Business Process Management Suite

GeneXus™ 15

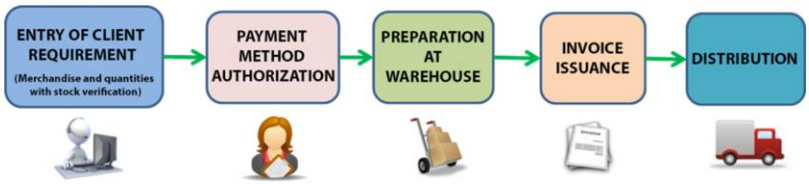
GeneXus

Brief theoretical introduction

- What is a business process?

  A business process is a set of logically related tasks which are performed to achieve a business result.

- Example



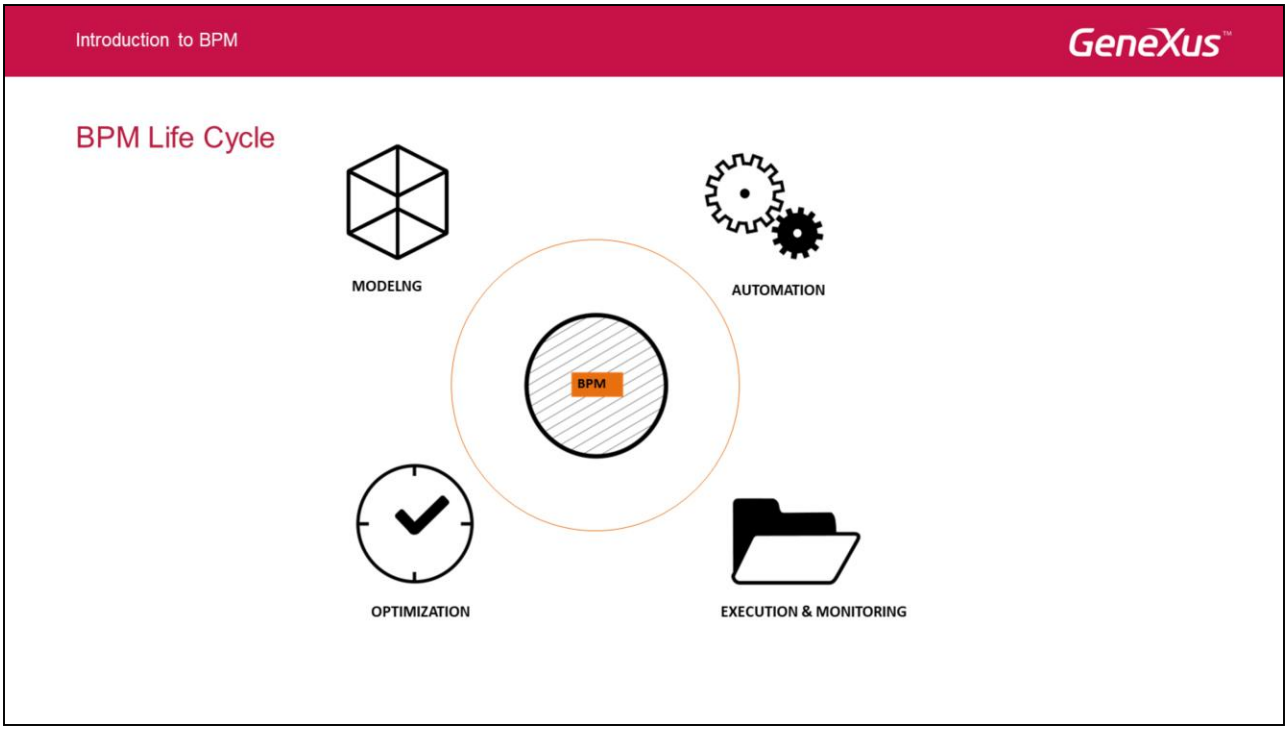
- What is BPM?

  The BPM (Business Process Management) is:
  - a methodology
  - a set of management tools

Um **processo de negócio** é um conjunto de tarefas relacionadas logicamente, executadas para alcançar um resultado de negócio.

O que se mostra no exemplo é um processo de negócio desenvolvido em uma empresa que vende mercadoria por atacado. Podemos observar claramente a cadeia de tarefas para executar o processo da venda.

BPM (Business Process Management) é uma metodologia focada em administrar e otimizar de forma contínua as atividades e os processos de negócios da organização, incluindo práticas e políticas de gestão, métricas e processos de controle, bem como ferramentas de software que dão suporte a todas essas atividades.

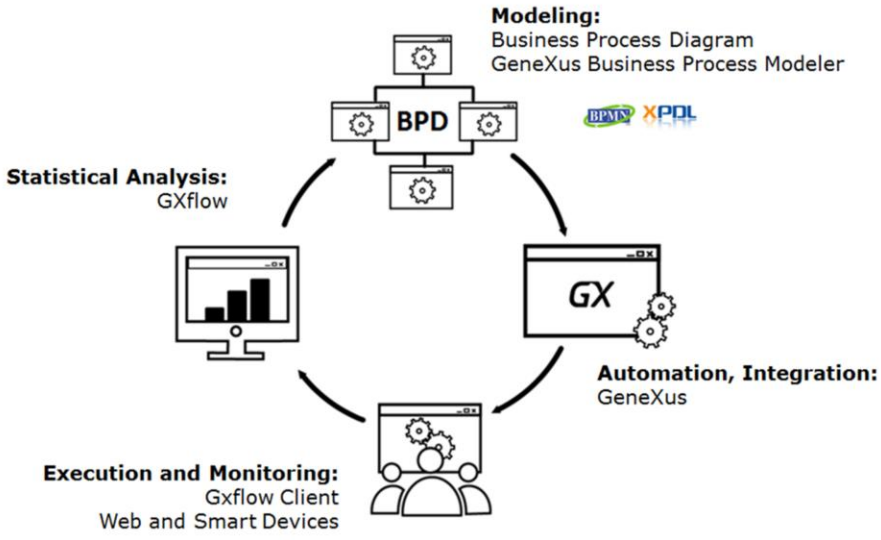


O **ciclo de BPM** começa na etapa de **modelagem**, onde descrevem-se os processos de negócio da organização. Geralmente utilizam-se ferramentas gráficas para compreender melhor os processos.

Logo vem a etapa de **automatização** destes processos de negócio para o qual utilizamos ferramentas de software.

A continuação começa na **execução e monitoramento** destes processos onde analiza-se o quão eficientes são, quais são os gargalos e que melhorias podemos implementar para melhorar no próximo passo para obter os resultados desejados.

Como é um ciclo, a metodologia orientada a processos é um **projeto que não tem fim** e o foco é a **melhora contínua** da organização.


Modeling:
Business Process Diagram
GeneXus Business Process Modeler
BPD
BPMN
XPDL
Statistical Analysis:
GXflow
GX
Automation, Integration:
GeneXus
Execution and Monitoring:
Gxflow Client
Web and Smart Devices

1. Produtividade:

Algo que ajuda muito em tudo isto é ter a possibilidade a todo momento de saber em qual estado está um processo e quem tem as tarefas em um certo momento. Tudo isto resulta no aumento da produtividade uma vez que poupa tempo e gestão operacional para determinar como proceder com determinado processo e, obviamente, isso resulta em economia de dinheiro.

2. Processos de negócio otimizados

O trabalho sobre o processo de negócio que a empresa diz para fazermos uma análise introspectiva e estamos continuamente otimizando-os.

3. Maior visibilidade e controle:

Outro tema importante é que ganha-se em controle pois em qualquer momento podemos saber o status dos processos e atuar pró-ativamente por exemplo para delegar trabalho ou reassignar tarefas que não foram processadas.

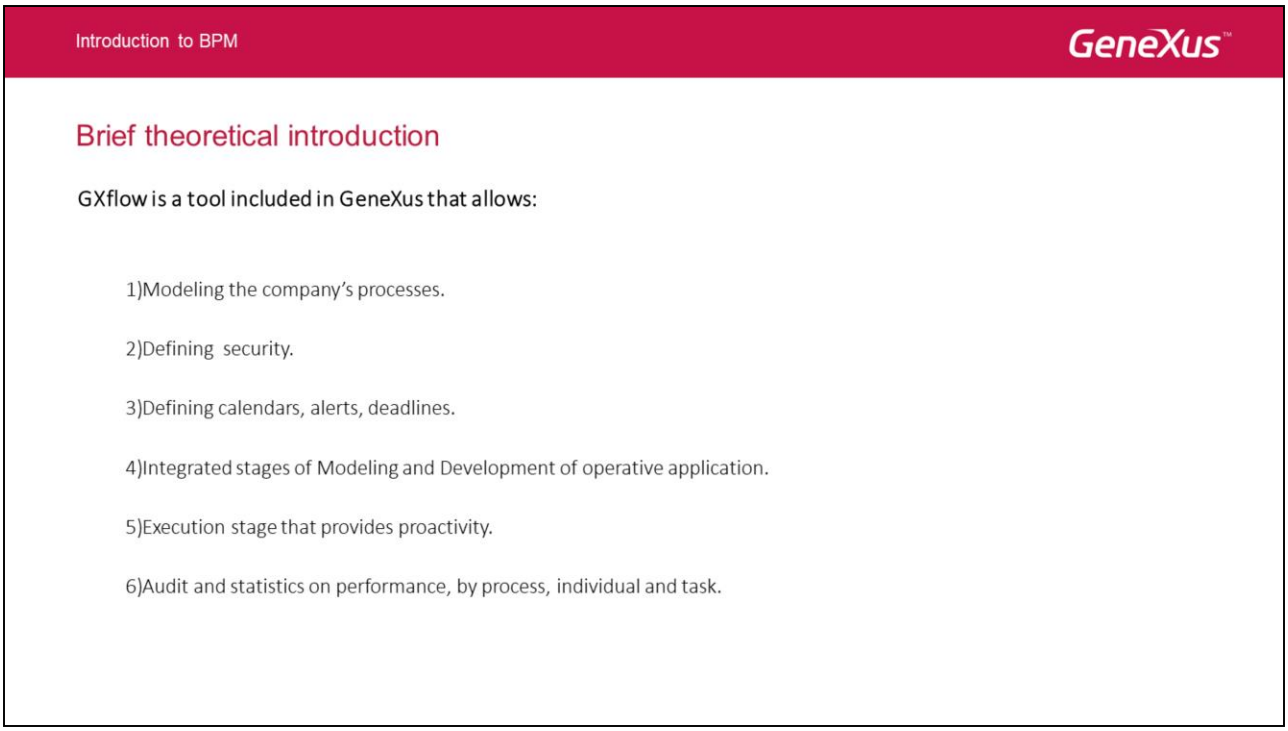
4. Melhora na comunicação:

Tendo claramente definido como devem-se fazer as coisas e que cada um tem claro o papél que debe cumprir, também conhece como desenvolve-se toda melhora enormemente a comunicação entre os atores e agiliza muito a passagem de informação em distintas etapas.

5. Flexibilidade para alterações

Tendo claramente definido como estão todas as etapas de um processo, nos permite avaliar o impacto de qualquer mudança que seja feita, ou sendo uma mudança para otimizar ou uma mudança obrigatória porque tem que agregar muitas tarefas por mudanças em regulações. Ser capaz de medir esse risco a mudanças dá à empresa maior flexibilidade e, portanto, pode ir evoluindo seus processos de mudanças, ditar os mercados com os quais interage.

Do ponto de vista do desenvolvimento o ganho vem pelo lado que vamos “declarar” mais e programar menos, vamos separar toda a lógica de controle de fluxos de nossa programação a qual será muito mais fácil de entender o que fizemos e portanto muito mais fácil de manter.



GXflow é uma ferramenta integrada a GeneXus que nos permite e oferece:

1) **Modelar os processos da empresa :** A diagramação dos processos nos dá a vantagem de ser capaz de alterar a ordem das suas tarefas, remover ou incluir novas tarefas e / ou alterar as condições da sua aplicação, sem tocar o código dos próprios objetos**.**

2 Definir segurança : Os roles que serão definidos e quais as tarefas que podem ser executadas. Isso evita a necessidade de incluir código para objetos de segurança.

3) Definir calendários, alertas, deadlines;

4) Etapas de Modelagem e Desenvolvimento de aplicação operativa integradas : No GeneXus X é acessível e fácil de relacionar os objetos GeneXus desenvolvidos que implementam a execução operacional com os diagramas que modelam os processos. Vamos ver como os objetos são arrastados para os diagramas e da praticidade de ter a modelagem dos processos integrados com os aplicativos operacionais.

5) Etapa de execução que oferece pró-atividade : Cada usuário quando executa o processo, verá a lista das tarefas pra fazer (não tem que olhar o trabalho pendente na aplicação desenvolvida).

6) Auditoria e estatísticas de desempenho por processo, pessoas e tarefas: O GXflow permite ver o que cada usuário está fazendo, quanto tempo leva cada tarefa, etc**.**

Brief theoretical introduction

Steps for working with GXflow:

MAY BE IN PARALLEL

- Create GeneXus objects to describe reality and processes
- Create business process diagrams to model processes
- Relate GeneXus objects to business process diagrams elements
- Execute process

Basic concepts to create business process diagrams

- How do we create a business process diagram?

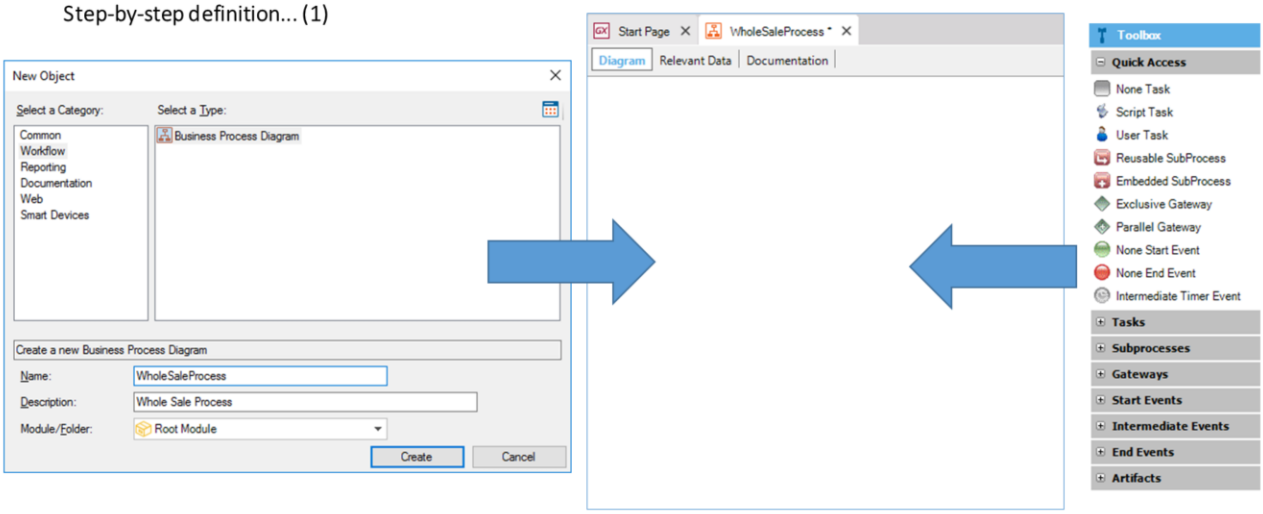
  - By creating a **Business Process Diagram** object using the GeneXus IDE
  - By using the Business Process Modeler (stand alone tool)

- How many business process diagrams will we define in our KB?

  As many as the company has.

Basic concepts to create business process diagrams

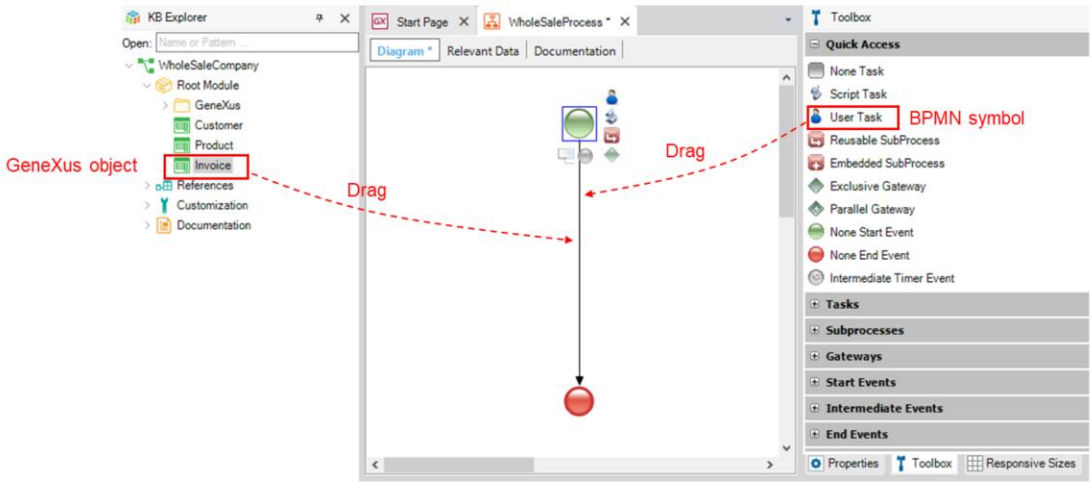
Step-by-step definition... (1)



A notação que se utiliza para os diagramas é BPMN (Business Process Modeling Notation ou Notação para o Modelagem de Processos de Negócio).

Basic concepts to create business process diagrams

Step-by-step definition...(2)



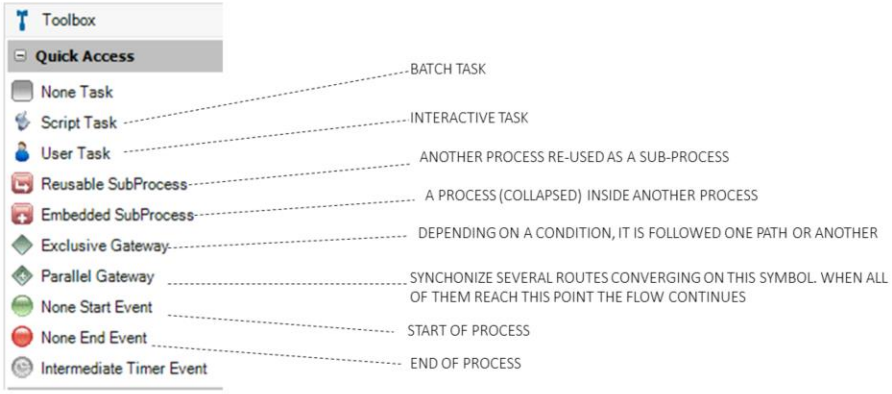
Si desde “Folder View” se arrastra un objeto transacción o web panel al diagrama, se estará agregando una “tarea / actividad interactiva” al flujo, **y la misma ya quedará con dicho objeto asociado para la etapa de ejecución**. También dicha “tarea / actividad interactiva” agregada al diagrama, quedará automáticamente nominada con el mismo nombre que la transacción o web panel que fue arrastrado.

Si en cambio se agrega una “tarea/actividad interactiva” al diagrama arrastrando el símbolo correspondiente desde la toolbox disponible para confeccionar diagramas de procesos de negocios, **luego será necesario asociar a dicha tarea un objeto transacción o web panel definido en la KB**. Para ello, simplemente después de haber agregado la tarea en el diagrama y teniéndola seleccionada, habrá que editar sus propiedades (F4 para abrir el diálogo de propiedades) y completar la propiedad **Web Application** con el objeto que corresponda, así como la propiedad **Name** con el nombre que se le quiera dar a la “tarea/actividad interactiva”.

GeneXus

Basic concepts to create business process diagrams

Description of symbols

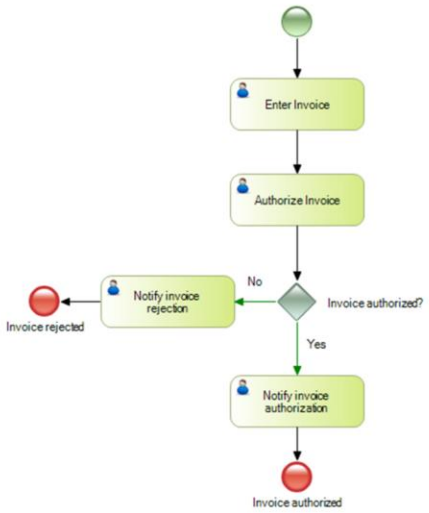


No se describirán todos los símbolos disponibles en la toolbox de workflow, sino los símbolos básicos que debemos conocer en un principio.

GeneXus

Basic concepts to create business process diagrams

Step-by-step definition ... (3)



Aquí vemos que hemos ido confeccionando el diagrama de proceso que creamos. Le hemos agregado tareas interactivas y una condición.

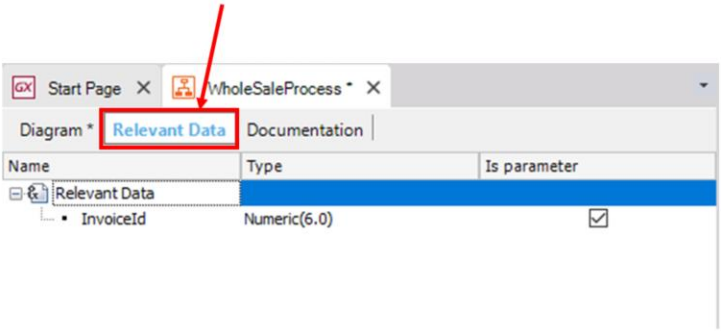
Vemos que con doble clic en las rutas verdes que salen de la condición, se abre el editor de condiciones para editar cada condición (así en caso de cumplirse una u otra se seguirá con cierto flujo de actividades u otro).

En lo que sigue veremos el concepto de “Datos Relevantes” que es fundamental para comprender qué información podemos involucrar en las condiciones

GeneXus

Basic concepts to create business process diagrams

RELEVANT DATA ≈ GLOBAL VARIABLE IN DIAGRAM



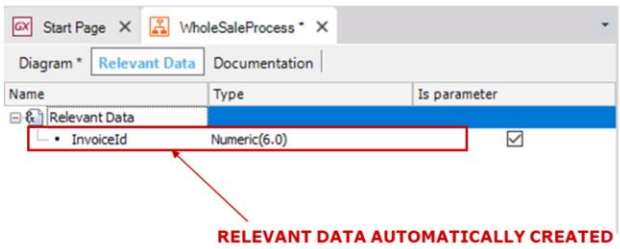
O conceito de “Dados Relevantes” é utilizado para especificar “variáveis globais” em um processo.

Este conceito permite administrar a passagem de informação entre as tarefas e que a informação seja conhecida em todo o fluxo.

Basic concepts to create business process diagrams

RELEVANT DATA (Contd.)

- When we drag a transaction from "KB Explorer":

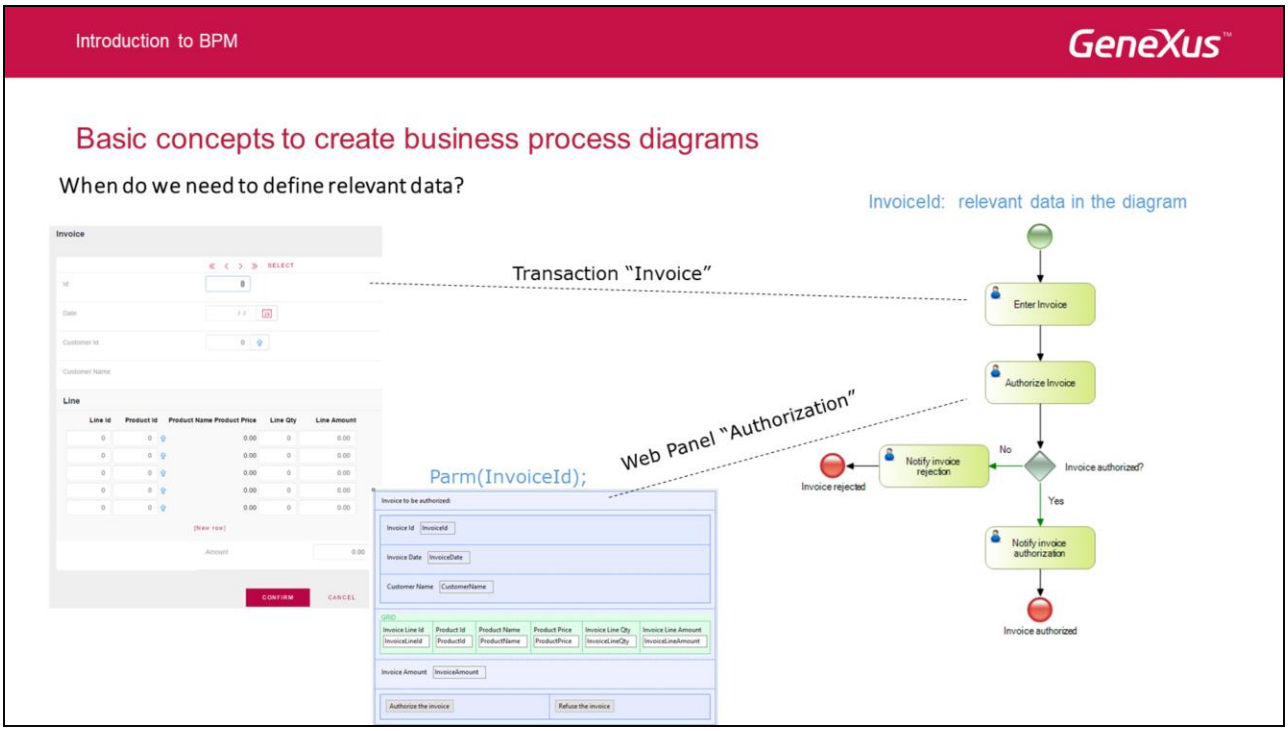


- Same name and data type that Transaction Primary Key

Para los datos relevantes que se definen automáticamente con igual nombre y tipo que las claves primarias de las transacciones (como en este ejemplo: InvoiceId), hay una correspondencia automática entre el dato relevante y el atributo PK.

Esto es, cuando modelamos el diagrama "el dato relevante es el dato global conocido en ese contexto" y en los objetos GeneXus que desarrollamos asociados a las actividades del diagrama "recibimos en parm al atributo PK" (tratándose de la misma información o bien en el contexto del diagrama, o en el contexto del desarrollo de la funcionalidad respectivamente). En otras palabras hay un mapeo automático entre el dato relevante de igual nombre y tipo al de un atributo clave primaria y dicho atributo clave primaria.

Si bien la mayoría de Datos Relevantes en un diagrama de proceso suelen corresponderse con claves primarias, también hay casos en los que surge la necesidad de definir explícitamente otros datos relevantes. A continuación veremos ejemplos.



Inicialmente hemos explicado que cuando trabajamos en GeneXus con workflow, realizamos los siguientes pasos básicamente:

- Crear objetos GeneXus
- Crear diagramas de procesos de negocios
- Asociar objetos GeneXus a diagramas de procesos de negocios
- Ejecutar proceso

Así es que venimos confeccionando un diagrama de proceso en nuestra KB; teníamos previamente definidos algunos objetos GeneXus que hemos arrastrado al diagrama, y vamos desarrollando otros objetos e incorporándolos al diagrama también. Más adelante veremos como asignar roles a las diferentes actividades del diagrama. Por ahora la idea es que contamos en la KB con una transacción "Invoice", la cual fue arrastrada al diagrama como primer actividad interactiva del proceso y más adelante definiremos que esta tarea interactiva podrá ser ejecutada por las vendedoras de la empresa.

