

# GeneXus Server

Cómo automatizar procesos para mejorar el ciclo de Desarrollo de las aplicaciones



Lo primero que debemos hacer es situarnos en el escenario y el problema que queremos resolver.

Actualmente, se requiere que el desarrollo de software sea ágil y continuo. ¿Por qué? Porque el mundo y los negocios están en continuo cambio y crecimiento, y por lo que el desarrollo de software debe acompañar ese ritmo de cambio y crecimiento.

Esto significa hacer más rápido el ciclo de desarrollo de las aplicaciones y realizar entregas continuas de alta calidad.

Para lograrlo, se propone un conjunto de prácticas que agrupan el desarrollo de software y las operaciones de TI. Dentro de estas prácticas, hay un punto muy importante, y es la posibilidad de integrar en los procesos en forma automática, tareas de codificación, construcción y testeado.

En nuestro caso también hemos adoptado esta práctica que permite entregar GX en forma más rápida, liberando upgrades cada 2 meses y conociendo en cada momento el estado del producto.

Entonces, en estos minutos les quiero hablar sobre el concepto de Integración continua, y contarles cómo trabajar con pipelines de Integración Continua asociados a Jenkins desde GX y GXserver. Vamos a ver también cómo adaptarlos a diferentes ambientes en caso de utilizar diferentes servidores.

# DevOps

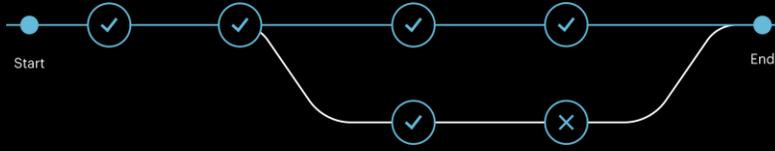


Entonces, vamos a empezar por el concepto de DevOps

¿Qué se entiende por DevOps? Es ese conjunto de prácticas que agrupan el desarrollo de software y las operaciones de TI

Con respecto a los procesos relacionados con DevOps, es necesario ver cuáles son las tareas necesarias para lograr un proceso ágil y automatizar todas las tareas posibles.

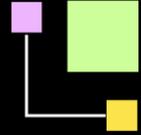
---



Para eso es necesario definir Pipelines.

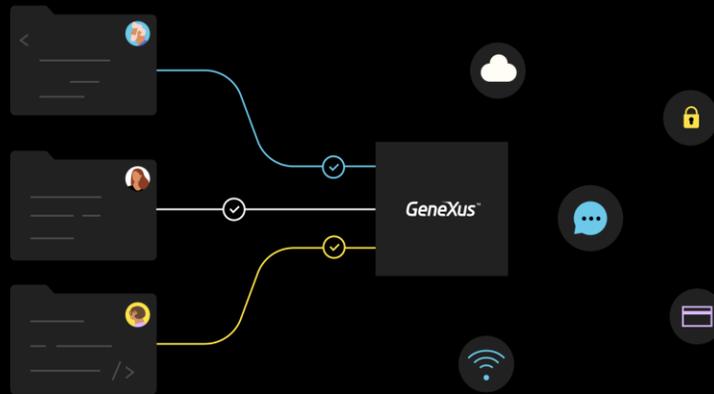
En software, un Pipeline representa una cadena de elementos de procesamiento dispuestos de modo que la salida de cada elemento sea la entrada del siguiente. Esto implica que se divide la entrega de software en varias etapas, lo que permite validar las nuevas funciones y evita que los errores afecten a los usuarios.

# Integración continua



Pasemos ahora al concepto de Integración Continua ¿Qué se entiende por Integración continua?

Es la integración en los procesos de forma automática, de tareas de codificación, construcción y testeo.



La integración continua debe garantizar que si un desarrollador escribe código, ese código se prueba y funciona primero individualmente y luego se integra en todo el sistema, y también debe garantizar que el código escrito no rompa nada en el sistema.

El objetivo de la Integración Continua es que el trabajo se integre con frecuencia, de manera que, en lo posible, se eviten los conflictos y si aparecen son pequeños y fáciles de solucionar.

Todo esto debe hacerse de forma automática. Por lo tanto, la integración continua es un punto muy importante en el proceso de DevOps.



# GeneXus™ 17

Bien.. entonces, ¿a dónde queremos llegar?

A partir de GeneXus 17, es posible crear y monitorear procesos de integración continua tanto desde el propio IDE de GeneXus como desde la consola web de GeneXus Server.





# GeneXus™ Server

Para integrar el desarrollo, contamos con GeneXus Server, que junto con las tareas de MSBuild, permite automatizar todos los pasos relacionados con la integración continua de la aplicación.

Para cada tarea que se puede ejecutar desde GeneXus, hay una tarea MSBuild que permite hacer lo mismo pero en forma automática.

Por supuesto, para habilitar esta funcionalidad, es necesario realizar ciertas configuraciones. Toda la información necesaria se encuentra en nuestro wiki:

---



# GeneXus & Jenkins



Antes de ver una demo, hablemos un poco de la integración de GeneXus con Jenkins.

