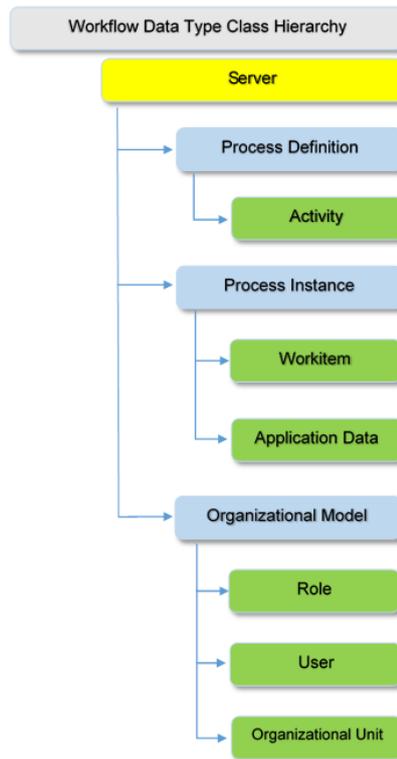


## API Workflow avanzado

La mayor parte de la API está organizada en una jerarquía, con el Server en la parte superior. Eso significa que el Server representa el punto de entrada a todas las funciones que hemos visto hasta ahora, incluidos los modelos, la ejecución de esos modelos y la información de usuario de las personas que los ejecutan. En este video en particular, nos centraremos en este objeto y las características que nos permite incluir en nuestro código GeneXus

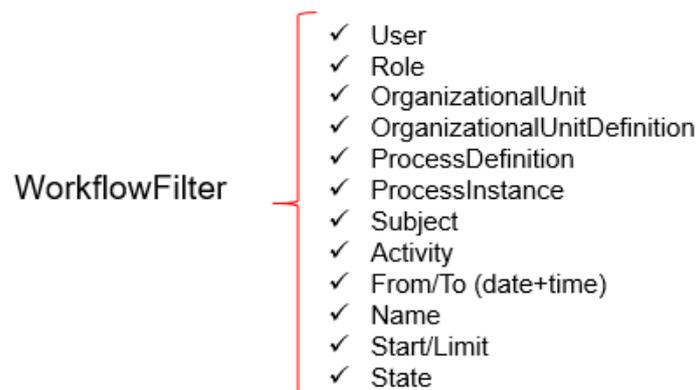


**WorkflowServer** tiene una serie de propiedades que nos permiten obtener información sobre el entorno de ejecución. Estas incluyen:

- La sesión de trabajo actual del usuario registrado
- La información del usuario y
- La configuración global para todo el servidor.

Además de esas propiedades, el objeto Workflow Server nos da acceso a una serie de métodos y funcionalidades, que son:

- **Connect and disconnect:** Conectar y desconectar en una sesión de GXflow, proporcionando el nombre de usuario y contraseña. Esta sesión es la que usará el resto del código en GeneXus para ejecutar y el usuario será el que regrese en `Server.CurrentUser`, por lo que algunas funcionalidades que se describen a continuación podrían fallar si el usuario actual no tiene los privilegios adecuados.
- **ListProcessDefinitions:** Lista todas las definiciones de proceso con una serie de filtros, en un orden particular, o accediendo directamente a una única definición de proceso usando su ID o nombre.
- **ListActivities:** Lista todas las actividades en un proceso, usando algunos filtros adicionales o accediendo directamente a uno usando su identificación o nombre.
- **ListProcessInstances:** Lista todas las instancias (o ejecuciones) de procesos de flujo de trabajo usando algunos filtros, o accediendo solo a uno usando su ID.
- **ListWorkitem:** Lista todos los workitems utilizando algunos filtros o accediendo a uno solo.
- **GetCalendar:** Permite acceder a todos los calendarios definidos en un Servidor, con sus definiciones de días festivos y días laborables, así como acceder a uno solo si conocemos su id o nombre
- **GetNode:** Permite el acceso a todos los nodos de proceso, es decir, obtener información sobre cada actividad, cada condición, nodo de evento y los vínculos entre ellos.
- **GetOrganizationalModel:** Permite acceder al modelo organizacional definido en el Servidor
- **GetEventsRepository:** Permite acceder al repositorio de eventos, para acceder a la información de registro del Servidor.
- **GetDocumentsrepository:** Permite acceder al repositorio de contenido de los documentos en el servidor.



Para varias características, el tipo de datos WorkflowFilter es una parte importante del uso adecuado de la funcionalidad. El objeto es un súper conjunto de todos los filtros posibles que podría ser interesante para nosotros incluir al acceder a una serie de instancias, workitems o algunos objetos. Sin él, nos veríamos obligados a filtrar la aplicación en la colección completa, lo que puede ser extremadamente ineficaz en entornos de producción que a menudo funcionan con cientos de procesos y millones de workitems. No todos los filtros se aplican a todas las funciones, pero es útil saber cuáles se aplican en cada lugar para maximizar nuestro rendimiento en el uso de la API. Sin embargo, también es importante saber que el uso de un filtro que no se aplica a una función no desencadena un error.

