

Manejo de versiones

GeneXus[®]

Manejo de Versiones

El desarrollo de software es un trabajo en equipo y cierto grado de confusión es inevitable.



Manejo de Versiones

Conclusión:

- El ciclo de desarrollo es un proceso dinámico que requiere control de los cambios realizados a los objetos del proyecto.

Se necesita:

- Marcar hitos en el desarrollo de la aplicación
- Tener líneas de desarrollo paralelas
- Administrar el ciclo de vida de la aplicación (SCM)



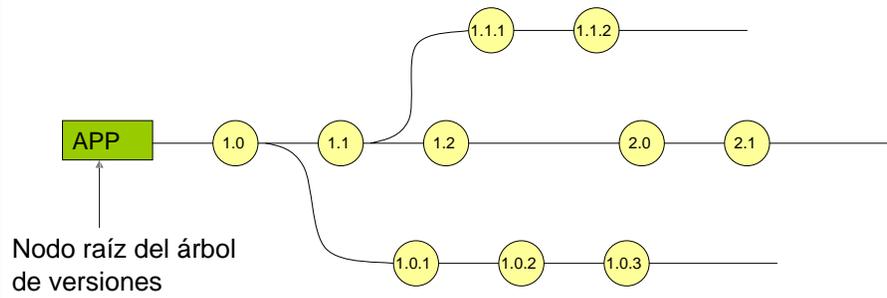
Durante el proceso de construcción de la aplicación, es necesario marcar hitos en el desarrollo de la misma, entendiendo como hitos la “congelación” del desarrollo en un determinado momento especial en el proceso. Esto se puede dar por ejemplo para liberar una versión a producción, congelar una versión entregada a un cliente, la necesidad de congelar un determinado estado especial de la aplicación, etc.

Además también vamos a querer tener distintas líneas de desarrollo de la aplicación, algo muy común por ejemplo cuando se quiere hacer variaciones del proyecto para un cliente o cuando se requiere que dos grupos de trabajo lo hagan en paralelo y necesitamos poder realizar una administración de todos estos elementos.

Lo que necesitamos básicamente es administrar el “ciclo de vida” de la aplicación durante el desarrollo. Varias de estas funcionalidades entran en lo que en el mundo del software se conoce como SCM (Software Configuración Management)

Manejo de Versiones

Solución: Manejo de versiones de la aplicación



GeneXus[®]

Se comienza el desarrollo siguiendo una línea principal de desarrollo (línea del medio – Trunk), lugar donde se agregan las funcionalidades requeridas y se utilizan prototipos para probarlas .

En determinados momentos de este ciclo surge la necesidad de establecer un checkpoint en el proceso, ya sea por la liberación de una versión, la entrega de una versión a un cliente, la necesidad de congelar un determinado estado de una aplicación, etc. Entonces lo que hacemos es congelar el producto en ese momento creando por ejemplo la versión 1.0 que se la entregamos a un cliente y se continúa el proceso de desarrollo principal.

En determinado momento surge la necesidad de realizar correcciones sobre la versión entregada al cliente (1.0) por lo que es necesario abrir una nueva línea de desarrollo para incluir estas correcciones sobre lo que era la versión 1.0 sin afectar la línea de desarrollo principal que siguió creciendo desde el momento de la congelación de la versión 1.0.

Entonces se crea lo que se conoce como Development Version o branch, que es simplemente una nueva línea de desarrollo paralela a la principal.

Luego durante el transcurso del proyecto vuelven a aparecer requerimientos de este tipo, ya sea de determinación de checkpoints como la necesidad de abrir nuevas líneas de desarrollo, entonces por ejemplo creamos la versión 1.1, o la 1.0.1 que vendría a ser un congelado de la línea de desarrollo abierta a partir de la versión 1.0 y así sucesivamente hasta tener por ejemplo la situación planteada en el diagrama.

