

## Evolução dos objetivos de Genexus Orientado a Quarta Dimensão

Preparado por: Breogán Gonda e Juan Nicolás Jodal [1]  
Artech, fevereiro de 2009

O que podemos dizer, 15 anos depois, dos objetivos de GeneXus? [2]: Como tem evoluído?, Como tem sido o seu cumprimento?

Os objetivos de GeneXus têm evoluído constantemente e sempre orientado a outros mais exigentes na medida em que nossas investigações, o desenvolvimento tecnológico geral e um melhor conhecimento da realidade vão sendo aconselhados.

Os fatos mostram um grau muito bom de cumprimento dos objetivos.

### As Quatro Dimensões

Tem se conseguido uma boa maneira de visualizar os objetivos e de avaliar os cumprimentos introduzindo quatro dimensões:

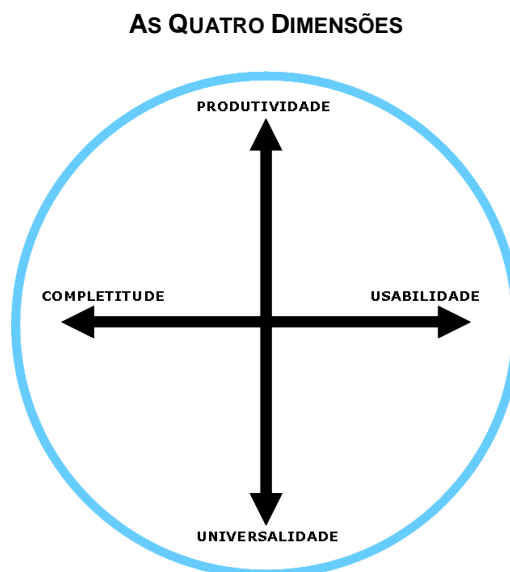
1. **Compleitude** (Será de um 100% se tanto a base de dados como todos os programas necessários são gerados por GeneXus)
2. **Produtividade** (É o aumento potencial de produtividade obtível com uma boa utilização de GeneXus com relação a que obteriam bons desenvolvedores utilizando manualmente linguagens de programação comuns (como COBOL ou RPG originalmente, Java ou C#hoyo).
3. **Universalidade** (Será de 100% se podem ser geradas aplicações para qualquer plataforma viva).
4. **Usabilidade** (Quantifica a facilidade de uso: será 100% se qualquer pessoa puder utilizá-lo, sem maior capacitação específica).

É interessante ver como evoluíram os objetivos e os alcances desde a primeira versão de GeneXus liberada em fins de 1989.

Sempre é difícil representar quatro dimensões sobre um plano. A seguinte, é uma representação razoável e ilustrativa (embora não rigorosa).

Representamos cada dimensão por uma flecha, segundo o seguinte diagrama e, para cada uma destas flechas atribuímos o valor que a avaliação de GeneXus lhe designa, segundo essa dimensão.

Arbitrariamente colocamos as dimensões da seguinte maneira: **Compleitude** apontando para a esquerda, **Produtividade** para cima, **Universalidade** para baixo e **Usabilidade** para a direita, de acordo com o seguinte gráfico.



## Objetivos no momento da liberação da primeira versão de Genexus

Quando a primeira versão de GeneXus foi liberada já tinha importantes alcances em relação à Completitude, à Produtividade e à Usabilidade.

### Completitude:

Na primeira versão teve-se como objetivo gerar 70% dos programas. Para cada um deles eram gerados 100% do código. Desta maneira se evitava a necessidade de modificar manualmente os programas gerados.

Esta característica foi sempre considerada essencial: dado que o sistema tem pleno conhecimento para gerar tanto a Base de Dados, como aqueles programas (os que caem dentro dos 70% que GeneXus era capaz de gerar), também tem conhecimento para efetuar a manutenção automática de tudo o que gera (Base de Dados –estrutura e conteúdo- e programas).

