

Introducción Teórica

GeneXus[®]

Herramientas y Metodologías

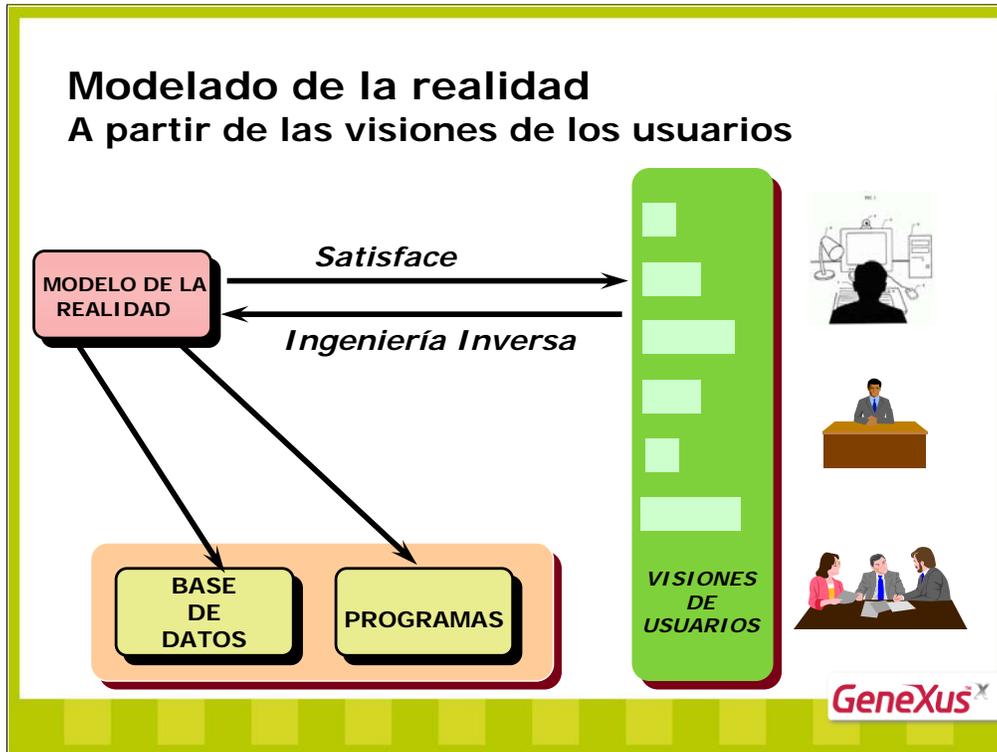


Nuestra tarea como profesionales de la informática consiste en desarrollar y mantener aplicaciones para apoyar al usuario en su actividad. Para realizar esta tarea existen diferentes herramientas y metodologías.

GeneXus es una herramienta para el desarrollo de aplicaciones sobre bases de datos. Su objetivo es permitir la implantación de aplicaciones en el menor tiempo y con la mejor calidad posible.

A grandes rasgos, el desarrollo de una aplicación implica tareas de análisis, diseño e implementación. La vía de GeneXus para alcanzar el objetivo anterior es liberar a las personas de las tareas automatizables (como el diseño de la base de datos), permitiéndoles así concentrarse en las tareas realmente difíciles y no automatizables (como comprender los problemas del usuario).

GeneXus emplea una metodología que tiene un enfoque muy diferente al de las metodologías más comúnmente utilizadas. Por tanto, aprender a utilizar GeneXus adecuadamente va más allá de conocer un nuevo lenguaje: lo más importante es **aprender su metodología**.

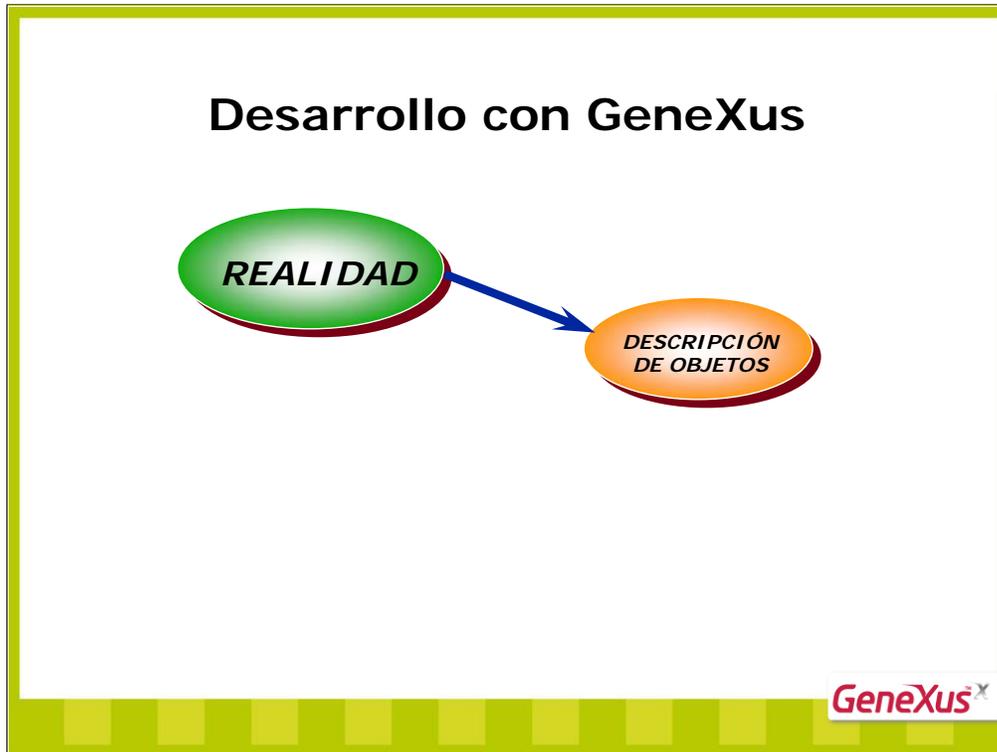


El primer problema al que nos enfrentamos en el desarrollo de aplicaciones es la obtención del conocimiento de la realidad.

