estamos no mundo da matemática, e a pergunta é: **dado um conjunto de visões de dados, existe um modelo relacional mínimo que as satisfaz?**

Finalmente, agora já temos todas as perguntas relativas a este assunto, ou seja, formulamos rigorosamente o problema.

Neste estado existe um sem-número de ferramentas que podem nos ajudar a resolvê-lo.

No caso concreto que utilizamos, além das ferramentas informáticas normais, técnicas e ferramentas da matemática, da lógica e da inteligência artificial.

Os trabalhos realizados nos conduziram à resolução do problema. Um subproduto importante foi nos introduzirmos em um mundo novo e promissor, o da prototipagem rápida.

Usamos muito a prototipagem em nossas pesquisas e, portanto, viabilizamos seu uso para nossos clientes com o GeneXus.

Utilizamos muito a prototipação em nossas pesquisas

6.

Podemos gerar automaticamente os programas de que precisamos? Alguns deles? Todos? Quais? Que vantagens esses programas teriam com relação aos escritos manualmente?

No começo não pensávamos na geração de programas, mas o fracasso que a informática sempre teve com as chamadas “bases de dados estáveis” foi nos convencendo de que devíamos, em algum momento, encarar este problema, já que de outro modo nossos clientes estariam sempre pressionados por grandes custos de manutenção.

Não tínhamos experiência na geração automática de programas, mas o assunto não era novo:

Quase desde seu início, a informática vem encarando a questão da geração automática de programas.

Durante muito tempo, os geradores de programas foram bastante primitivos

Durante muito tempo, os geradores de programas foram bem primitivos e, acima de tudo, orientados para a geração de relatórios simples a partir de arquivos planos.

Na segunda metade da década de 80, houve grandes avanços e esses geradores começaram a gerar tanto programas “batch” como transacionais e a interagir com bases de dados.

Estes geradores se baseavam em “templates”, ou “esqueletos” que, a partir de uma abordagem do tipo “preencher as lacunas”, para aos

poucos ir adquirindo mais e mais sofisticação, chegavam a resolver uma parte razoável das necessidades de uma instalação.

Ficou claro para nós que a geração automática de programas em algum momento se tornaria um assunto essencial.

Ficou claro para nós que a geração automática de programas iria se tornar um assunto essencial

7. Existem as bases de dados estáveis?

As “bases de dados estáveis” são um assunto recorrente na informática. A ideia era a seguinte:

Se conseguirmos “a base de dados correta” para uma determinada empresa, essa base de dados se manterá estável no futuro. Como consequência, com o passar do tempo nos limitaremos a escrever programas que utilizem essa base de dados.

Se isso não for possível significa que não conseguimos definir “a base de dados correta”.

Muito já se escreveu sobre esse assunto.

Mas a premissa é falsa. Somente se uma organização estiver paralisada ou morta poderá ter um modelo estável!

Então é bom não desperdiçar esforços na busca desses tipos de modelos estáveis, e sim se preparar para trabalhar com os modelos possíveis, reais, instáveis.

8.

Como trabalhar bem com bases de dados instáveis?

O ponto anterior nos sugere esta pergunta, mas, ao analisá-la, acabamos desmembrando-a nas seguintes:

- **Como reorganizar a base de dados quando esta sofrer mudanças estruturais?**
- **Como modificar os programas para que possam funcionar bem com a nova base de dados?**

A primeira destas perguntas nos leva a um profundo estudo de como transformar o conteúdo de uma base de dados com a velha estrutura em um novo conteúdo, com a nova estrutura.

O problema teórico inicial é: **é possível fazer essa conversão sem perda de dados?**

Se a resposta for sim, aparecem outras perguntas: o que devemos fazer para realizar a conversão sem perder dados? E podemos gerar automaticamente os programas que realizem essa conversão?

A segunda pergunta tem uma resposta óbvia: se somos capazes de gerar os programas, vamos gerar novamente todos os programas!

O uso da força bruta nos dá quase sempre uma primeira solução

O uso da força bruta quase sempre nos dá uma primeira solução, mas quando temos milhares de programas, não parece uma boa so-

lução, mesmo considerando o grande e constante aumento de potência do hardware.

Certamente esta resposta não nos convence muito e nos sugere outra pergunta: diante de modificações na base de dados, **podemos determinar que programas se vêm afetados para poder gerá-los novamente?**

Se conseguirmos isso, será excelente, mas a situação nos leva a uma nova pergunta: **existem programas gerados para a velha estrutura que funcionam corretamente com a nova e que, ainda assim, poderiam ser substituídos agora por outros, mais eficientes?**

9. Os deuses também brincam: as “tabelas ampliadas”

Em um domingo de agosto de 1986, em Nova York, quando ainda não pensávamos em montar uma empresa e um produto, e sim em empacotar um conjunto de descobertas científicas e tecnológicas e licenciá-las para terceiros, descobrimos algo esteticamente formidável: as “tabelas ampliadas”.