Era claro, no entanto, que o ideal seria gerar 100% dos programas. Se isso fosse conseguido, GeneXus seria capaz de efetuar automaticamente a manutenção de todo o sistema, com uma diminuição dramática de custos (tempo e dinheiro). A tecnologia de que se dispunha nesse momento não o permitia.

### Productividade

A Produtividade aumentava de maneira muito importante, dado que o desenvolvedor não precisava dedicar sua atenção a um conjunto de tarefas tradicionais: análise de dados, projeto da base de dados, projeto e escritura dos programas.

As provas básicas dos sistemas consistem em verificar a correção das especificações pela via de protótipos vivos, completos, muito fáceis de usar e oportunos.

O objetivo desde o primeiro momento foi de um aumento potencial da produtividade em 500% sobre programar manualmente com as linguagens disponíveis (no caso RPG e COBOL).

### Usabilidade

A Usabilidade aumentava comparando-a com a de programar manualmente (naquela primeira versão sobre o sistema operativo IBM OS/400, o Sistema de Gerência de Base de Dados nativo do computador IBM AS/400, sua linguagem de comandos e uma linguagem de programação que podia ser RPG ou COBOL).

Por que aumentava a usabilidade? Porque o desenvolvedor não precisava nenhum conhecimento detalhado de ditos elementos. Permitia então, que o desenvolvedor se independentizasse destes elementos de baixo nível e pudesse dedicar sua atenção a entender e resolver conceitualmente o problema do cliente:

dedicar-se a resolver os problemas reais e não os que apresentam as limitações da tecnologias que utiliza.

Não era fácil quantificar este aumento da usabilidade, mas era importante porque permitia tanto aos velhos usuários, acostumados à tecnologia obsoleta, como aos novos usuários com pouca ou nenhuma experiência; usar a nova tecnologia imediatamente.

Os objetivos se cumpriram. Qual foi a reação dos clientes?. Por que contrataram GeneXus?. Quais foram as características que mais valorizaram?

Os principais elementos que levaram à contratação pelos primeiros clientes foram:

- GeneXus viabilizava o desenvolvimento de aplicações para o novo computador IBM AS/400 para técnicos com muito pouco conhecimento desse novo computador e da sua tecnologia.
- GeneXus permitia um aumento muito grande na produtividade dos desenvolvedores.
- **Em particular não foi considerada como uma vantagem essencial por parte dos potenciais clientes o fato de que GeneXus promettesse a manutenção automática de todo o que gerava.** Realmente ninguém avaliava seriamente esta vantagem porque ninguém acreditava que a manutenção automática fosse possível!

Usando GeneXus um pouco, os clientes mantiveram a avaliação positiva anterior: técnicos com muito pouco conhecimento do AS/400 desenvolviam sem dificuldades e com grande produtividade seu trabalho.

**No entanto a maior mudança foi que os clientes passaram a avaliar como uma vantagem fundamental que GeneXus mantinha automaticamente tudo o que gerava.**

Ao mesmo tempo, GeneXus só pretendia gerar 70% dos programas necessários. Este fato havia sido acolhido como razoável no momento da contratação, mas depois passava a ser considerado como uma grande limitação para os clientes. Por quê? Porque agora estavam claras várias vantagens de GeneXus:

- Forte aumento da produtividade da programação automática sobre a manual.
- Enorme vantagem da manutenção automática sobre a manutenção manual.
- Só o que era gerado automaticamente poderia ser mantido também automaticamente.

Como consequência de tudo isto, os usuários quiseram evitar o desenvolvimento manual dos 30% dos programas restantes. Mas, o mais importante, a necessidade de manter manualmente para sempre todos os programas desenvolvidos manualmente.

Como resultado do anterior ocorreram duas coisas importantes:

- Todos compreenderam muito bem que em nenhum caso deviam modificar manualmente os programas gerados, para preservar a capacidade de manutenção automática.
- Os clientes exerceram sobre a ARTech uma forte pressão para conseguir que GeneXus gerasse 100% dos programas.

Gerar 100% dos programas era um objetivo muito bom por várias razões. Algumas destas razões eram muito claras:

Manutenção automática de 100% da aplicação. Aumento da produtividade consequente.