Nadie dentro de la empresa conoce los requerimientos y el alcance de la aplicación a desarrollar como un todo. Entonces, ¿cómo logramos obtener el conocimiento de la realidad de una forma lo suficientemente objetiva y detallada al mismo tiempo, que nos permita construir un modelo corporativo?

Este conocimiento se encuentra en cada una de las visiones de los usuarios. Cada usuario conoce bien los objetos con los que trabaja cotidianamente, la información que se maneja en ellos, las reglas que deben seguirse, los cálculos que deben realizarse.

Por lo tanto, el punto de partida de la metodología GeneXus es: **describir las visiones de los usuarios** para modelar el sistema; y a partir del modelo de la realidad definido, **GeneXus construye el soporte computacional** -base de datos y programas- en forma totalmente **automática**.



Utilizando GeneXus, la tarea básica del analista es la descripción de la realidad. Sólo el ser humano puede desarrollar esta tarea ya que sólo él puede entender el problema del usuario.

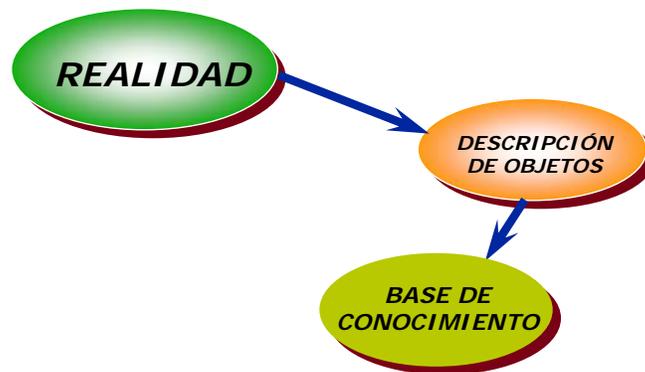
El analista GeneXus trabaja en alto nivel, en vez de realizar tareas de bajo nivel como: diseñar archivos, normalizar, diseñar programas, programar, buscar y eliminar los errores de los programas.

Para comenzar el desarrollo de una aplicación con GeneXus, el primer paso consiste en crear un nuevo proyecto o **base de conocimiento**.

Una vez creada una nueva base de conocimiento (en inglés: knowledge base; abreviado: KB), el siguiente paso es describir las visiones de los usuarios. Para ello se deben identificar los objetos de la realidad (prestando atención a los sustantivos que los usuarios mencionan en sus descripciones, como por ejemplo: clientes, productos, facturas) y pasar a definirlos mediante **objetos** GeneXus.

Con la definición de estos objetos, GeneXus puede extraer el conocimiento y diseñar la base de datos y los programas de la aplicación en forma **automática**.

Desarrollo con GeneXus



GeneXus^x

Utilizando GeneXus, la tarea básica del analista es la descripción de la realidad. Sólo el ser humano puede desarrollar esta tarea ya que sólo él puede entender el problema del usuario.

El analista GeneXus trabaja en alto nivel, en vez de realizar tareas de bajo nivel como: diseñar archivos, normalizar, diseñar programas, programar, buscar y eliminar los errores de los programas.

Para comenzar el desarrollo de una aplicación con GeneXus, el primer paso consiste en crear un nuevo proyecto o **base de conocimiento**.

Una vez creada una nueva base de conocimiento (en inglés: knowledge base; abreviado: KB), el siguiente paso es describir las visiones de los usuarios. Para ello se deben identificar los objetos de la realidad (prestando atención a los sustantivos que los usuarios mencionan en sus descripciones, como por ejemplo: clientes, productos, facturas) y pasar a definirlos mediante **objetos** GeneXus.

Con la definición de estos objetos, GeneXus puede extraer el conocimiento y diseñar la base de datos y los programas de la aplicación en forma **automática**.



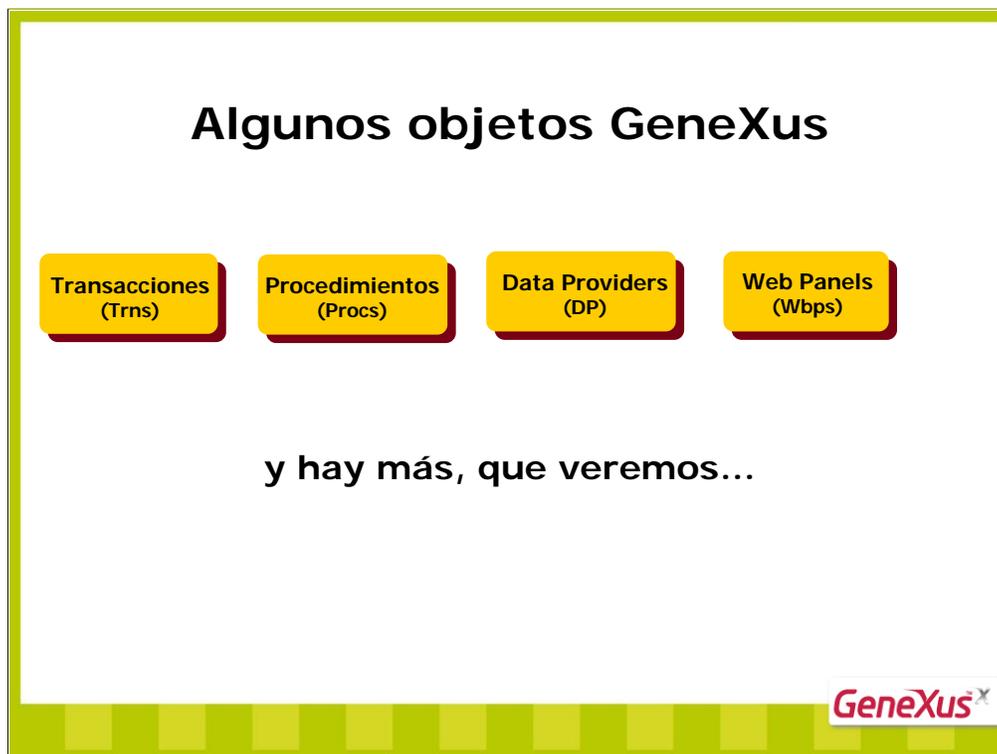
A partir de los objetos definidos en la base de conocimiento, GeneXus **genera automáticamente** tanto los **programas de creación / reorganización de la base de datos** como los **programas de la aplicación**.

