

Beyond Low-Code

Um olhar crítico sobre a categoria

By **Gastón Milano** | Outubro 2020

O ano de 2020 ficará na memória de todos nós por razões óbvias: COVID-19.

Não há dúvida de que a humanidade tem vivido um evento quase sem precedentes neste ano.

Além da saúde das pessoas em todo o mundo, o COVID-19 também coloca governos, indústrias e empresas à prova. Também acelerou os processos de transformação digital e evidenciou algumas características que são necessárias para sobreviver às mudanças e que serão fundamentais para ter por conta do futuro.

Em todos os tipos de setores ficou evidente a necessidade de passar por uma transformação digital verdadeira e profunda. De um momento para o outro, a medicina teve que usar a telemedicina, a educação passou a dar aulas remotas, as compras obviamente se voltaram drasticamente para o formato on-line, os pagamentos sem contato floresceram, as interfaces de voz invisíveis são usadas cada vez mais e todos os tipos de coisas tradicionalmente físicas se tornaram virtuais e remotos. Assim, um número infinito de indústrias acelerou rumo a uma transformação digital que estava acontecendo, que todos sabíamos que ia acontecer, mas que antes do COVID-19 o fazia sem muita velocidade.

A indústria de software tem o desafio particular e a oportunidade de acompanhar e capacitar as indústrias vencedoras e apresentar soluções e alternativas para as indústrias perdedoras. Em particular, existe uma categoria relativamente nova de plataformas e ferramentas que tem a capacidade de gerar grandes mudanças nas empresas em um curto espaço de tempo: chama-se “Low-Code Platforms” e é uma categoria na qual GeneXus compete.

As plataformas “Low-Code” emergem como alternativa à construção de software de forma tradicional, um conjunto de ferramentas que permite criar sistemas utilizando vários editores, linguagens gráficas gerais e algum tipo de geração de código, no âmbito de uma procura de um maneira melhor, mais eficaz e mais eficiente de desenvolver software.

•A discussão sobre Low-Code

Embora **GeneXus** hoje pertença circunstancialmente a esta categoria (e é sempre bom pertencer a algum tipo de classificação), acredito que haja alguns conceitos no nome e no que a própria categoria significa, dos quais tenho interesse em discordar e me levam a escrever este artigo para poder compartilhar a discussão.

Indo ao que significa, “low-code” (“baixo nível de codificação necessário”), parece-me importante lembrar que seus objetivos de 1) construir sistemas de software de negócios com maior velocidade do que os mecanismos tradicionais e 2) que eles podem ser construídos por pessoas de diversos níveis de conhecimento técnico, é uma aspiração profunda e ambiciosa que compartilho plenamente.

No entanto, existem alguns aspectos que não partilho com quem fala frequentemente de “Low-Code”, e é sobre estes pontos que pretendo elaborar:

1) O nome

O nome “**Low-Code**” foca em um recurso que não é necessariamente um benefício: escrever pouco ou muito código não significa algo bom ou ruim em si mesmo. O fato de exigir pouco código não significa necessariamente que o projeto será sustentável, compreensível ou extensível após a conclusão.

O termo “Low-Code” coloca a atenção em como os sistemas são construídos com essas plataformas e deixa de fora o mais importante de tudo: o benefício de criar software com esses tipos de soluções.

Pode me chamar de quixotesco (sonhador), mas nestes tempos de crise gostaria de pertencer a alguma categoria cujo nome remete ao por que acreditamos que ferramentas como GeneXus são relevantes para o mundo.

Porque neste novo mundo não importa se escrevo pouco código ou não, importa muito mais se eu atingir meus objetivos no tempo necessário. Posso fazer 3 aplicações em 1 semana para resolver um problema real? Posso estender esses aplicativos ou eles são rígidos ou limitados? Posso criar e interconectar sistemas em poucos dias? Posso mudar rapidamente o sistema desenvolvido se a realidade mudar? Posso alternar facilmente entre as tecnologias se tiver que abandonar um provedor específico?

Se há uma coisa que sabemos, é que hoje, mais do que nunca, a realidade está mudando e continuará mudando cada vez mais rápido, talvez até mais rápido do que a própria tecnologia. Portanto, o foco em como construímos algo novo rapidamente é importante, mas muito mais importante é colocar o foco em como conseguimos adaptar e desenvolver sistemas em alta velocidade conforme o mundo muda enquanto estamos trabalhando.

Escrever pouco código não é nosso objetivo: nosso objetivo é construir ferramentas que permitam transformações úteis nas várias indústrias que podem ser **favorecidas por nossa plataforma e nossa expertise**.