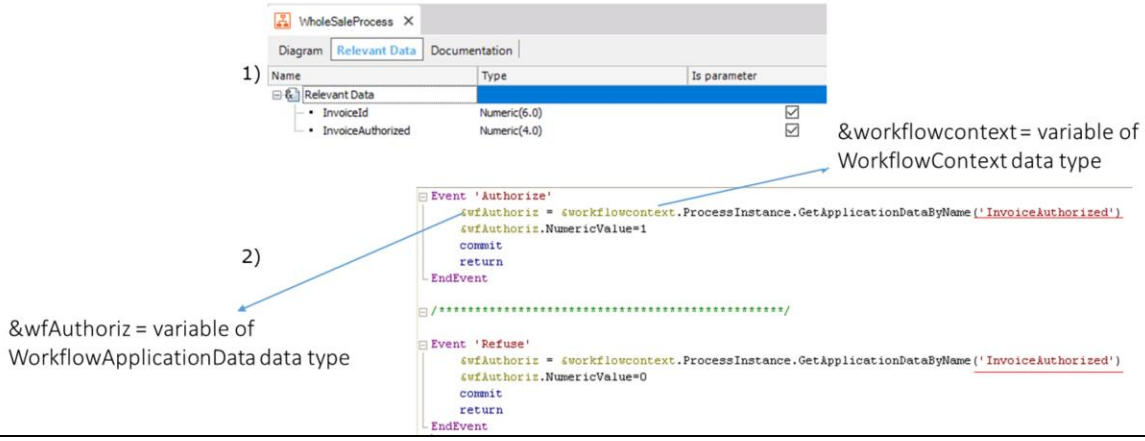
Cuando una vendedora ingrese una venta (a través de la transacción "Invoice"), entrará el cliente, la mercadería solicitada, las cantidades, la forma de pago solicitada por el cliente, grabará la venta con un número interno (no el nro de factura formal) y ahí terminará la primer actividad. Luego esa venta deberá ser evaluada por un supervisor (quien evaluará de qué cliente se trata, el monto, la forma de pago), y el supervisor deberá aceptar o denegar la venta. Esta segunda actividad interactiva del proceso fue agregada al diagrama arrastrando el web panel "Authorization".

Entonces, a nivel del diagrama de proceso contamos con el dato relevante InvoiceId (de igual nombre y tipo de dato que el atributo *InvoiceId*), lo cual significa que dicho dato se conoce a lo largo de todo el flujo de actividades. Y en lo que se refiere a los objetos GeneXus relacionados al diagrama, el web panel "Authorization" implementa la 2da actividad del proceso y recibe por parámetro al atributo *InvoiceId*. En el form del web panel "Authorization" se visualizarán los datos de la factura, el límite de crédito y saldo de límite de crédito del cliente y estará la posibilidad de ver la historia de facturas anteriores del cliente. Este web panel ofrecerá al supervisor 2 botones: "Authorize" y "Refuse".

Basic concepts to create business process diagrams

Step-by-step definition ... (4)

We create a new relevant data: **InvoiceAuthorized**



De modo que hemos definido un dato relevante en el diagrama de proceso (de nombre InvoiceAuthorized) y luego en el web panel “Authorization” se lee el dato relevante en variable &wfAuthoriz de tipo WorkflowApplicationData. El tipo de datos WorkflowApplicationData viene a significar “dato relevante”.

También se utiliza la variable &WorkflowContext, para poder acceder a los datos de la instancia del proceso mediante la siguiente consulta:

&WorkflowContext.ProcessInstance.GetApplicationDataByName('InvoiceAuthorized')

El método WorkflowProcessInstance viene a significar “instancia de proceso”. O sea que si analizamos la primer línea codificada en cualquiera de los 2 eventos, estamos recuperando en la variable &wfAuthoriz (de tipo “dato relevante”) el valor del dato relevante de nombre ‘InvoiceAuthorized’ (entre comillas va el nombre del dato relevante tal como se ha definido en el diagrama de proceso) perteneciente a la instancia del proceso que se está ejecutando.

Luego en la segunda línea codificada en ambos eventos, se está asignando a la variable &wfAuthoriz (de tipo “dato relevante”) el valor 0 o 1 según corresponda (utilizando la propiedad Numeric del tipo de dato WorkflowApplicationData, por estar asignando un valor número).

Por último en cada evento se hace un commit y return, porque ya se ha recuperado el valor del dato relevante, se le ha cargado el valor deseado y se desea culminar con la ejecución de esta actividad.

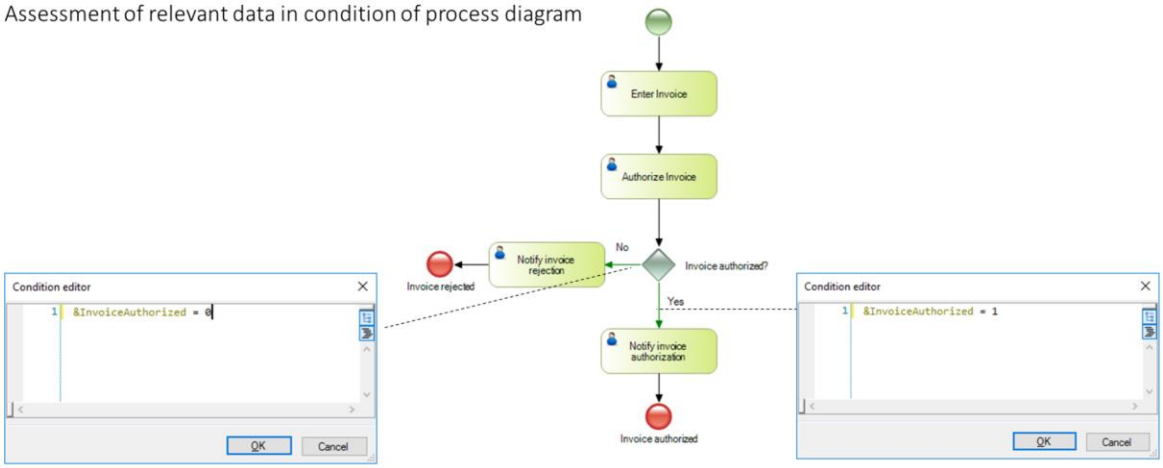
De esta manera entonces leemos y cargamos datos relevantes en los objetos GeneXus asociados a las actividades de un diagrama de proceso, empleando tipos de datos workflow, propiedades y métodos.

Sin embargo para datos relevantes con igual nombre y tipo que las claves primarias, esto no es necesario (sino que se reciben por parm los atributos primarios directamente y ya hay mapeo automático con el dato relevante correspondiente).

Basic concepts to create business process diagrams

Step-by-step definition ... (5)

Assessment of relevant data in condition of process diagram



Observe que en cada ruta verde que sale del rombo de condición, se evalúa el valor del dato relevante &InvoiceAuthorized.

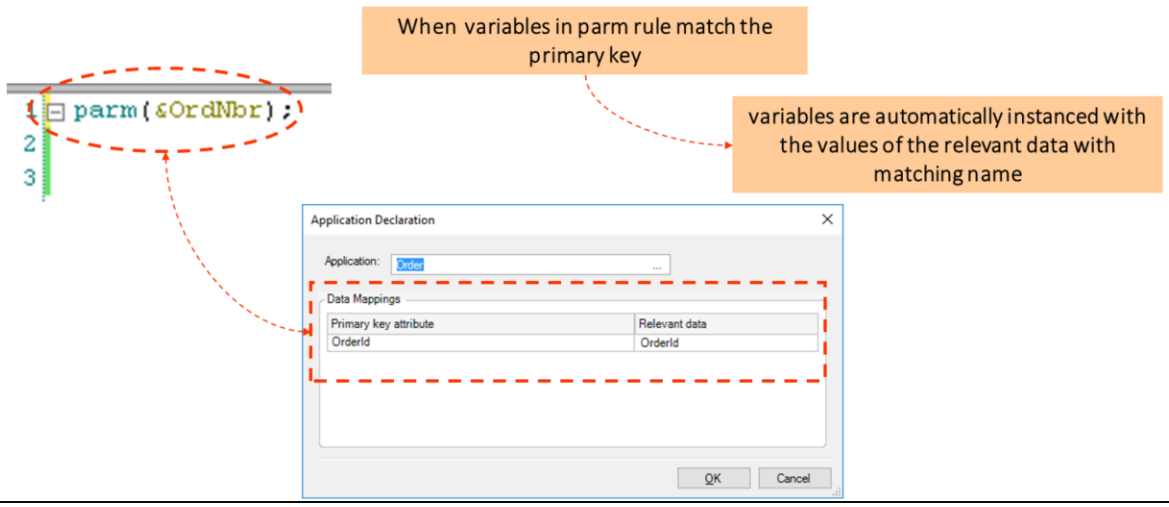
Si bien cuando definimos el nombre del dato relevante no incluimos el & (amperson), el diálogo de definición de datos relevantes muestra claramente un ícono &, dando la pauta de que se trata de variables:

De modo que para referirnos a un dato relevante en el flujo, lo referenciamos con & como se puede observar en las 2 imágenes capturadas correspondientes al editor de condiciones. Sin embargo para datos relevantes con igual nombre y tipo que las claves primarias, es posible referenciar tanto el dato relevante (con &) como el atributo clave primaria directamente.

GeneXus

Basic concepts to create business process diagrams

Relevant Data mapping in Transactions with parameters



Como é feito o mapeamento dos dados relevantes em transação quando recebe parâmetros?

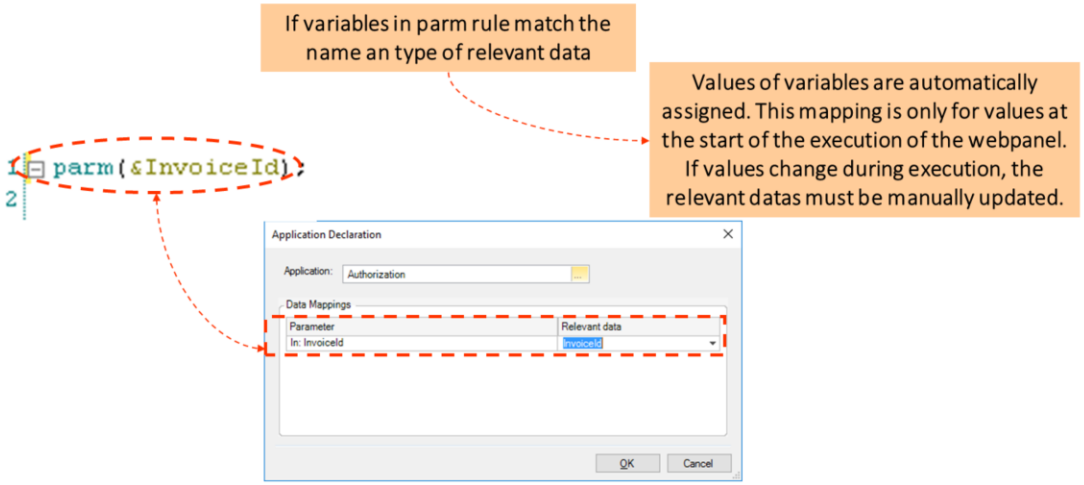
Quando as transações recebem por parâmetro a chave primária da transação e recebem-na diretamente no atributo o em variáveis baseadas na chave primária, é feito um mapeamento automático com os dados relevantes.

Nota: As variáveis precisam ter o mesmo nome dos atributos.

GeneXus

Basic concepts to create business process diagrams

Relevant Data mapping in Web panels with parameters



Como é feito o mapeamento dos dados relevantes em webpanels quando recebe parâmetros?

Se as variáveis ou atributos na regra parm coincidem com o mapeamento do dado relevante,

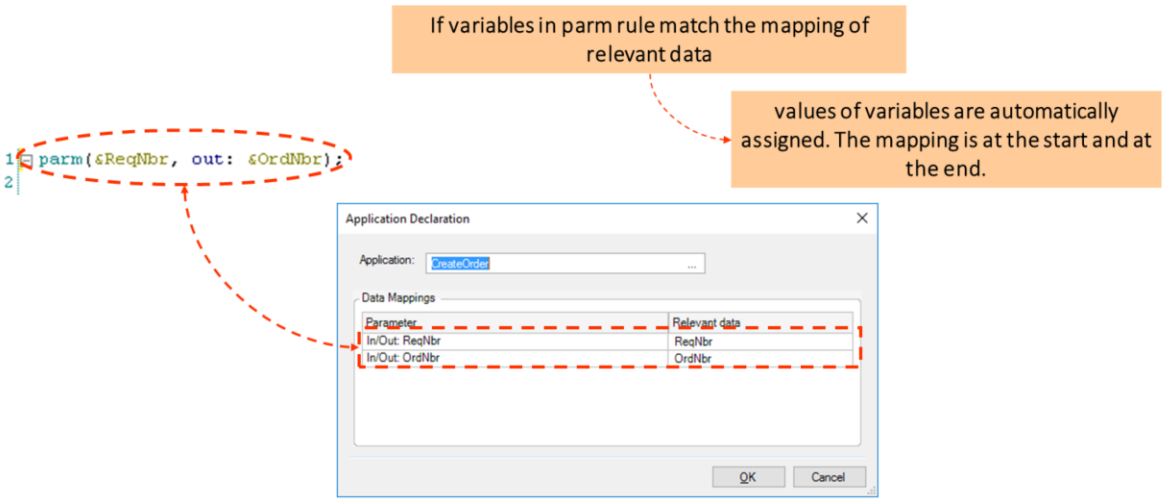
Os valores das variáveis são atribuídos automaticamente. Esse mapeamento é apenas para os valores na entrada do webpanel, na saída não se atualiza o valor;se mudou, é preciso atualizá-lo manualmente.

Nota: As variáveis precisam ter o mesmo nome dos atributos.

GeneXus

Basic concepts to create business process diagrams

Relevant Data mapping in Procedure with parameters



Como é feito o mapeamento dos dados relevantes em procedimentos quando eles recebem parâmetros?

Se as variáveis ou atributos na regra parm coincidem com o mapeamento do dado relevante,

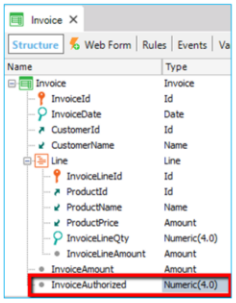
Os valores das variáveis são atribuídos automaticamente. O mapeamento é na entrada e na saída.

Nota: As variáveis precisam ter o mesmo nome dos atributos.

GeneXus

Basic concepts to create business process diagrams

Another possible solution to solve steps (4) and (5) – Step 1



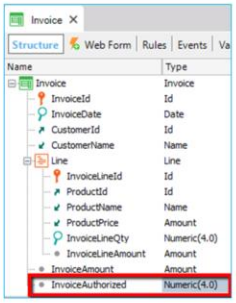
1)Define *InvoiceAuthorized* attribute in “Invoice” transaction

Creamos al atributo InvoiceAuthorized en la transacción Invoice y lo utilizamos para almacenar, en cada factura si la misma fue autorizada o no.

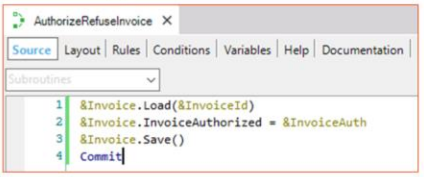
GeneXus

Basic concepts to create business process diagrams

Another possible solution to solve steps (4) and (5) – Step 2



2) Web panel "Authorization" invokes procedure AuthorizeRefuseInvoice who saves the value



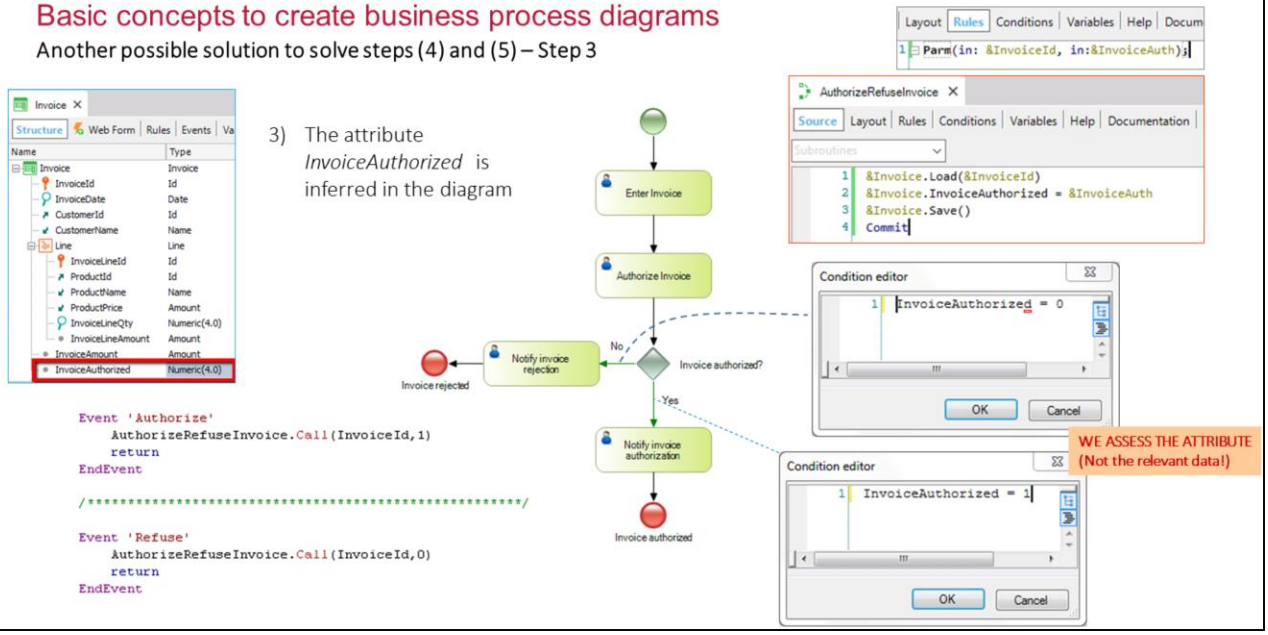

```
Event 'Authorize'
    AuthorizeRefuseInvoice.Call(InvoiceId,1)
    return
EndEvent

/*****************************************************/

Event 'Refuse'
    AuthorizeRefuseInvoice.Call(InvoiceId,0)
    return
EndEvent
```

Desde el webpanel Authorization Invocamos al procedimiento AuthorizeRefuseInvoice quien guarda el valor 1 o 0, en el atributo InvoiceAuthorized.

GeneXus

Basic concepts to create business process diagrams

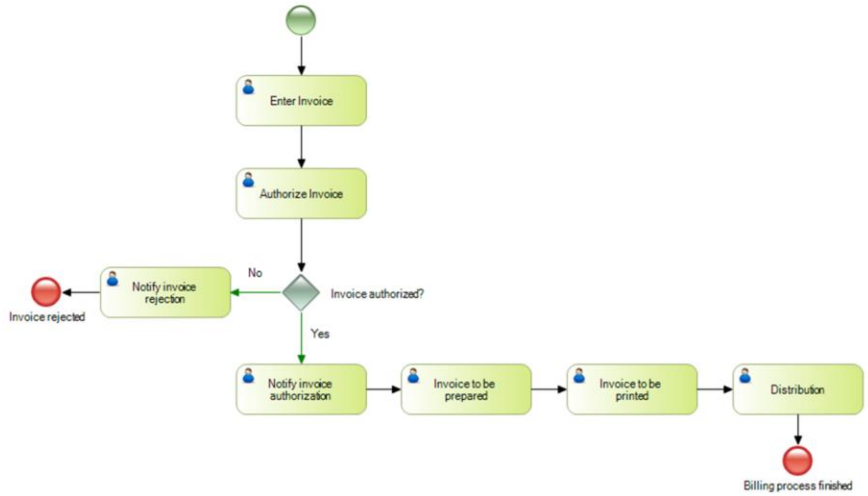
Another possible solution to solve steps (4) and (5) – Step 3



Con esta solución, no necesitamos definir el dato relevante InvoiceAuthorized, porque en un atributo tenemos la información de si la factura fue autorizada/rechazada…y en el diagrama de procesos podemos inferir atributos a través de llaves primarias, y evaluar el valor de los atributos directamente.

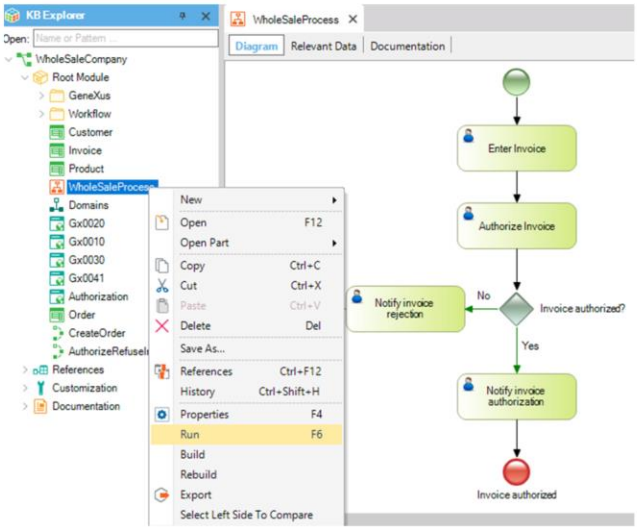
Basic concepts to create business process diagrams

Complete process diagram, corresponding to the process we are creating:



Basic concepts to create business process diagrams

To execute a process we must select the Run option



Para executar um processo com o prototipador de GXflow, deve-se selecionar a opção Run sobre ele. Ao fazer isso, o processo de Build de GeneXus se encarregará de realizar todas as ações necessárias para poder executar o processo.

Algumas dessas ações são:

- **Criação ou reorganização das tabelas de workflow**: essas tabelas são requeridas pelo motor de workflow. Elas serão criadas se se detectar que não existem no banco de dados do aplicativo, ou reorganizadas caso as tabelas existentes correspondam a uma versão anterior de Gxflow.

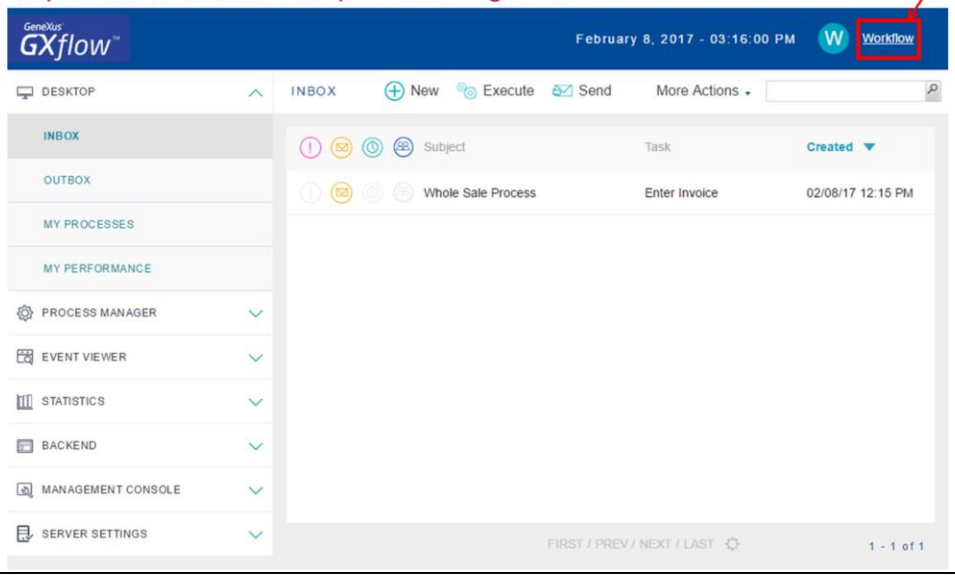
- **Deploy de processos:** disponibiliza os processos para que possam ser executados pelo motor de workflow. Essa operação é executada somente se diagramas foram criados ou modificados.

- **Deploy do runtime de workflow:** os binários e recursos que compõem o motor e cliente de Gxflow são integrados com os binários de nossa aplicação. Essa operação somente se executará pela primeira vez que se fizer uma build tendo diagramas de processo ou quando se detectar que na instalação de GeneXus há uma versão mais nova do runtime de GXflow.

- **Adaptação de objetos associados a tarefas de processos**: para cada um desses objetos, gera-se o código necessário para que possam ser invocados pelo motor de workflow.

Basic concepts to create business process diagrams

Execution:



Al ejecutar el diagrama se abre una ventana en el navegador, mostrando la ventana principal del Cliente de Workflow. Esta pantalla similar a una bandeja de entrada de mails. Allí identificaremos a la tarea pendiente de ser ejecutada y el proceso al cual pertenece.

La tarea pendiente es para el usuario **Workflow**, que es el usuario administrador utilizado por defecto en la prototipación. Para cambiar el usuario, debemos cambiar el modo de ejecución.

Hay dos métodos de ejecución:

- Prototyper
- StandardClient

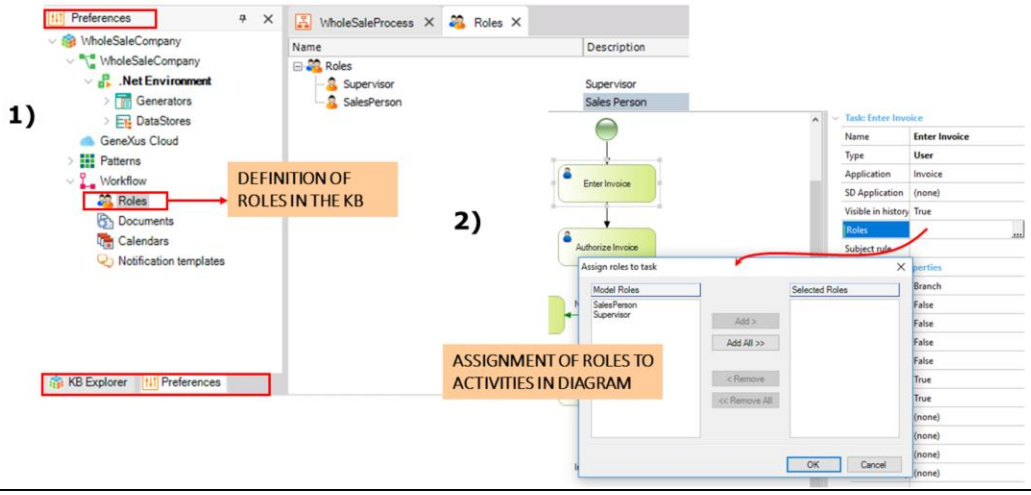
En la sección Preferences→Environment→Properties podemos configurar el modo que deseemos.

El método “Prototyper” está, por supuesto, orientado a la etapa de prototipación, y el método “StandardClient” está orientado a la etapa de producción. Las pruebas de seguridad no son importantes durante la etapa de prototipación, entonces cuando ejecutamos la aplicación con el método “Prototyper”, todos las instancias de prueba anteriores son borradas y existe la posibilidad de ejecutar todas las actividades sin controlar los roles.

Cuando ejecutamos la aplicación en el método “StandardClient”, nada es abortado al comienzo, y se deben crear usuarios (para el Login), cada uno con su lista de roles correspondiente, porque en este caso, la seguridad sí será controlada.

Basic concepts to create business process diagrams

Role definitions:



La definición de roles se hace al nivel de la KB, en la sección de Preferences.

Seleccionando cada actividad en el diagrama y presionando F4, en la propiedad **Roles** podemos asignar la lista de roles que van a ejecutar esa actividad.

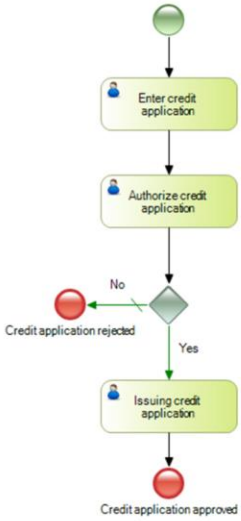
Una vez que los roles son asignados, para chequear la seguridad del proceso que modelamos, tenemos que cambiar el modo de ejecución. Para eso, en el grupo Workflow de las propiedades del Environment, seleccionamos la propiedad Execution Mode y asignamos el valor StandardClient.

Module 2

Process Modeling

GeneXus

What is business process modeling?