Luego, si un objeto de la realidad cambia, si se identifican nuevas o diferentes características del mismo, o si se encuentran objetos aún no modelados, el analista GeneXus debe reflejar dichos cambios en los objetos GeneXus que correspondan, y la herramienta se encargará automáticamente de realizar las modificaciones necesarias tanto en la base de datos como en los programas asociados.

La metodología GeneXus es una **metodología incremental**, pues parte de la base de que la construcción de un sistema se realiza mediante aproximaciones sucesivas.

En cada momento el analista GeneXus define el conocimiento que tiene y luego cuando pasa a tener más conocimiento (o simplemente diferente) lo refleja en la base de conocimiento y GeneXus se ocupará de hacer **automáticamente** todas las adaptaciones en la base de datos y programas.

Si GeneXus no fuera capaz de realizar **automáticamente** las modificaciones en la base de datos y programas conforme se realicen cambios que así lo requieran, el desarrollo incremental sería inviable.



Una vez creada una base de conocimiento, el siguiente paso consiste en comenzar a describir los objetos de la realidad mediante objetos GeneXus.

Los objetos GeneXus más importantes son:

Transacciones

Permiten definir los objetos de la realidad que el usuario manipula (ej: clientes, productos, proveedores, facturas, etc.). Son los primeros objetos en definirse, ya que a través de las transacciones, GeneXus infiere el diseño de la base de datos.

Además de tener por objetivo la definición de la realidad y la consecuente creación de la base de datos normalizada, cada transacción tiene asociada una pantalla para ambiente windows y otra para ambiente Web, para permitir al usuario dar altas, bajas y modificaciones en forma interactiva a la base de datos. El analista GeneXus decidirá si trabajar en ambiente windows, Web, o ambos, y GeneXus generará los programas para ello.

Procedimientos

Permiten recuperar información de la base de datos, y desplegarla ya sea en la pantalla, en un archivo o impresa en papel. Son los típicos listados o informes. Además, permiten la actualización de la información de la base de datos.

Data Providers

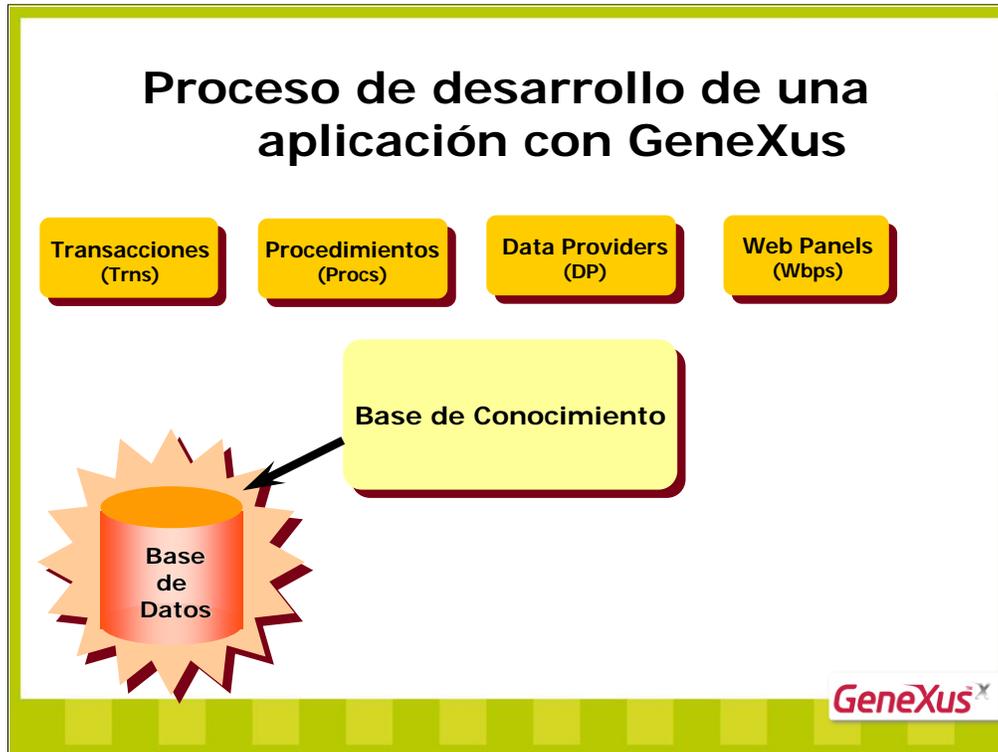
Permiten cargar y devolver datos jerárquicos para intercambio de información entre objetos de la misma aplicación o de otras aplicaciones.

Web Panels

Permiten al usuario realizar interactivamente consultas a la base de datos, a través de una pantalla. Ejemplo: un web panel permite al usuario ingresar un rango de caracteres, y muestra a continuación todos los clientes cuyos nombres se encuentran dentro del rango.