Não vamos dedicar muito espaço a este assunto, que é bem conhecido na comunidade GeneXus, já que este trabalho pretende pôr ênfase nas perguntas e não nas respostas. Mas vamos dizer o seguinte, que contém algumas simplificações não essenciais:

O que é uma “tabela ampliada”? Para cada registro de uma determinada tabela, existe um registro virtual formado pela concatenação

do registro original e todos aqueles de outras tabelas da base de dados que estejam direta ou indiretamente determinados por ele.

O conjunto desses registros será chamado de “tabela ampliada” da tabela original e aquela, “tabela base” associada à tabela ampliada.

Um domingo de agosto de 1986 descobrimos as tabelas estendidas

Qual é a importância das tabelas ampliadas?

As descrições expressas em termos de tabelas ampliadas se mantêm vigentes através das modificações estruturais da base de dados.

Como consequência, por mais que alguns programas deixem de ser corretos ou ótimos, devido a modificações na base de dados, as descrições das visões de usuários do GeneXus continuam válidas e, então, é possível propagar automaticamente as modificações, identificando os programas não válidos e gerando-os novamente a partir das descrições originais de suas visões.

Nunca tivemos dúvidas de que esta descoberta era esteticamente formidável. Com o tempo, a realidade nos mostrou que é muito mais do que isso.

Nunca tivemos dúvidas de que esta descoberta era esteticamente formidável

10. GeneXus

Nossa intenção original era licenciar a tecnologia obtida para grandes jogadores. A dura realidade nos provou que aquele era um propósito ingênuo: **que credibilidade podem ter ideias tão avançadas diante de grandes jogadores tecnológicos dos países mais desenvolvidos quando veem de um país que não tem uma forte tradição como produtor de tecnologia?**

Que credibilidade podem ter ideias tão avançadas quando procedem de um país que não tem tradição tecnológica?

O dilema era decidir entre publicar as descobertas realizadas e dar por encerrada a pesquisa ou criar uma empresa, um produto e tentar comercializá-lo, começando pelo Uruguai e os países vizinhos, para depois continuar com o resto do mundo.

Escolhemos a segunda opção. Consideramos que, através do tempo, se nosso produto realmente for útil para os nossos clientes, se a quantidade de clientes aumentarem constantemente, mesmo que não seja a grande velocidade, se conseguirmos dar suporte oportunamente às novas tecnologias que forem surgindo e se mantivermos um comportamento empresarial espartano, teríamos sucesso.

Propusemos-nos esse objetivo com o maior compromisso e, no final de 1988, fundamos a Artech, batizamos o nosso produto com o nome de GeneXus e decidimos lançar sua primeira versão no segundo semestre de 1989.

Algumas das perguntas que surgiram naquele momento:

Para que plataforma o GeneXus deverá gerar aplicações em uma primeira etapa?

O realista seria escolher inicialmente uma única plataforma e nos especializarmos nela até que conseguíssemos um volume empresarial mínimo que nos permitisse encarar outras.

Escolhemos o IBM AS/400.

Em que plataforma o GeneXus deverá funcionar em uma primeira etapa?

Era preciso uma plataforma que não requeresse grandes investimentos, que fosse eficiente e que assegurasse um desenvolvimento permanente.

Também era desejável a maior independência possível da plataforma para a qual geraríamos as aplicações, para mais tarde viabilizar a geração para outras plataformas.

Era desejável a maior independência possível da plataforma para a qual geraríamos as aplicações

Escolhemos PC com sistema operacional DOS.

Que problemas o GeneXus deverá resolver de imediato e quais serão adiados para o futuro?

Para nós era claro que, no princípio, ainda tínhamos carências teóricas e inconvenientes práticos que nos impediam de gerar aplicações completas.

Um objetivo era gerar automaticamente tudo o que fosse possível.

Um segundo objetivo era que pudéssemos manter automaticamente tudo o que gerássemos.

Um segundo objetivo era poder manter automaticamente tudo o que gerássemos

Por isso, descartamos o plano de gerar partes de programas e deixar ao desenvolvedor a tarefa de completá-los, dado que não poderíamos manter automaticamente o que o desenvolvedor escrevesse a mão.