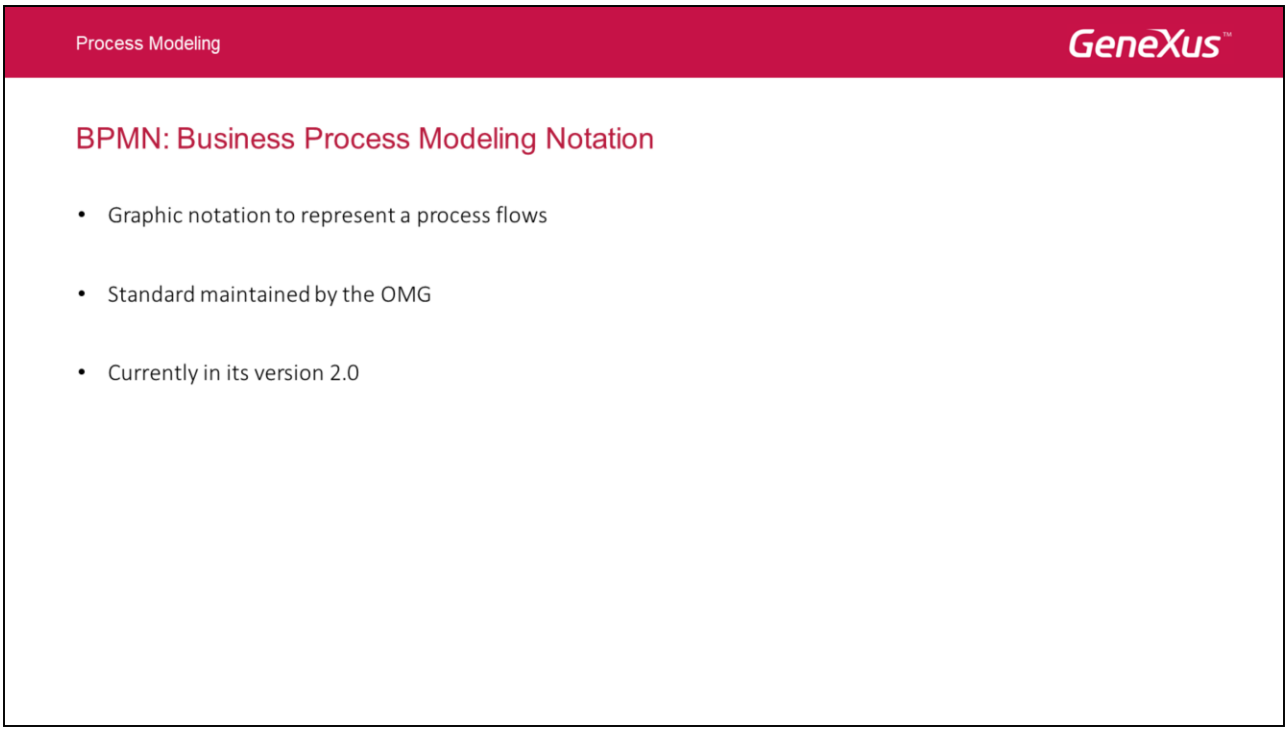


It is a representation of a **business process**

The business process may be:

- Analyzed
- Executed
- Improved

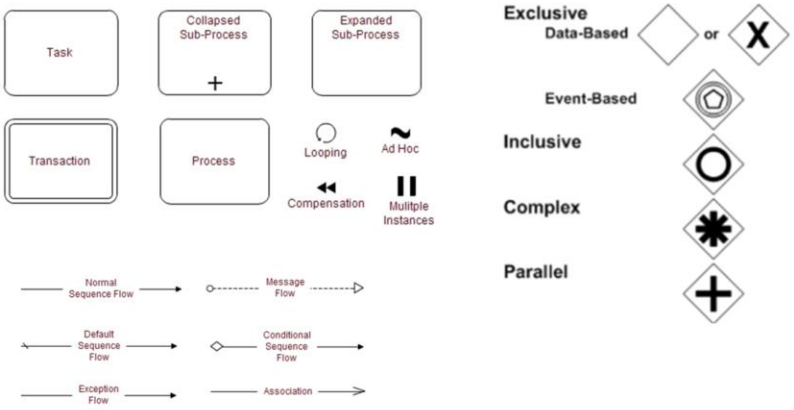
É a atividade de representar os processos de negócio de uma empresa com o objetivo de que estes possam ser analisados, executados e melhorados.



Para realizar esta atividade, vamos ver o padrão **BPMN (Business Process Modeling Notation)**, este padrão define **símbolos e regras** que nos permitem representar graficamente um processo de negócio.

GeneXus

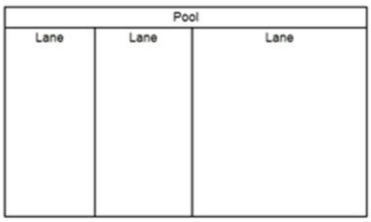
Symbols

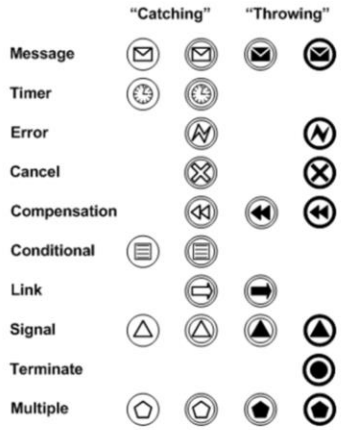


Os símbolos definidos pelo BPMN são monocromáticos, ferramentas como o GeneXus com base na norma, também adiciona cores, de modo que seja mais claro para o usuário.

GeneXus

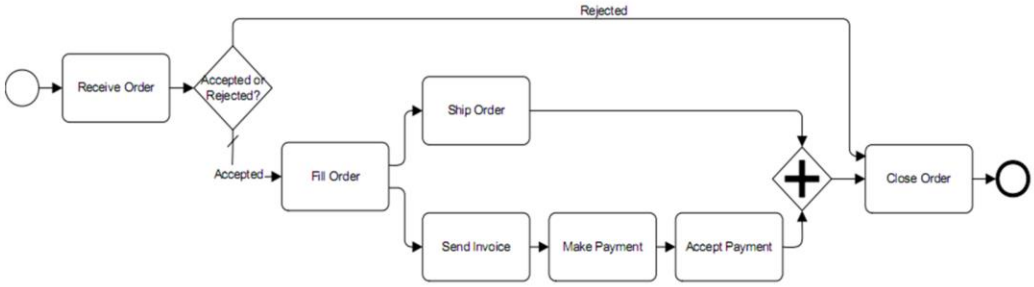
Symbols





GeneXus

Symbols



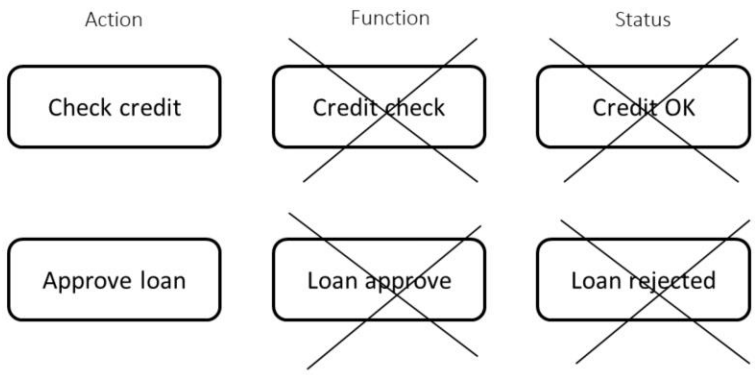
Activities

Uma atividade representa certo trabalho a ser feito dentro do processo de negócio.
Uma atividade leva certo tempo para ser executada e utiliza um ou mais recursos da organização.
Requer certo tipo de entrada e, normalmente, produz algum tipo de saída.
É representada graficamente com um retângulo com os vértices arredondados.

Uma atividade pode ser do tipo atômica ou composta.
O tipo atômico é conhecido como Tarefa.
O tipo composto de uma atividade é chamado Subprocesso.
Graficamente, diferenciam-se pelo símbolo de mais (+) que o Subprocesso utiliza.

Activities

- They represent actions implemented in the process.
- They DO NOT represent statuses or functions.
- Their names must be of the type Verb – Noun.

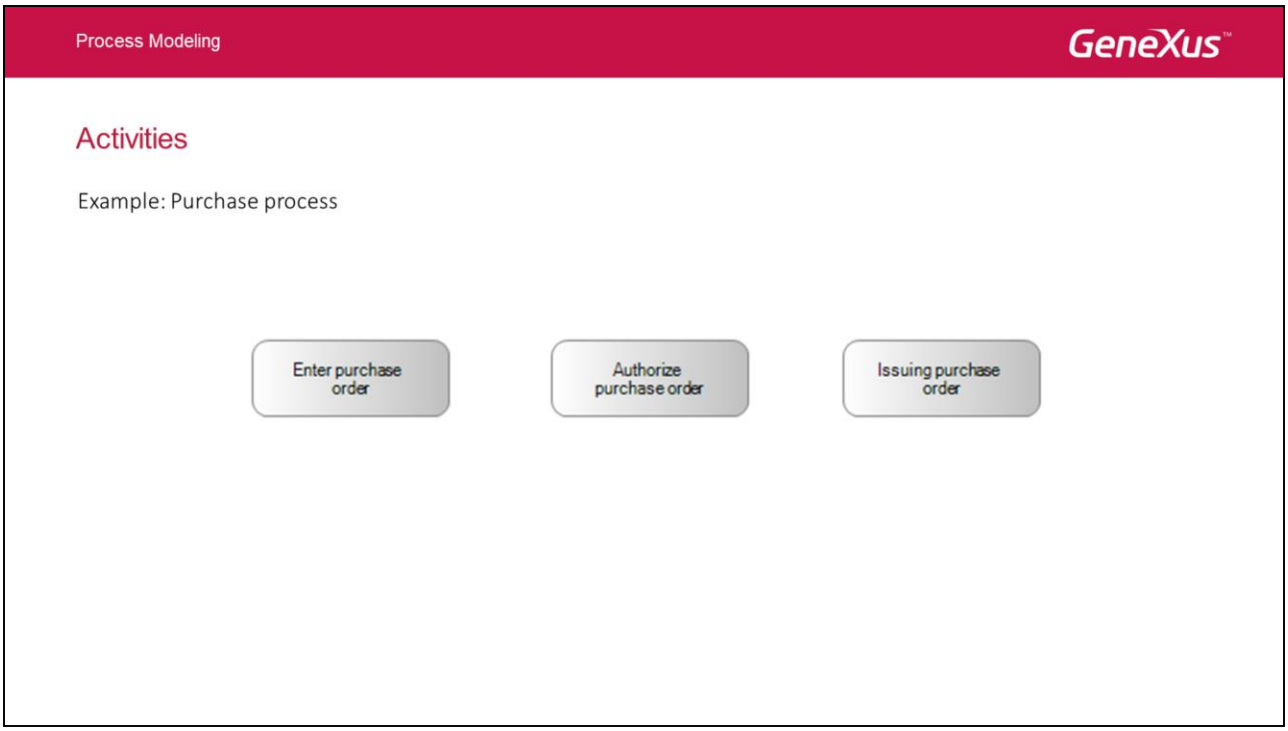


As atividades representam **ações** realizadas no processo.

NÃO representam estados ou funções

Seus nomes devem ser da forma Verbo – Substantivo

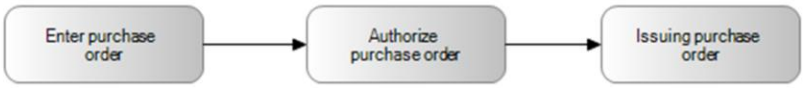
Através destes exemplos podemos ver **como descrever** as atividades dos processos.



Então, de volta ao nosso exemplo, temos de representar três atividades:

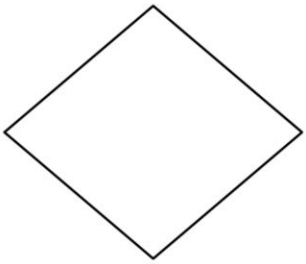
Introduzir Pedido de Compras
Autorizar Pedido de Compras
Gerar Ordem de Compras

Sequence Connectors



Para indicar a ordem de precedência destas atividades utilizamos o símbolo Conector de Sequência.

Gateways



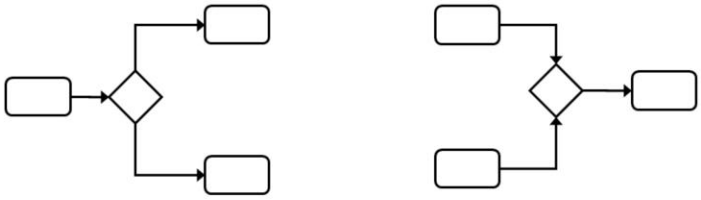
Até este ponto o processo que definimos é linear, o que significa que processam-se as três atividades uma a continuação da outra sem exceções. Mas em nosso exemplo após a autorização da tarefa há dois caminhos possíveis, um é gerar a ordem de compras e o outro é que o processo termine com base na decisão tomada durante a autorização do pedido.

Os Gateways são elementos de modelagem que controlam como um processo Diverge ou Converge. Quando o fluxo não precisa ser controlado, então os gateways não são necessários.

Todos os Gateways são representados por um losango, utilizando-se distintos ícones em seu interior para distinguir o tipo de Gateway.

Gateways

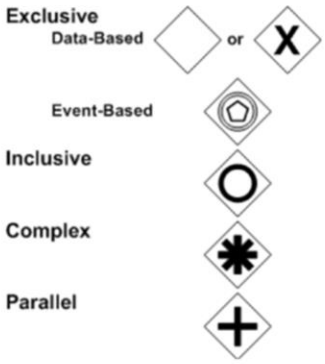
They are used to control branching and merging of flow paths in a process



Para isto vamos utilizar o símbolo **Gateway**, que nos permite ramificar ou unificar o processo.

Gateways

BPMN defines several types:

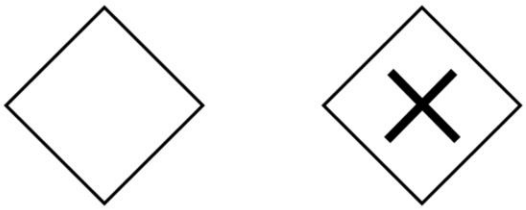


GeneXus

Exclusive branching

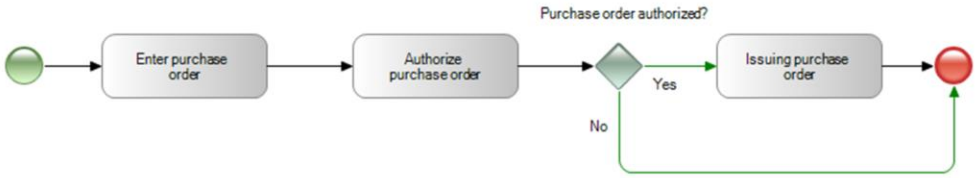
Exclusive gateway

It is used when the path of the process flow depends on the assessment of a condition of the true / false type.



Em nosso processo vamos utilizar um gateway de tipo Exclusive para determinar se gera a ordem de compras ou não.

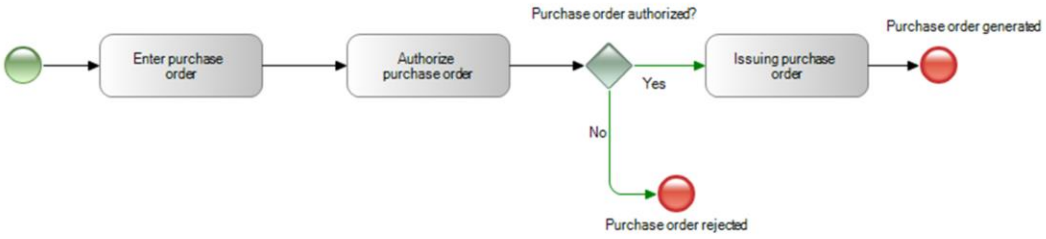
Gateways



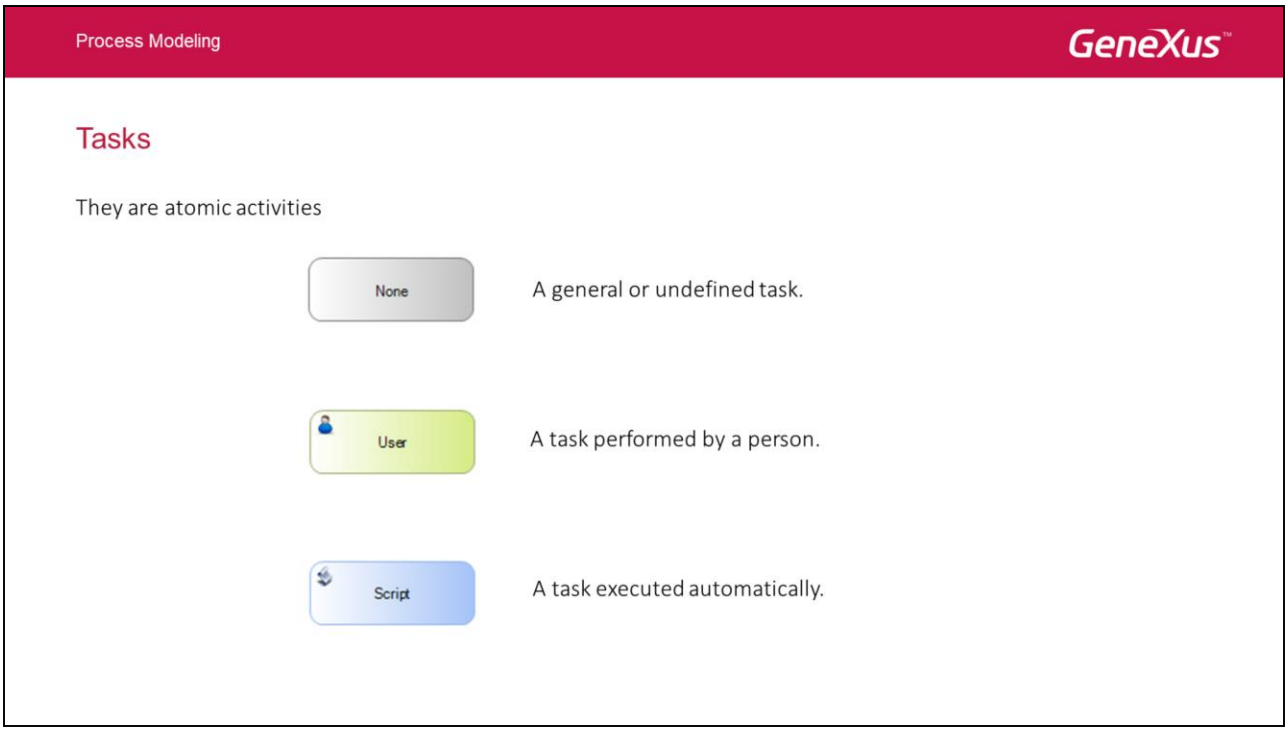
Colocamos em seguida a tarefa de autorização e antes da geração da ordem de compra.

Como vemos estamos utilizando o mesmo símbolo de fim para ambos caminhos, BPMN estabelece que o correto é utilizar um para cada caminho.

Gateways



Desta forma podemos conhecer como terminou o processo, neste caso se o pedido foi autorizado ou recusado.



Os diferentes tipos de tarefas que existem para a modelagem dos processos são:

None

Uma tarefa geral ou indefinida. Normalmente, é utilizada em uma etapa inicial da modelagem (quando não se tem informação suficiente para determinar o tipo).

User

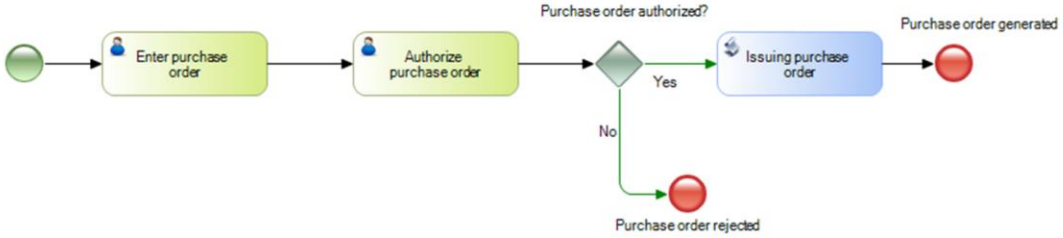
É uma típica tarefa de workflow na qual uma pessoa tem que executar certo trabalho utilizando um determinado programa. As pessoas podem ver suas tarefas através de uma interface do tipo de Caixa de Entrada.

Script

Uma tarefa script é aquela que envolve a execução de um programa e não requer a participação de pessoas. A tarefa é executada automaticamente pelo motor de workflow quando está pronta para executar. Quando termina a execução do programa, o motor completa a tarefa.

Example

Purchase process



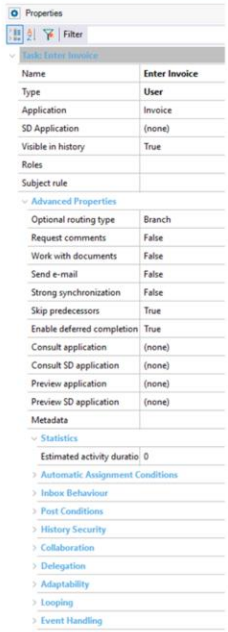
Exemplo: Processo de compras

1. Um funcionário ingressa um pedido de compras
2. O pedido é autorizado pelo **chefe do funcionário**
3. Se o pedido é autorizado, **gera-se automaticamente** a ordem de compras ou do contrário o processo termina

Em nosso exemplo tanto a tarefa de ingresso do pedido de compras, como a tarefa de autorização devem ser realizadas por usuários, portanto atribuímos o tipo **User.** Para o caso da geração da ordem como é automática não requer a interação de um usuário, portanto atribuímos o tipo **Script**.

GeneXus

Task Properties



As tarefas estão associadas a diferentes propriedades de configuração. A seguir, há uma lista com algumas delas.

Task Properties

Application / SD Application

- Object that will be invoked upon executing the task.

Consult Application

- It shows the resulting information from the execution.

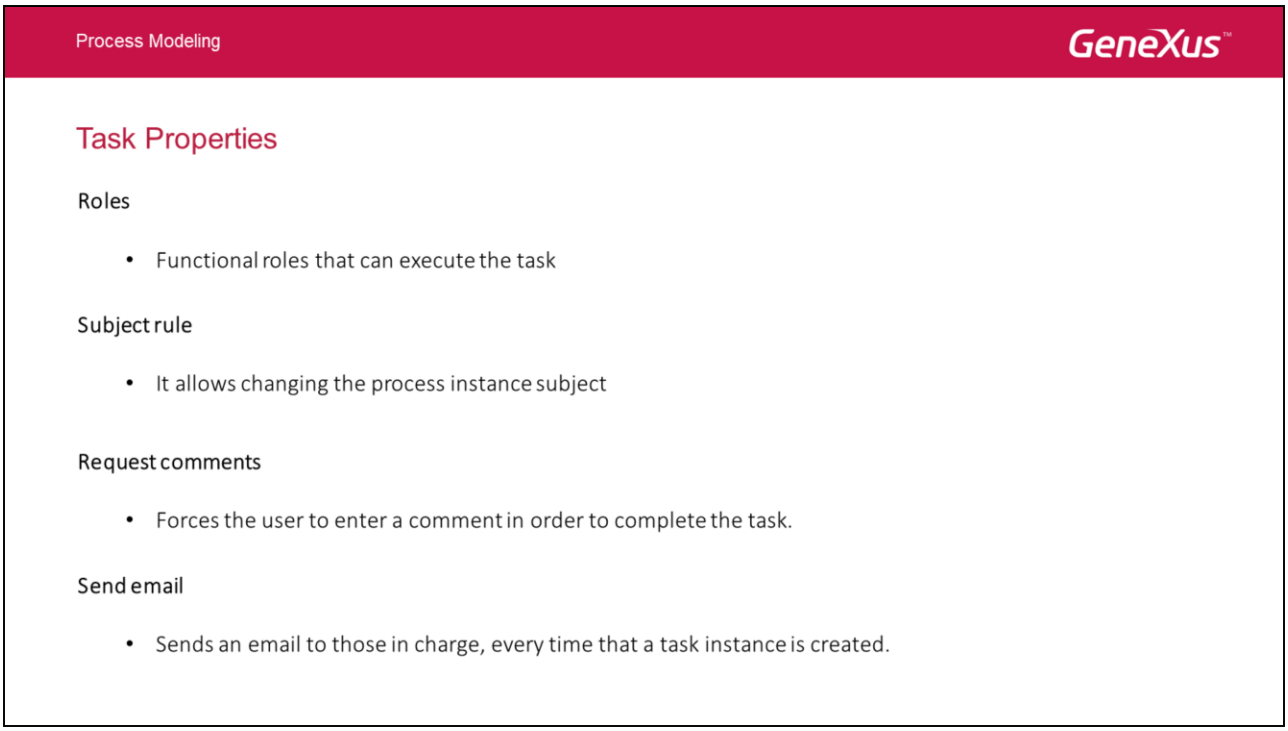
Preview Application

- It shows the basic data for the user to decide whether to take the task.

- **Application:** Permite indicar o objeto GeneXus que se invocará ao executar a tarefa.

- **Consult Application:** Permite indicar o objeto GeneXus que se invocará do histórico do processo para mostrar a informação resultante da execução.

- **Preview Application:** Permite indicar o objeto GeneXus que se invocará da opção Preview de uma tarefa para decidir se se deve ou não pegar a tarefa.



- **Roles:** Indicam-se os papéis que terão permissões para executar essa tarefa. A definição dos papéis realiza-se em Preferences/Workflow/Roles.

- **Subject Rule:** Permite modificar o assunto da instância de um processo. Podem ser utilizados caracteres, atributos, dados relevantes. Por exemplo, pode-se colocar "Customer registration number“ + CustomerId + &relevantData.

- **Request comments:** Indica se o usuário terá que inserir comentários antes de finalizar a tarefa.

- **Send email:** Indica se se enviará um e-mail aos responsáveis de uma tarefa a cada vez que se criar uma instância dela, tendo a possibilidade de usar o modelo automático de envio de e-mail ou criar um personalizado.

Uma atividade pode ser atômica ou composta.

O tipo atómico é conhecido como **Tarefa**.

O tipo composto de uma atividade é chamado **Subprocesso**.

Graficamente diferenciam-se pelo símbolo de mais (+) que utiliza o Subprocesso.

Sub-processes

- The are composite activities.
- There are two types:
  - Reusable: they enable us to reuse a process defined in an another diagram.
  - Embedded: they enable us to define the sub-process embedded within the diagram containing it.

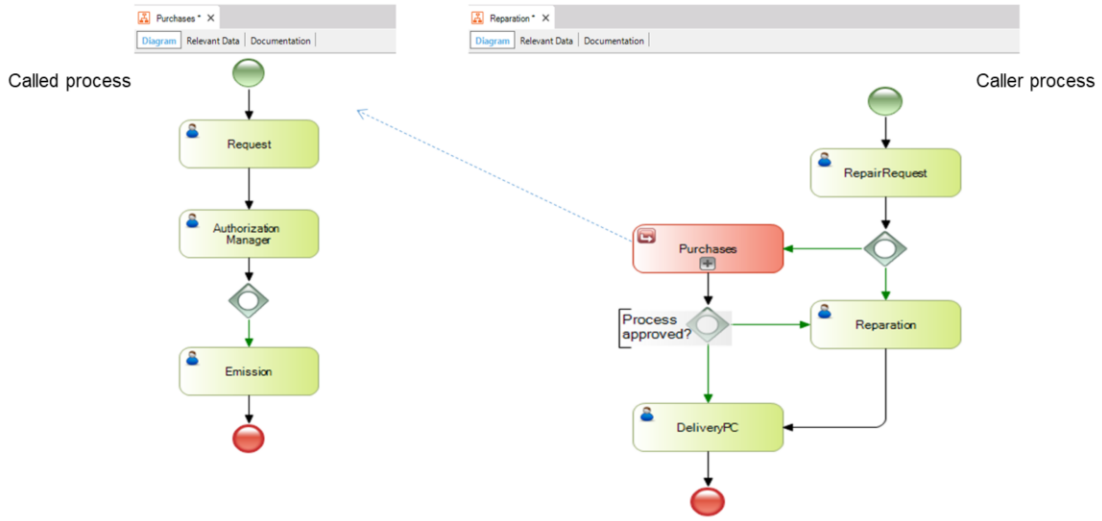
A finalidade dos subprocessos é poder encapsular um processo como uma tarefa de outro processo. Isto implica duas vantagens: a melhora da visibilidade das definições de processos e a reutilização dos mesmos.