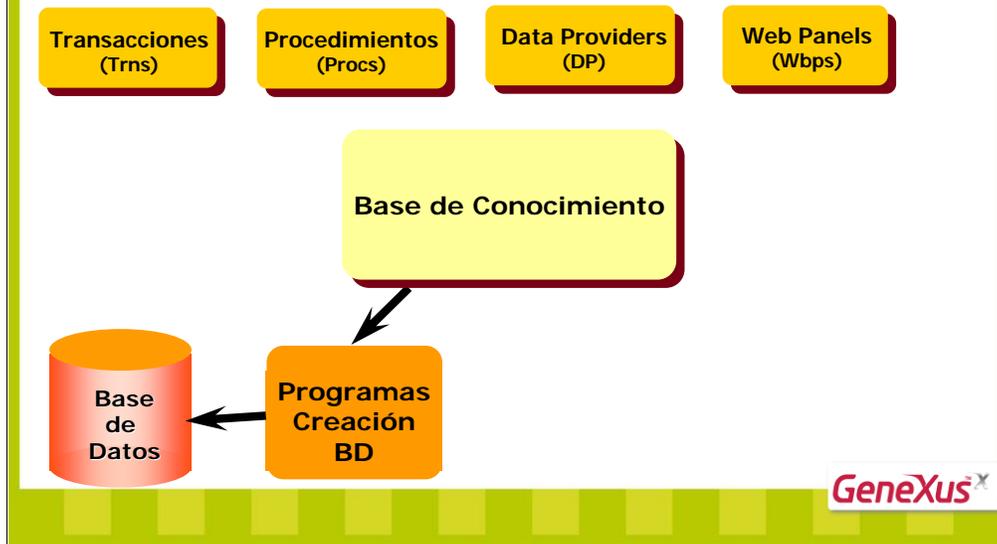
Son objetos web muy flexibles que se prestan para múltiples usos. No permiten la actualización de la base de datos, sino solo su consulta.

Proceso de desarrollo de una aplicación con GeneXus

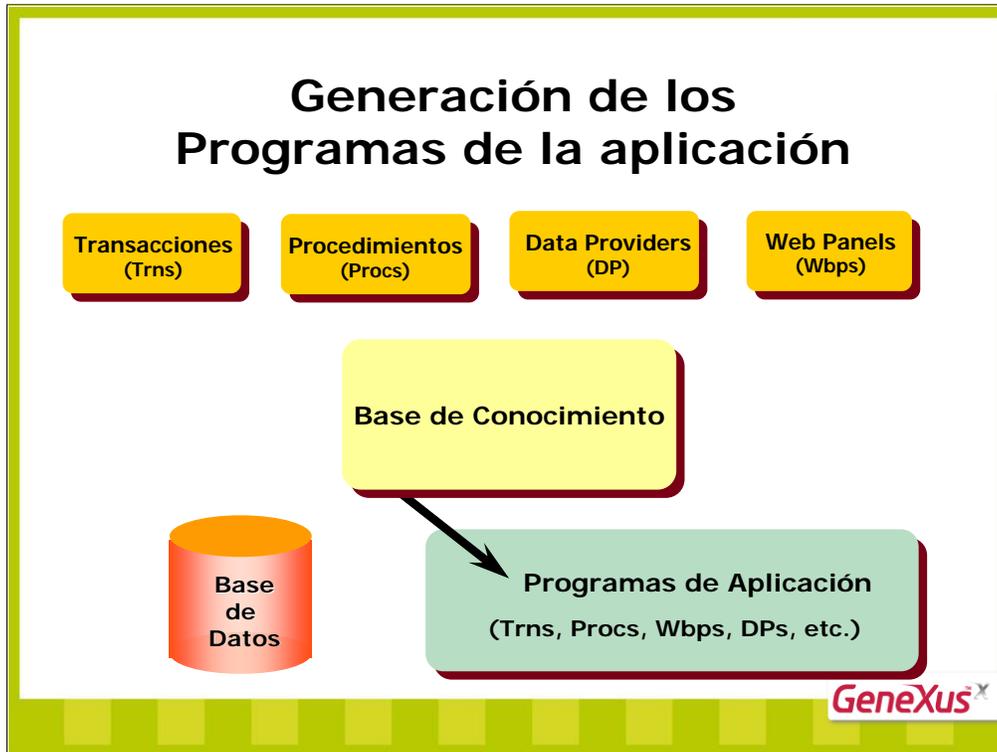


Los primeros objetos que se definen son las transacciones, ya que es a partir de ellas que GeneXus extrae el conocimiento necesario para diseñar el modelo de datos normalizado (en 3era. forma normal). Luego se van definiendo los demás objetos que correspondan.

Creación de la Base de Datos

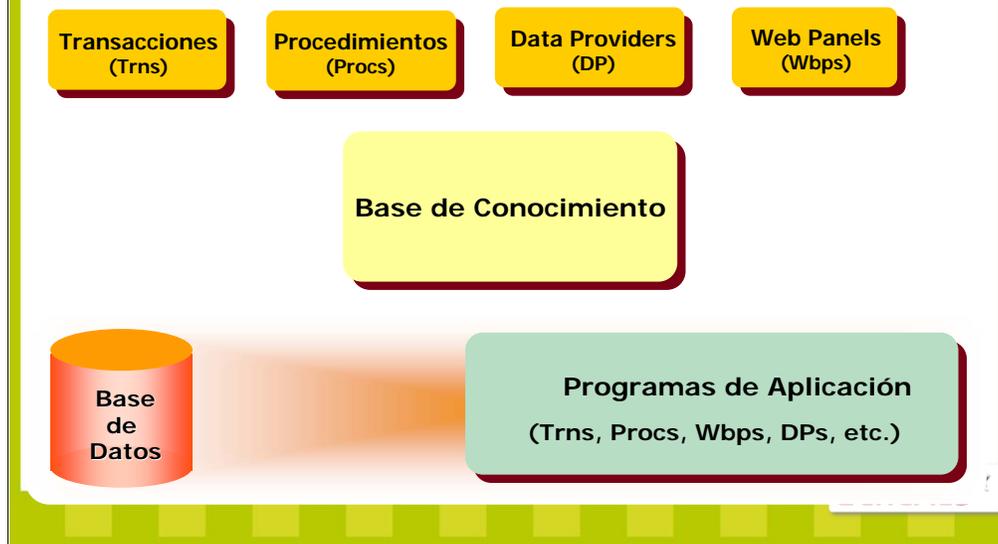


GeneXus **genera automáticamente** los programas necesarios para crear la base de datos y los ejecuta. De esta manera obtenemos la base de datos **creada por GeneXus en forma automática**.