- El filtro **User** : Se aplica a todas las entidades reguladas por la presencia de un usuario, como workitems asignados, eventos activados o documentos asignados.
- El filtro **Rol**: Se aplica a entidades reguladas por roles, como candidatos para una actividad o usuarios con algún rol asignado.
- El filtro **OrganizationalUnit**: Se aplica a las entidades reguladas por ellas, como usuarios o roles.
- El filtro **OrganizationalUnitDefinition**: Se aplica a la lista de organizaciones.
- El filtro **ProcessDefinition**: Se aplica a todas las entidades que incluyen ProcessDefinition. Estos incluyen workitems, instancias de procesos, actividades y roles de candidatos.
- El filtro **ProcessInstance**: Se aplica a las entidades definidas por una instancia, que incluyen documentos, workitemsy usuarios asignados.
- El filtro **Subject**: Se aplica al tema de las instancias y los workitems. Este filtro distingue entre mayúsculas y minúsculas.
- El filtro **Activity**: Se aplica a workitems y roles candidatos de esa actividad.
- Los filtros **From/To**: Definen un límite inferior y superior para la fecha de creación de documentos, eventos, instancias de proceso y workitems.

- El filtro **Name**: Se aplica al nombre descriptivo de los tipos de documentos, el título de los documentos y los nombres de los roles, restricciones, usuarios, tareas y definiciones de procesos. Todos estos filtros distinguen entre mayúsculas y minúsculas.
- **Start/limit**: Se utilizan para mejorar el rendimiento de obtener varios elementos, para usarlos en funcionalidades como informes o paginación. Start especifica el desplazamiento del conjunto de datos recuperado, mientras que Limit especifica la cantidad de objetos devueltos. La mayoría de los DBMS modernos admiten esta funcionalidad a nivel del servidor, en lugar del controlador, mejorando el ancho de banda y el tiempo de transferencia con la base de datos, aunque pueden existir algunas restricciones para su uso. Por ejemplo, SQL Server admite esta funcionalidad desde 2012, pero requiere que se especifique un pedido.
- **State**: Incluye objetos que se ven afectados por el motor de workflow , por lo que se puede utilizar para filtrar definiciones de procesos, instancias, workitems y documentos. Este filtro es bastante complejo, ya que tiene una serie de valores posibles, y no todos se aplican a los diferentes tipos de objetos, así que profundicemos un poco más:

Cuando se utilizan con definiciones de proceso, se pueden habilitar (es decir, se pueden crear nuevas instancias) o inhabilitar (para que no se puedan crear nuevas instancias)

<b>WorkflowProcessDefinitionState</b>
ENABLED
DISABLED

Cuando se utilizan con instancias de procesos, pueden ser:

**OPEN\_RUNNING**: Cuando una instancia se crea y ejecuta normalmente.

**OPEN\_NOTRUNING\_SUSPENDED**: Cuando se ejecuta la funcionalidad de Suspenden en la instancia, el estado cambia a NOTRUNING\_SUSPENDED. Además, todos los subprocesos y workitems activos se cambian a suspendidos para garantizar que ninguna parte de la instancia permanezca activa.

**CLOSED\_COMPLETED**: este es el estado utilizado cuando una instancia se completa normalmente.

**CLOSED\_ABORTED**: Similar a suspendido, cuando se ejecuta la funcionalidad de Abortar la instancia, dicha instancia y todos sus componentes se configuran en este estado.

**CLOSED\_TERMINATED**: este estado se utiliza cuando la instancia finaliza de forma inesperada, especialmente en una fecha límite

Cuando se usa en un workitems, el filtro de estado puede ser:

**OPEN\_ACTIVE\_READY:** Cuando se crea el workitem

**OPEN\_ACTIVE\_ASSIGNED:** Cuando ha sido asignado o tomado por un usuario

**OPEN\_ACTIVE\_INPROCESS:** Mientras el workitem está en ejecución, es decir, después de que se tomó pero antes de que se envié.

**CLOSED\_COMPLETED:** Cuando el elemento de trabajo se devolvió y se consideró completado.

Volviendo a los filtros:

Los filtros **CreatedFrom** y **CreatedBy** funcionan de manera similar a From/To. De hecho, From/To se mantienen por motivos de compatibilidad.

Los filtros **EndedFrom** y **EndedBy** funcionan de manera similar a los anteriores, pero para objetos que tienen una fecha de finalización, como workitems e instancias. Es importante detallar que las instancias que están abiertas actualmente serán reducidas por estos filtros si se especifica. En caso de que esté interesado en las instancias abiertas actualmente, debe usar el filtro State en su lugar.

**Search**, se puede utilizar para filtrar una combinación de diferentes atributos en diferentes entidades: para workitems, filtra por subject, su nombre, el nombre de su proceso, el nombre de la tarea o el nombre del usuario; para los usuarios, filtra por código, nombre o correo electrónico; para restricciones, filtra por código o valor; para procesos, filtra por nombre y descripción; por ejemplo, filtra por subject,, el nombre de su proceso o el nombre de su usuario y para los documentos, su título o su responsable.