Outras só ficaram claras anos depois:

Portabilidade das aplicações de umas plataformas para outras.

Comercialização de Bases de Conhecimento, etc.

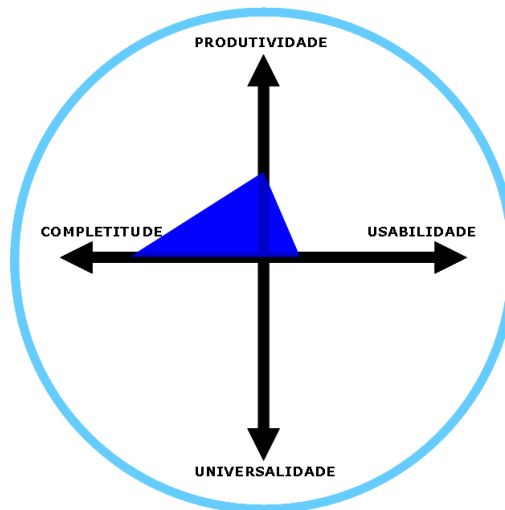
Mas o objetivo de gerar e manter automaticamente toda a aplicação não era um objetivo fácil de conseguir (nenhuma outra ferramenta o fazia – nem o faz hoje – em todo o mundo). Seria possível?

Para gerar 100% de uma aplicação, primeiro é necessário poder descrevê-la total e fielmente. Naquele momento GeneXus não tinha o suficiente poder expressivo para isso.

Como poderia ser adicionado rapidamente poder expressivo a GeneXus? GeneXus era 100% declarativo, se se agregasse uma linguagem de programação – por exemplo, uma linguagem de 4ª Geração – ganharia-se poder expressivo. Mas ao mesmo tempo, perderia-se a capacidade de manter automaticamente tudo o que se gerava, porque nas linguagens procedurais [3] conhecidas, os programas fonte não se mantinham válidos ante modificações da Base de dados.

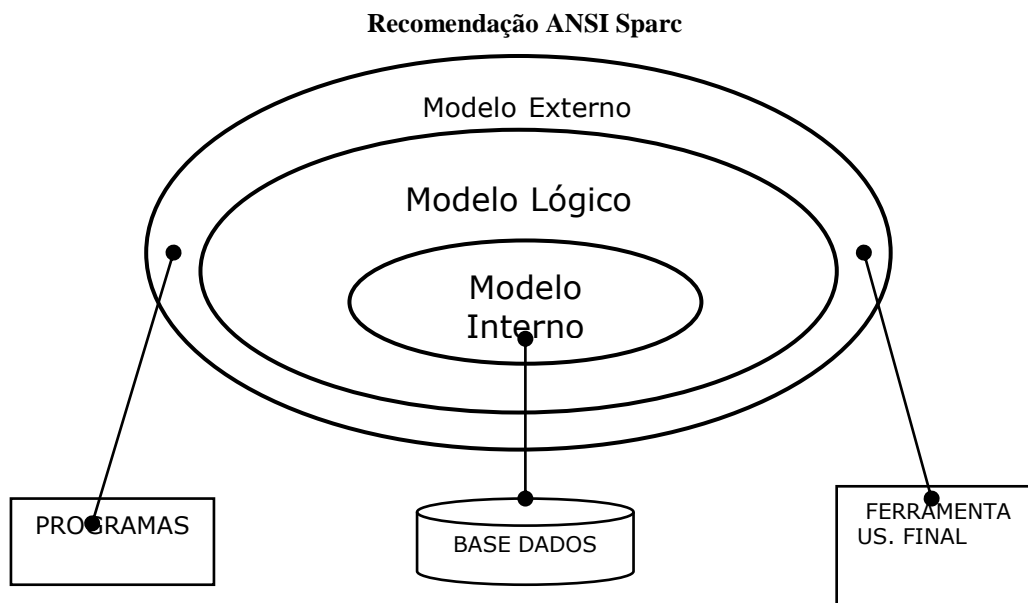
O que se precisava era uma linguagem processual, cujos programas fonte se mantivessem válidos frente modificações na base de dados (e que fosse do maior nível possível).