Com base nisso, há dois tipos de processos:

Reutilizável: permitem reusar um processo definido em outro diagrama

Embebido: permitem definir o subprocesso incorporado dentro do diagrama que o contém

Reusable Subprocess

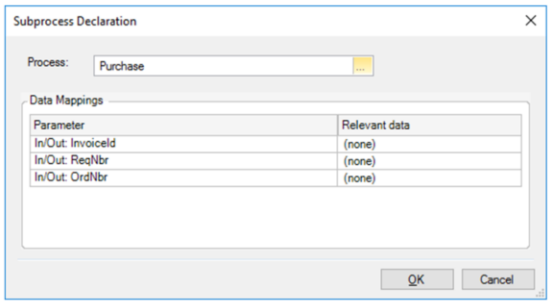


Um subprocesso reusável (processo chamado) é um processo que é invocado de outro processo (processo chamador). Os dados relevantes do processo chamado são independentes dos dados do processo chamador, mas existe entre eles um mecanismo de passagem de parâmetros.

Mapping of relevant data between subprocesses

It allows communicating the data of a process with data of the other one in two ways:

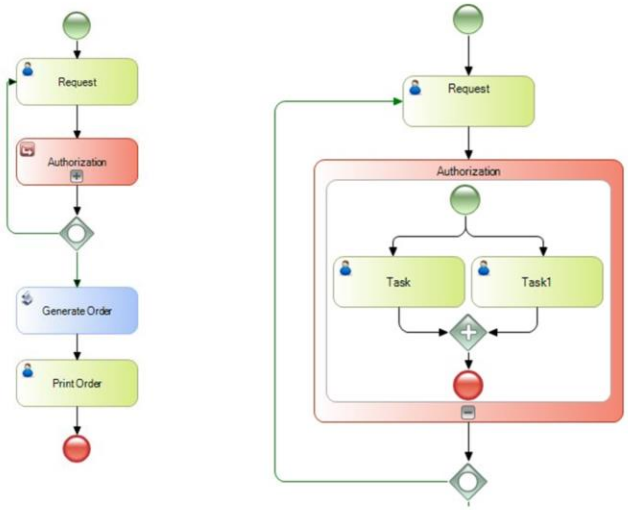
- From the route that arrives at the subprocess
- From the dialog box to select the subprocess



Quando se chama um subprocesso muitas vezes, é preciso trocar informação ou receber informação desse processo. Para a comunicação dos dados, são utilizados os dados relevantes.

Para acessar a configuração de quais dados relevantes serão trocados, pode-se partir da rota que chega ao subprocesso ou da caixa de diálogo de seleção do processo.

Embedded Subprocesses



Subprocessos embebidos

Quando usá-los? Caso se tenha um diagrama complexo, por exemplo, pode-se detectar um grupo de atividades que pode ser encapsulado em um subprocesso embebido, tendo como resultado um diagrama mais compacto com uma visão global e entendível do processo ou, se quisermos ver um detalhe em particular, é possível fazer uma expansão inline (+) e conhecer as atividades embebidas, o que nos dará vantagem na documentação do processo.

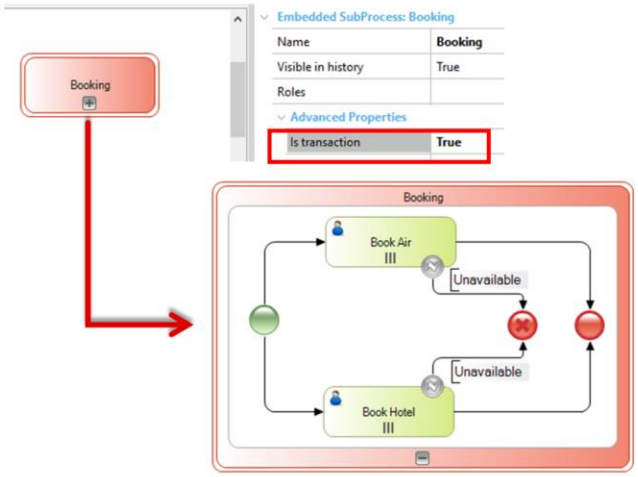
Outra opção relaciona-se a quando se quer fazer algo a um conjunto de tarefas: colocar um timer, por exemplo. Nesse caso, em vez de criar outro processo para agrupar essas tarefas, se faz dentro do mesmo.

Suas principais características são:

- Permitem-nos definir um subprocesso dentro de um processo pai, e não é reusável por outros processos.
- Os dados relevantes do processo pai estarão disponíveis para o subprocesso sem a necessidade de realizar mapeamentos.
- Não é possível iniciar eventos do tipo triggers (um timer, por exemplo) já que só o processo pai decide quando eles devem ser iniciados.

Tip: Para editar um subprocesso embebido, pode-se fazer dentro do processo pai ou mediante o clique duplo no subprocesso, isso abre o subprocesso em uma nova aba, e é útil para editar processos complexos ou que se aninham em vários níveis de subprocessos.

Transactional Subprocess

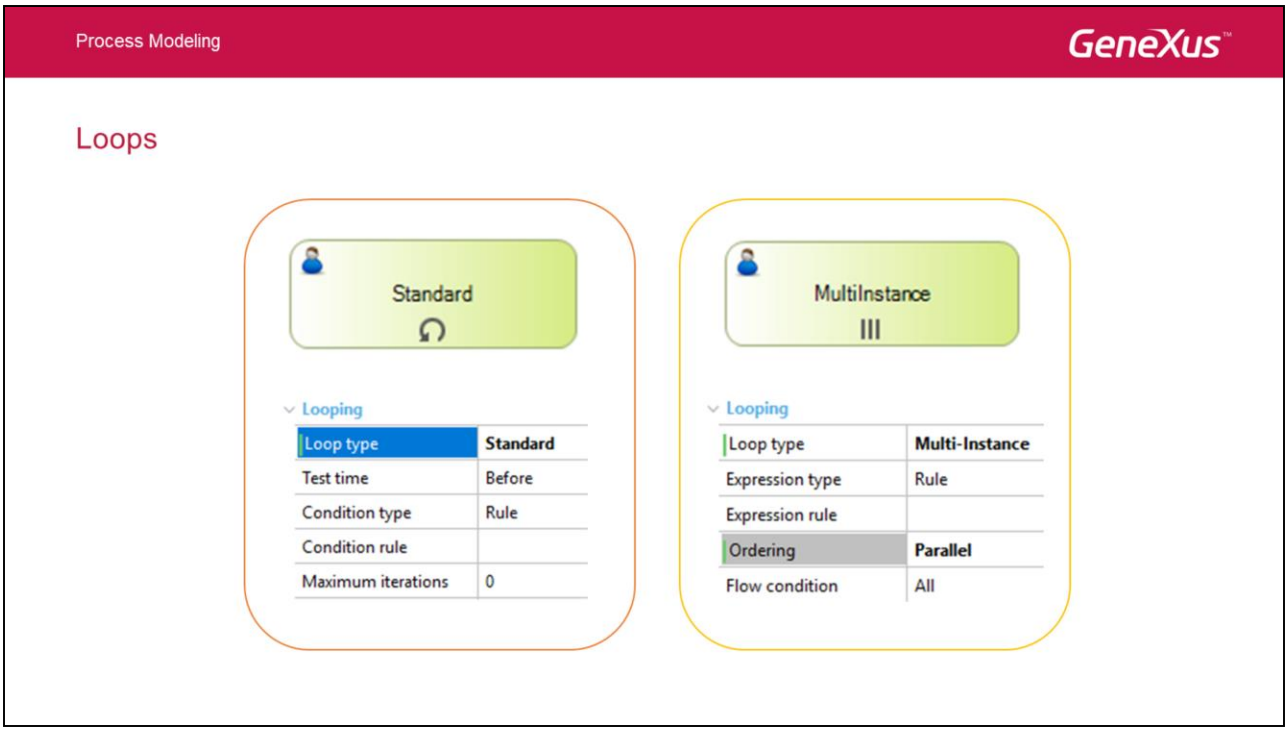


Subprocesso transacional

Um subprocesso embebido pode ser considerado uma unidade lógica (transação), ou seja, todas as tarefas que estão dentro do subprocesso devem ser concluídas, caso contrário, o processo se desfaz, desfazendo todas as atividades que o compõem (rollback).

Para definir um subprocesso transacional, é preciso habilitar a propriedade 'Is transition' do subprocesso, que se identificará por ter uma borda dupla.

Como se vê no exemplo, temos um subprocesso que faz reservas de voos e hotéis. Se ocorrer algum erro de falta de disponibilidade em qualquer uma das reservas, o fluxo avançará a um evento de finalização do tipo Cancel. Isso acionará o rollback do processo e qualquer atividade de reserva que tenha sido concluída ficará como "undone".



Existem várias maneiras de definir bucles no modelagem de processos.

Os bucles podem ser definidos em apenas uma atividade (seja tarefa ou subprocesso), para isso, é preciso modificar a propriedade Looping type que pode ter os seguintes valores:

Standard

Uma atividade do tipo bucle padrão possui uma expressão booleana que é avaliada depois de cada ciclo do bucle. Se a expressão ainda for correta, o ciclo continua.

Existem duas variantes do bucle que refletem as construções de programação “While” e “Until”.

Em um bucle while, a expressão é avaliada antes de executar a atividade, o que significa que a atividade pode não ser executada nenhuma vez.

Em um bucle until, a expressão é avaliada depois que se tenha executado a atividade, o que significa que a atividade se executará ao menos uma vez.

O tipo de bucle é configurado com a propriedade “Test Time” (Before = While, After = Until).

A condição (booleana) pode ser definida mediante uma regra ou mediante um procedimento (para casos complexos)

Multi-Instance

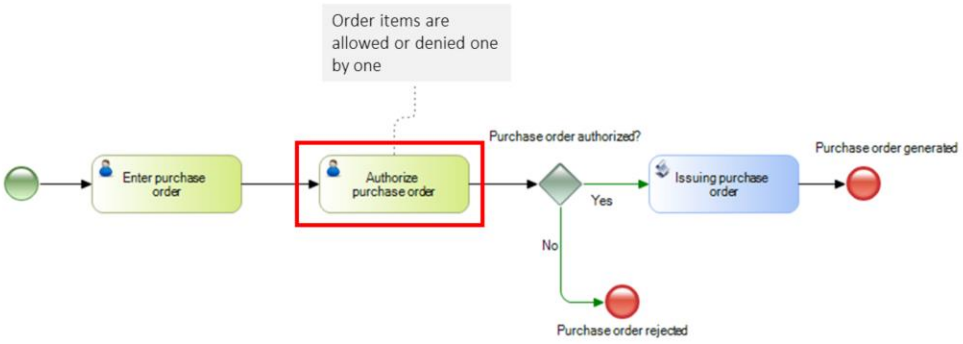
Um bucle Multi-Instance reflete a construção de programação “for each”.

A expressão de um bucle Multi-Instance é uma expressão numérica que avalia só uma vez antes de realizar a atividade. O resultado da avaliação da expressão é um número inteiro que especificará o número de vezes que a atividade deverá se repetir.

Como acontece com o bucle padrão, a expressão pode definir-se com um regra ou com um procedimento. Há também duas variantes do bucle Multi-Intance dependendo se as diferentes instâncias executam de forma sequencial ou em paralelo, o que é configurado na propriedade Ordening.

Example

Purchase process



Processo de compras

1. Um funcionário ingressa um pedido de compras
2. Cada ítem do pedido é autorizado separadamente pelo chefe do funcionário
3. Se ao menos um dos ítems do pedido foi autorizado então gera-se a ordem de compras correspondente

É necessário um mecanismo para declarar que uma atividade será executada potencialmente mais de uma vez

O número de vezes que será executada não é conhecido durante o desenho

Standard Looping

- The activity will be executed sequentially, provided that a condition of the true/false type is fulfilled.

- Properties involved:
  - Condition rule
  - Test time

A atividade será executada sequencialmente enquanto cumpra-se uma condição de tipo verdadeiro/falso

Outras propriedades:

Condition rule: permite especificar a condição que deve cumprir para que siga executando a atividade (implementação)

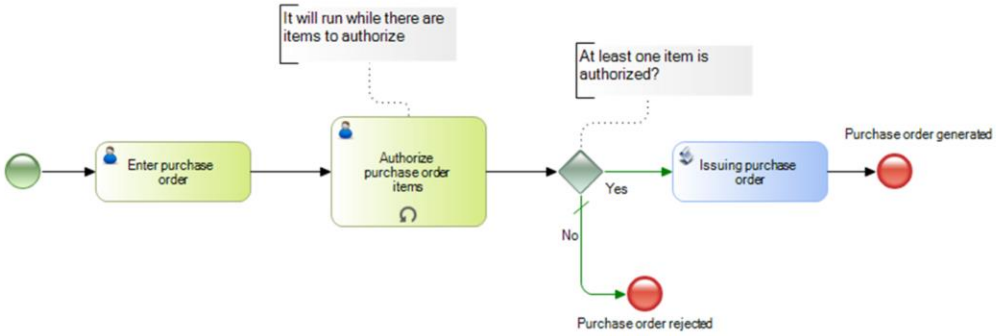
Test time: especifica o momento em que é avaliada a condição

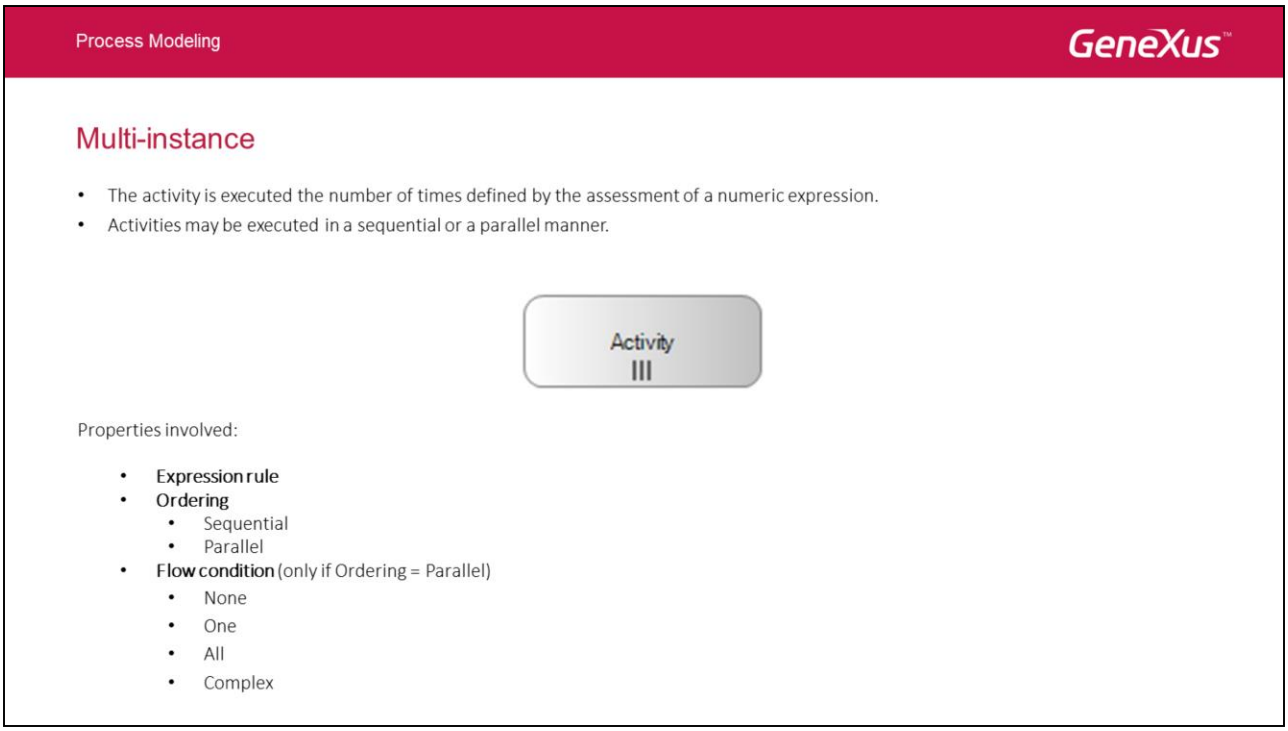
Before: a avaliação é feita antes de executar a atividade

After: primeiro executa-se a atividade e após avalia-se a condição

GeneXus

Standard Looping





A atividade será executada um número de vezes que será determinado mediante a avaliação de uma expressão numérica

As atividades podem executar em forma sequencial ou paralela

Outras propriedades:

Expresion rule: expressão numérica que indica o número de vezes que executará a atividade (implementação)

Ordering

Sequential: executará de forma sequencial

Parallel: executará de forma paralela

Flow condition (aplica somente se Ordering = Parallel)

Determina como os caminhos se unificam logo que diferentes instancias de uma atividade sejam executadas

Valores:

None: não há unificação. Logo que cada instância termina, o fluxo que que segue a mesma continua de modo que este fluxo se repite para cada instancia.

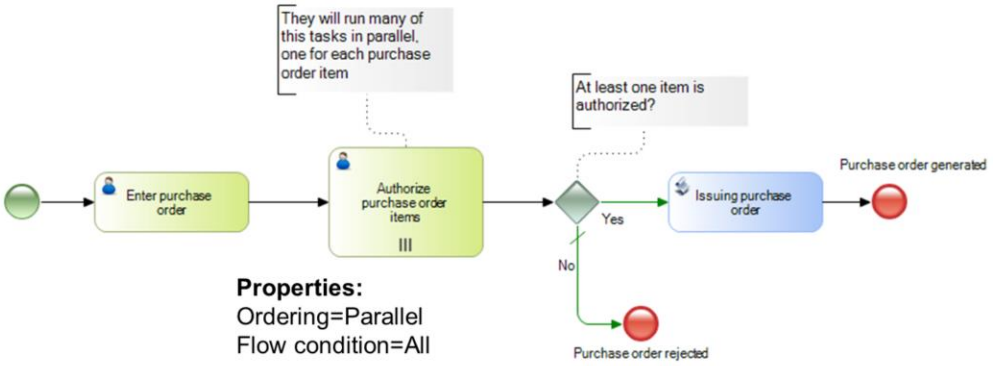
One: não há unificação. O processo continua logo que a primeira das instâncias termine de ser executada. E é executado apenas o fluxo que parte dela.

All: similar a um parallel Gateway, o processo continua somente se todas as instânicas da atividade finalizem.

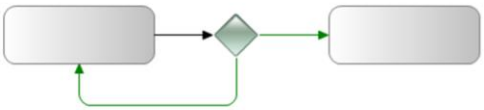
Complex: permite especificar uma condição que indicará quando o processo pode continuar. A condição é avaliada cada vez que uma instancia é finalizada.

GeneXus

Multi-instance

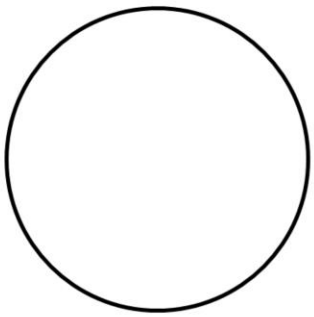


Loops



Os bucles também podem ser criados utilizando fluxos de sequência que se conectam com elementos “anteriores” do diagrama.

Events



Até agora, nós não mencionamos onde começa e termina o processo, para isso vamos utilizar o símbolo **Evento.**

BPMN Events

They represent occurrences taking place during the process

Types of events:

Os eventos representam **acontecimentos que ocorrem durante o processo**.
Há três tipos de eventos:

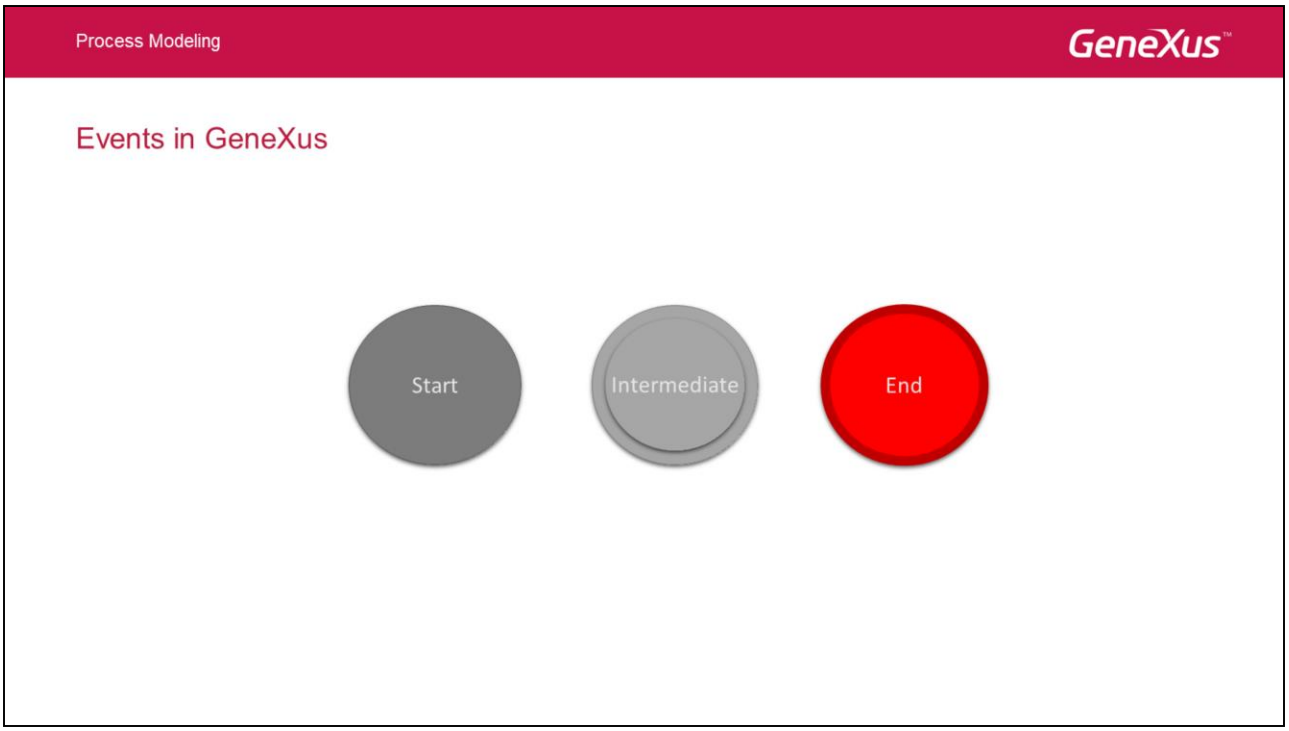
Início: Permite indicar onde inicia o processo. Um evento inicial **NÃO** pode ter conectores de entrada.
Intermediário: Permite indicar um acontecimento entre o início e o fim de um processo.
Fim: Permite indicar onde termina o processo. Um evento final **NÃO** pode ter conectores de saída.

Estes diferem no curso, os de **início tem linha fina**, os **intermediários tem linha fina dupla** e terminará com uma **linha grossa.**

BPMN Events

For each event type, BPMN defines sub-types to specify the nature of an event.

Para cada um destes três tipos BPMN define **subtipos** que permitem especificar a **natureza do evento**.



Os eventos são representados com um círculo. É algo que “acontece” durante o curso do processo de negócio. Esses eventos afetam o fluxo do processo e costumam ter uma causa (trigger) ou um impacto (resultado). Os eventos representados com um círculo com centro aberto permitem aos marcadores internos diferenciar diferentes trigger e resultados.

Há três tipos de eventos, baseados no momento em que afetam o fluxo:

Start

Como seu nome indica, um evento Inicial mostra onde um processo inicia. Sua utilização é opcional. Caso **NÃO** seja utilizado, qualquer atividade da qual **NÃO** chega conectores será considerada **Inicial**.

Recomenda-se sua utilização, pois faz o diagrama mais legível. BPMN estabelece que os eventos de início sejam representados com um círculo de traço fino (a cor é acrescentada por GeneXus).

Um evento inicial **NÃO** pode ter conectores entrantes.

Intermediate

O evento Intermediário indica que algo acontece (um evento) em algum lugar entre o início e o final de um processo.

Afeta o fluxo do processo, mas **NÃO** inicia ou termina (diretamente) o processo. BPMN estabelece que os eventos intermediários sejam representados com um círculo e traço duplo (a cor é acréscimo de GeneXus).

End

O evento de Fim indica que um processo vai terminar.

Sua utilização é opcional. Caso **NÃO** seja utilizado, qualquer atividade que **NÃO** tenha conectores salientes marca o fim desse caminho no processo (pode haver vários caminhos paralelos).

Um evento Final **NÃO** pode ter conectores salientes. BPMN estabelece que os eventos de fim se representem com um círculo de traço grosso (a cor é acrescentada por GeneXus)

Events in GeneXus

Os eventos podem ser classificados como “Catch” ou “Trhow”, dependendo se pegam ou disparam o evento.

Tipos de Evento:

None

É um evento que não tem um tipo definido.

É um evento intermediário, pode ser utilizado para estabelecer um marco no processo.

Timer

Quando é utilizado como um elemento independente, permite especificar um “delay” no processo, isso é, uma espera por um tempo determinado.

Quando é utilizado como um anexo a uma atividade, permite modelar quanto se quer esperar por essa atividade (deadline).

Conditional

Este evento é utilizado para esperar até que certa condição seja verdadeira.

Signal

Este evento é utilizado para enviar ou receber sinais dentro ou fora do processo (neste último caso, o alcance é a hierarquia de processos).

Error

O evento error interrompe o processo ou exige uma correção.

Quando é intermediário, é utilizado anexado a uma atividade e permite definir um caminho alternativo se existir a possibilidade de ocorrência de um erro ao executar tal atividade. Caso seja do tipo final, interromperá o processo.

Events in GeneXus

Message

Este evento envía o recibe un mensaje desde otra entidad de negocio o participante (otra Pool).

Este evento envia ou recebe uma mensagem da outra entidade do negócio ou participante (outra Pool).

Compensate

Este evento es utilizado para deshacer actividades en el caso de que un subproceso transacción sea cancelado o necesite deshacerse (rollback). En un evento catch solamente pueden estar adjuntos a actividades. Cuando es throw se puede colocar en el flujo normal del proceso

Link

El evento link siempre se utiliza de a par: (throw-catch), es necesario que ambos eventos tengan el mismo nombre para hacer el vínculo entre ellos. Su objetivo es crear un flujo de secuencia virtual.

Solo puede haber un evento link del tipo catch y existir muchos eventos link del tipo trhow con el mismo nombre. Son utilizados únicamente dentro de un mismo proceso. Para comunicación entre procesos se recomienda el uso de señales. O evento link é sempre utilizado aos pares: (throw-catch), é necessário que ambos os eventos tenham o mesmo nome para azer o vínculo entre eles. Seu objetivo é criar um flujo de sequencia virtual. Pode haver apenas um evento link do tipo catch e muitos eventos link do tipo throw com o mesmo nome. São utilizados unicamente dentro de um processo. Para comuncação entre processos, recomenda-se o uso de sinais.