Jenkins es un servidor gratuito y de código abierto que ayuda a automatizar partes del proceso de desarrollo de software, con Integración Continua. En el proceso de configuración del entorno, que mencionamos anteriormente, se establece la conexión entre GeneXus Server y Jenkins.

Esta integración de GeneXus con Jenkins permite crear, administrar y monitorear desde un servidor Jenkins, pipelines dentro de una instancia de GeneXus Server o directamente desde dentro del propio IDE de una KB cliente.

---

## Demo

Ya tenemos la KB TravelAgency hosteada en GeneXus Server

Si observamos el menú, vemos que se agregó la opción Continuous Integration. Desde aquí podemos ver los pipelines creados para dicha Kb, con toda su información relacionada.

Vamos entonces a crear un nuevo Pipeline. Vemos que el nombre de la Kb asociada ya está definido porque estamos dentro de la KB TravelAgency

Elegimos entonces

- La versión de la KB para la cual crear el pipeline
- El environment que se tenga definido en la Kb
- Ponemos un nombre al pipeline. En este ejemplo ponemos NewPipeline
- Indicamos además la frecuencia con la cual queremos que se ejecute
- Y la versión de GeneXus a utilizar

También tenemos opciones para indicar.

Por ejemplo, si lo único que quiero es hacer un Build y compilación de la KB, o también Deploy to cloud, y en ese caso crea la base de datos, compila, ejecuta y además ejecuta los tests que tenga definidos en la KB.

En este ejemplo estamos indicando que queremos hacer deploy to cloud y

---

ejecutar los tests que tenga definidos en la KB.

Una vez que el pipeline se crea aparece en la lista de pipelines disponibles, si bien aún no se ejecutó.

Se indica cuando se va a ejecutar y también lo puedo ejecutar en forma manual con la opción Run.

También puedo indicar si quiero ejecutar los tests y hacer Force rebuild/

Una vez que el pipeline se esta ejecutando, el estado cambia a Building...

Además de trabajar con los pipelines asociados a una Kb en particular, también es posible trabajar con todos los pipelines definidos a nivel del server, asociados a todas las Kbs, e incluso se puedo filtrar por los míos.

Ya podemos ver que nuestro nuevo pipeline termino de ejecutarse y el resultado fue satisfactorio.

Cuando ingresamos a un pipeline, vemos toda su información asociada: la kb a la que esta asociado, la versión, el usuario que lo creó, las opciones elegidas para ejecutar

Es posible también abrir la kb, editar el pipeline y acceder también a la aplicación que quedo generada por el deploy to cloud.

Podemos también ir directamente a Jenkins donde también veo el pipeline definido, y ver la ejecución de cada uno de los pasos

Por ejemplo, se pueden ver los cambios que se registraron en la KB cuando se hizo el Build. Esto se ve en un log que genera Jenkins.

También podemos ver el resultado del Build, y recordemos que en este ejemplo se indicó que se generara el deploy to cloud y se ejecutaran los tests asociados a la KB... y aquí podemos ver el resultado de esos tests, si fue satisfactorio o no. En esta Kb hay incorporados Unit test y Tests de UI, por lo tanto, en el output vemos el resultado de cada uno de ellos.

Ahora bien, todas estas opciones podemos definirlas desde el ide de GeneXus, a través del nuevo tab Continuous Integration agregada al cuadro Team Development.

Ese tab nos permite ver la misma información que veíamos recién desde la consola de GeneXus server.

Vemos todos los pipelines definidos... el estado de cada uno... la cantidad de ejecuciones que tuvo cada uno y su estado

También podemos ver información sobre los cambios que hubo en la ejecución de

ese pipeline, del detalle de los commits que se realizaron

Acá por ejemplo se ve que se ejecutaron 3 commits, por lo tanto, se realizó update de la KB antes de realizar el Build

Podemos también ejecutar desde aquí el pipeline en forma manual... y también ir a Jenkins para ver el detalle como vimos anteriormente.

Desde GX también podemos crear nuevos pipelines, como lo hicimos desde la consola del server: seleccionar la versión, el environment, cada cuanto queremos que se ejecute y las opciones que se desean ejecutar (si hacer o no deploy to cloud, y ejecutar o no los tests definidos en la KB).



# Ejecución en otros ambientes

Bien, ahora que ya sabemos cómo trabajar con pipelines desde GX y GXserver asociados a Jenkins, pensemos lo siguiente:

¿Podremos definir y ejecutar pipelines en diferentes ambientes? La respuesta es si.