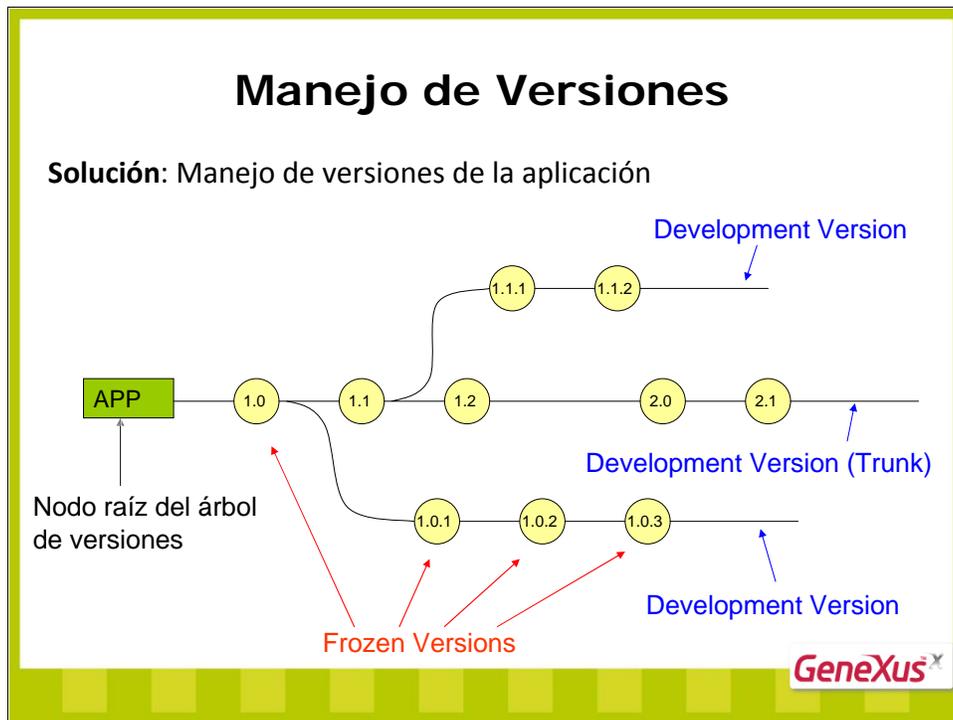
Estas situaciones forman parte de la operativa normal en el desarrollo de una aplicación y es necesario administrar fácilmente este proceso.

Para ello se introduce el concepto de Manejo de Versiones. Las versiones se clasifican en:

- Development Versions, representan las líneas de desarrollo de la aplicación las cuales son independientes entre si, existe una línea principal y varias paralelas, la principal vendría a ser lo que se conoce como Trunk y las demás serían lo que en SCM se conoce como Branches
- Frozen Versions (también conocidas como Labels en SCM), representan los congelados creados en determinados momentos del proceso sobre las DV para determinar ciertos checkpoints (liberación de versión, entrega a cliente, congelar estado, etc.)

Manejo de Versiones

Solución: Manejo de versiones de la aplicación



Se comienza el desarrollo siguiendo una línea principal de desarrollo (línea del medio – Trunk), lugar donde se agregan las funcionalidades requeridas y se utilizan prototipos para probarlas .

En determinados momentos de este ciclo surge la necesidad de establecer un checkpoint en el proceso, ya sea por la liberación de una versión, la entrega de una versión a un cliente, la necesidad de congelar un determinado estado de una aplicación, etc. Entonces lo que hacemos es congelar el producto en ese momento creando por ejemplo la versión 1.0 que se la entregamos a un cliente y se continúa el proceso de desarrollo principal.

En determinado momento surge la necesidad de realizar correcciones sobre la versión entregada al cliente (1.0) por lo que es necesario abrir una nueva línea de desarrollo para incluir estas correcciones sobre lo que era la versión 1.0 sin afectar la línea de desarrollo principal que siguió creciendo desde el momento de la congelación de la versión 1.0.

Entonces se crea lo que se conoce como Development Version o branch, que es simplemente una nueva línea de desarrollo paralela a la principal.

Luego durante el transcurso del proyecto vuelven a aparecer requerimientos de este tipo, ya sea de determinación de checkpoints como la necesidad de abrir nuevas líneas de desarrollo, entonces por ejemplo creamos la versión 1.1, o la 1.0.1 que vendría a ser un congelado de la línea de desarrollo abierta a partir de la versión 1.0 y así sucesivamente hasta tener por ejemplo la situación planteada en el diagrama.