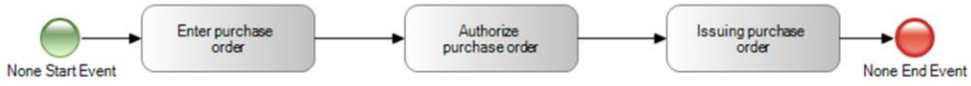
Cancel

Sirve para cancelar un subproceso transaccional. Un evento de finalización cancel indica que se debe realizar un rollback a dicho subproceso. Serve para cancelar um subprocesso transacional. Um evento de finalização cancel indica que se deve realizar um rollback em tal processo.

Terminate

Un evento “terminate” da por terminado el proceso. Esto es, a diferencia del evento final “none”, no solo termina el camino actual sino todo los caminos paralelos que puedan existir. Um evento “terminate” dá por terminado o processo. Isso é, diferente do evento final “none”, não apenas termina o caminho atual, mas todos os caminhos paralelos que possam existir.

Start and End Events

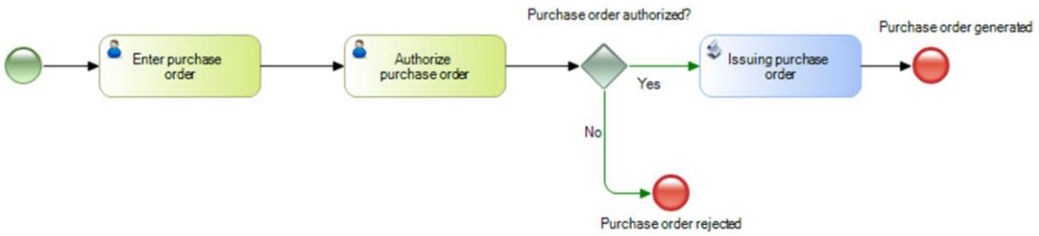


Então agregamos a nosso processo o início e o fim para indicar onde começa e onde termina.

GeneXus

Process Termination

- The “None” termination event is enough to indicate the end of a process, when we have **only one active path**



O evento de finalização “None” é atingido para indicar o fim de um processo somente se tem um caminho ativo

Process Termination

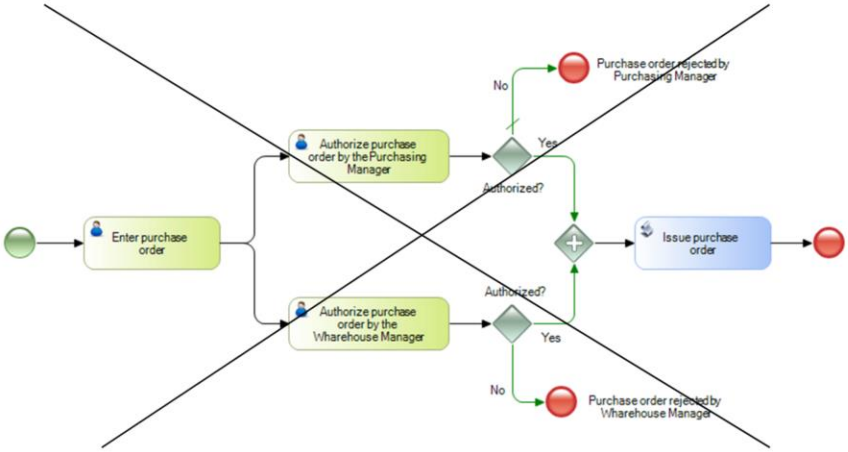
- What happens when we have two parallel authorizations?

¿O que acontece se temos duas autorizações em paralelo?

Se o fluxo do processo chega a determinado ponto, requer-se que o processo finalize, independentemente se tem outros caminhos paralelos que estão ativos

En este ejemplo si uno de los gerentes no autoriza la orden de compra, el proceso debe terminar, independientemente de si el otro gerente la aprobó o no.

Process Termination



La solución usando None End Events no funciona, porque este símbolo solamente finaliza el flujo que llega al mismo y el resto de los flujos continúan activos.

En el ejemplo, la orden de compra se emitiría igual si uno de los gerentes no la aprueba.

Process Termination

Terminate End Event

- Finalizes a process by interrupting all activities that remain active at that point.

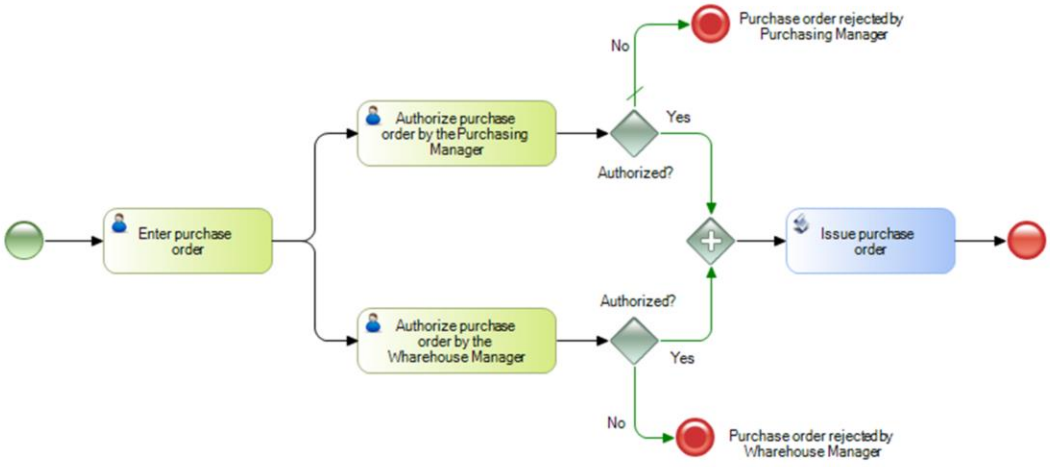
Teminate End Event

Finaliza um processo interrompendo todas as atividades que estão ativas nesse momento

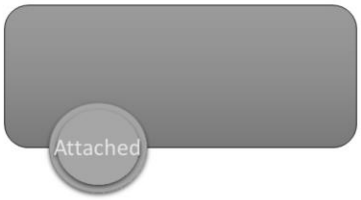
Se o processo é utilizado como subprocesso, NÃO finaliza-se o processo pai

GeneXus

Process Termination



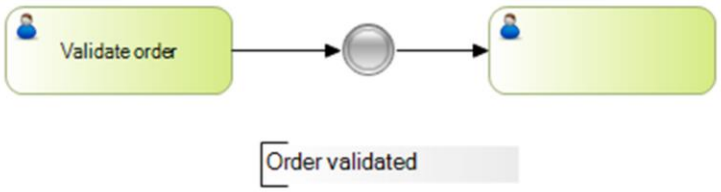
Attached Intermediate Events



É possível anexar eventos às margens de uma atividade (tarefa ou subprocesso) para modelar exceções ao fluxo normal do processo. Isso permite que, se durante o período em que a atividade estiver ativa ocorrer um evento, será possível disparar um fluxo alternativo com a possibilidade de interromper a atividade.

GeneXus

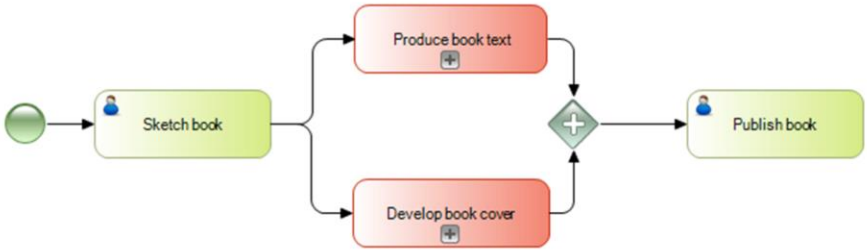
Indication of Milestones



Order validated

Signals

- They provide communication between processes



Signals

Example: Communication between processes

- Subprocess *Produce book text*

- Subprocess *Develop book cover*

Signals

Intermediate Event Signal (BPMN)

Throw
(to trigger the signal)

Catch
(to catch the signal)

Signal Intermediate Event

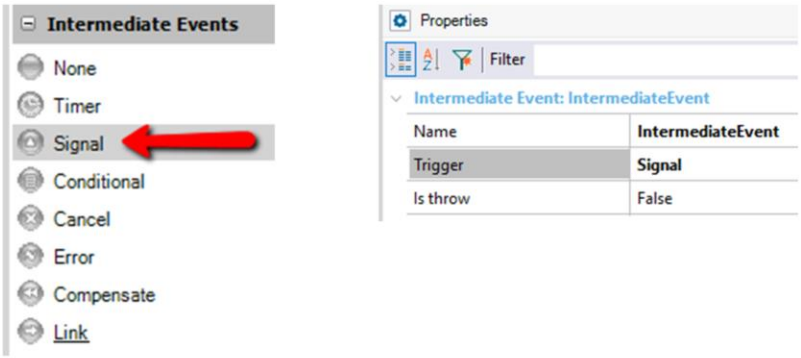
Permite sinalizar a ocorrência de um evento que poderá ser detectado em qualquer parte do processo, subprocessos e inclusos processos avós

Existem dois tipos:

- Throw (para disparar o sinal)
- Catch (para capturar o sinal)

Signals

Intermediate Event Signal (GeneXus)



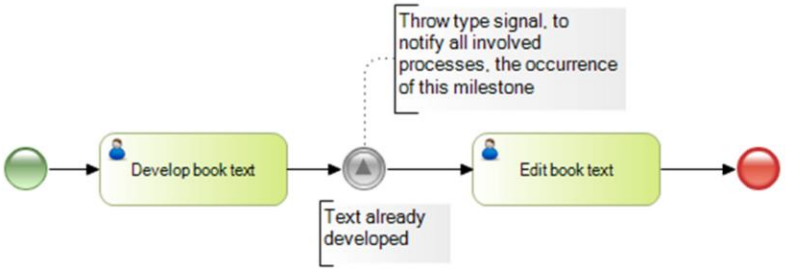
Em GeneXus, usamos um evento intermediário do tipo sinal e configuramos suas propriedades:

- **Trigger** = *Signal*
- **Name** = Os sinais são identificados por seu nome
- ***Is Throw*** = Se é um *Throw* temos que configurar a propriedade *Is Throw* = *true*

Signals

Example: Communication between processes

- Subprocess *Produce book text*

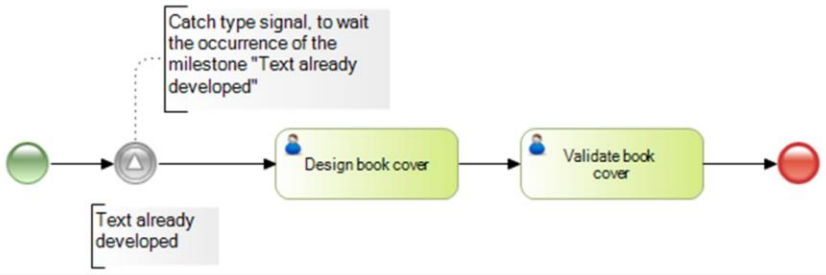


Se dispara una señal, para indicar que el hito de que el texto ya ha sido desarrollado.

Signals

Example: Communication between processes

- Subprocess *Develop book cover*

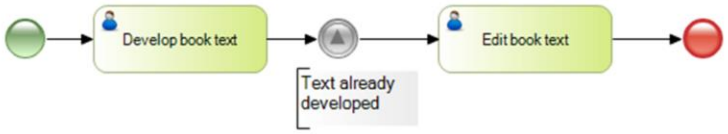


Se captura la señal para ver si ocurrió el hito de que el texto ya ha sido desarrollado.

Signals

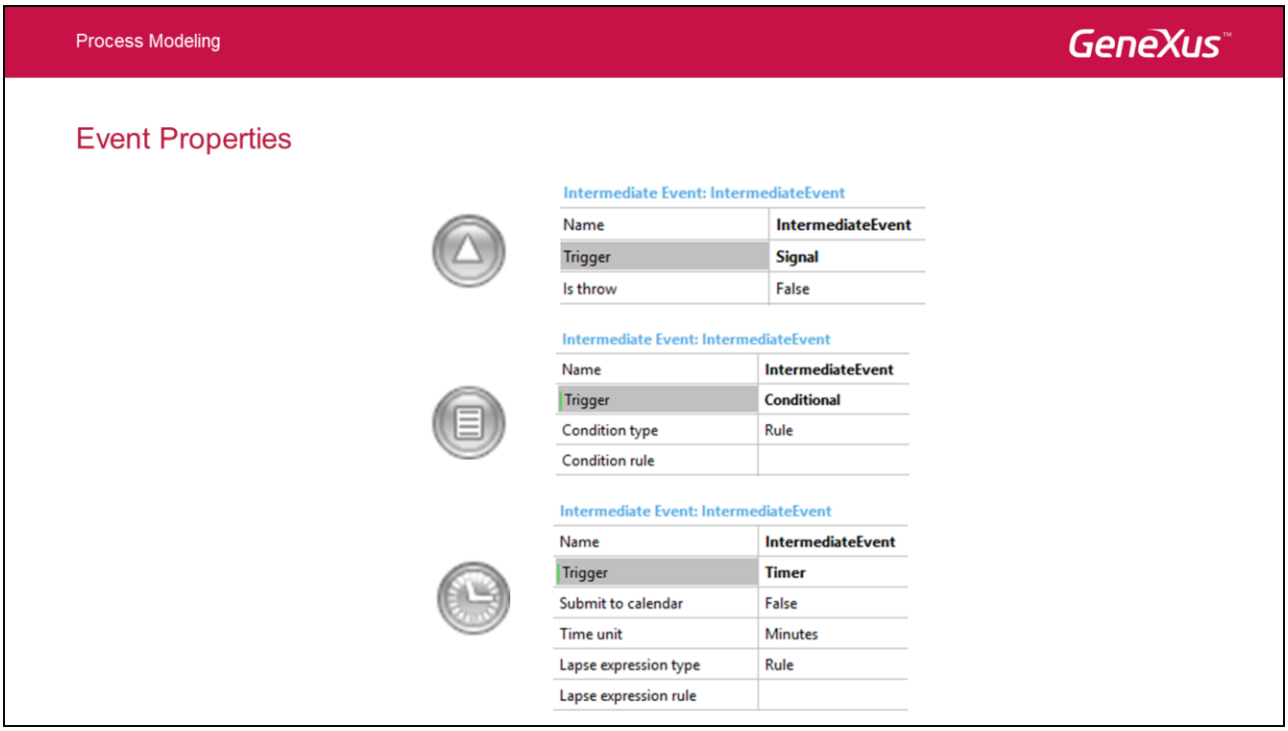
Example: Communication between processes

- Subprocess *Produce book text*



- Subprocess *Develop book cover*

Los procesos se comunican entre sí utilizando las señales.



Todos os eventos têm a propriedade Nome e Trigger, que indica qual é o tipo de evento.

Adicionalmente e dependendo do tipo de evento, podem existir outras propriedades.

Event Signal

A propriedade “Is throw” indica se está disparando ou esperando pelo evento.

O nome do evento é utilizado para discriminar entre diferentes sinais, isso é, se tenho um evento de sinal do tipo “catch” e nome “Signal”, espera-se por um evento de sinal de tipo “throw” e mesmo nome.

Event Conditional

Um evento condicional está associado a uma condição. Essa condição pode ser expressa com uma regra ou mediante o chamado a um procedimento.

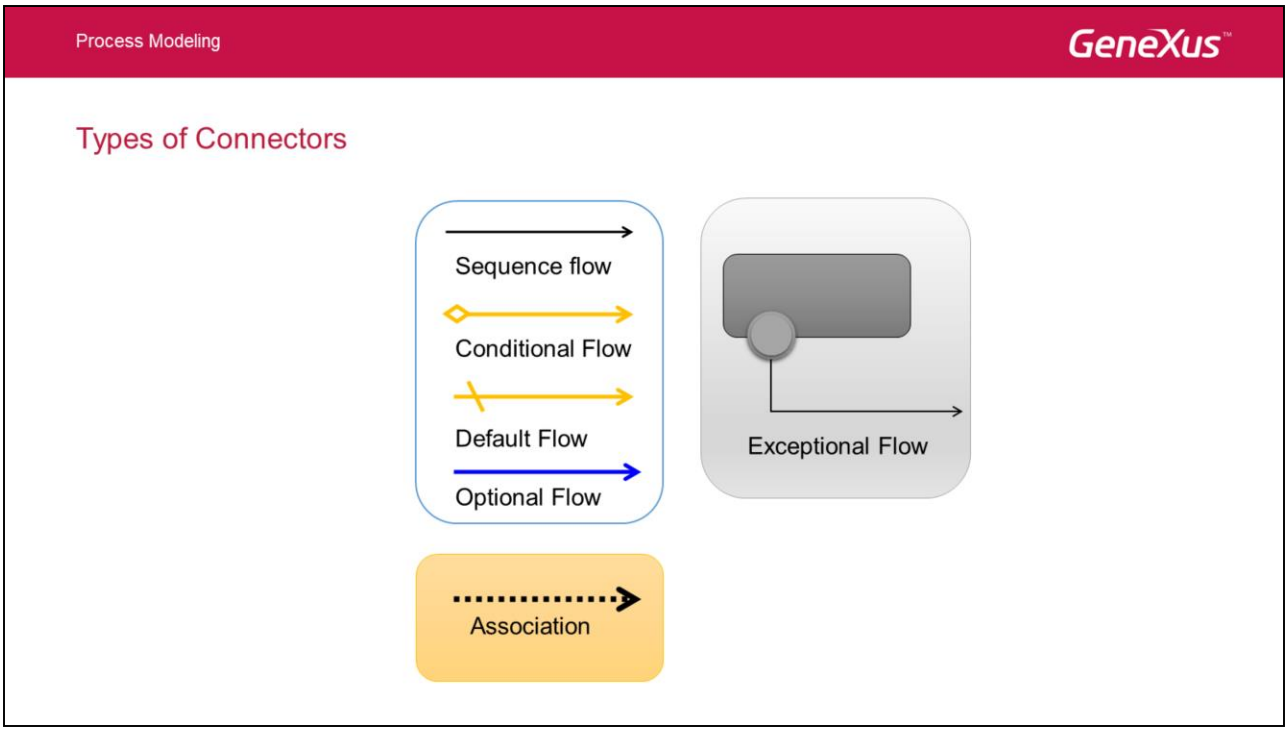
As regras de condição de um evento condicional são avaliadas pelo motor de workflow quando são modificados os dados relevantes da instância de processo à qual o evento pertence ou de um dado relevante compartilhado em um hierarquia de processos.

Como consequência, a expressão a definir tem que estar baseada em dados relevantes, já que se utilizasse atributos, a regra nunca chegaria a avaliar.

Os eventos condicionais sempre são do tipo “catch”.

Event Timer

Um evento de tempo está associado a uma expressão numérica (ou a um procedimento) que indicará o tempo que deve passar para que ocorra o evento (contando desde o momento de sua criação). Pode-se especificar a unidade de tempo e se está sujeito a um calendário particular.



Os elementos do diagrama de processos conectam-se entre eles em um diagrama para criar o esqueleto básico da estrutura de um processo de negócio.

Há três tipos de conectores que fazem essa função. Esses conectores são:

Fluxo Sequencial

O fluxo sequencial é representado por uma linha sólida com uma cabeça de flecha sólida e é utilizado para mostrar a ordem (a sequência) em que as diferentes atividades serão executadas no Processo.

Fluxo Condicional

O fluxo condicional é representado por um linha sólida com uma ponta de flecha sólida. Se o fluxo parte de uma atividade, no começo da linha utiliza-se um minilosango. Se o fluxo parte de uma comporta, então o minilosango não é utilizado porque o tipo de fluxo fica implícito.

Um fluxo condicional está associado a uma regra de condição que é avaliada no tempo de execução para determinar se tal fluxo deve ou não continuar.

Fluxo Default

Este fluxo é utilizado em exclusive ou inclusive gateways e é utilizado para modelar que é o fluxo que deve continuar quando os fluxos restantes condicionais salientes não avaliam como verdadeiro no tempo de execução.

Para diferenciá-lo, utiliza-se uma barra diagonal que o corta transversalmente no começo da linha.

Types of Gateways

Existem diferentes tipo de gateways que podem ser utilizados tanto para convergência, como para divergência. A seguir, os diferentes tipos são detalhados:

- **Exclusive:** *Divergência:* o fluxo seguirá apenas um dos caminhos que tem origem no Gateway, a escolha será feita avaliando as condições que se definem para o Gateway (terá uma por caminho). *Convergência:* quando múltiplos caminhos chegam ao Gateway, ele simplesmente deixa o fluxo avançar sem avaliar nenhuma condição nem realizar nenhuma sincronização.

- **Event:** *Divergência:* o fluxo avança por apenas um dos caminhos que tem origem no Gateway, conforme a ocorrência de um evento especificado.
Convergência: tem o mesmo comportamento que o Gateway Exclusive.

- **Inclusive:** *Divergência:* o fluxo seguirá todos os caminhos originados no Gateway que cumpram as condições do Gateway (terá uma para cada caminho).

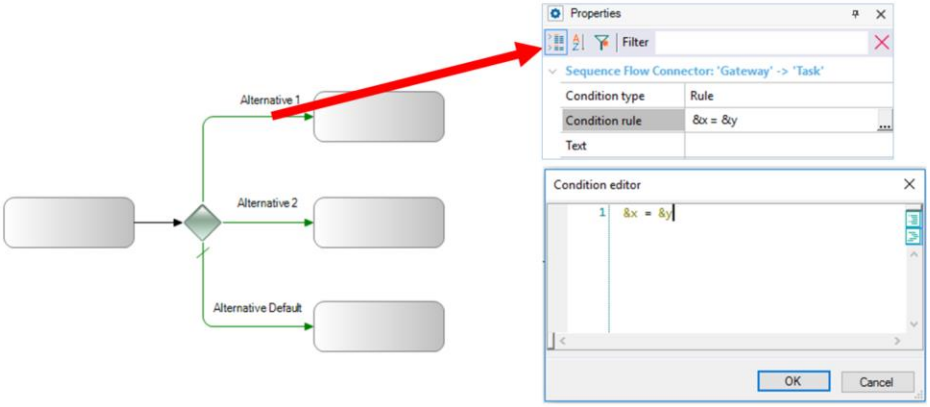
Convergência: realiza uma sincronização de todos os caminhos por onde o fluxo do processo efetivamente chegará ao gateway.

- **Parallel:** *Divergência:* o fluxo avança por todos os caminhos originados no Gateway.

Convergência: realiza uma sincronização de todos os caminhos que chegam ao Gateway.

A seguir, veremos detalhadamente a utilização de cada um deles em diferentes exemplos de modelagem de processos.

Exclusive Gateway



Flow -> One of the routes

Os **Exclusive** gateways (decisões) são lugares dentro de um processo de negócio nos quais a sequência de fluxo pode tomar dois ou mais caminhos alternativos. Para uma determinada instância do processo, pode-se tomar apenas um dos caminhos.

Uma decisão não é uma atividade da perspectiva do processo do negócio, é um tipo de porta de conexão que controla a sequência de fluxo entre as atividades.

Pode ser pensado como uma pergunta que se faz neste momento do processo e que tem um conjunto definido de respostas (apenas uma delas pode ser verdadeira em uma determinada instância).

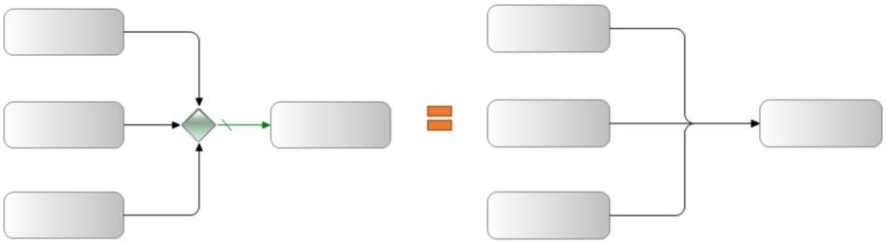
A cada conector saliente é associado uma regra de condição.

Durante a execução do gateway, avaliam-se as regras condicionais de cada uma dos conectores salientes. A primeira que avalie como verdadeira será a que determina o caminho a seguir.

Se nenhuma das regras avaliar como verdadeiro, então se seguirá pelo caminho default (o conector que está marcado com uma linha transversal).

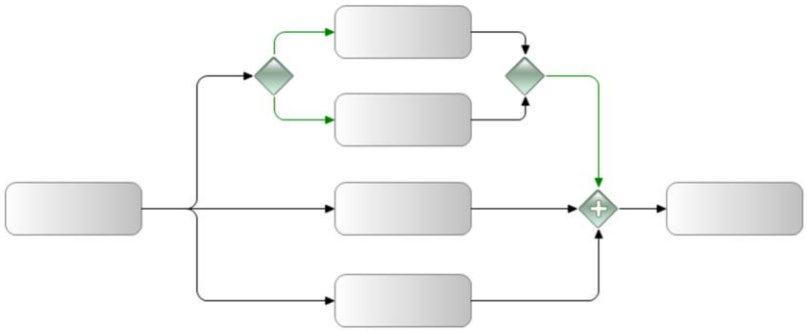
O conector default não é obrigatório; neste caso, deve-se assegurar que em cada instância ao menos uma das regras avalie como verdadeiro.

Exclusive Gateway (Synchronization)



Os **Exclusive** gateway também podem ser utilizados para sincronização embora sua utilização poucas vezes seja necessária para a modelagem, já que geralmente pode ser modelado sem eles (como se vê na figura da direita) obtendo o mesmo comportamento.

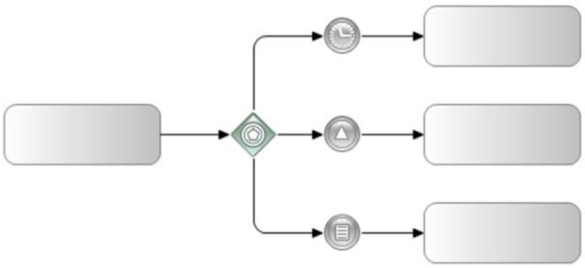
Exclusive Gateway (Synchronization)



Há certas situações em que um **Exclusive** gateway é requerido para sincronizar.

No exemplo da figura, se o **Exclusive** não fosse utilizado para sincronizar o resultado de um gateway anterior, **Parallel** gateway teria quatro conectores da sequência de entrada. No entanto, apenas três das quatro sequências de fluxo poderiam passar por vez, logo, o processo ficaria obstruído nesse **Parellel** gateway.

Event Gateway

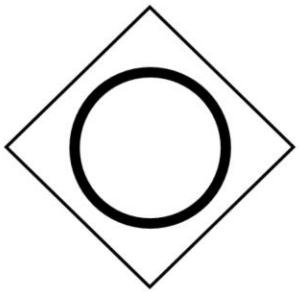


Flow -> One of the routes

Um gateway do tipo **Event** é um caso particular de um **Exclusive** gateway (no sentido de que apenas um dos caminhos possíveis pode ser escolhido), no qual a decisão de que caminho seguir depende da ocorrência de um evento e não da avaliação de um regra de condição.

É importante esclarecer que se nunca ocorrer nenhum dos eventos especificados, o processo ficará estancado, por esse motivo, recomenda-se utilizar sempre um evento **Timer** como uma das alternativas.

Inclusive Gateway

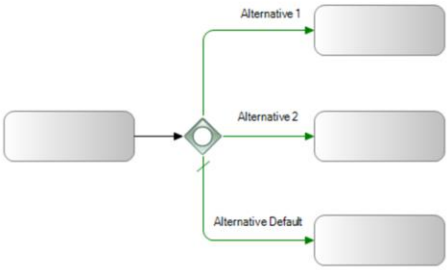


Um **Inclusive** gateway é semelhante ao **Exclusive** no sentido de que cada conector está associado a uma regra condicional. A diferença é que todas as regras sã avaliadas e segue-se por todas aquelas que avaliam como verdadeiro.

