

# Beyond Low-Code

Una mirada crítica a la categoría.

Por **Gastón Milano** | Octubre 2020

El año 2020 quedará en la memoria de todos nosotros por razones obvias: COVID-19.

No hay dudas que a la humanidad le ha tocado vivir un acontecimiento casi sin precedentes este año.

Además de la salud de las personas alrededor del mundo, el COVID-19 puso a prueba también a los gobiernos, a las industrias y a las empresas. También aceleró los procesos de transformación digital e hizo obvias algunas características que son necesarias poseer para poder **sobrevivir a los cambios y que serán fundamentales a tener en cuenta en el futuro.**

En todo tipo de sectores se puso al desnudo la necesidad de transitar rápidamente **una verdadera, una profunda, transformación digital.** De un momento para otro la medicina tuvo que usar telemedicina, la educación dar clases remotas, las compras obviamente se volcaron drásticamente a formato online, los pagos contactless florecieron, las interfaces invisibles de voz se usan cada vez más y todo tipo de cosas tradicionalmente físicas pasaron a ser virtuales y a distancia. Así, un sin fin de industrias aceleraron hacia una transformación digital que estaba sucediendo, que todos sabíamos que iba a pasar, pero que antes del COVID-19 lo hacía sin mucha velocidad.

La industria del software tiene el particular desafío y la oportunidad de **acompañar y potenciar** las industrias ganadoras y presentar soluciones y alternativas a las industrias perdedoras. En particular, hay una categoría de plataformas y herramientas relativamente nueva que tiene la capacidad de generar grandes cambios en las empresas en poco tiempo: recibe el nombre de **“Low-Code Platforms”** y es una categoría en la que GeneXus compete.

Las plataformas de “Low-Code” emergen como una alternativa a construir software de manera tradicional, un conjunto de herramientas que permiten crear sistemas usando diversos editores, lenguajes en general gráficos y algún tipo de generación de código, como parte de una **búsqueda de una mejor, más eficaz y más eficiente, forma de desarrollar software.**

## •La discusión alrededor de Low-Code

Si bien **GeneXus** hoy pertenece circunstancialmente a esta categoría (y siempre es bueno pertenecer a algún tipo de clasificación) creo que hay algunos conceptos en el nombre y en lo que la propia categoría significa con los que me interesa discrepar y me llevan a escribir este artículo para poder compartir la discusión.

Yendo a lo que significa, “low-code” (“bajo nivel de codificación requerido”), me parece importante rescatar que sus objetivos de 1) construir sistemas de software de negocios con mayor velocidad que mecanismos tradicionales y 2) que puedan ser construidos por personas de diverso nivel de conocimiento técnico, es una aspiración profunda y ambiciosa que comparto totalmente.

## 1) El nombre

El nombre “**Low-Code**” se enfoca en una característica que no necesariamente es un beneficio: Escribir poco o mucho código no significa algo bueno o malo de por sí. Que requiera poco código no implica forzosamente que el proyecto vaya a ser mantenible, comprensible o extensible luego de terminado.

El término “Low-Code” pone la atención en cómo se construyen los sistemas con dichas plataformas y deja de lado lo más importante de todo: el beneficio que da crear software con este tipo de soluciones.

Llámenme quijotesco, pero en estas épocas de crisis me gustaría pertenecer a alguna categoría cuyo nombre haga referencia al porqué creemos que **herramientas como GeneXus son relevantes para el mundo.**

Porque en este nuevo mundo no importa si escribo poco código o no, importa mucho más si obtengo mis objetivos en los tiempos necesarios. ¿Puedo hacer 3 aplicaciones en 1 semana para resolver un problema real? ¿Puedo extender esas aplicaciones o son rígidas o limitadas? ¿Puedo crear e interconectar sistemas en pocos días? ¿Puedo cambiar rápidamente el sistema desarrollado si cambia la realidad? ¿Puedo cambiar entre tecnologías fácilmente si hay que abandonar algún proveedor particular?

Si hay algo que sabemos, es que hoy más que nunca la realidad cambia y seguirá cambiando cada vez con más velocidad, tal vez hasta más rápidamente que la tecnología misma. Entonces, el foco en cómo construimos algo nuevo rápidamente es importante, pero mucho más importante es poner el foco en cómo logramos **adaptar y evolucionar** sistemas a gran velocidad a medida que el mundo cambia mientras estamos trabajando.

