

# A gênese do GeneXus: algumas reflexões

por Breogán Gonda \*  
Copyright © GeneXus, Fevereiro 2019

Whitepaper



GeneXus™

**The  
power  
of  
doing.**

# A gênese do GeneXus: algumas reflexões

Por Breogán Gonda \*

Copyright © GeneXus, Fevereiro 2019

**Em 1984**, uma grande empresa no Brasil nos encarregou de reprojeter totalmente sua área de TI. O cliente queria desenvolver todos os seus sistemas interagindo com um único banco de dados corporativo.

O desafio era grande: na época, todos falavam sobre sistemas e bancos de dados corporativos, mas a realidade era muito diferente e todas as empresas continuaram a usar vários "subject databases".

Cada subject database foi usado para suportar uma pequena família de aplicativos. Cada subject database foi atualizado com total independência dos outros. Como consequência, a consistência geral era impossível. As empresas sabiam disso e, portanto, não combinavam dados de diferentes subject database.

Resumindo: havia um sistema de computação operacional que operava sob certas condições restritivas, mas não havia computação corporativa.

O cliente nos apresentou um grande desafio: um único banco de dados centralizado para atender a todas as suas necessidades.

O cliente pensava, com bom senso, que poderia obter qualquer informação que fosse necessária, em qualquer momento, a partir daquele banco de dados corporativos.

Não parecia haver nenhuma restrição nos recursos necessários, porém havia um tempo máximo de desenvolvimento de um ano.

A tarefa testou nossas convicções, nossa experiência, nossas metodologias e nos confrontou com um conjunto de problemas adicionais. Todos os obstáculos foram superados e o projeto terminou com sucesso.

Tendo enfrentado tantas dificuldades nos fez aprender muitas coisas e nos encorajou a continuar. A questão era fazer projetos mais semelhantes ou construir sua própria metodologia e uma ferramenta para apoiá-lo? Este último foi o caminho escolhido.

Participamos de vários projetos para diferentes empresas, enquanto continuamos nossa pesquisa e, finalmente, lançamos nosso primeiro GeneXus em 1989.

Hoje, 34 anos depois daquele projeto, chegamos a um conjunto de reflexões sobre um evento que foi importante para o nosso desenvolvimento tecnológico e empresarial.

Tivemos um importante contrato para dar um grande passo adiante, constituíamos uma equipe muito boa e, embora muitas pessoas em todo o mundo falassem ou escrevessem muito sobre computação corporativa, não havia, ou pelo menos muito poucas, realizações de sucesso na área.

Encaramos a questão com grande seriedade e responsabilidade, o que nos levou a enfrentar um conjunto de problemas:

## PROBLEMAS

**Base de dados.** Assumindo que fomos bem-sucedidos: nossa primeira estimativa indicava que o banco de dados necessário

teria mais de 500 tabelas (no momento da conclusão da nossa consultoria, ele tinha 750), nossas metodologias funcionariam para um problema tão grande? Havia que dar respostas a questões importantes::

**Exploração da solução.** O cliente queria que sua alta gerência e / ou assistentes pudessem, em quase todos os casos, obter as consultas de que precisavam a qualquer momento sem a ajuda de especialistas.

Se disponibilizaria a linguagem SQL que, do ponto de vista do usuário não informatizado, é um nível bastante baixo, porque para formular a maioria das consultas, deve-se indicar qual tabela deve obter cada dado e como combinar essas tabelas.

**Construção da solução.** Construção da solução: Como projetar um banco de dados tão grande? Era nossa especialidade, éramos professores universitários, pensávamos que conhecíamos bem o assunto, assessoramos sobre o assunto a várias grandes corporações brasileiras (petrolífera, financeira, industrial, governamental, comercial ...), porém nenhuma delas pretendia uma base de dados corporativa muito grande, muito complexo e que requereria um alto regime transacional, mas todos eles tinham várias "subject databases", cada uma ajudando-as a resolver uma pequena família de problemas.

**Tamanho do modelo de dados.** Tamanho do modelo de dados. Como raciocinar, na fase de projeto, com um modelo de dados de mais de 500 tabelas? Como visualizá-lo? Como normalizá-lo ou se devemos renunciar à normalização?