Não se conseguiu nenhuma solução imediata e se colocou o assunto em primeiro lugar na lista de investigação.

**OBJETIVOS NO MOMENTO DA PRIMEIRA VERSÃO DE GENEXUS (1990)****Objetivos de 1992: 100% de Completitude**

Depois de considerar e descartar muitas alternativas se chegou a uma colocação simples do problema:

**Solução através do Sistema de Gerência de Base de Dados.** A solução mais razoável (e independente de GeneXus ) era a implementação por parte dos fabricantes de Sistemas de Gerência de Base de Dados da recomendação ANSI Sparc do Esquema de três modelos (Externo, Lógico, Interno). Todas as aplicações interagiriam com o Modelo Externo, que não se altera frente a modificações na Base de Dados.



As tendências que se percebiam na época tornavam muito pouco provável esta solução: o mercado dos Sistemas de Gerência de Base de Dados, onde haviam atuado múltiplas empresas com diferentes produtos, competindo permanentemente em um clima de inovação e entusiasmo, adotava como standard o SQL em uma versão nada inovadora.

O standard implicou uma quase congelação das funcionalidades e a sobrevivência de uns poucos de todos aqueles fabricantes.

Conclusão: não havia (nem tem) um caminho por aqui.

**“Base de Dados Estáveis”.** Muitos teóricos postularam as “bases de dados estáveis, bem desenhadas a priori” e muito se falou sobre isso. (Obviamente se a base de dados é estável o problema que estamos tentando resolver não existe: como não tem modificações estruturais na base de dados, não tem repercussões delas sobre os programas).

No entanto, estes conceitos não se compadecem com a realidade: só podem existir Bases de Dados Estáveis em empresas ou organizações decadentes, que perderam toda capacidade de inovação.

Conclusão: se trata de uma abordagem que não tem nenhum contato com a realidade prática.

**Linguagem Procedural cujos programas sejam independentes da estrutura da Base de Dados.** A terceira opção era desenhar e desenvolver uma linguagem Procedural tal, que a validade de seus programas fonte não fosse afetada pelas mudanças na base de dados.

Adotamos esta abordagem e, algo que é muito difícil ou impossível, tomado isoladamente, resultou totalmente possível em um ambiente baseado em conhecimento como GeneXus: por que colocar manualmente nos programas elementos (nomes de tabelas, arquivos e similares) que podem ser inferidos automaticamente (no momento oportuno, que neste caso é o da geração dos programas)?

A linguagem Procedural de GeneXus age só sobre o Modelo Externo (cujos elementos não são afetados pelas mudanças na Base de Dados) e não utiliza elementos físicos de baixo nível como tabelas, arquivos, etc., e seus programas fonte são imunes às modificações estruturais da Base de Dados.

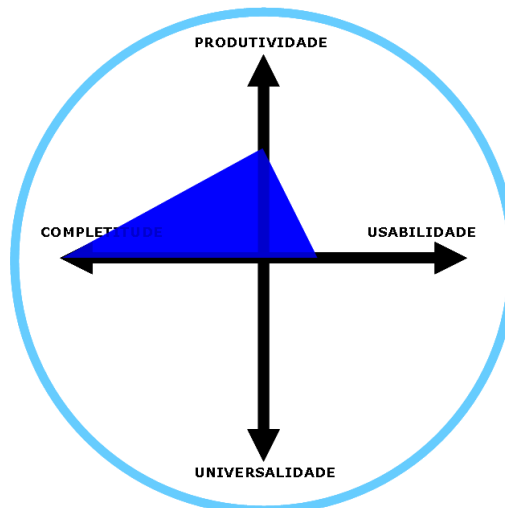
A inclusão desta linguagem Procedural de alto nível permitiu resolver com GeneXus 100% do problema.

Agora GeneXus era capaz de gerar e manter automaticamente a Base de Dados (estrutura e conteúdos) e 100% dos programas.