Escribir poco código no es nuestro objetivo: nuestro objetivo es construir herramientas que permitan transformaciones útiles en las diversas industrias que puedan ser favorecidas por **nuestra plataforma y nuestro expertise.**

Mientras no exista otra categoría seguiremos hablando de “Low-Code”, incluso de “**Multi-Experience**” pero la realidad es que me sentiría más cómodo en otra categorías como: **High Performance Platforms, Evolutionary Development Platforms o Knowledge Driven Platforms.**

## 2) Lenguajes Gráficos

Otra cosa que me genera curiosidad es que parecería que el tener lenguajes gráficos es un elemento esencial en esta categoría.

Es verdad que muchas veces una imagen vale más que mil palabras, pero es aún más cierto que el **conocimiento profundo y de larga duración está escrito en palabras y fórmulas**. Plantear que el lenguaje gráfico en sí mismo y en soledad es lo que permitirá evolucionar y almacenar conocimiento de negocio de un sistema, es por demás inocente y optimista, por no decir totalmente incorrecto. La diversidad de realidades, de situaciones, de necesidades, de personas y la diversidad de formas de conocimiento no pueden ser simplificadas o reducidas de tal forma.

Es cierto que un lenguaje gráfico muchas veces, y fácilmente, puede volver más atractiva una herramienta. También es cierto que una interfaz gráfica puede bajar el nivel de conocimiento necesario para llevar adelante algunos tipos de soluciones más estándares, pero no está claro, cuando llega la hora de crear, entender y conservar conocimiento, si es lo mejor. Hay estudios y análisis en ambas direcciones, y hoy es seguro que no hay un consenso de que algo completamente gráfico sea el camino ideal, en particular para la creación de sistemas sofisticados que evolucionan a lo largo del tiempo.

Sólo sabemos que si hay algo que seguro no va perdurar en el tiempo, ese algo es el código. Sea poco o mucho, el código desaparecerá, cambiará y estamos convencidos de que el conocimiento no puede quedar almacenado ni en líneas de código, ni en dibujos. El conocimiento debe quedar en algún tipo de base de conocimiento **abstracta, a la vez flexible e independiente** de la tecnología, para poder permitir evolucionar las soluciones de software que se desea generar.

## • Las cosas buenas del Low-Code

Dicho esto, creo que la categoría “Low-Code” está ayudando a darnos cuenta de lo que necesitamos en la industria de software.

¿Porqué es necesaria la categoría ?

Porque en este mundo lleno de incertidumbres en todo sentido tenemos que evolucionar hacia nuevas formas de construcción de software. Lo que sigue son mis porqué esta categoría es necesaria (más allá del nombre)

### **Por Tiempo**

Porque los tiempos que necesitan hoy las empresas, sin importar su industria, para construir soluciones no pueden ser atendidos con metodologías de construcción de software tradicionales

### **Por Gente**

Porque construir experiencias múltiples para los sistemas implica involucrar un sinfín de especialistas, muchas veces inviables por razones de costo.

### **Por el futuro**

Porque evolucionar sobre código escrito manualmente insume largos tiempos y costos muchas veces difíciles de afrontar por las empresas.

Porque seguramente las personas que construyan el sistema hoy no serán las personas que los mantendrán en el futuro. ¿Realmente podemos dejar el conocimiento del sistema de mi empresa, en código que nadie va a comprender en unos años?

### **Por la complejidad actual**

Porque hacer un sistema hoy implica casi por defecto hacer un sistema de información complejo que implica la cohesión y comprensión de diversas tecnologías. Aún en la presencia de expertos de alta performance la falta de automatización de tareas lleva a un sin fin de costosos errores.

Richard Feynman decía: **“hay una diferencia importante entre saber el nombre de las cosas y saber lo que las cosas realmente son”**.

Entiendo que el nombre de la categoría no es el mejor. Creo que incluso lo que algunos piensan que ese nombre significa tampoco es lo mejor, pero también es cierto que la esencia de lo que está trayendo esta categoría al mercado es algo que estábamos esperando hace mucho tiempo. Podemos tener acuerdos y discrepancias con esta nueva categoría, pero lo importante es que existe y es cada vez más fuerte. Como diría Breogán Gonda, uno de los fundadores de GeneXus, **“Bienvenida sea la competencia”**.



MONTEVIDEO - URUGUAY

CIUDAD DE MÉXICO - MÉXICO

MIAMI - USA

SÃO PAULO - BRASIL

TOKYO - JAPAN

Av. Italia 6201- Edif. Los Pinos, P1

Hegel N° 221, Piso 2, Polanco V Secc.

8950 SW 74th Ct, Suite 1406

Rua Samuel Morse 120 Conj. 141

2-27-3, Nishi-Gotanda

Shinagawa-ku, Tokyo, 141-0031

(598) 2601 2082

(52) 55 5255 4733

(1) 201 603 2022

(55) 11 4858 0300

(81) 3 6303 9381

(81) 3 6303 9980