Certamente, um conjunto de erros humanos seria cometido. Seria bom ter ferramentas para nos ajudar: mas elas não existiam!

**As fontes de conhecimento.** Supondo que

obteríamos ou construiríamos as ferramentas necessárias: precisaríamos de matéria-prima para fazê-las funcionar. Essa matéria-prima era o conhecimento, em todos os níveis e de forma rigorosa, da realidade do usuário (minas

Seria bom ter ferramentas para nos ajudar: mas elas não existiam!

de dados, regras de negócio, regras gerais de consistência, etc.).

Para construir o modelo ER - Entity Relationship, que pretendíamos, quem na empresa cliente tinha esse conhecimento dos diferentes objetos da realidade e seus relacionamentos, com a objetividade e detalhe necessários?: NINGUÉM.

**A manutenção do sistema.** Dado o tamanho da solução e supondo que a construíssemos seguindo estritamente uma boa metodologia de desenvolvimento, o que ocorreria ao retirar os consultores? A metodologia seria escrupulosamente seguida? Quando seria necessário recorrer à documentação, as informações obtidas seriam fiel, precisas e atualizadas?

É quase impossível que uma metodologia usada para construir uma solução continue a ser aplicada integralmente ao longo do tempo, a menos que seja suportada por uma **ferramenta de desenvolvimento** que guie

e obrigue a conformidade com suas regras.

## SOLUÇÕES

**Descrever com precisão e rigor.** O problema só poderia ser resolvido e esta solução

O GeneXus infere o modelo de dados, o esquema do base de dados e os programas aplicativos.

ser mantida válida ao longo do tempo, se pudéssemos descrever a realidade do cliente de maneira precisa e rigorosa.

**Fontes de conhecimento.** Concluímos que a fonte básica de conhecimento são as visões de dados do usuário, expressas em uma estrutura de referência sólida e simples.

**Inteligência artificial.** As soluções manuais não resolviam um problema tão grande, por isso recorreremos à Inteligência Artificial.

Pode-se pensar que isso significa simplesmente o uso de certas ferramentas. Essa é uma maneira simplista e incorreta de ver as coisas.

Desde já analisamos as linguagens mais usadas no momento que eram LISP e PROLOG e apareceram duas posições: utilizar LISP porque, para ele, havia uma pluralidade de processadores de linguagem e, além disso, era a linguagem mais antiga e mais usada, ou escolher PROLOG uma nova

linguagem, com maiores pretensões, mas com pouco uso e para o qual havia poucos processadores de linguagem. Optamos pelo PROLOG e foi uma boa decisão.

A Inteligência Artificial parecia muito promissora e muitas empresas, especialmente nos EUA, estavam tentando desenvolver sistemas especialistas, especialmente para diagnósticos. Em um diagnóstico, uma confiabilidade de 95% ou 97%, por exemplo, é excelente (a de um especialista humano muito bom é geralmente menor). Mas na construção de grandes modelos de dados e sistemas, é terrível: você precisa de 100%!

Mas havia outro detalhe: nós éramos engenheiros de computação, com um bom conhecimento matemático, mas não sabíamos nada sobre Inteligência Artificial.

## UM NOVO MUNDO

O problema nos levou a um novo mundo. O mundo do conhecimento puro!

**Learning by doing.** Quando não sabemos como resolver um problema, mas o estudamos e finalmente o resolvemos, conseguimos muito mais do que isso: **Learning by doing** é hoje a maneira de estarmos atualizados e poder utilizar de imediato e com grande proveito as novas tecnologias.

**Rigor.** Trabalhar com conhecimento puro tem um conjunto de requisitos que podemos sintetizar em uma palavra: **rigor**.

Deveríamos e devemos (os pesquisadores, aqueles que participam de alguma maneira na concepção e construção de GeneXus) trabalhar com um nível muito alto de abstração, a fim de expressar cada problema com toda a precisão, como um problema de matemática e lógica pura.

Todas as nossas especificações devem ser

totalmente rigorosas e baseadas em um quadro de referência claro e sólido:

Representar o **significado** dos diferentes elementos, a fim de operar automaticamente com ele.

Adotar uma **regra básica**, a realidade é consistente, logo qualquer representação que fizermos dela deve ser consistente.

Implementar **operadores poderosos** para serem usados automaticamente.