Recomenda-se o uso dos caminhos default para evitar que o processo fique obstruído em qualquer situação.

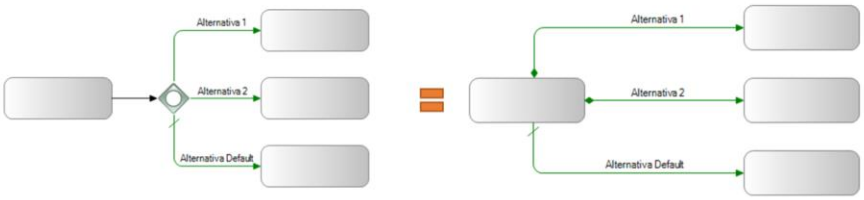
GeneXus

Inclusive Gateway (Branching)



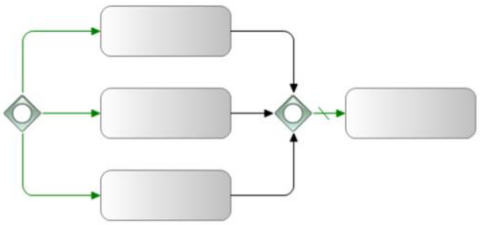
Flow -> All routes

Inclusive Gateway (Branching)



Pode-se modelar este mesmo comportamento sem utilizar um **Inclusive** gateway. Em seu lugar, podemos utilizar sequências de fluxo do tipo comercial que também estão associadas a uma regra de condição.

Inclusive Gateway (Synchronization)

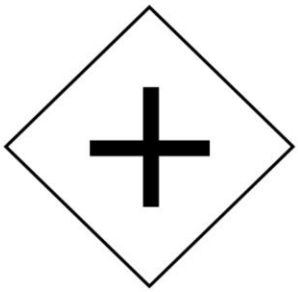


Um **Inclusive** gateway também pode ser utilizado para sincronizar.

Neste caso, o gateway espera por todas as sequências de fluxo que foram selecionadas em um passo anterior.

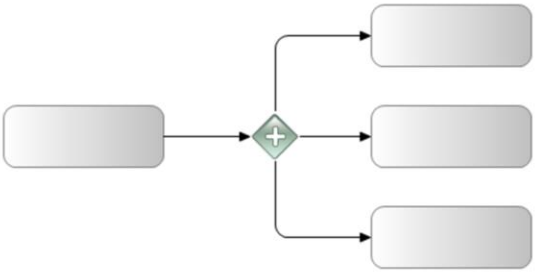
Atualmente, o motor de Gxflow suporta apenas essa modalidade quando o **Inclusive** gateway que sincroniza está emparelhado com outro **Inclusive** gateway, como mostra a figura.

Parallel Gateway



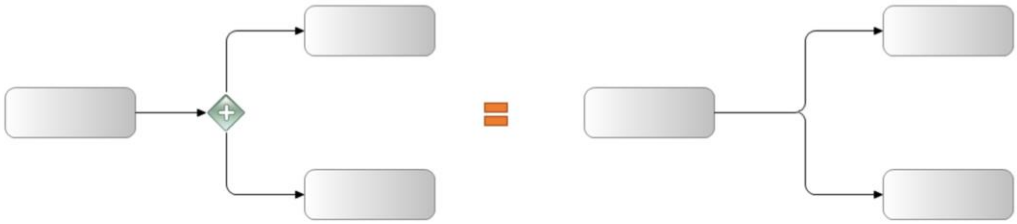
Um **Paralell** gateway permite criar fluxos paralelos (ou sincronizá-los quando se utiliza nessa outra modalidade).

Parallel Gateway (Branching)



Simplesmente divide o fluxo em dois ou mais caminhos paralelos, não avalia nenhuma condição.

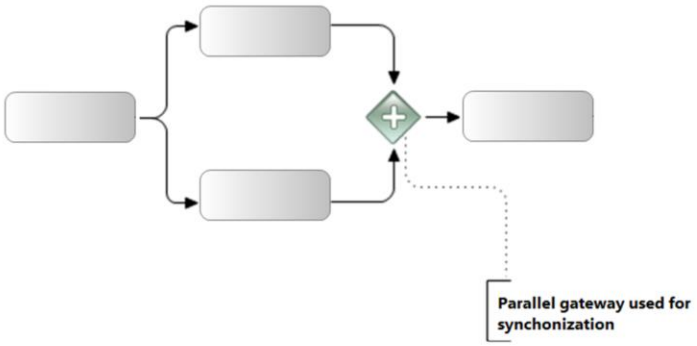
Parallel Gateway (Branching)



Pode-se modelar este mesmo comportamento sem utilizar um **Paralell** gateway, como mostra a figura. No entanto, o uso do gateway pode esclarecer o diagrama em determinas circunstâncias.

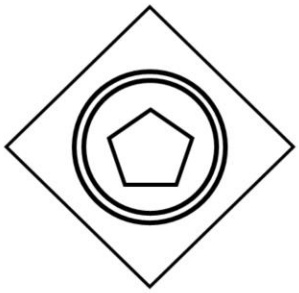
GeneXus

Parallel Gateway (Synchronization)



Quando o **Parallel** gateway é utilizado para sincronização, esta espera por todas as sequências de fluxo entrantes para então poder seguir adiante.

Gateway Event

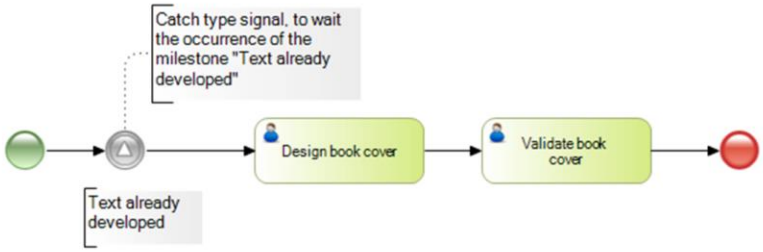


Um gateway de tipo **Event** é um caso particular de um **Exclusive** gateway (no sentido que somente um dos caminhos possíveis pode ser escolhido) em que a decisão de qual caminho a seguir depende da ocorrência de um evento e não da avaliação de uma regra de condição.

Está claro que se nunca ocorrer nenhum dos eventos especificados, o processo será parado, recomenda-se utilizar sempre um evento **Timer** como uma das alternativas.

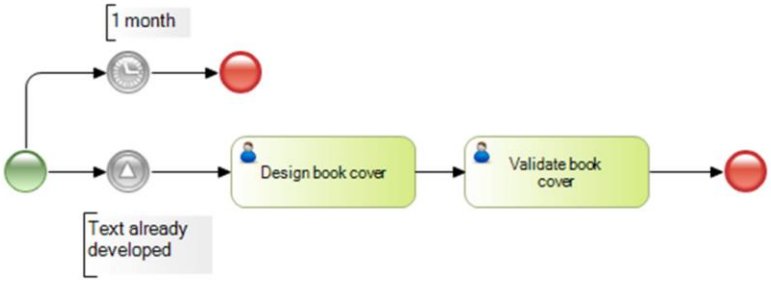
Gateway Event

The process below may be blocked indefinitely awaiting the arrival of the signal.

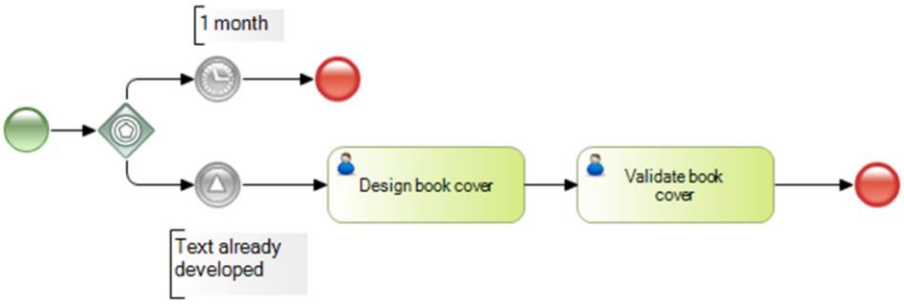


Event-based branching

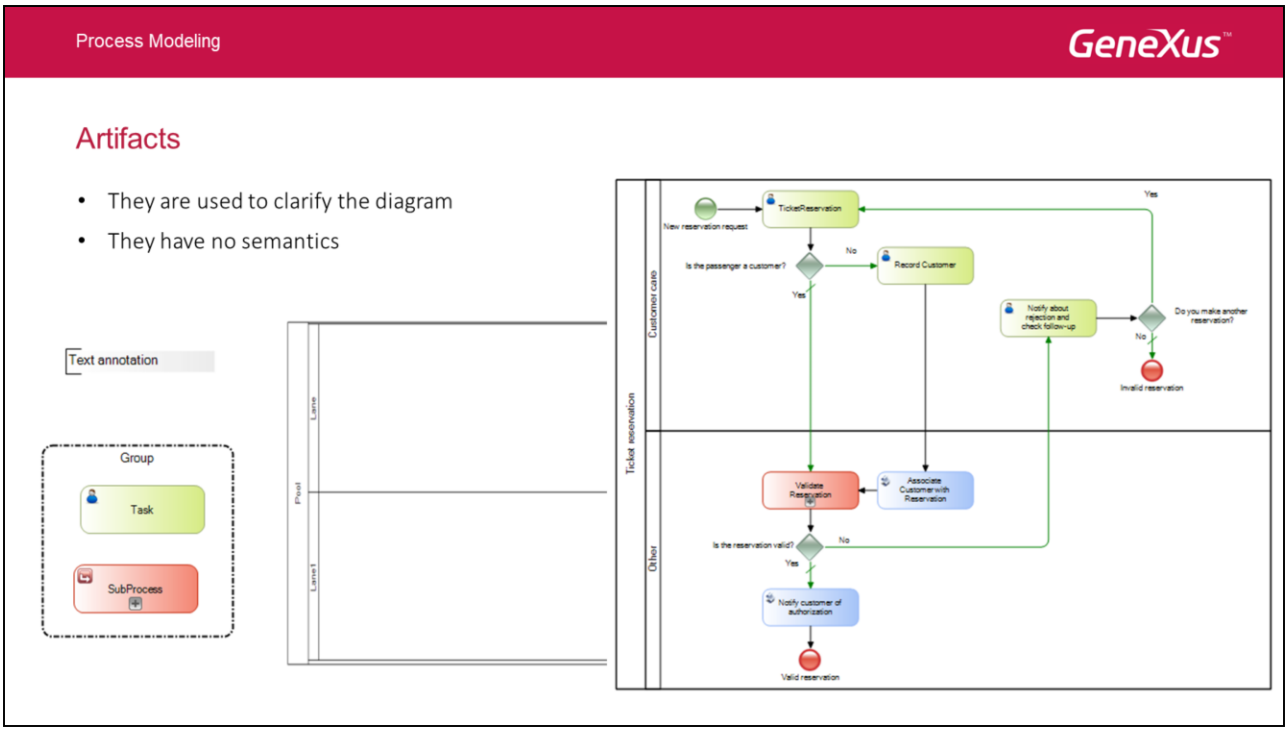
We avoid the blocking by defining a maximum waiting time of one month.



Event Gateway



Cuando utilizamos un gateway del tipo evento, el camino que continúe depende del tipo de evento que se capture.



Os Artifacts são elementos utilizados dentro dos diagramas para dar mais claridade a ele. Os artifacts disponíveis são:

• **Text annotation:** As anotações de texto são um mecanismo para que o modelador possa proporcionar informação adicional para o leitor do diagrama.

Graficamente, representa-se com um retângulo aberto com um linha negra sólida.

Uma anotação de texto pode-se conectar a um determinado elemento do diagrama utilizando uma associação (flecha pontilhada).

• **Group:** Um grupo é um artefato que proporciona um mecanismo visual para agrupar elementos de um diagrama de maneira informal.

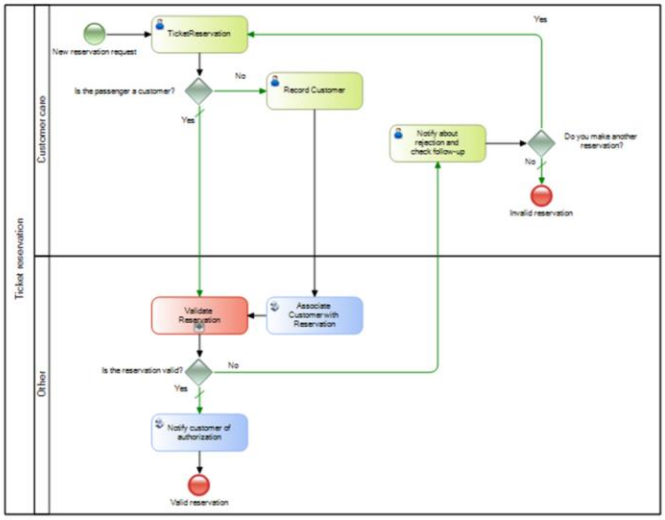
Graficamente, é representado com um retângulo de vértices arredondados formado por uma linha pontilhada.

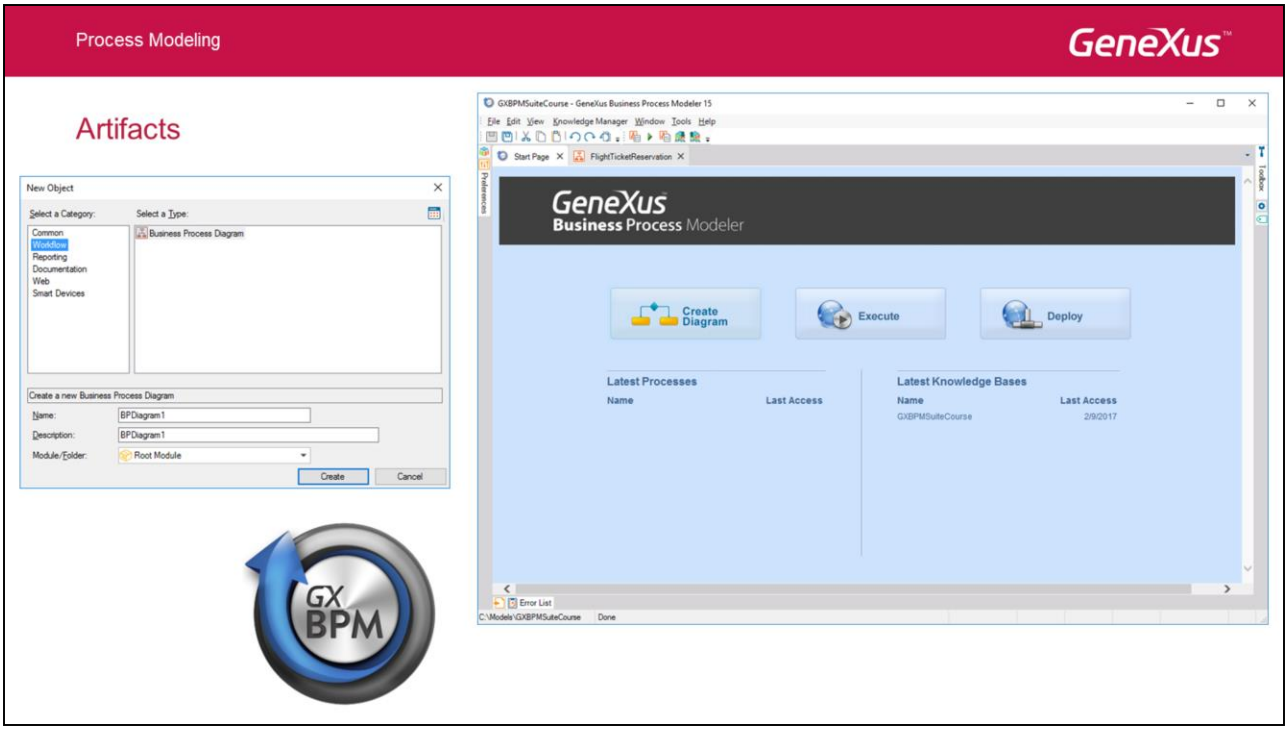
- **Pool:** Representa o processo de um participante e é utilizado quando o diagrama envolve duas entidades de negócio ou participantes separados. As atividades dentro dos contendores são consideradas processos autônomos, por isso o fluxo de mensagens definem-se como o mecanismo para ter comunicação entre dois participantes.
- **Lane:** Permite associar as atividades com uma função ou papel específico da empresa.

GeneXus

Artifacts

Pool and Lanes example





Para **modelar processos de negócio** pode-se utilizar o objeto Business Process Diagram que encontra-se em GeneXus mas também pode-se utilizar a ferramenta **GeneXus Business Process Modeler** (genexus.com/gxbpm) que é distribuida de forma independente do GeneXus. Esta ferramenta foi pensada para resolver dois cenários, os quais descrevem a continuação.

Cenário 1: fase de análise

O primeiro na fase inicial do projeto onde teremos os analistas, que são funcionais que entendem muito bem do negócio da empresa e conhecem de negócio mas não necessariamente tem um perfil técnico. E o que precisam é uma ferramenta que lhes permita captar esse conhecimento, documentá-lo e formalizá-lo de forma que possa ser automatizado, neste caso com GXflow.

Cenário 2: modificação de processos que já estão em produção

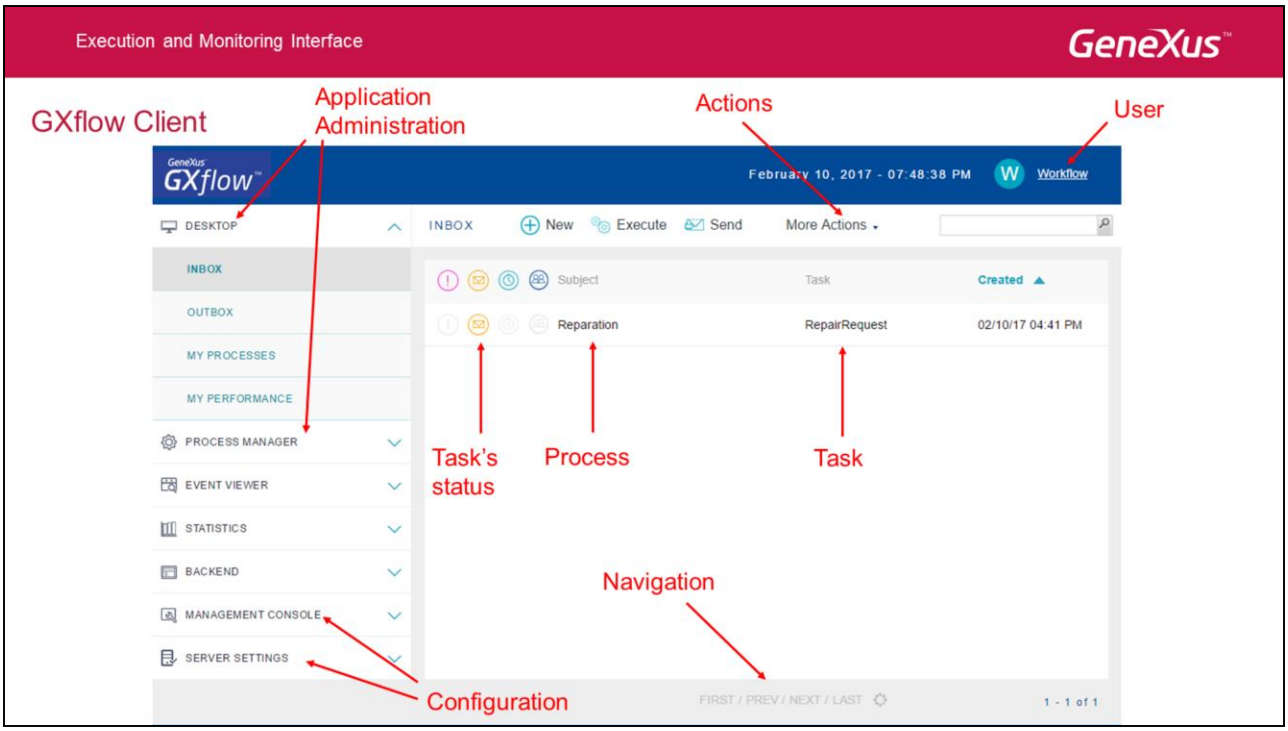
O segundo cenário está localizado em outro extremo do projeto, quando o sistema está em produção ou sendo implantado. Por exemplo um ERP que está sendo implantado em uma empresa necessita-se parametrizar determinadas regras de negócio.

GeneXus 15

Module 3

Execution and Monitoring Interface

Neste Modulo, estudaremos os componentes do cliente GXflow.



Esta é a interface do cliente GXflow: à esquerda, temos as aplicações; à direita, à medida que se seleciona uma aplicação, podemos ver a informação que ela contém. Essas aplicações são:

Área de trabalho

- Caixa de Entrada, Caixa de Saída, Meus processos, Meu rendimento

Administrador de Processos

- Processos, Tarefas, Eventos de Negócio, Definições de Processos, Calendários de Processos, Participantes, Telas de notificação

Visor de Eventos

- Eventos

Estatísticas

- Análise de processos, Análise de tarefas, Performance de processos, Performance de tarefas, Performance de equipes

Console de Administração

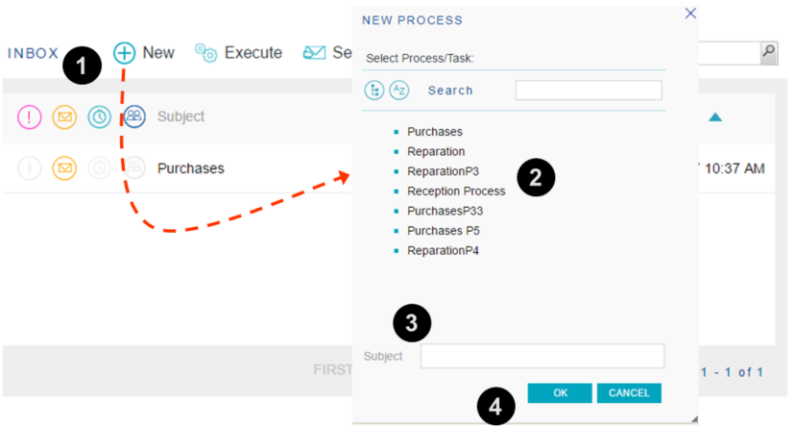
- Modelo Organizacional
  - Usuários, Papéis, Definições de Unidades Organizacionais, Unidades Organizacionais

Configurações do Servidor

- Principal, Segurança, Avançados

Inbox

Start of new process instances: Only users who have roles associated with the start element



1- New Button
2- Select process/task
3- Subject
4- Confirm

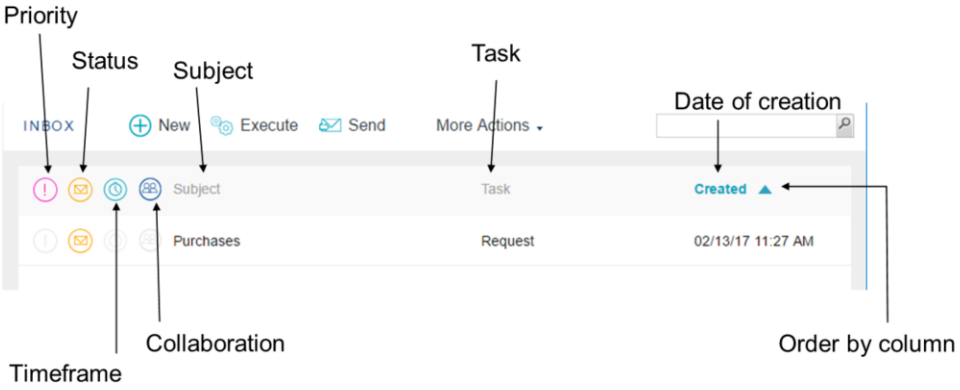
Essa é a caixa de entrada na qual os usuários interagem com as tarefas, executando-as e concluindo-as.

Poderão iniciar instâncias de processo cujo símbolo inicial é um evento Start, uma tarefa Interativa o um tarefa Batch; além disso, o usuário logado deve ter seu papel associado a algum desses símbolos.

Execution and Monitoring Interface

GeneXus

Inbox



O quadro com as tarefas a serem executadas compõe-se pelas seguintes colunas:

Prioridade: Esta coluna indica a prioridade da instância do processo. Pode ser alta, normal ou baixa; se for normal, não vemos nenhum ícone.

Estado: Esta coluna indica o estado de uma tarefa.

Prazo: Indica o tempo que resta até que a tarefa expire.

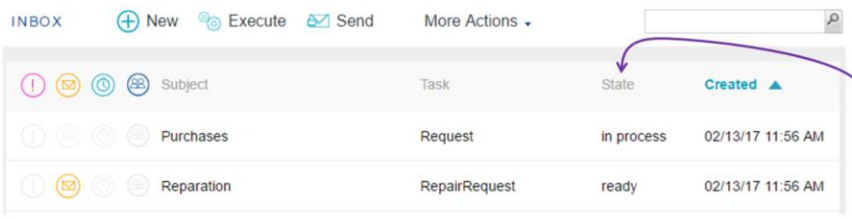
Colaboração: Indica que há outros usuários colaborando na tarefa.

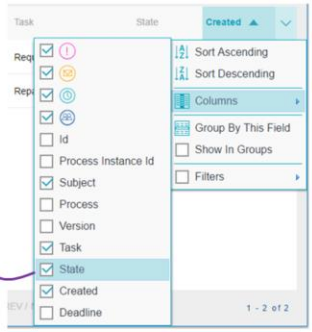
Assunto: Indica o assunto da instância de processo.

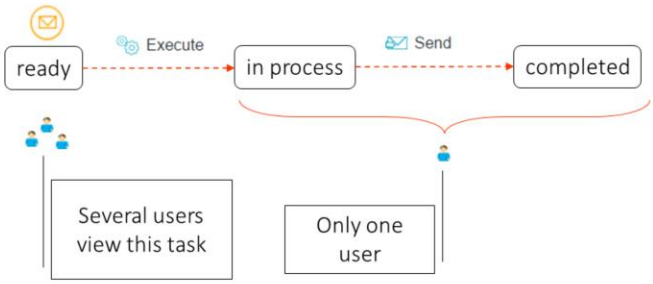
Tarefa: Indica o nome da tarefa que se está processando.

Data de criação: Indica a data de criação da tarefa.