Luego, **GeneXus genera** programas de aplicación para interactuar con la base de datos previamente creada.

Resultado final en la Etapa de Desarrollo



Una vez creada la base de datos y generados los programas, contamos con una aplicación pronta para ejecutar.

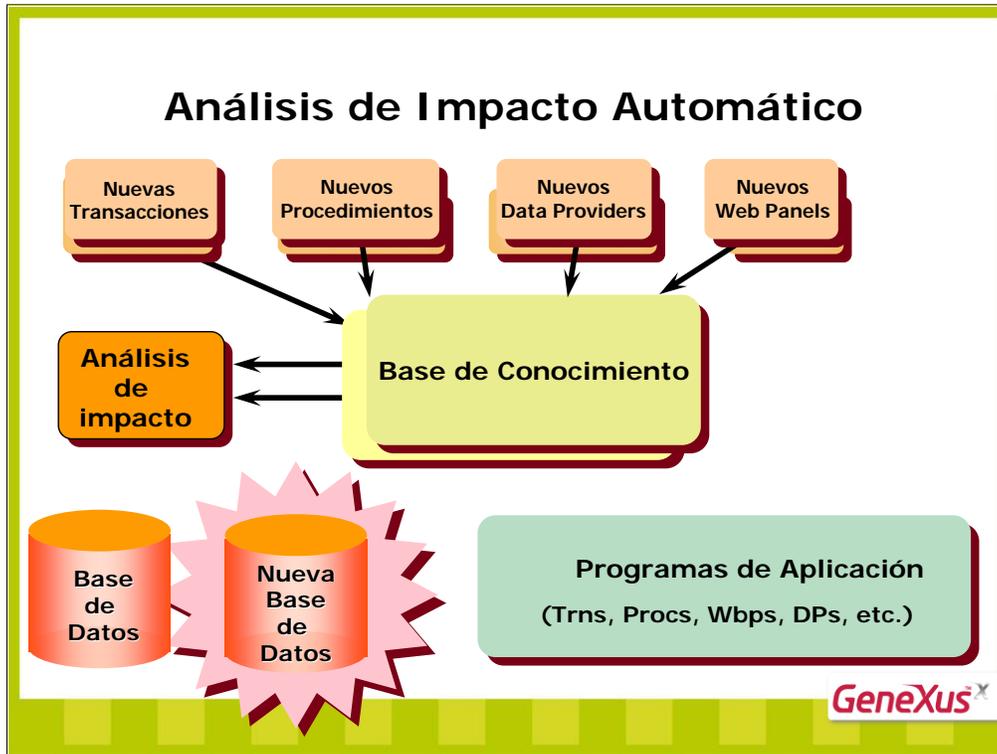


Durante el ciclo de vida de la aplicación, surgirá repetidamente la necesidad de hacer modificaciones en la base de conocimiento, ya sea porque las visiones de los usuarios cambian, porque se deben hacer correcciones, o simplemente agregar nuevo conocimiento.

Las modificaciones que se realicen sobre la base de conocimiento serán **analizadas por GeneXus para evaluar si es necesario efectuar cambios en la base de datos** (por ejemplo: modificación/creación de tablas/índices), o no.

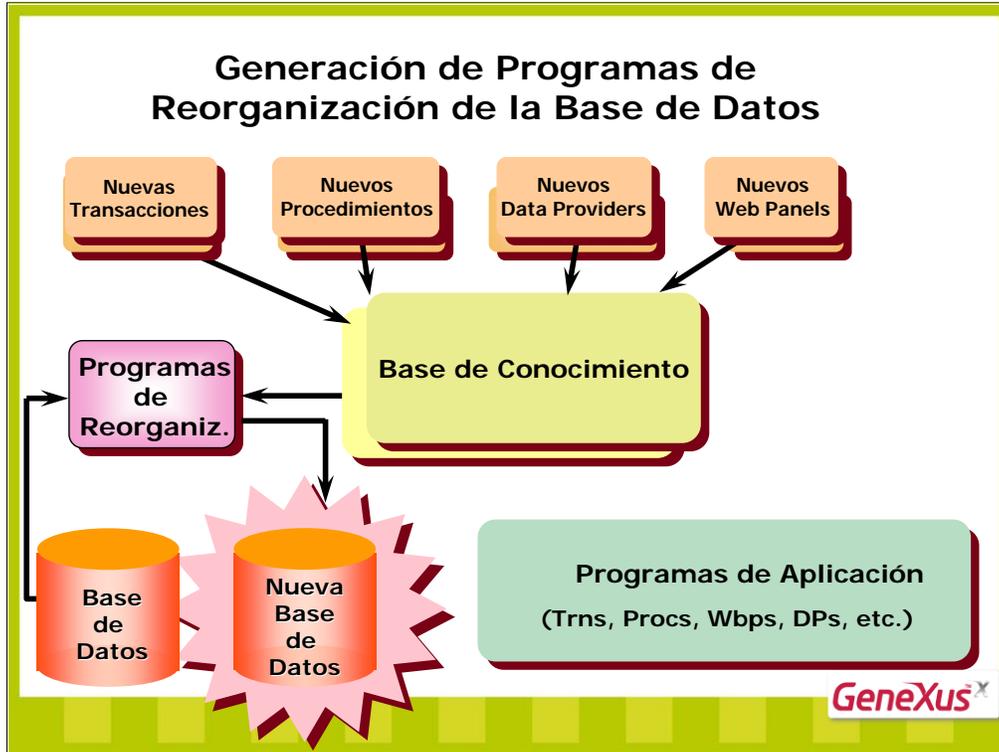
En caso de detectar cambios para efectuar en la base datos, GeneXus detallará los mismos en un **reporte de análisis de impacto (IAR: Impact Analysis Report)**, que es un reporte que explicita todos los cambios sobre tablas, índices, datos, etc. que habría que realizar para reflejar la nueva realidad.