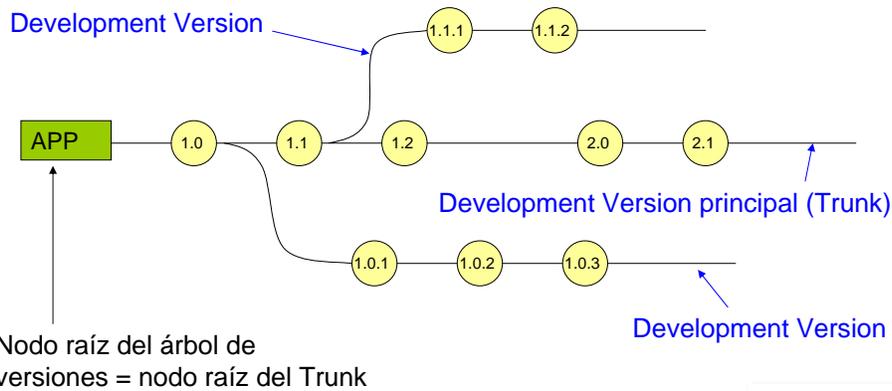
Estas situaciones forman parte de la operativa normal en el desarrollo de una aplicación y es necesario administrar fácilmente este proceso.

Para ello se introduce el concepto de Manejo de Versiones. Las versiones se clasifican en:

- Development Versions, representan las líneas de desarrollo de la aplicación las cuales son independientes entre si, existe una línea principal y varias paralelas, la principal vendría a ser lo que se conoce como Trunk y las demás serían lo que en SCM se conoce como Branches
- Frozen Versions (también conocidas como Labels en SCM), representan los congelados creados en determinados momentos del proceso sobre las DV para determinar ciertos checkpoints (liberación de versión, entrega a cliente, congelar estado, etc.)

Manejo de Versiones

Development Version: (Branch) Línea de desarrollo de la aplicación. Pueden haber varias líneas que transcurran paralelas.



Las development version son las líneas de desarrollo de la aplicación, es decir el lugar donde efectivamente creamos y modificamos la aplicación.

En el ciclo de vida de una aplicación participa una línea de desarrollo principal, es decir, donde comienza el proceso de desarrollo de la aplicación y en la cual normalmente se van a estar haciendo las modificaciones requeridas en el avance del proyecto. En SCM esta línea de desarrollo se conoce con el nombre de Trunk.

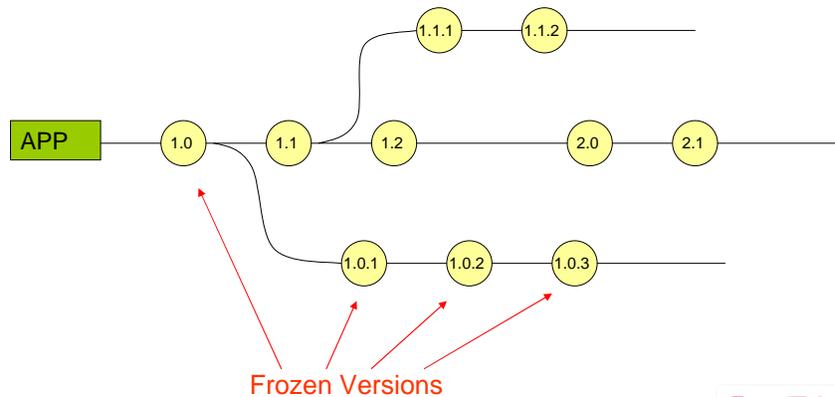
Además de esta línea principal podrán existir una o varias líneas de desarrollo secundarias, totalmente independientes de la línea principal e independientes entre si. En SCM estas líneas de desarrollo secundarias se conocen como Branches y son usadas en general para realizar correcciones o pequeños agregados sobre versiones congeladas o liberadas de la aplicación, o para liberar una versión especial para un cliente.

El desarrollo en cada una de estas development version es independiente, teniendo cada versión sus propios objetos, su propia base de datos, ambientes para generar la aplicación, etc.

Una Development Version, es entonces, una copia de la KB editable e independiente.

Manejo de Versiones

Frozen Version: Versión no modificable, es una “foto” de la aplicación en un momento dado.



GeneXus[®]

Una Frozen Version permite almacenar en forma estática momentos especiales de la KB. Es el elemento que utilizamos para marcar distintos hitos en el proceso, como por ejemplo el “cierre” de una versión para liberarla a los clientes.

Se obtiene a partir de una versión en desarrollo (development version), “congelándola” para obtener una “foto” en un determinado momento.