Implementation cycle of a task



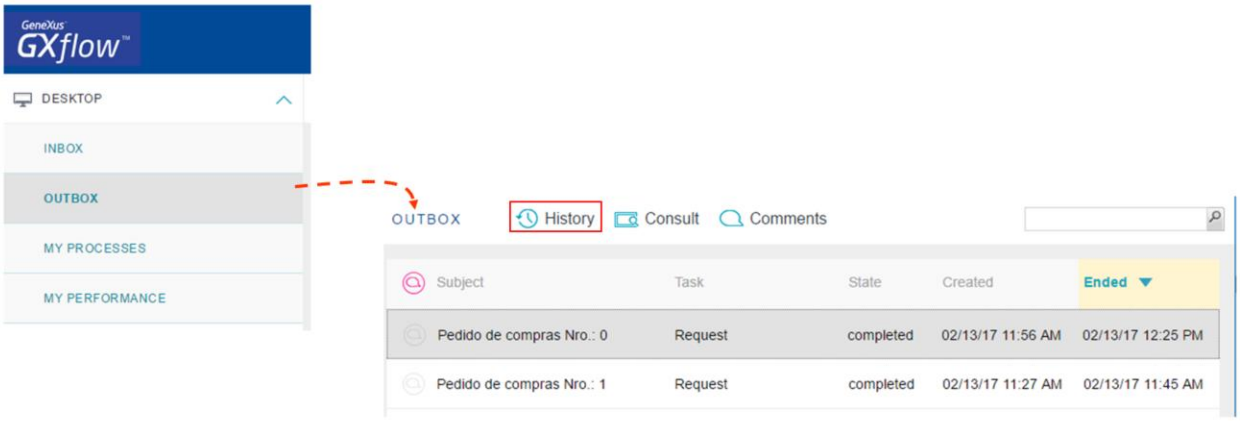




GeneXus

Outbox

- Tasks completed by the user

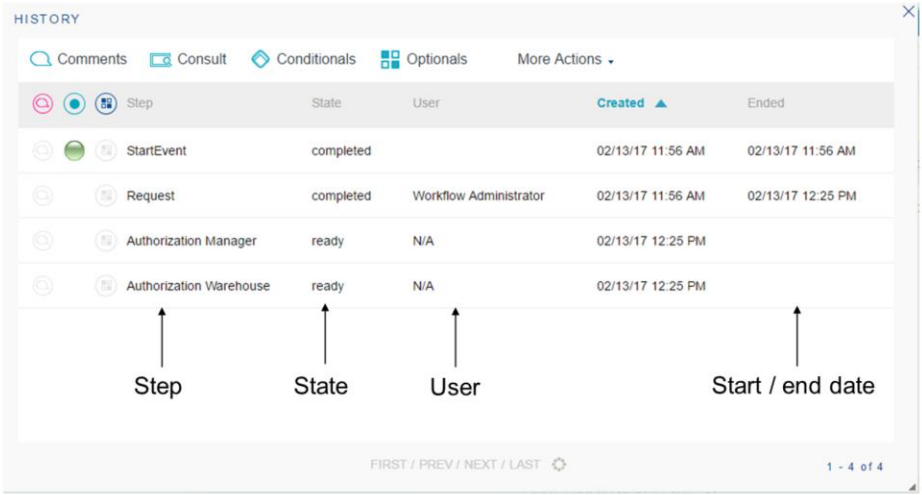


A caixa de saída permite-nos ver as tarefas que o usuário que logado processou. Também se pode consultar a tarefa, se foram inseridos comentários e o histórico.

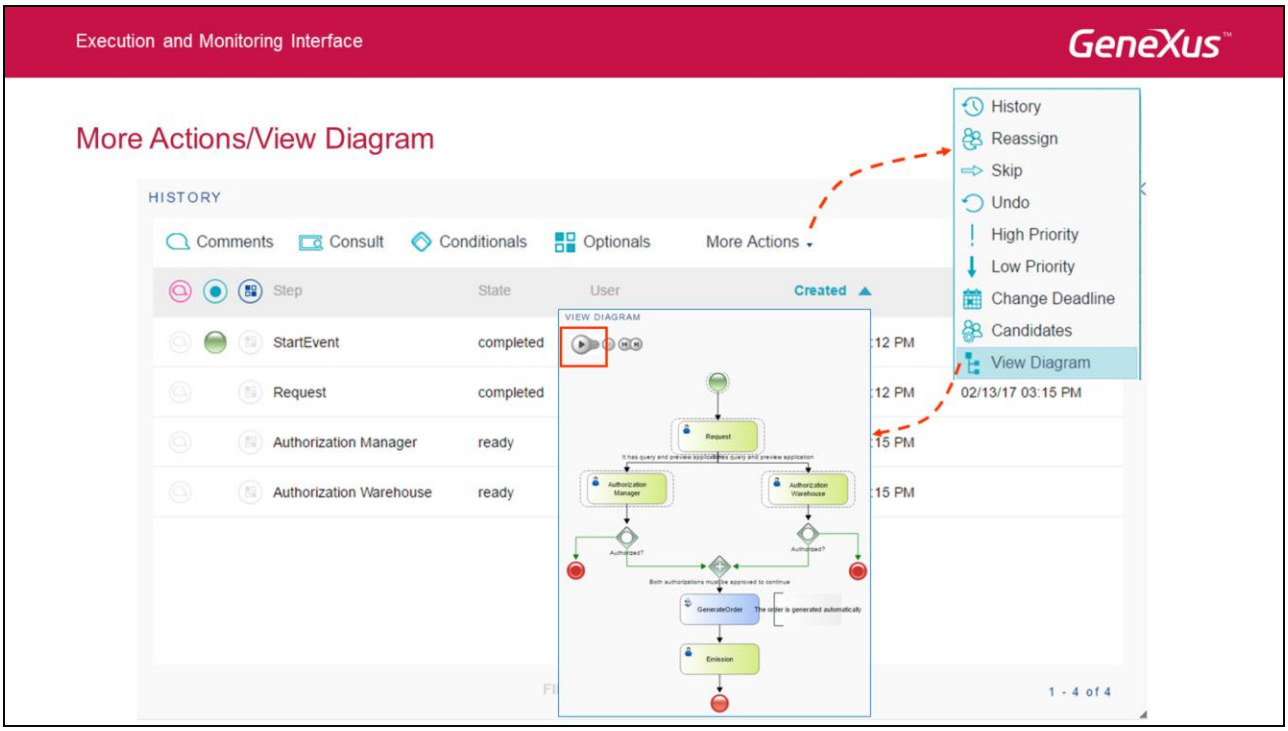
GeneXus

History Log

- Historical data of process execution



O histórico de uma instância de processo permite-nos visualizar como se deu o fluxo na medida em que ela foi sendo processada. Por exemplo, que valor as condições tomaram, que caminho especial foi escolhido, que usuário processou cada tarefa, em que data se deu o início e o fim.



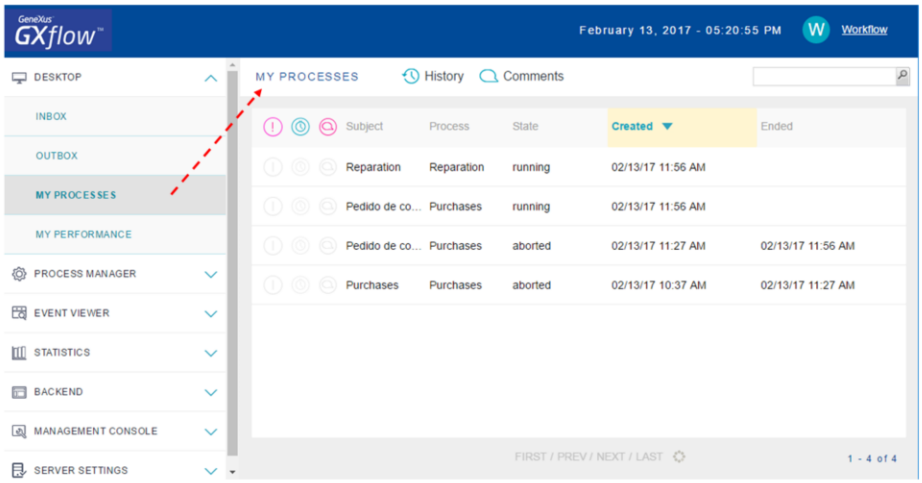
También é possível ter uma representação gráfica do processo e visualizar a ordem em que foram executadas as diferentes tarefas apenas clicando no botão “Ver diagrama”, como se vê na imagem.

Ao clicar no botão de reprodução, o usuário inicializará a animação que mostrará passo a passo as tarefas que foram geradas para a instância do processo consultada.

Além disso, ao posicionar o ponteiro do mouse sobre alguma das tarefas, aparecerá um pequeno quadro de texto com informação sobre a tarefa.

My Processes

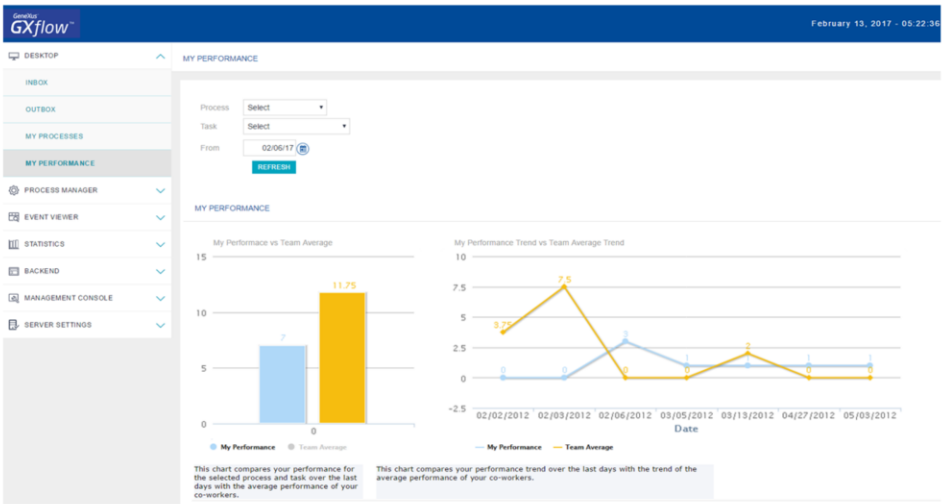
- Process instances created by the user



A aplicação de “Meus processos” permite-nos visualizar as instâncias que o usuário criou, comentários que foram inseridos e o histórico do processo.

My Performance

- Compare user performance

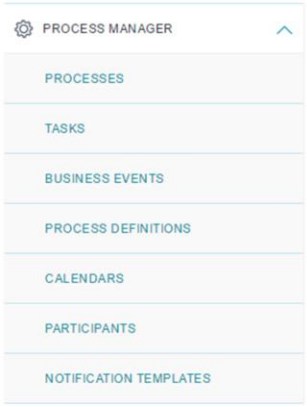


Esta aplicação permitirá que usuário, baseado em uma data, compare seu rendimento para os processos e tarefas selecionados com o rendimento de seus companheiros de trabalho que compartilham um mesmo papel ou tarefa.

GeneXus

Process Manager

- Manages everything related to processes (instances, definitions, statuses, etc.)

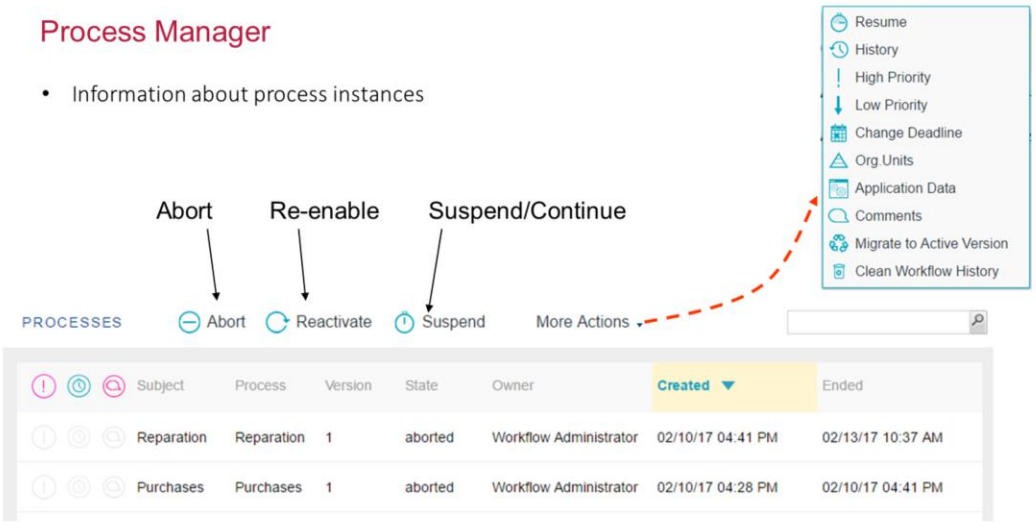


O administrador de processos conta com as aplicações de administração relacionadas aos processos de negócios.

GeneXus

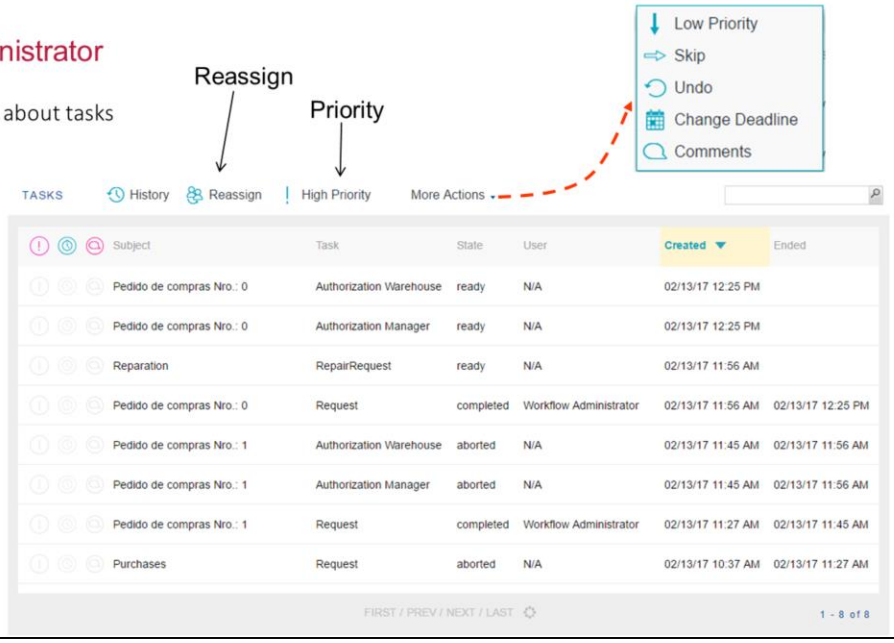
Process Manager

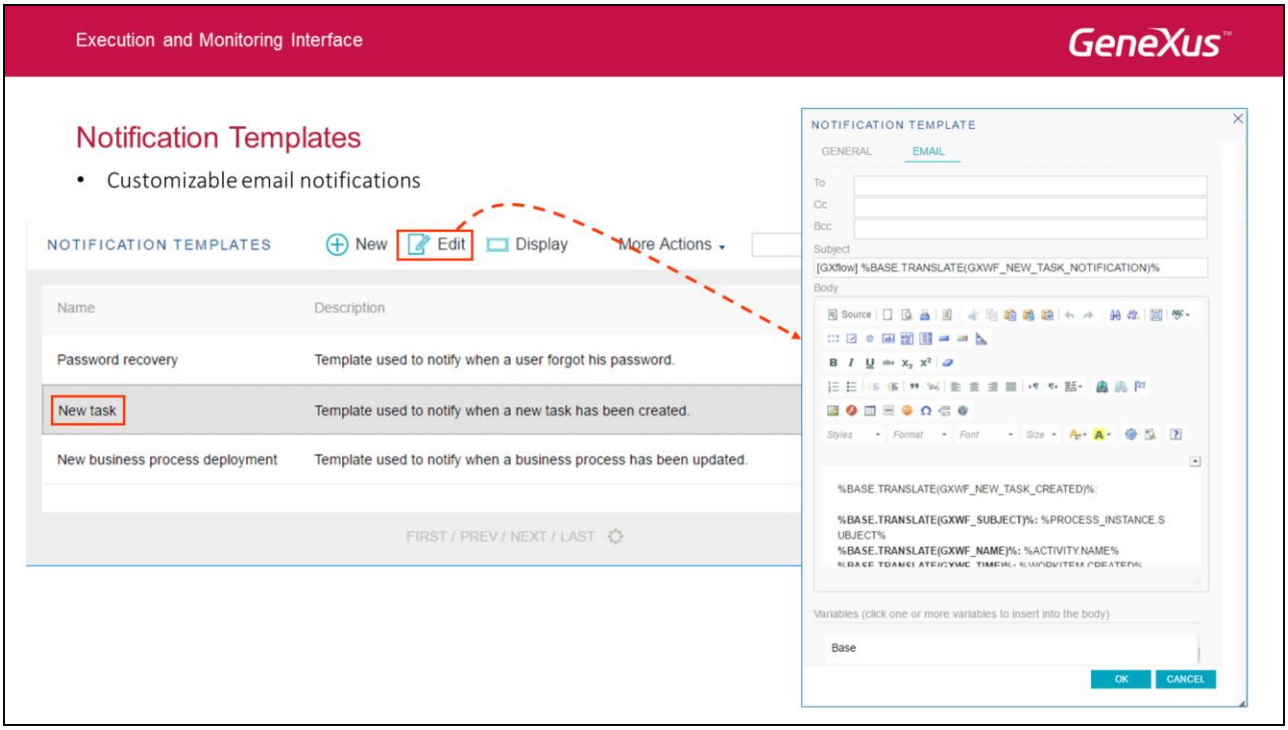
- Information about process instances



Tasks administrator

- Information about tasks

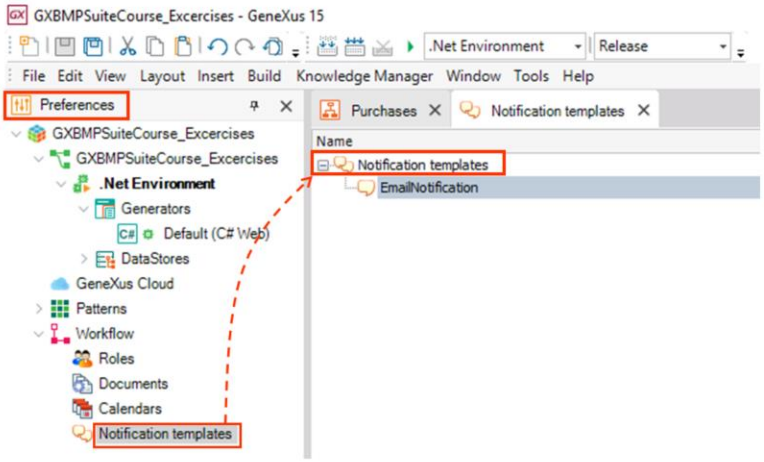




Existem modelos pré-definidos para a notificação de e-mail, que podem ser personalizados por meio de variáveis de processo. O usuário administrador poderá editar o destinatário, assunto e corpo de mensagem, como se vê na imagem.

GeneXus

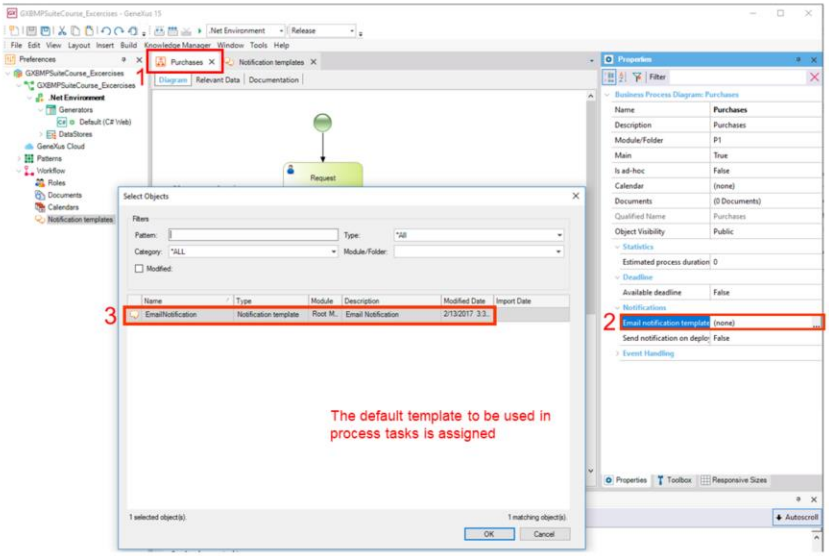
Creating a template in GeneXus



Outra opção é um novo modelo.

Para isso, é preciso ir a Preferences – Workflow - Notification templates, no momento de defini-lo, é preciso indicar o nome e uma descrição.

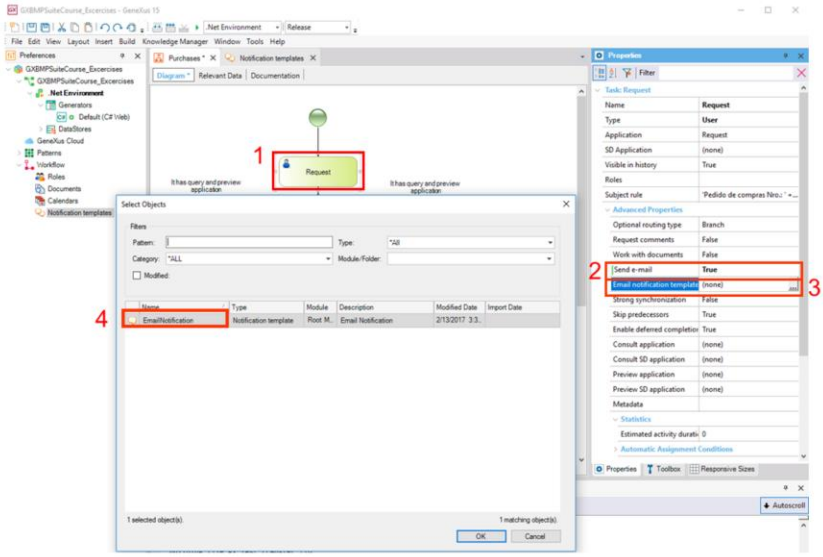
Associating the template at diagram level



Os modelos de notificação podem ser associados ao diagrama, deixando-os disponíveis para todas as tarefas do processo.

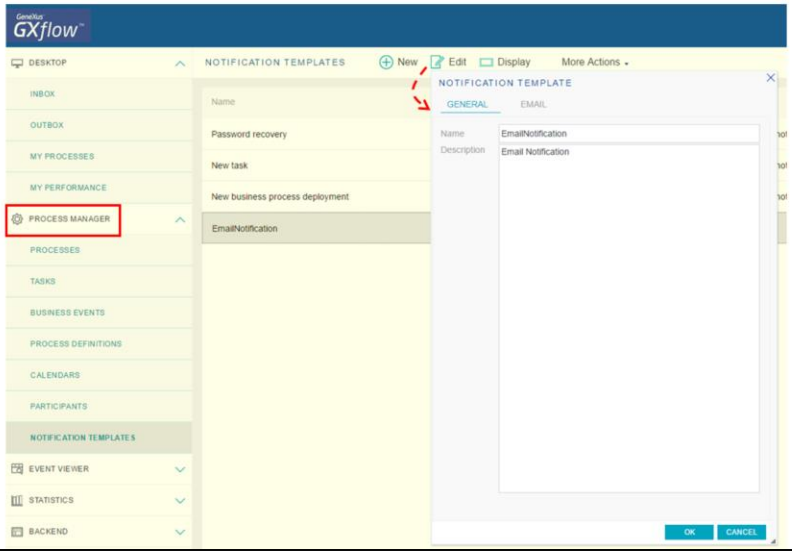
Também é possível enviar uma notificação ao fazer um deploy do processo, para isso, tem que marcar a propriedade «Send Notification on deploy» como True.

Associating the template at task level



Os modelos de notificação podem ser associados diretamente a cada tarefa, para isso, é preciso marcar a propriedade «Send Email» em True, posteriormente, escolhe-se o modelo.

Editing the template in GXflow Client



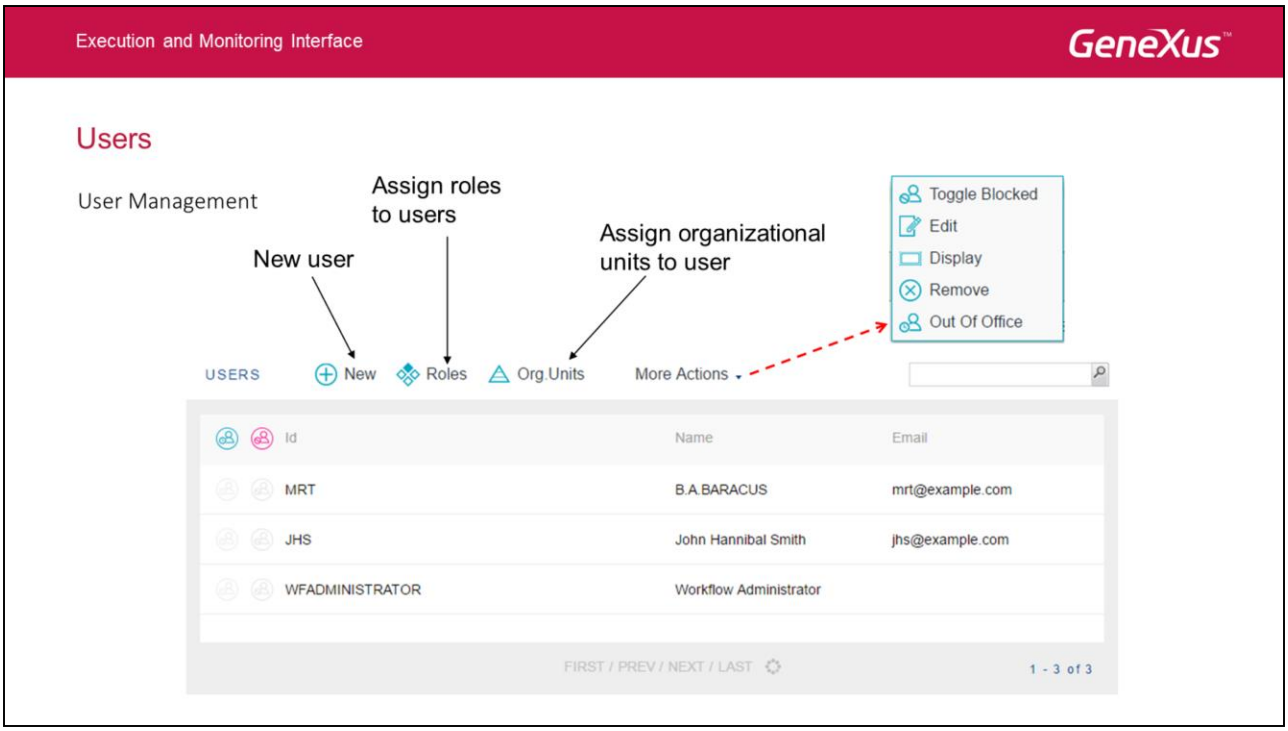
Definido o modelo na KB, pode-se editá-lo no Cliente de GXflow, tendo a possibilidade de configurar o conteúdo da mensagem, assunto e destinatários.

GeneXus

Management Console

- Management of Users, Roles and Organizational Units

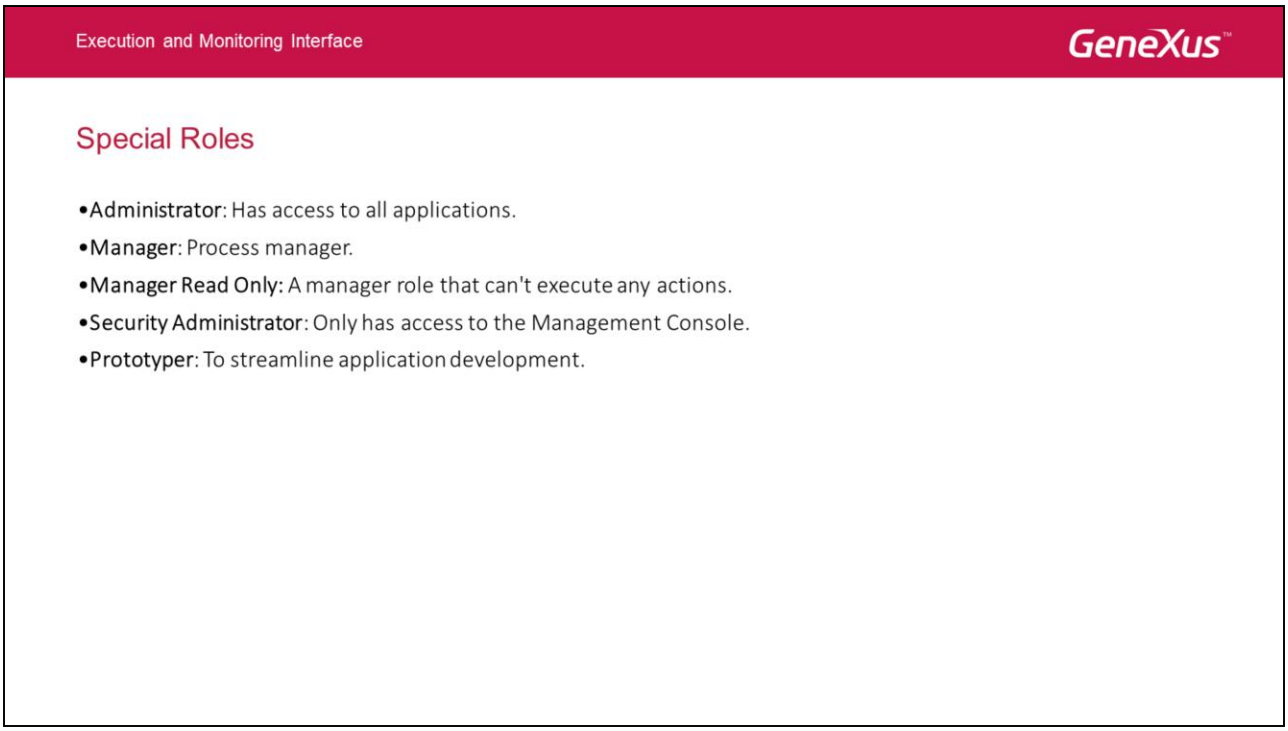
Mediante essa aplicação, manejam-se os usuários, papéis, unidades organizacionais e sua definição. Isso nos permite restringir as tarefas que os usuários veem e podem processar.