Desta maneira se conseguia 100% da Completitude e se assumia o compromisso de mantê-lo sempre no futuro. Isto aumentava substancialmente as vantagens de usar GeneXus. Os usuários assim o entenderam.

De todas formas, GeneXus era uma ferramenta que gerava basicamente aplicações para uma só plataforma: o computador IBM AS/400. No entanto, não havia nenhuma restrição teórica para generalizá-lo a outras plataformas.

#### OBJETIVOS DE 1992: 100% DE COMPLETITUDE



#### Objetivos de 1995: suporte da arquitetura Cliente / Servidor

Em 1995 ocorreu algo vastamente esperado e postergado: uma forte expansão da arquitetura Cliente / Servidor.

A ARTech lançou geradores Cliente / Servidor para os Sistemas de Gerência de Base de Dados mais importantes da época: IBM DB2, IBM DB2 para AS/400, Informix, Microsoft SQL Server e Oracle.

Estas novas capacidades tiveram uma acolhida muito boa e o segmento de clientes usuários da arquitetura Cliente / Servidor passou a ser rapidamente o de maior crescimento.

**Paralelamente, de repente e sem que muitos o esperassem, a liberação para fins comerciais da Internet constituiu um tremendo êxito e a informática toda começou a se transformar de uma maneira vertiginosa.**

Até então os sistemas eram previsíveis e estruturados, para alguns poucos milhões de usuários em todo o mundo. Estes usuários os utilizavam sem nenhum grau de liberdade e depois de ser especialmente treinados para isso.

Os desenvolvedores – relativamente poucos em todo o mundo – mantinham para si todas as decisões e os graus de liberdade.

Frente a essa realidade tradicional, aparecia a Internet, acessível desde o começo por muito mais gente de qualquer parte do mundo, que não podia ser treinada e com um nível de liberdade muito maior. GeneXus muito rapidamente lançou seu primeiro Gerador para Web.

Qual era a situação em fins de 1995?

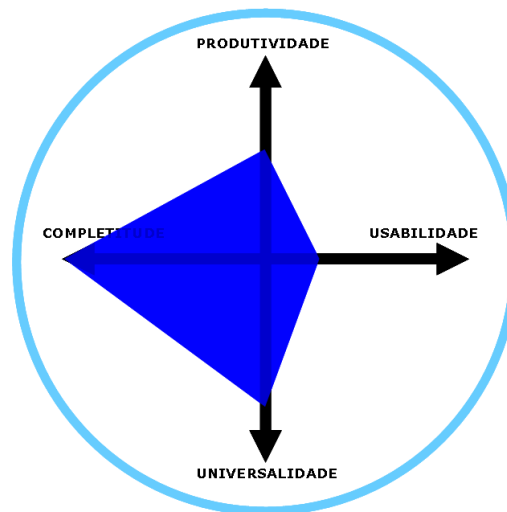
GeneXus mantinha sua **Compleitude**: 100% dos programas eram gerados e mantidos automaticamente.

A **Produtividade** se mantinha muito alta (potencialmente ao redor de 500% da que se poderia chegar a obter programando manualmente).

A **Universalidade** tinha se incrementado substancialmente ao incorporar as arquiteturas Cliente / Servidor e Web. O que ficava fora do alcance de GeneXus? Basicamente aplicações para mainframes (mas, já era notório que não existia uma tendência a desenvolver novas aplicações para eles).

A **Usabilidade** se mantinha nos níveis anteriores.