**Pipeline**

Definition **Pipeline script**

Script

```
24
25
26
27      /* tools */
28      genexus = tool name: "${env.gxInstallationId}", type: "genexus"
29      msbuild = tool name: "${env.msbuildId}", type: "msbuild"
30      ciscripts = tool name: 'CIScripts', type: 'com.cloudbees.jenkins.plugins.customtools.CustomTool'
31      defaultkbpath = tool name: 'DefaultKBPath', type: 'com.cloudbees.jenkins.plugins.customtools.CustomTool'
32
33      /* additional variables */
34      sqlIntegratedSecurity = "${env.sqlserverCredentials == 'none' ? true : false}"
35      buildScript = "${env.ciscripts}\\Build\\General.msbuild"
36      localSettings = "${env.ciscripts}\\KBSettings\\${env.LocalSettingsFile}"
37      relativeTestResultsFile = "Tests\\TestResults.xml"
38      fullTestResultsFile = "${WORKSPACE}\\${relativeTestResultsFile}"
39      KBPath = "${env.defaultkbpath}\\${env.localKbPath}"
40
```

Use Groovy Sandbox

[Pipeline Syntax](#)

Por defecto, GeneXus y GeneXus Server arman automáticamente un pipeline que hace Build y Deploy to cloud, pudiendo incluso ofrecer un ambiente de prueba básica del deploy to cloud, lo cual resuelve de por sí un escenario importante.

Ahora bien, en dicho pipeline figuran llamadas a scripts MSbuild, y se pueden modificar tanto los pipelines como dichos scripts.

Esto hace que a partir del pipeline inicial generado por GeneXus, es posible modificarlo y resolver otros escenarios



**Jenkins** cd Blog Documentation **Plugins** Community Subprojects About English Download

Sort Relevance  
 Relevance  
 Most installed  
 Trending  
 Title  
 Release date

Categories [Show all](#)

- Platforms
  - iOS Development
  - NET Development
  - Android Development
  - Ruby Development
- User interface
  - User Interface
  - List view columns
- Administration
  - Agent Management
  - Page Decorators
  - Authentication and User Management
  - Cluster Management
  - Command Line Interface
- Source code management
  - Source Code Management
  - Source Code Management related

**Browse** Find plugins...

Search by **algolia**

1 to 50 of 1836

Previous **1** 2 3 4 5 6 ... 36 37 Next

<p><b>Mailer</b>                      Installs: 277858                      Released: 6 months ago</p> <p>Built Tools, email                      This plugin allows you to configure email notifications for build results</p> <p>Andres Rodriguez                      Jesse Glick                      (4 other contributors) <b>Ma</b></p>	<p><b>Script Security</b>                      Installs: 277766                      Released: about a month ago</p> <p>api-plugin, Security                      Allows Jenkins administrators to control what in-process scripts can be run by less-privileged users.</p> <p>Jesse Glick                      Kohsuke Kawaguchi                      (7 other contributors) <b>SS</b></p>	<p><b>Credentials</b>                      Installs: 277574                      Released: 3 months ago</p> <p>api-plugin                      This plugin allows you to store credentials in Jenkins.</p> <p>Daniel Beck                      Jesse Glick                      (5 other contributors) <b>Cr</b></p>	<p><b>SSH Credentials</b>                      Installs: 276667                      Released: 3 months ago</p> <p>api-plugin                      Allows storage of SSH credentials in Jenkins</p> <p>Daniel Beck                      Jesse Glick                      (4 other contributors) <b>SC</b></p>	<p><b>SCM API</b>                      Installs: 276559                      Released: about a month ago</p> <p>api-plugin, Source Code Management                      This plugin provides a new enhanced API for interacting with SCM systems.</p> <p>Jesse Glick                      Manuel Racema Soto                      (7 other contributors) <b>SA</b></p>
<p><b>Structs</b>                      Installs: 275425                      Released: 4 months ago</p> <p>api-plugin                      Library plugin for DSL plugins that need names for Jenkins objects.</p> <p>Jesse Glick                      Kohsuke Kawaguchi                      (8 other contributors) <b>St</b></p>	<p><b>JUnit</b>                      Installs: 274980                      Released: about 7 hours ago</p> <p>Build Reports                      Allows JUnit-format test results to be published.</p> <p>Andrew Bayer                      Andres Rodriguez                      (12 other contributors) <b>JU</b></p>	<p><b>Pipeline: Step API</b>                      Installs: 274013                      Released: 2 months ago</p> <p>api-plugin, Miscellaneous                      API for asynchronous build step primitive.</p> <p>Jesse Glick                      Kohsuke Kawaguchi                      (7 other contributors) <b>PS</b></p>	<p><b>Matrix Project</b>                      Installs: 273303                      Released: 3 months ago</p> <p>Miscellaneous                      Multi-configuration (matrix) project type.</p> <p>Antonio Mufiz                      Jesse Glick                      (2 other contributors) <b>MP</b></p>	<p><b>Display URL API</b>                      Installs: 273039                      Released: 4 months ago</p> <p>api-plugin, User Interface                      Provides the DisplayURL Provider extension point to provide alternate URLs for use in notifications</p> <p>James Dumay                      Michael Neale                      (9 other contributors) <b>DU</b></p>

También es posible modificar los pipelines para hacer otras tareas, usar otros plugins, y en definitiva todo Jenkins.



GeneXus + ???



Y por último mencionar también que toda la integración de Jenkins, con MSbuild y GeneXus son plugins de código abierto, y así como está hecha esta integración con Jenkins puede surgir en el futuro (o incluso crearlo ustedes) la integración con otros motores

---

# GeneXus™

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)  
[training.genexus.com/certifications](http://training.genexus.com/certifications)