E nossos usuários? (os desenvolvedores): Não seria realista exigir que eles trabalhassem permanentemente com um alto nível de abstração. Devemos **esconder a complexidade** para que os desenvolvedores trabalhem em um nível alto, de maneira direta e o mais simples possível, com elementos concretos e exigindo uma abstração normal.

**Conhecimento puro.** Basear-nos em conhecimento puro significa não apoiar-se em elementos físicos ou tecnológicos de nenhum tipo (que podem mudar e certamente mudarão com o tempo).

Apenas no momento da geração do código, incluir automaticamente os elementos físicos e / ou tecnológicos necessários.

Isso permite criar um alto nível de abstração, totalmente independente de variáveis casuísticas. Todas as especificações são armazenadas neste alto nível.

**Futureproofing.** O conhecimento se armazena e se opera com ele em uma Base de Conhecimento que possui uma enorme capacidade de inferência.

**O GeneXus infere o modelo de dados, o esquema do base de dados e os programas aplicativos.**

Uma das consequências mais importantes é que, como esse conhecimento é independente da tecnologia utilizada, sempre podemos gerar automaticamente um sistema usando outra configuração tecnológica. Sim-

plesmente devemos utilizar, em cada caso, a versão GeneXus que suporta a configuração tecnológica escolhida.

Isto é, o Futureproofing é operado: com o GeneXus, os sistemas são protegidos contra as mudanças determinadas pelo desenvolvimento contínuo da tecnologia. **Não há necessidade de reprogramar nada manualmente!**

### **Manutenção e evolução automáticas.**

Uma consequência menos espetacular que a anterior, mas não menos importante, é a manutenção automática dos sistemas. A capacidade de inferência do GeneXus permite originalmente a geração automática do sistema (base de dados e programas) e, quando ocorrem mudanças, propaga-as automaticamente.

O desenvolvedor precisa apenas atualizar, em um nível conceitual, os elementos descritivos que foram alterados. O GeneXus reportará o impacto dessas mudanças e, se aceito, as propagará automaticamente.

## **INTEGRAÇÃO**

Ninguém, nenhuma empresa, por maior que seja hoje e em qualquer área, é auto-suficiente. É cada vez mais necessário complementar as descobertas, a experiência e os produtos dos outros.

É bom:

- Facilitar a integração com outros produtos ao máximo para resolver os problemas reais dos clientes da maneira mais completa, rápida e econômica possível.

Não desenvolva nada que possa ser comprado. Definir "pode ser comprado" considerando a qualidade da solução a ser incorporada, seu tempo de disponibilidade e seu custo / oportunidade.

O GeneXus aumentou muito, nas últimas

versões, sua capacidade de integração.

## RESUMO E FUTURO

Temos separados a administração do conhecimento da administração da tecnologia. Tudo o que foi dito sobre a nossa tecnologia e, em particular, do uso automático do conhecimento puro poderia ser resumido em uma frase:

**"Nós alcançamos uma administração automática muito boa do conhecimento dos sistemas de negócios".**

Mas o mundo evolui, onde está o limite desses sistemas de negócios? Hoje, empresas nos mais diversos campos estão se posicionando como empresas de software com especialização em tais áreas.

O GeneXus representa o conhecimento de forma rigorosa e permanente e, como resul-

tado, opera automaticamente com ele. Que tipo de conhecimento? Qualquer um! Devemos sempre trabalhar duro, mas não há limites!

É por isso que será bom evoluir permanentemente: nossos clientes precisarão disso. Será necessário o acréscimo contínuo de novos tipos de aplicações - impensáveis até recentemente - que tornarão os sistemas cada vez mais complexos.

**A sólida base científica e tecnológica do GeneXus nos permitirá, incrementalmente, continuar construindo as soluções que se tornam necessárias.**

---

**\*Breogán Gonda.** Engenheiro, desenvolvedor de software, consultor, pesquisador, professor. Atualmente presidente do conselho da GeneXus.

A sólida base científica e tecnológica do GeneXus nos permitirá, incrementalmente, continuar construindo as soluções que se tornam necessárias.

Descubra como GeneXus  
pode ajudá-lo com o seu negócio.

[www.genexus.com](http://www.genexus.com)