Enquanto não houver outra categoria, continuaremos falando sobre “Low-Code”, até mesmo “**Multi-Experience**”, mas a realidade é que me sentiria mais confortável em outras categorias, como: **High Performance Platforms, Evolutionary Development Platforms o Knowledge Driven Platforms**.

2) Linguagens Gráficas

Outra coisa que me deixa curioso é que parece que ter linguagens gráficas é um elemento essencial nesta categoria.

É verdade que muitas vezes uma imagem vale mais que mil palavras, mas é ainda mais verdade que o **conhecimento profundo e duradouro se escreve em palavras e fórmulas**. Expôr que essa linguagem gráfica por si só é o que permitirá que um sistema evolua e armazene o conhecimento do negócio é inocente e otimista, senão totalmente incorreto. A diversidade de realidades, de situações, de necessidades, de pessoas e a diversidade de formas de conhecimento não pode ser simplificadas ou reduzidas desta forma.

É verdade que uma linguagem gráfica muitas vezes, e facilmente, pode tornar uma ferramenta mais atrativa. Também é verdade que uma interface gráfica pode diminuir o nível de conhecimento necessário para realizar alguns tipos de soluções mais padronizadas, mas não está claro se é o melhor quando chega a hora de criar, compreender e preservar o conhecimento. Existem estudos e análises que vão nos dois sentidos e hoje certamente não há consenso de que algo totalmente gráfico é o caminho ideal, principalmente para a criação de sistemas sofisticados que evoluem ao longo do tempo.

Só sabemos que, se há algo que certamente não vai durar no tempo, esse algo é o código. Seja pouco ou muito, o código vai desaparecer, vai mudar e estamos convencidos de que o conhecimento não pode ser armazenado nem em linhas de código nem em desenhos. O conhecimento deve permanecer em algum tipo de base de conhecimento abstrata, muitas vezes **flexível e independente da tecnologia**, a fim de permitir que as soluções de software a serem geradas evoluam.

• **As coisas boas sobre o Low-Code**

Dito isso, acho que a categoria “Low-Code” está nos ajudando a perceber o que precisamos na indústria de software.

Por que a categoria é necessária?

Porque neste mundo cheio de incertezas em todos os sentidos temos que evoluir para novas formas de construção de software. A seguir, explico por que essa categoria é necessária (além do nome)

Por tempo

Porque os tempos que as empresas precisam hoje, independente de seu setor, de construir soluções não podem ser atendidos com as metodologias tradicionais de construção de software.

Por pessoas

Porque construir múltiplas experiências para os sistemas implica envolver infinitos especialistas, muitas vezes inviáveis por questões de custo.

Porque os tempos de treinamento em todas as novas tecnologias que precisam ser levadas em consideração para um sistema complexo são muito altos.

Para o futuro

Porque evoluir em código escrito manualmente leva muito tempo e custos que costumam ser difíceis para as empresas.

Porque certamente as pessoas que constroem o sistema hoje não serão as pessoas que farão a manutenção dele no futuro. Podemos realmente deixar o conhecimento do sistema da minha empresa, em um código que ninguém vai entender daqui a alguns anos?

Pela complexidade atual

Porque fazer um sistema hoje quase que por padrão implica fazer um sistema de informação complexo que envolve a coesão e a compreensão de várias tecnologias. Mesmo na presença de especialistas de alto desempenho, a falta de automação de tarefas leva a inúmeros erros custosos.

Richard Feynman disse: **“Há uma diferença importante entre saber o nome das coisas e saber o que as coisas realmente são.”**

Eu entendo que o nome da categoria não é o melhor. Acho que mesmo o que alguns pensam que esse nome significa não é o melhor também,, mas também é verdade que a essência do que essa categoria está trazendo para o mercado é algo que estávamos esperando há muito tempo. Podemos ter acordos e divergências com essa nova categoria, mas o importante é que ela existe e está se fortalecendo. Como diria Breogán Gonda, um dos fundadores de GeneXus, **“Bem-vindo a concorrência”**.



MONTEVIDEO - URUGUAY

Av. Italia 6201- Edif. Los Pinos, P1

(598) 2601 2082

CIUDAD DE MÉXICO - MÉXICO

Hegel N° 221, Piso 2, Polanco V Secc.

(52) 55 5255 4733

MIAMI - USA

8950 SW 74th Ct, Suite 1406

(1) 201 603 2022

SÃO PAULO - BRASIL

Rua Samuel Morse 120 Conj. 141

(55) 11 4858 0300

TOKYO - JAPAN

2-27-3, Nishi-Gotanda

(81) 3 6303 9381

Shinagawa-ku, Tokyo, 141-0031

(81) 3 6303 9980