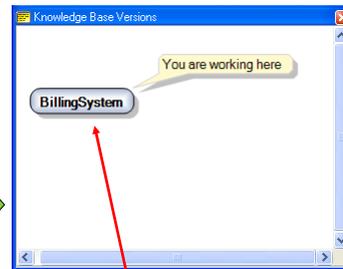
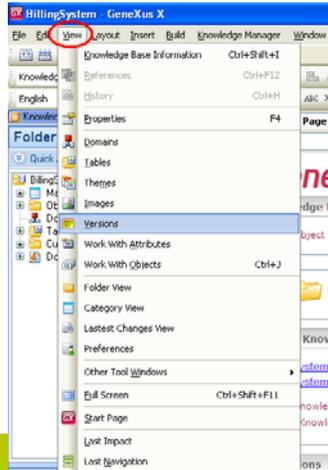
La versión obtenida es Read Only, es decir que objetos de la misma no podrán ser modificados, ni tampoco sus propiedades. Sí será posible realizar acciones relacionadas con la generación de la aplicación, como por ejemplo la creación de la base de datos o la regeneración de los programas.

Cuando congelamos una versión es porque determinamos que la misma está en un estado consistente y sería conveniente guardar dicho estado. Por ejemplo, congelamos una versión X para dársela a los clientes, en determinado momento, mientras se continúa con el proceso de desarrollo, un nuevo cliente requiere de la aplicación, entonces lo que hacemos es generar la misma en la versión X, que sabemos tiene un estado correcto y se la instalamos al nuevo cliente.

Los objetos si bien no pueden ser modificados, pueden ser abiertos para distintas consultas o para realizar comparaciones con otras versiones de la aplicación.

Manejo de Versiones

Para ver el árbol de versiones, abrimos la ventana **Knowledge Base Versions** (View/Versions):



Nodo raíz del árbol de versiones = nodo raíz del Trunk

GeneXus[®]

Manejo de Versiones

Surge la necesidad de “congelar” versiones, para fijar hitos en el proyecto. Para eso creamos Frozen Versions (copia de sólo lectura de la aplicación).



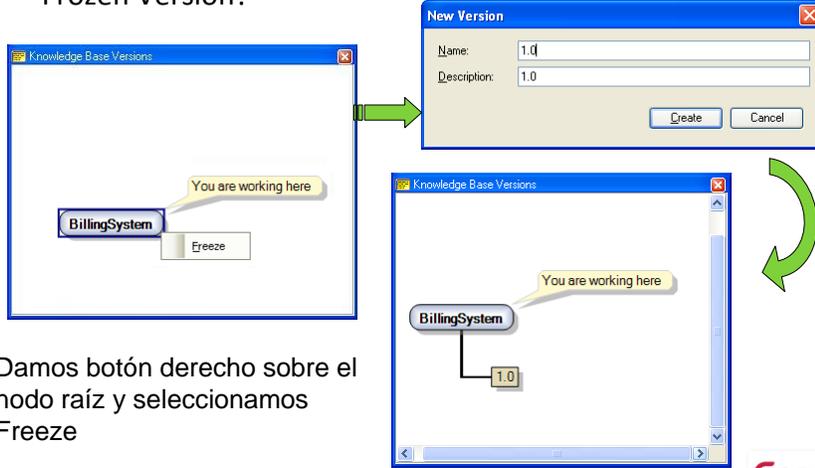
GeneXus[®]

Las Frozen Version sirven para:

- Analizar (no modificar) objetos, propiedades, environments, etc.
- Como fuente de un Reporte de Análisis de Impacto de la base de datos
- Para crear la base de datos
- Para regenerar todos los programas

Manejo de Versiones

¿Cómo “congelamos” una Development Version, para crear una Frozen Version?

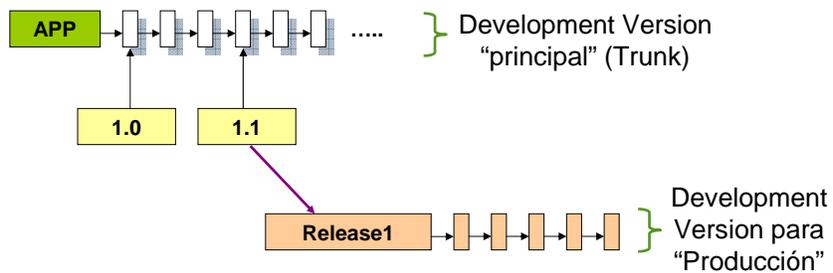


Damos botón derecho sobre el nodo raíz y seleccionamos Freeze

GeneXus[®]

Manejo de Versiones

Ahora queremos poner la aplicación en Producción. Para eso partir de una Frozen Version 1.1 creamos una Development Version "Release 1".