**WorkflowPriority**, se puede utilizar para filtrar workitems instancias por su prioridad establecida, las opciones son:

- **High priority**, prioridad alta.
- **Normal priority**, normal y
- **Low priority** baja

Otro objeto que tenemos que considerar al programar con GeneXus en la API es **WorkflowContext**. Generalmente, los objetos que deben usarse como Gateways requieren los

siguientes parámetros: Process Definition, Process Instance y Workitem, para identificar en qué parte del flujo de trabajo nos encontramos.

**parm(in:&WorkflowProcessDefinition, in:&WorkflowProcess, in:&WorkflowWorkitem, out:&ConditionalCode);**

Desde las últimas versiones de GeneXus, hemos introducido WorkflowContext , el cual se define automáticamente en cada transacción y webpanel que se llama desde workflow e incluye:

**ProcessDefinition:** La definición de proceso que se está ejecutando actualmente

**ProcessInstance:** La instancia de proceso que se está ejecutando actualmente

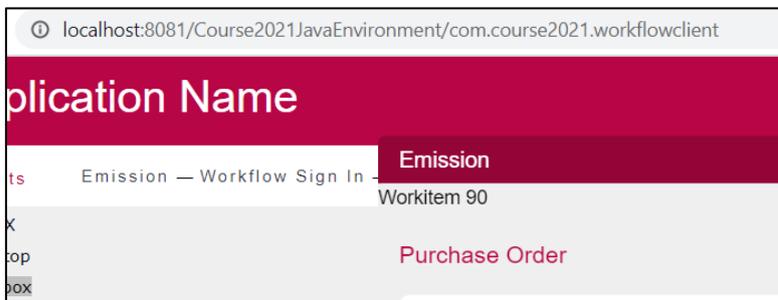
**Workitem:** El elemento de trabajo actual

Y también la funcionalidad para verificar los derechos de un usuario con una actividad en particular.

Esto significa que el mismo código en el mismo objeto se comportará de manera diferente si se ejecuta dentro de la bandeja de entrada o como una URL externa

```
Event Start
    msg(!"Workitem "+&WorkflowContext.Workitem.Id)
EndEvent
```

Bandeja de entrada



URL externa



El tipo de datos WorkflowUser amplía las propiedades que definen a un usuario en Workflow, que incluyen:

- **UserName:** Debe ser único.
- **FullName**
- **Email** (que es opcional y no único).
- **AccessLevel**, que define si el usuario puede ejecutar esta API sobre otros usuarios o solo sobre sí mismo.
- Un conjunto de roles, **OrganizationalUnits**, **Restrictions**, **ExternalCode**, que se utiliza para interactuar con otros módulos, como **GeneXus Access Manager** y si el usuario se encuentra actualmente conectado, bloqueado o "Fuera de oficina". En caso de que se encuentre fuera de oficina, las propiedades también incluyen su sustituto.

Además de esas propiedades, el tipo de datos incluye muchas funciones.

Puede obtener la worklist actual del usuario, **GetWorklist**, o la lista de procesos que el usuario tiene derecho a iniciar, utilizando los filtros descritos anteriormente. Siempre que su usuario tenga suficiente nivel de acceso, también puede obtener los roles definidos dentro de la unidad organizacional que tiene el usuario. Puede asignar o eliminar roles, unidades organizacionales o restricciones del usuario.

Como administrador, también puede bloquear o habilitar un usuario, así como configurarlo como "Fuera de la oficina", en cuyo caso también puede seleccionar otro usuario como sustituto.

Finalmente, puede forzar la reconstrucción de la worklist del usuario (**RebuildWorklist**). Al hacerlo, se volverán a aplicar todas las reglas relacionadas con los roles candidatos para los workitems que pueda tener el usuario, considerando las restricciones y unidades organizacionales adecuadas para, en esencia, actualizar la lista de workitems que un usuario puede tomar y ejecutar.

La instalación del servidor está configurada de forma predeterminada para realizar estos cálculos para cualquier cambio en los derechos de los usuarios, como pueden consumir muchos recursos y tiempo en entornos con decenas de roles, cientos de usuarios. y millones de workitems, tiene la opción de deshabilitar la reconstrucción automática y activarla manualmente bajo ciertas condiciones (por ejemplo, cuando un usuario inicia sesión)

-WorkflowRole	}	✓ Name/Value
-WorkflowRestriction		✓ AddUser
-WorkflowOrganizationalUnit		✓ RemoveUser

El último conjunto de tipos de datos que nos gustaría discutir son Workflowrole, WorkflowRestriction y WorkflowOrganizationalUnit. Incluso cuando son conceptualmente diferentes y tienen diferentes casos de uso, comparten la característica de que todos son modificadores para definir lo que un usuario puede o no puede hacer.

Como tal, todos tienen un nombre o valor, y también las funcionalidades para agregar y eliminar usuarios a sus metadatos