#### OBJETIVOS DE 1995 SUPORTE DA ARQUITETURA CLIENTE / SERVIDOR



#### Objetivos 2001: aumento radical da Universalidade

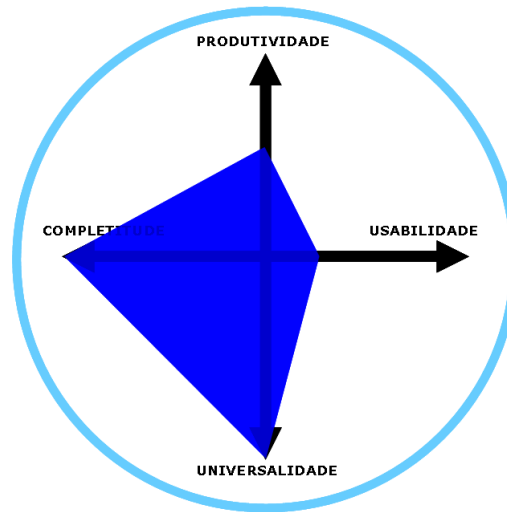
Na segunda metade da década dos 90 apareceu o conceito de “plataformas” de execução de aplicações n-camadas, orientadas para a rede e que ofereciam algo que os usuários pediam: atualização automática das versões dos programas nos PCs clientes.

As duas plataformas rivais Java e .NET rapidamente dividiram uma boa parte do mercado (que com sua competência permanente fizeram crescer). Java e .NET compartilhavam o mercado fundamentalmente com a arquitetura Cliente / Servidor, enquanto começava o crescimento da arquitetura Web e diminuía substancialmente a participação dos mainframes.

GeneXus lançou muito oportunamente seus geradores para Java e .NET e paralelamente aperfeiçoava fortemente seus geradores para a arquitetura Web.

Com isto manteve seus bons indicadores no que se refere a **Compleitude, Produtividade e Usabilidade** e aumentou substancialmente sua **Universalidade**: agora GeneXus gerava para todas as plataformas para as que, realmente estavam sendo desenvolvidas aplicações novas.

#### OBJETIVOS 2001 AUMENTO RADICAL DA UNIVERSALIDADE



#### Objetivos de 2004: aumento radical da Productividade

A informática tem mudado muito desde a liberação da Internet: existem muitos mais usuários (muitas centenas mais). **Estes usuários, em geral, não são treináveis.**

As Bases de Dados já não são só as Bases de Dados físicas que temos dentro da empresa, senão Bases de Dados Estendidas, que envolvem clientes, fornecedores, serviços Web públicos ou privados, etc. Simultaneamente, nossas Bases de Dados são acessadas por outros, convenientemente autorizados, mas das formas mais diversas.

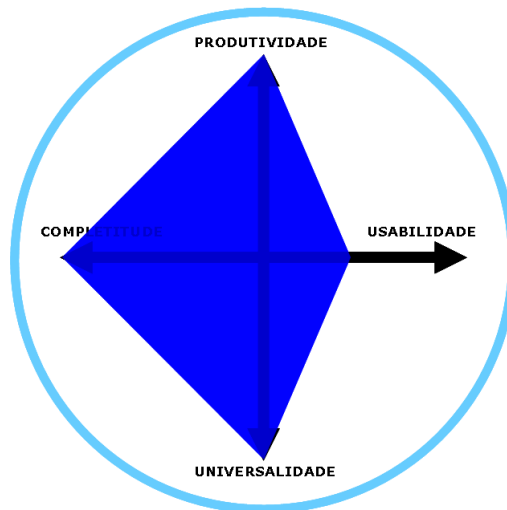
Os dispositivos que agem como terminais têm se diversificado acrescentando Hand Helds, Palms, Pocket PCs, telefones celulares, etc.

As redes têm se generalizado, incluindo agora velocidades muito maiores, conexões sem fio de curta distância (Wi Fi), conexões sem fio de distância média (Wi Max), conexões telefônicas de alta velocidade, associadas às redes de telefonia celular (GPRS, EDGE).

Mas o fundamental é que agora nossas aplicações devem servir a muitos milhões de usuários potenciais, que têm um grau de liberdade muito maior que de costume e que não são treináveis. Como consequência as aplicações são muito mais complexas, precisam ser muito mais estudadas e elaboradas, de maneira a esconder toda essa complexidade e apresentar-se frente ao usuário final com interfaces muito simples, muito intuitivas, muito fáceis de usar (por usuários geralmente não catalogados e, por isso, não treináveis).