En la versión de "Producción" se van produciendo variaciones debido a arreglos, pero no se agregan funcionalidades nuevas. Las mismas son agregadas en la línea de desarrollo principal.

Las Development Version sirven para:

Trabajar en una línea de desarrollo paralela a la principal

Como fuente o destino de una operación de Revert desde una Frozen Version de Backup

Manejo de Versiones

Creamos la Frozen Version 1.1 como vimos anteriormente...

The image illustrates the process of creating a new frozen version in a knowledge base. It consists of three main components:

- Left Panel:** A window titled "Knowledge Base Versions" showing a tree structure. The root node is "BillingSystem", which has a child node "1.0". A yellow callout bubble above "BillingSystem" says "You are working here". A "Freeze" button is visible next to the "BillingSystem" node.
- Top Panel:** A "New Version" dialog box with "Name: 1.1" and "Description: 1.1". It has "Create" and "Cancel" buttons. A green arrow points from the "Freeze" button in the left panel to this dialog.
- Right Panel:** The same "Knowledge Base Versions" window after the action. The tree structure now includes a new node "1.1" above the "1.0" node. A red arrow points to the "1.1" node, which is labeled "Frozen Version 1.1". A green curved arrow points from the "1.1" node back to the "BillingSystem" node.

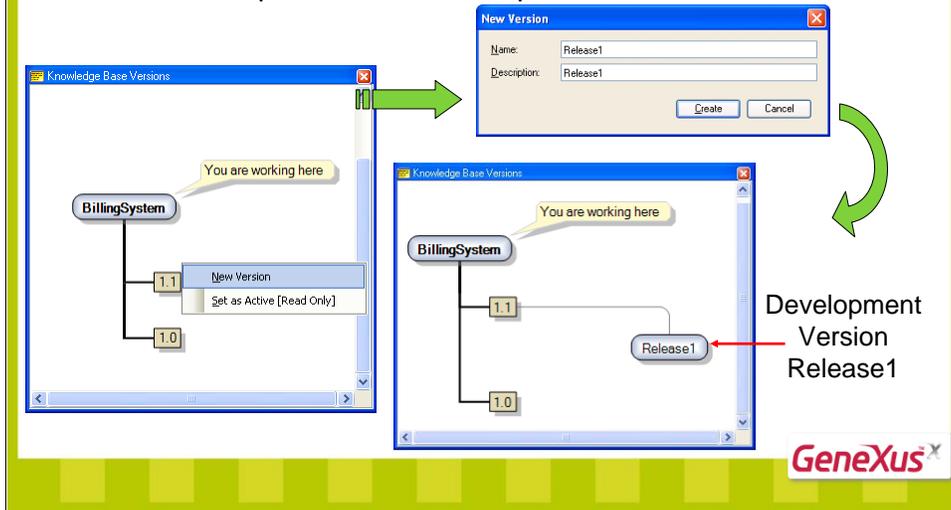
Below the left panel, the text reads: "Botón derecho sobre el nodo raíz y click en Freeze".

The "GeneXus" logo is located in the bottom right corner of the screenshot area.

Nótese que las Frozen Version más nuevas, se muestran más arriba en el árbol de versiones.

Manejo de Versiones

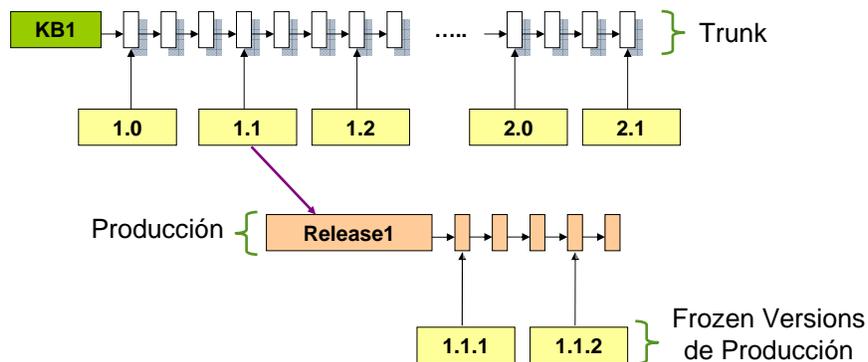
Damos botón derecho sobre la Frozen Version 1.1 y seleccionamos New Version para crear la Development Version Release 1.



El tiempo que se demora en crear una nueva Development Version es proporcional al tamaño de la KB.