Não sabíamos como descrever programas procedurais, como por exemplo, processos “batch” ou rotinas casuísticas que não pudessem ser derivadas das transações.

Poderíamos introduzir uma linguagem de 4a geração, que resolveria os problemas anteriores, mas isso significaria desistir da manutenção automática desses programas. Decidimos não fazer isso.

Então resolvemos implementar aquilo que podíamos gerar e manter automaticamente sem restrições: transações, consultas e relatórios simples.

Estimamos que o primeiro GeneXus gerava e mantinha automaticamente 70% dos programas de uma instalação e se encarregava do design, da geração e da manutenção automáticos da base de dados. O resto requeria forte pesquisa adicional.

Estimamos que o primeiro GeneXus gerava e mantinha automaticamente 70% dos programas

Poderíamos gerar e manter automaticamente 100% das aplicações?

Era nosso objetivo e estávamos trabalhando para isso, mas achávamos que tínhamos bastante tempo para conseguir.

Gerar e manter automaticamente 70% dos programas parecia um grande sucesso: sómente alguns geradores podiam gerar coisas parecidas, mas nenhum podia oferecer manutenção automática.

Para nós era claro que a manutenção automática era uma característica muito importante

e única do GeneXus, mas pensávamos que os clientes contratavam o GeneXus por seu forte aumento de produtividade no desenvolvimento, enquanto que a manutenção automática era uma característica nova, inesperada e vista com certo ceticismo fora da Artech.

Pensávamos que os clientes contratavam GeneXus unicamente por seu forte aumento de produtividade no desenvolvimento

A realidade nos surpreendeu quando vários clientes nos expressaram suas opiniões:

“Gerar automaticamente 70% dos programas nos ajuda muito, e damos muito valor a isso.”

“Ter que escrever a mão 30% dos programas é uma restrição que aceitamos confiando que vocês irão eliminá-la no futuro.”

“No entanto, o que é inaceitável para nós é que ter que manter manualmente aqueles programas que o GeneXus não gera.”

Como satisfazer nossos clientes? Como podemos ter a certeza de poder gerar aplicações completas? Com uma linguagem procedural.

Nossa pergunta foi: **podemos construir uma linguagem procedural cujas descrições (programas fonte nessa linguagem) não se tornem inválidas diante de modificações estruturais na base de dados?**

A resposta foi SIM e isso nos permitiu lançar o primeiro GeneXus completo.

A partir daí, as perguntas não pararam de aparecer...

Podemos gerar aplicações para outras plataformas e para outras arquiteturas?

Podemos identificar padrões em nossas descrições e, a partir deles, gerar automatica-

mente os objetos GeneXus que respondem a elas?

Podemos estender o GeneXus para além da Artech? Podemos viabilizar para que a comunidade o faça?

... e muitas mais.

Formular novas perguntas é uma tarefa permanente e cada dia mais pessoas participam dela: a equipe de pesquisa e desenvolvimento, toda a Artech, as empresas de software, os clientes em geral, toda a comunidade GeneXus.

**Formular novas perguntas é
uma tarefa permanente e a
cada dia mais pessoas
participam dela**

11. Respostas

Em todos estes anos de trabalho, surgiram muitas perguntas, que em geral trouxeram respostas, muitas respostas. No entanto, podemos sintetizá-las em um princípio:

É possível “descrever” em vez de “programar”!

E uma convicção:

Nunca devemos perder nossa liberdade de pensar!

12.

Agradecimentos

Queremos agradecer especialmente a muitas pessoas e empresas: à nossa equipe, aos nossos clientes, a toda a comunidade GeneXus.

Mas também com toda a humildade, queremos agradecer àqueles que, sem serem clientes e sem integrar a nossa comunidade, e às vezes sem acreditar no GeneXus, nos apresentam permanentemente novas dificuldades e desafios, porque sempre aprendemos com essas dificuldades e desafios.

O mundo é maravilhoso, e tudo aquilo que aumenta nossa capacidade de conhecê-lo é um privilégio e uma grande oportunidade!

labelas
geração
muitas companhia
como
manutenção
tempo
grandes

cada
fazer
pergunta
conjunto
tempo
sempre
tabelas

como
muito
automaticamente

fizemos
modelo
comunidade

problemas
aplicações
linguagem
informação
perspectiva

automaticamente
comunidade
descrições
princípio
mundo
mundo
muito
mundo
Como
resolução
objetivo
possível

gerar
modelos
problema
clientes
princípio
mundo
muito
mundo
Como
resolução
objetivo
possível

grandes
bases
tabelas

muitas

programas

GeneXus