Asimismo, en el **reporte de análisis de impacto** se informan los eventuales problemas que los cambios en cuestión podrían ocasionar, como inconsistencias o redundancias.



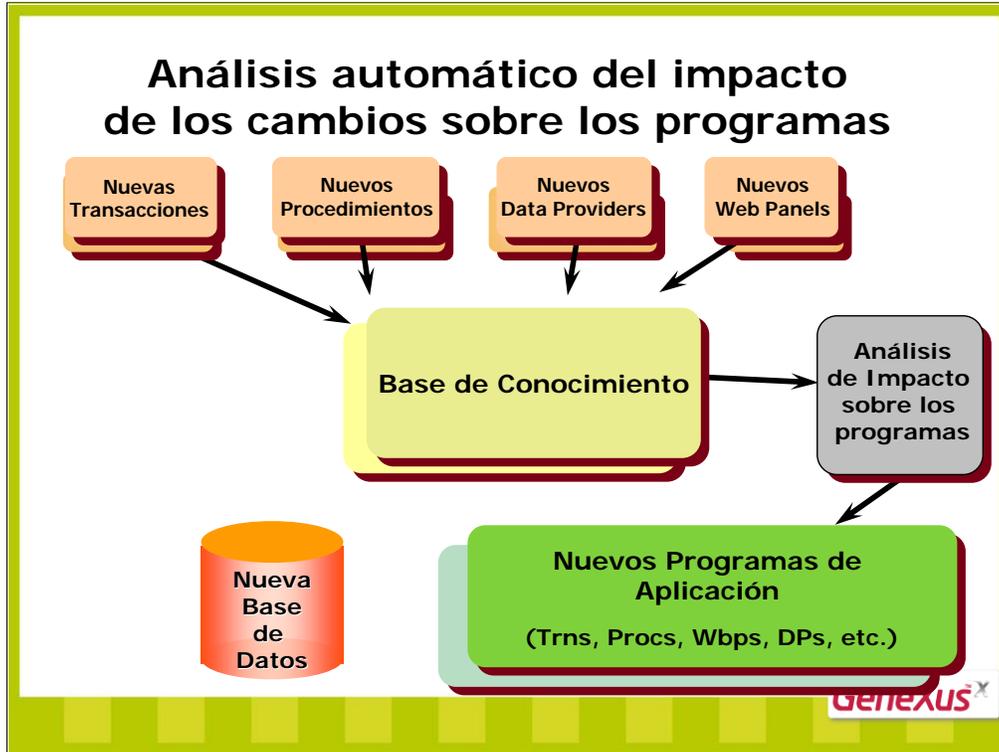
Algunas veces la nueva base de datos coincide con la anterior. Otras veces esto no ocurre, y la base de datos debe sufrir alguna modificación para representar la nueva realidad.

El analista debe estudiar el **reporte de análisis de impacto** y resolver si desea realizar efectivamente los cambios en la base de datos, o renunciar a ello dejando la base de datos como estaba.

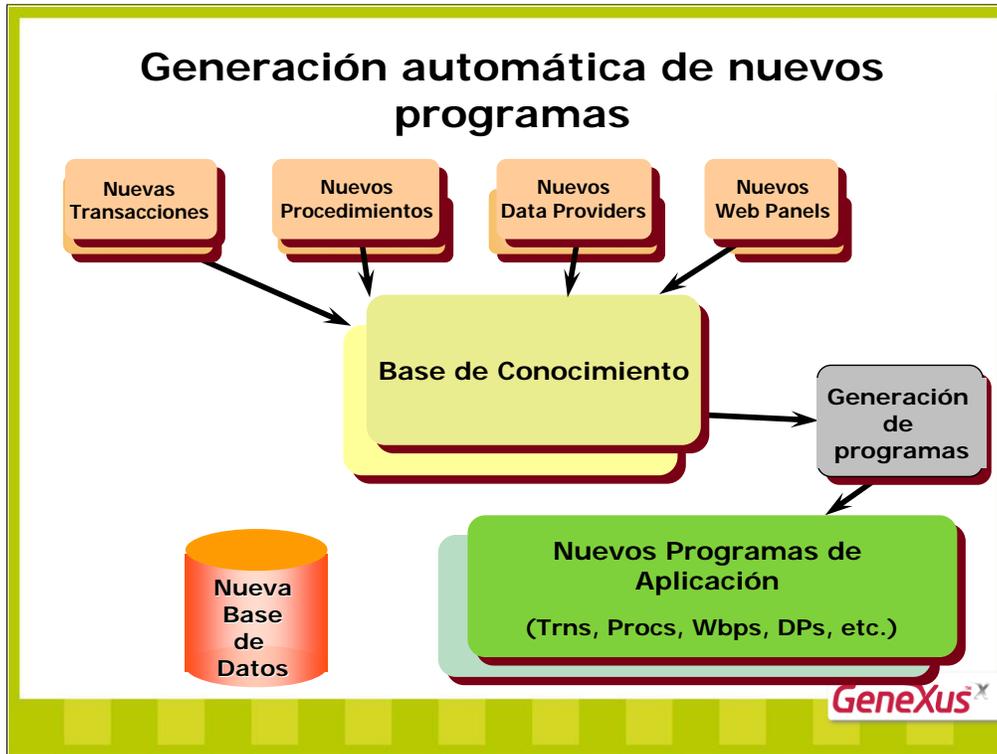


Si el analista opta por aplicar los cambios propuestos, decimos que optó por **reorganizar** la base de datos. Utilizamos este término para referirnos a la acción de aplicar cambios físicos sobre la base de datos.