Ante isto a ARTech concluiu que um aumento potencial de produtividade de 500% sobre a produtividade de desenvolver a mão em uma linguagem de baixo nível comum (como Java ou C#, por exemplo) não será mais suficiente em um futuro próximo. Então assumiu-se o objetivo de aumentar a produtividade substituindo o tradicional 500% por 2000%, o que se consegue com dois aumentos de produtividade de 100% cada um sobre a da versão imediatamente anterior, o primeiro na versão liberada a fins de 2005 e o segundo na versão a ser liberada em meados de 2007.

Paralelamente se seguiram cuidando as outras dimensões e, em particular a Universalidade, acrescentando oportunamente suporte para diferentes plataformas ou elementos de plataformas que foram surgindo e passaram a ser importantes no mercado como, por exemplo, PostgreSQL, MySQL, AJAX.

**OBJETIVOS DE 2004: AUMENTO RADICAL DA PRODUCTIVIDADE****O futuro: aumento radical da Usabilidade**

E agora? Qual é a próxima inovação? O que podemos esperar de GeneXus?

GeneXus resolveu muito bem suas três primeiras dimensões:

A **Completitude** e a **Universalidade** chegaram ao ótimo e haverá que trabalhar permanentemente para mantê-las ante as inovações tecnológicas que se aproximam.

**Produtividade:** o aumento da produtividade que está sendo adicionado é tal, que permitirá desenvolver com GeneXus - em tempo hábil e com custos razoáveis – as aplicações cada vez mais complexas que os diferentes usuários requererão nos próximos anos. Isto será muito importante porque não era mais possível fazê-lo a mão com linguagens de programação comuns como Java e C#.

Qual é, em resumo, a situação com respeito à Quarta Dimensão (a usabilidade)?

A **Usabilidade** é boa, mas não se pode comparar seu nível com o das outras três dimensões: GeneXus é uma ferramenta para desenvolvedores com uma boa formação algorítmica, por exemplo, porque seu componente Procedural se mantém importante. Para aumentar de forma substancial sua Usabilidade, para que uma pessoa qualquer, com boa formação geral, possa usá-lo em benefício de suas atividades normais sem uma capacitação específica cara deve facilitar muito mais ainda o seu uso.

Mas qual é a situação atual? A Usabilidade de GeneXus tem sido importante desde o princípio e hoje continua sendo como fica claro no seguinte:

Não é imprescindível conhecer, quando se começa um desenvolvimento, qual será a plataforma de execução, (Hardware, Sistema Operativo, Sistema de Gerência de Base de Dados, Arquitetura, Linguagem de Programação a utilizar).

Os desenvolvedores não precisam nunca um conhecimento detalhado da plataforma de execução. Em particular, esta característica, facilita a reciclagem de velhos desenvolvedores acostumados a tecnologias obsoletas e a incorporação de novos desenvolvedores sem experiência.

O Cliente ganha muito em liberdade porque em qualquer momento pode resolver a mudança da plataforma de execução e transformar com GeneXus suas aplicações na nova.

Novas tecnologias podem aparecer em meio a um grande projeto e serem utilizadas imediatamente sem traumas.

GeneXus integra automaticamente os diferentes elementos de uma aplicação, garante sua permanente consistência e mantém uma documentação completa, ativa e sempre atualizada.

A correção dos sistemas se verifica provando as especificações via prototípica viva, completa, oportuna.

Tudo isto é muito importante e a maior parte destas características são únicas. Onde está, então, o problema? Não existe nenhum problema, mas existe sim uma restrição: GeneXus deve ser utilizado sempre por desenvolvedores profissionais.



É defendível no médio prazo a idéia de que todas as aplicações devam ser construídas por desenvolvedores profissionais?

Nossa opinião é que não: a necessidade de utilizar unicamente desenvolvedores profissionais é consequência das limitações da tecnologia!