Manejo de Versiones

También podemos crear nuevas Frozen Versions tanto en la rama del desarrollo principal como en la rama de Producción.



GeneXus[®]

Como las líneas de desarrollo del Trunk (Desarrollo) y de Release 1 (Producción) son paralelas, los cambios en una no afecta a la otra.

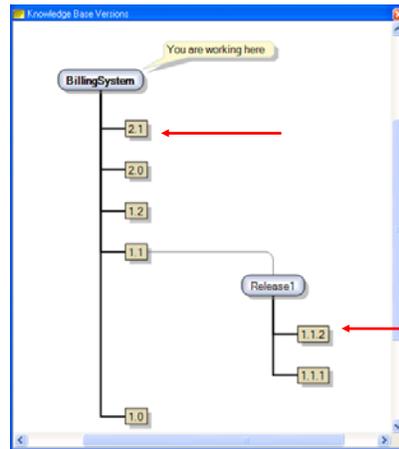
Ambas versiones son entonces totalmente independientes y podemos requerir congelarlas por diferentes motivos. Por ejemplo, en el caso de la rama de Producción, para fijar un estado luego de ciertos arreglos que tuvimos que hacer.

De acuerdo a la metodología adoptada, en el ciclo de desarrollo principal, es donde se agregan nuevas funcionalidades, arreglos, cambios importantes a la aplicación, prototipado y testing. Es más frecuente que se necesita "fotos" en esa etapa viva del desarrollo de la aplicación.

En la rama del Release1, los cambios son menores, más bien arreglos circunstanciales que no agregan funcionalidad. En este caso, es menos frecuente la necesidad de crear Frozen Versions, pero puede ser igualmente necesario.

Manejo de Versiones

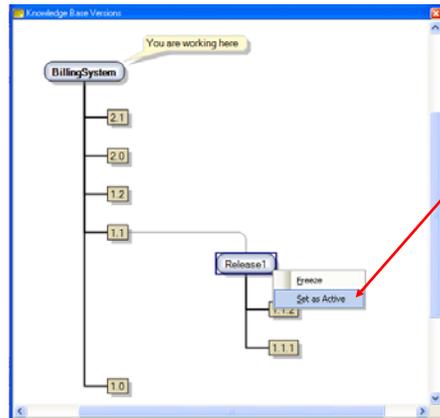
Nótese que las Frozen Versions más nuevas, se muestran más arriba en el árbol de versiones



GeneXus[®]

Manejo de Versiones

Luego de ciertos arreglos en Producción, nos interesa generar la aplicación. Para eso marcamos la DV Release 1 como activa.



Damos botón derecho sobre el nodo Release1 y elegimos "Set Active"

GeneXus[®]

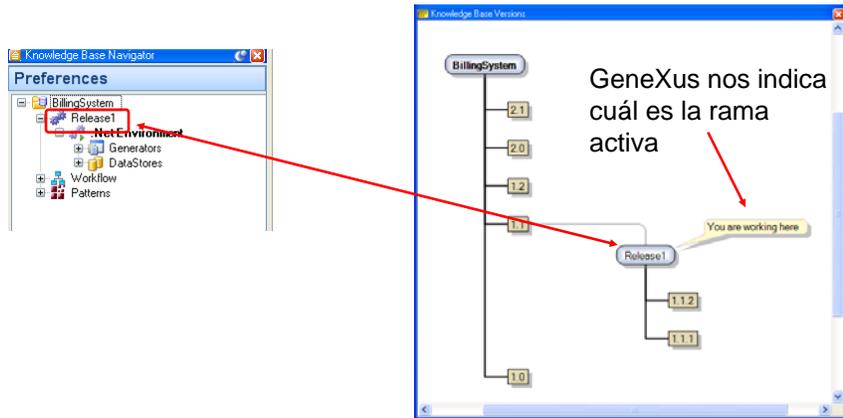
GeneXus genera automáticamente los programas y las estructuras de la BD, partiendo de la versión que esté activa.

Se puede marcar como activa una versión en desarrollo o una versión congelada. En éste último caso, no podremos hacer ninguna modificación a la misma, solamente utilizarla para generar la aplicación o para realizar un impacto a la base de datos, o para comparar versiones.

Solamente puede haber una versión activa a la vez.

Manejo de Versiones

La Development Version que esté activa, será la que se utilizará para generar la aplicación al hacer un Build (F5)



GeneXus nos indica cuál es la rama activa

GeneXus^x