GeneXus generará los programas que implementan las modificaciones sobre las estructuras físicas de la base de datos, y mediante su ejecución nos brindará la nueva versión de la base de datos con los cambios efectuados.



Ya sea que se requiera reorganizar la base de datos o no, considerando las nuevas definiciones introducidas, GeneXus estudiará el impacto de los cambios sobre los programas actuales.



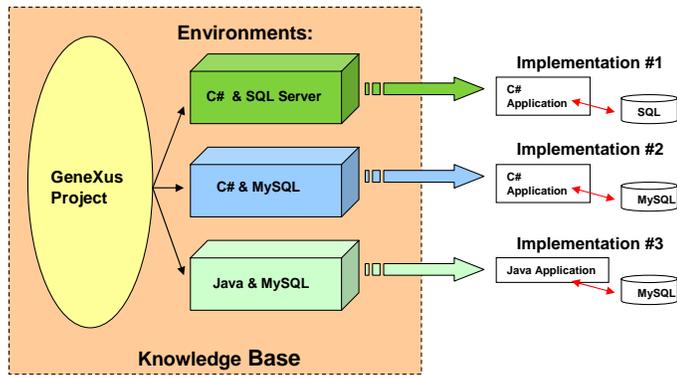
Por último, GeneXus proseguirá con la **generación/regeneración** de los **programas de aplicación** que sean necesarios, obteniendo así una nueva versión de la aplicación.



De modo que nuevamente contaremos con una aplicación pronta para ejecutar, con los cambios aplicados.

Environments

- El lugar donde se almacena la información para generar la aplicación en cierta plataforma de ejecución se llama Environment.

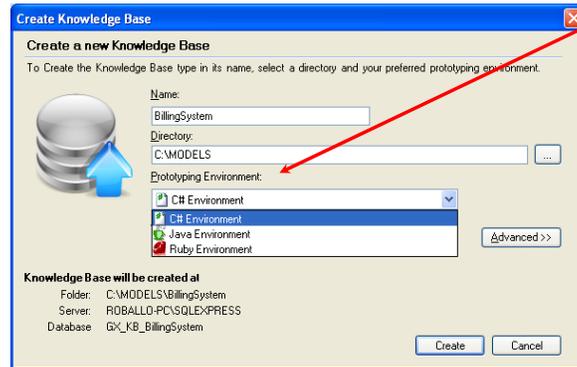


- El uso de varios Environments permite distintas implementaciones de la misma aplicación.

GeneXus[®]

Environments

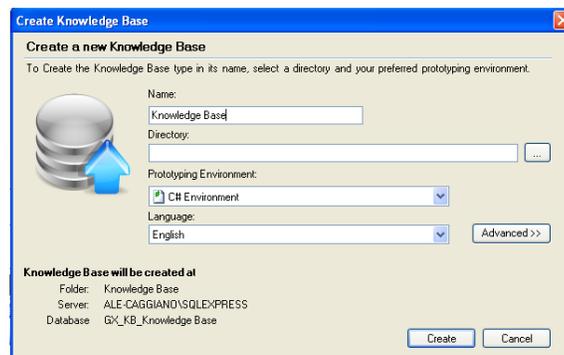
- Cuando se crea una base de conocimiento (KB), GeneXus pide al usuario que seleccione el Environment con el que va a trabajar:



- Con estos datos, se crea automáticamente un 'Environment'.

GeneXus[®]

Para crear una base de conocimiento, se debe seleccionar en la barra de menú de GeneXus, el ítem **File / New Knowledge Base**. A continuación aparecerá un diálogo como el siguiente:



Se deberá indicar;

- Nombre de la Knowledge Base: en nuestro caso será BillingSystem.
- Directorio donde se creará.
- El 'Environment' por defecto: vea como se muestran lenguajes de programación. GeneXus los utilizará para crear los programas asociados a la base de datos. Las opciones ofrecidas son: C# Environment, Java Environment y Ruby Environment.
- Language: Idioma en el que aparecerán los botones, mensajes, etc.

Al momento de crear la KB se comienza a definir el ambiente de implementación (Environment) cuyas definiciones serán luego completadas al momento de ejecutar la aplicación (nombre de la base de datos, servidor, forma de conexión, etc).

Environments

- Para ver el 'Environment' creado, seleccionamos la ventana de **Preferences** del **Knowledge Base Navigator**:

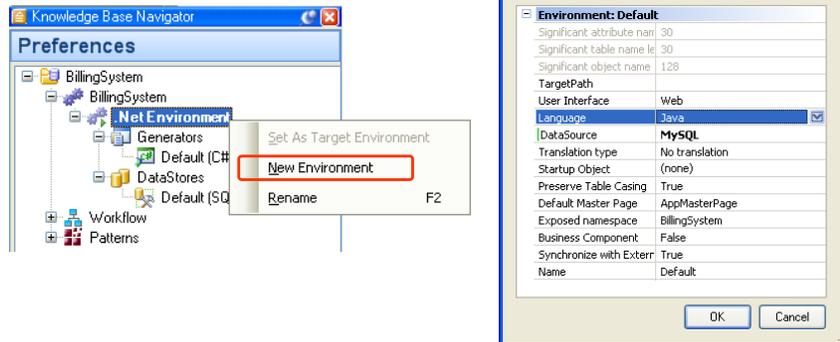
The image shows two windows from the Knowledge Base Navigator. The left window is titled 'Preferences' and displays a tree view of project components. The component '.Net Environment' is selected, and a red arrow points to it from the text 'Default Environment'. The right window is titled 'Properties' and shows the configuration for the selected '.Net Environment'. A red box highlights the 'User Interface', 'Language', and 'DataSource' properties.