Levantando essas limitações podemos pensar em um aporte muito maior de usuários da informação, de usuários que conhecem o negócio – o assunto que seja – e que não conhecem (nem pretendem conhecer) os detalhes de baixo nível que hoje são necessários para construir um sistema:

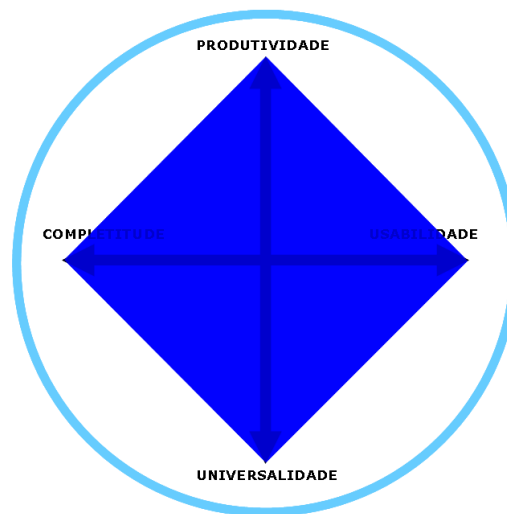
Cada vez será mais importante o conhecimento que se tenha sobre o problema que se quer resolver e menos o da tecnologia necessária para resolvê-lo. Cada vez os sistemas serão mais diversificados: em pouco tempo os sistemas comerciais baseados em contabilidade, compras, vendas, nômima, estoque ou bem ERPs, CRMs, etc., constituirão uma proporção menor das aplicações que os usuários (os muitos centenas de usuários) necessitam. Onde está o conhecimento para construir as novas aplicações?, pois muito mais nestes usuários que nos desenvolvedores profissionais.

Para isso GeneXus deverá evoluir muito tornando-se mais “usável”: muito mais fácil de utilizar por usuários que precisam resolver um problema (que conhecem bem) e não são desenvolvedores profissionais.

Em particular, é necessário tornar GeneXus mais amigável, diminuindo todo o possível o nível de abstração necessário para utilizá-lo, ocultando a complexidade e, em particular, aumentando seu componente declarativo e adicionado, provavelmente, componentes gráficos intuitivos e fáceis de usar, de maneira de minimizar a necessidade de utilizar seu componente Procedural.

**Esta é a grande tarefa para os próximos anos!**

#### O FUTURO: AUMENTO RADICAL DA USABILIDADE



---

<sup>1</sup> **Breogán Gonda e Juan Nicolás Jodal** são Engenheiros em Computação, formados pela Faculdade de Engenharia da Universidade da República, Uruguay.

Desenvolveram uma ampla atividade internacional de consultoria, docência e investigação.

Suas áreas principais de investigação são Bases de Dados Relacionais, Inteligência Artificial, tratamento automático do conhecimento e desenvolvimento automático de Aplicações.

Receberam o Prêmio Nacional de Engenharia 1995, outorgado pela Academia Nacional de Engenharia do Uruguai.

São sócios fundadores e, respectivamente, Presidente e Vice-Presidente da ARTech, empresa que desenvolve e comercializa em todo o mundo o produto GeneXus.

-----

---

<sup>2</sup> **Genexus é um sistema baseado em conhecimento** e, em particular, é um sistema que se embasa em uma muito boa capacidade de tratamento automático do conhecimento dos sistemas de negócios.

Com uma rigorosa fundação matemática, GeneXus, partindo de visões de usuários, captura todo o conhecimento contido nelas e o sistematiza em uma Base de Conhecimento.

A base de conhecimento de GeneXus tem uma grande capacidade de inferência lógica: em qualquer momento é capaz de proporcionar qualquer conhecimento que foi armazenado nela ou que pode ser inferido logicamente de aqueles armazenados na mesma. Baseado nesta capacidade de inferência é capaz de projetar, gerar e manter, de forma 100% automática, a base de dados e os programas necessários para satisfazer todas as visões de usuários conhecidas em um determinado momento.

-----

<sup>3</sup> **Procedural / Procedural:** A palavra espanhola correta é “Procedural”, no entanto na informática não é usada e sim substituída por Procedural, que é a palavra inglesa que se utiliza em todo mundo. Neste trabalho utilizaremos “Procedural”.

---