Environment: .Net Environment	
Significant attribute name	30
Significant table name length	30
Significant object name	128
TargetPath	CSharpModel
User Interface	Web
Language	C#
DataSource	SQL Server
Translation type	No translation
Startup Object	(none)
Preserve Table Casing	True
Default Master Page	AppMasterPage
Exposed namespace	BillingSystem
Business Component	False
Synchronize with Extern	True
Name	.Net Environment

GeneXus[®]

Environments

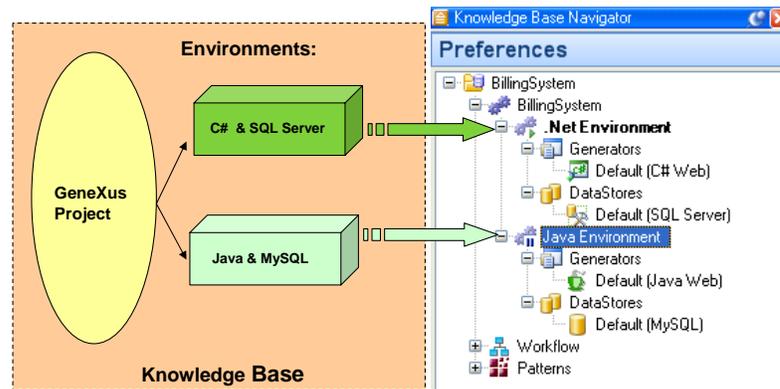
- Para tener implementaciones en distintas plataformas, creamos varios 'Environments'.



GeneXus[®]

Environments

- Vemos los 'Environments' creados:

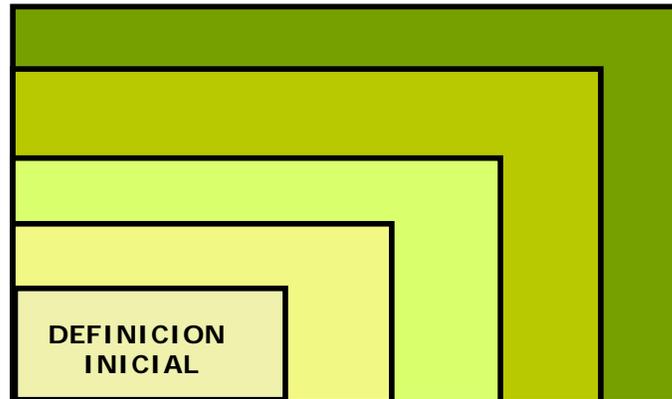


GeneXus[®]

El environment activo está señalado en negrita y tiene el ícono de "Play"

Metodología Incremental

Construir una aplicación mediante aproximaciones sucesivas.



GeneXus[®]

La construcción automática de la base de datos y programas, permite a GeneXus aplicar esta metodología de desarrollo, conocida como **metodología incremental**.

Como ya hemos explicado, este proceso se realiza mediante aproximaciones sucesivas.

Metodología Incremental

Para aplicar un desarrollo incremental, puedo manejar distintas **versiones** de la aplicación.

- Versión para prototipación.
- Versión para poner en producción.

GeneXus[®]

Sobre el versionado (de qué se trata, cómo realizarlo, etc.) volveremos al final. Por ahora alcanza con saber que al crear la base de conocimiento, los programas que estaremos ejecutando, programas reales, serán una versión de la aplicación, de prueba. Una vez que decidamos que esa versión está lista para ser puesta en producción, alcanzará con hacer otra '**versión**' de la aplicación, y ¡listo!.



Ventajas de la Prototipación

- Permite ver resultados en forma temprana.
- Permite el seguimiento de los requerimientos del usuario.
- Detección de errores en forma temprana.
- Logra el compromiso de los usuarios con el desarrollo.
- Sistemas de mejor calidad.

GeneXus[®]

Toda comunicación es susceptible de errores:

- El usuario olvida ciertos detalles
- El analista no toma nota de algunos elementos
- El usuario se equivoca en algunas apreciaciones
- El analista malinterpreta algunas explicaciones del usuario

Como la implementación de sistemas es habitualmente una tarea que insume bastante tiempo, y muchos de estos problemas sólo son detectados en las pruebas finales del sistema, el costo en tiempo y dinero de solucionarlos se torna muy grande. Sabido es que la realidad no permanece estática, por lo que no es razonable suponer que se pueden mantener congeladas las especificaciones mientras se implementa el sistema. Sin embargo, debido al tiempo que suele insumir la implementación, muchas veces esto se hace y se acaba implementando una solución relativamente insatisfactoria.

El impacto de estos problemas disminuiría mucho si se consiguiera probar cada especificación inmediatamente y saber cuál es la repercusión de cada cambio sobre el resto del sistema. Una primera aproximación a esto, ofrecida por diversos sistemas, es la posibilidad de mostrar al usuario formatos de pantallas, informes, etc., animados por menús. Esto permite ayudar al usuario a tener una idea de qué sistema se le construirá, pero al final siempre se presentan sorpresas.

Una situación bastante diferente sería la de poner a disposición del usuario para su ejecución, una aplicación funcionalmente equivalente a la deseada hasta en los mínimos detalles. Y esto es lo que ofrece GeneXus! Un **prototipo GeneXus** es una aplicación pronta funcionalmente equivalente a la aplicación de producción.

Así es que la aplicación puede ser totalmente probada antes de ponerse en producción; y durante las pruebas, el usuario final puede probar de una forma natural no solamente formatos de pantallas, informes, etc., sino también fórmulas, reglas del negocio, estructuras de datos, etc., y trabajar con datos reales.

Esto solo es posible gracias a la construcción automática que realiza GeneXus del soporte computacional (base de datos y programas).