A aplicação de administração de usuários permite cadastrar usuários, atribuir-lhes papéis, bloqueá-los ou desbloqueá-los, editar e mostrar sua informação, eliminá-los, acrescentar-lhes unidades organizacionais e configurar que estarão fora do escritório.

Os usuários são nomeados automaticamente e, se não estiverem nomeados, não será possível atribuir-lhes um papel.

No que diz respeito ao bloqueio, o motor bloqueará os usuários se tentarem entrar como uma senha inválida mais de três vezes (pode-se modificar configurando a política de segurança).



Além dos papéis criados pelo usuário para administrar a segurança das tarefas, GXflow conta com papéis especiais que podem ser atribuídos a qualquer usuário:

Administrator: Quando um usuário com este papel acessa o cliente GXflow, visualiza e pode realizar todas as ações em todas as aplicações, seja no plano de fundo, no administrador de processos, no visor de eventos, no console de administração ou nas preferências do servidor.

Manager: Quando um usuário com este papel acessa o cliente GXflow, visualiza apenas a aplicação administração de processos e pode realizar todas as ações relacionadas com a instâncias de processos.

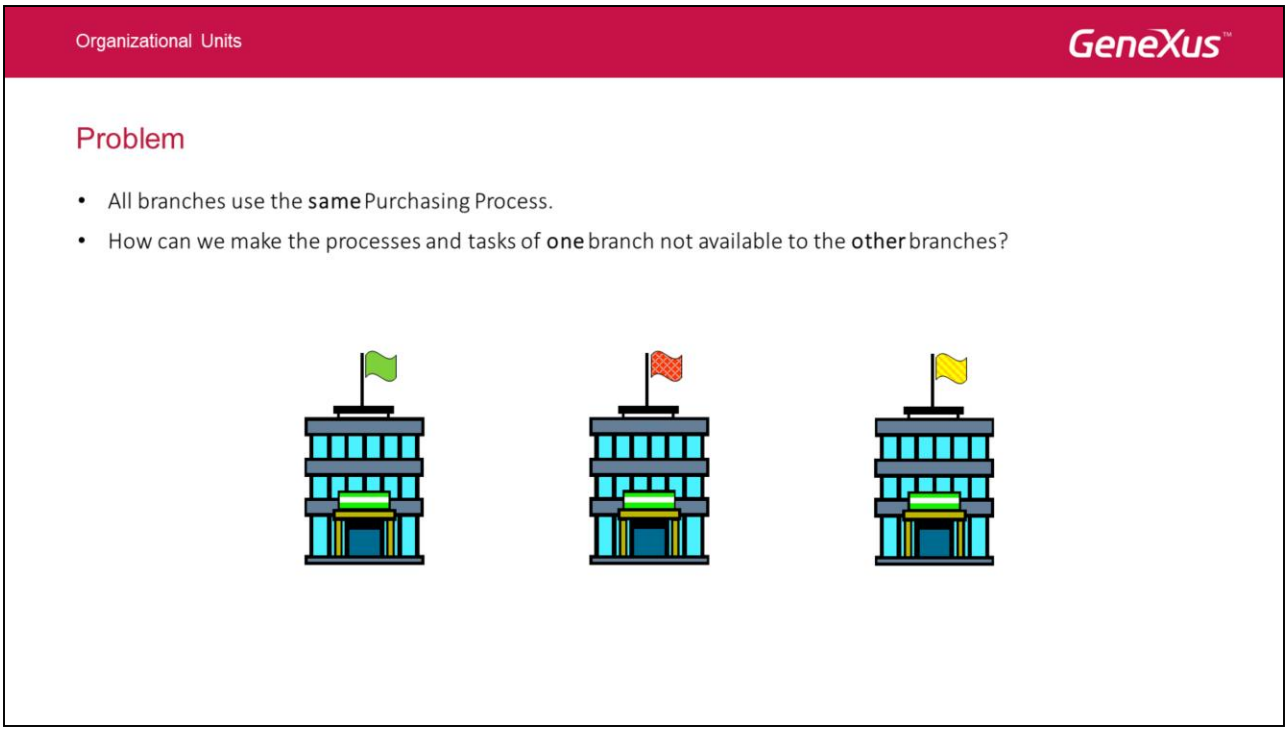
Manager Read Only: Como o manager, o usuário pode ver apenas a aplicação de administração de processos, não pode, contudo, realizar ações.

Security Administrator: Quando um usuário com este papel acessa o cliente GXflow, pode ver apenas o console de administração.

Prototyper: Este papel permite executar as tarefas sem importar o papel que lhe tenha sido atribuído. Isso permite facilitar o desenvolvimento.

Module 4

Organizational Units



Muitas vezes, encontramos dentro das empresas certas restrições que permitem que nem todas as pessoas que trabalham na empresa possam ver as instâncias dos mesmos processos.

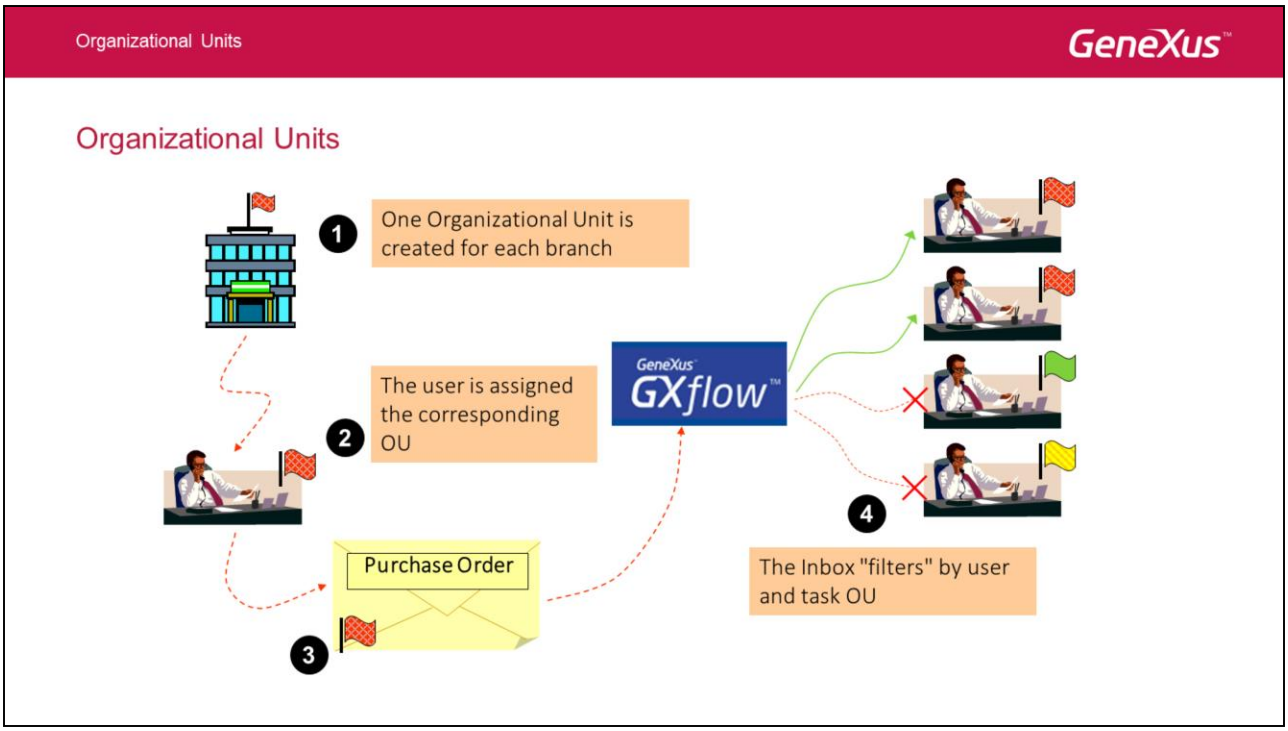
Imagine uma empresa com várias Filiais, por exemplo, e todas essas Filiais utilizam o mesmo processo de Compras.

Como fazemos para que os processos e tarefas de **UMA** filial **NÃO** estejam disponíveis para as demais filiais?

Para resolver essa questão, contamos com o conceito “Unidades Organizacionais”.

As unidade Organizacionais (UO) permitem definir grupos dentro de uma organização, como filiais, departamentos, equipes. Definidas as unidades, cria-se uma instância de cada uma delas e, dentro de cada instância, seus membros são associados a ela.

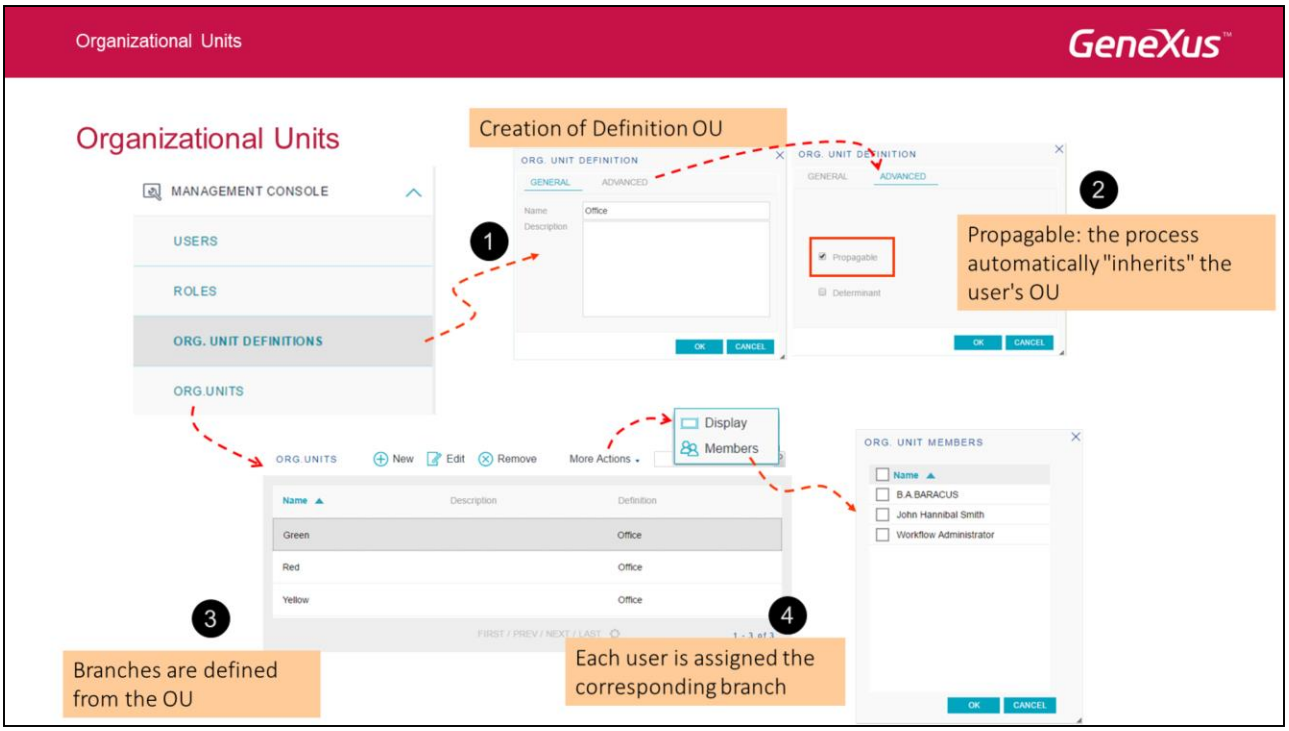
Em nosso caso, a empresa conta com três filiais, a filial *Green*, a filial *Red* e a filial *Yellow*. Dentro da *Green*, há duas pessoas para a área de compras; em *Red*, cinco pessoas, e a filial *Yellow*, por sua vez, conta com quatro pessoas.



Quais são os passos que deveríamos seguir para poder implementar este caso?

Em primeiro lugar, estabelecemos que a definição da unidade organizacional será *Offlice*.

1. Depois, define-se uma UO para cada filial da empresa, neste caso *Red*, *Green* e *Yellow*.
2. Definidas as UO, a cada usuário é atribuída a UO correspondente.
3. A partir desse momento, o processo herdará a UO do usuário que o inicia.
4. Assim, na caixa de entrada de cada usuário não se filtrará apenas pela tarefa, mas também pela UO.



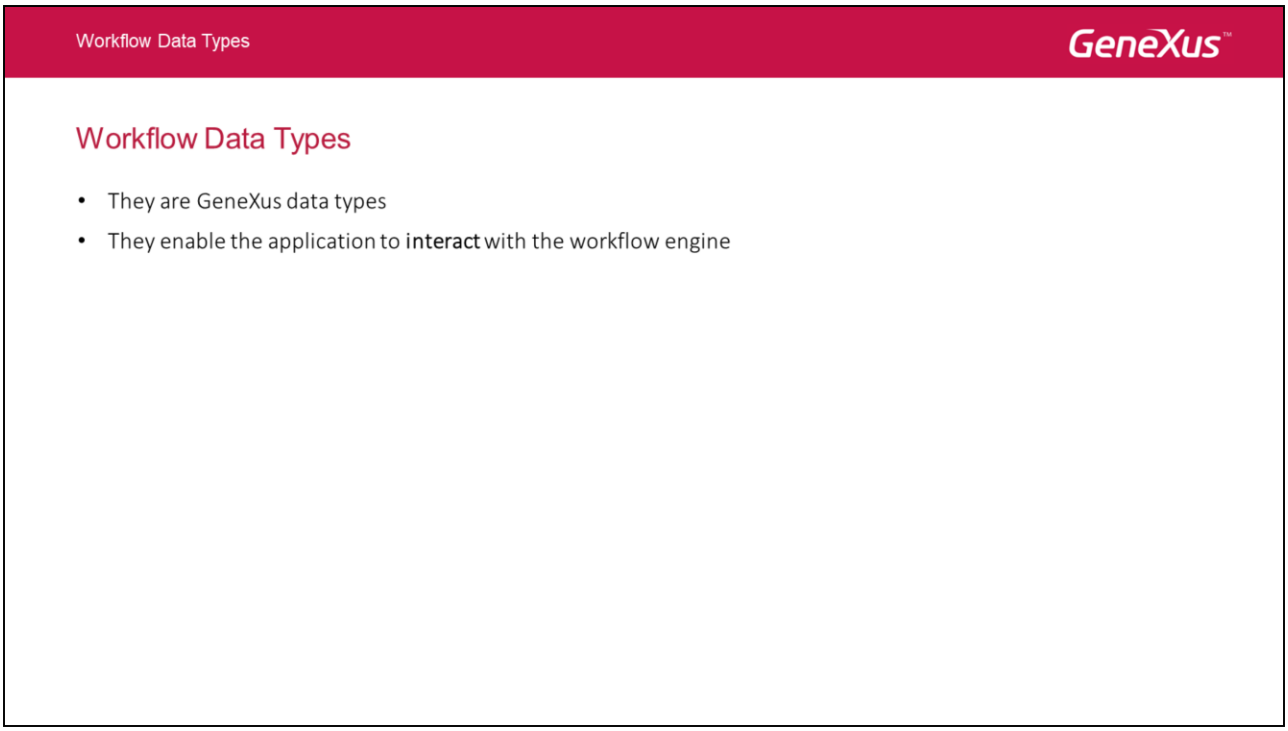
Para implementar isso, devemos ir ao cliente de GXflow e entrar com o Usuário Administrador ou com qualquer usuário que tenha permissão para isso. Em seguida, é preciso ir ao **Management Console - Organizational Model** e fazer o seguinte:

1. Em **Organizational Unit Definitions**, criar um nova chamada *Office*.
2. Indica-se sim no **Propagable** , para que o processo herde de forma automática a UO do usuário.
3. Da opção **Organizational Units ,** são definidas as diferente filiais com as quais a Empresa conta, neste caso, *Red*, *Green* e *Yellow*.
4. A cada filial são atribuídos os usuários correspondentes.

A partir desse momento, fica definido que cada filial verá apenas os processos de compra que pertencem a ela.

Module 5

Workflow Data Types

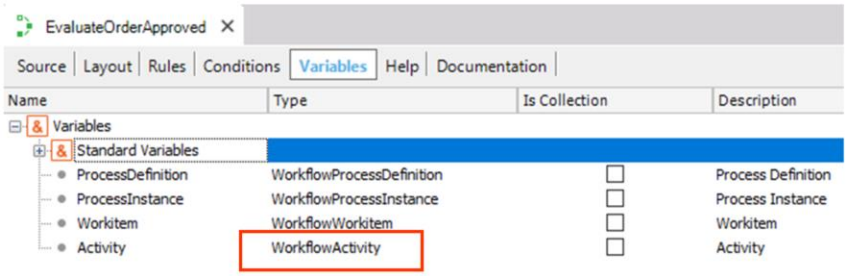


Os tipo de dados de Workflow são um conjunto de tipos de dados GeneXus que permite interagir com o motor de GXflow.

GeneXus

Workflow Data Types

- Workflow Prefix
- Defined as **variables** in GeneXus objects.



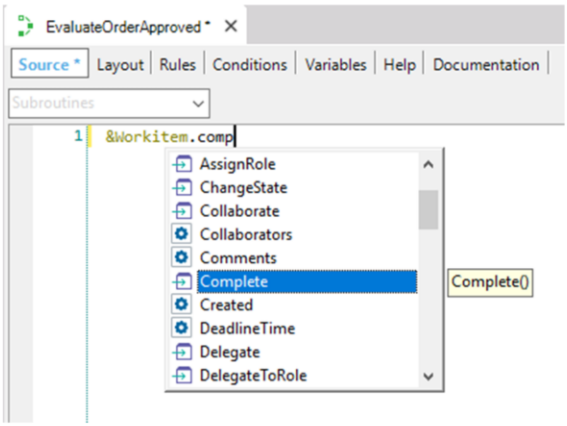
Todos os tipo de dados Workflow tem o mesmo prefixo para diferenciá-los do resto de tipos de dados que GeneXus suporta.

Para poder usar um determinado tipo de dado Workflow, deve-se definir primeiramente uma variável baseada em tal tipo de dado (procedimento igual ao de qualquer tipo de dado GeneXus).

GeneXus

Workflow Data Types

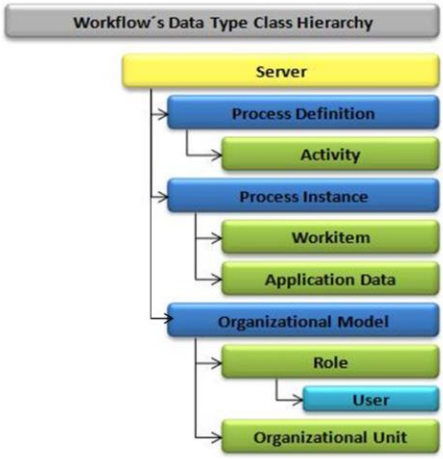
- Intellitips that facilitate use



Como acontece com os outros tipos de dados de GeneXus, conta-se com a informação contextual (intellitips) que ajuda na utilização dos dados.

GeneXus

Hierarchy of Workflow Data Types



Os tipos de dados Workflow organizam-se em uma hierarquia de tipos por meio da qual nos é permitido navegar partindo da ponta da entrada principal, que é o tipo de dados Server.

A partir do tipo de dados Server, e utilizando métodos e propriedades, podemos acessar outros tipos e assim sucessivamente.

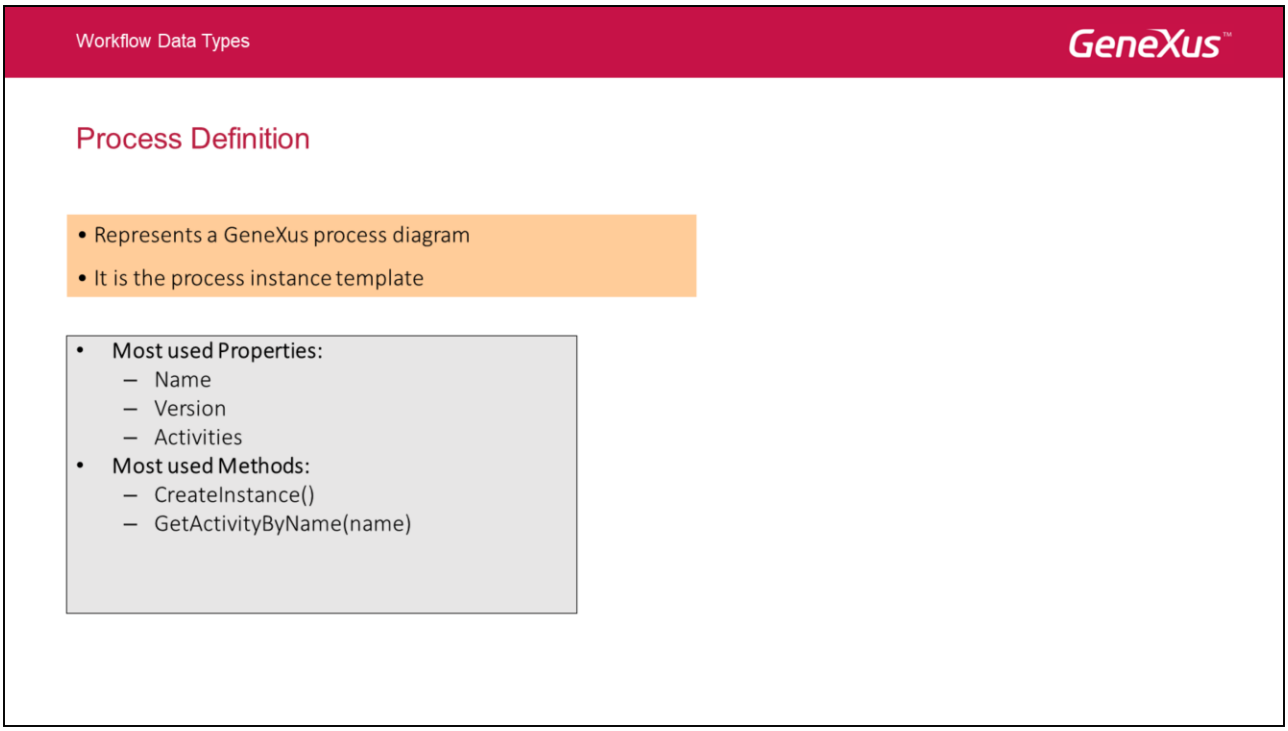
GeneXus

Basic Types

- **ProcessDefinition** – Data type: WorkflowProcessDefinition
- **ProcessInstance** – Data type: WorkflowProcessInstance
- **Workitem** – Data type: WorkflowWorkitem
- **Context** – Data type: WorkflowContext
- **Application Data** – Data type: WorkflowApplicationData
- **Error** – Data type: WorkflowError

Os tipos de dados básicos de workflow, isso é, os mais utilizados, estão listados no slide.

Por uma questão de comodidade, omitiremos o prefixo “Workflow” comum a todos os tipos que vamos ver.



O tipo Process Defintion corresponde a um diagrama de processos publicado.

A rigor, corresponde a uma determinada versão de um diagrama, já que a cada vez que se publica um digrama, cria-se um novo Process Defintion com o número da versão incrementado em um.

Algumas das propriedades mais importantes são as seguintes:

Name

Permite obter o nome do processo.

Version

Permite acessar o número da versão.

Activities

Permite obter uma coleção com as atividades que compõem o processo.

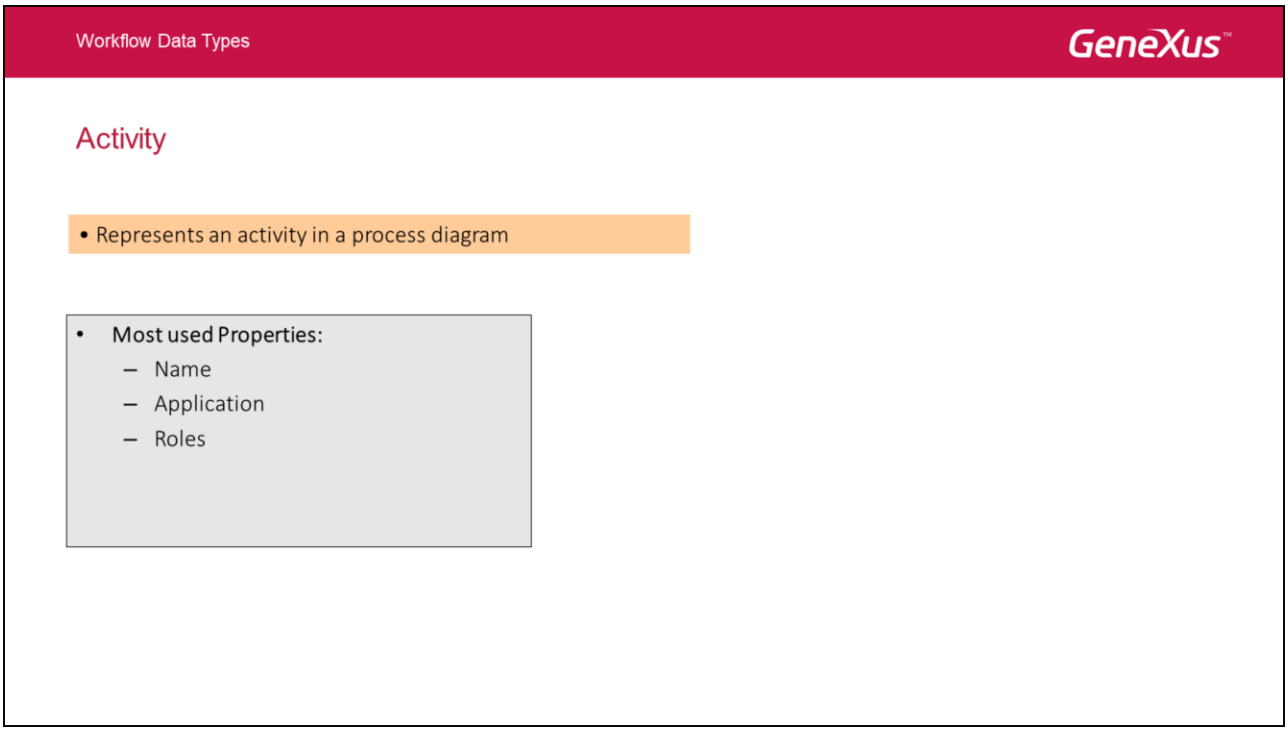
Dentro dos métodos mais importantes temos:

CreateInstance

Permite criar uma nova instância de processo baseada nesta definição.

GetActivityByName

Permite recuperar uma atividade a partir de seu nome.



O tipo Activity representa uma atividade de um diagrama de processo, ou seja, o que um diagrama representa com um retângulo de vértices arredondados e significa certo trabalho a ser feito em nosso processo.

Algumas das possibilidades disponíveis são:

Name

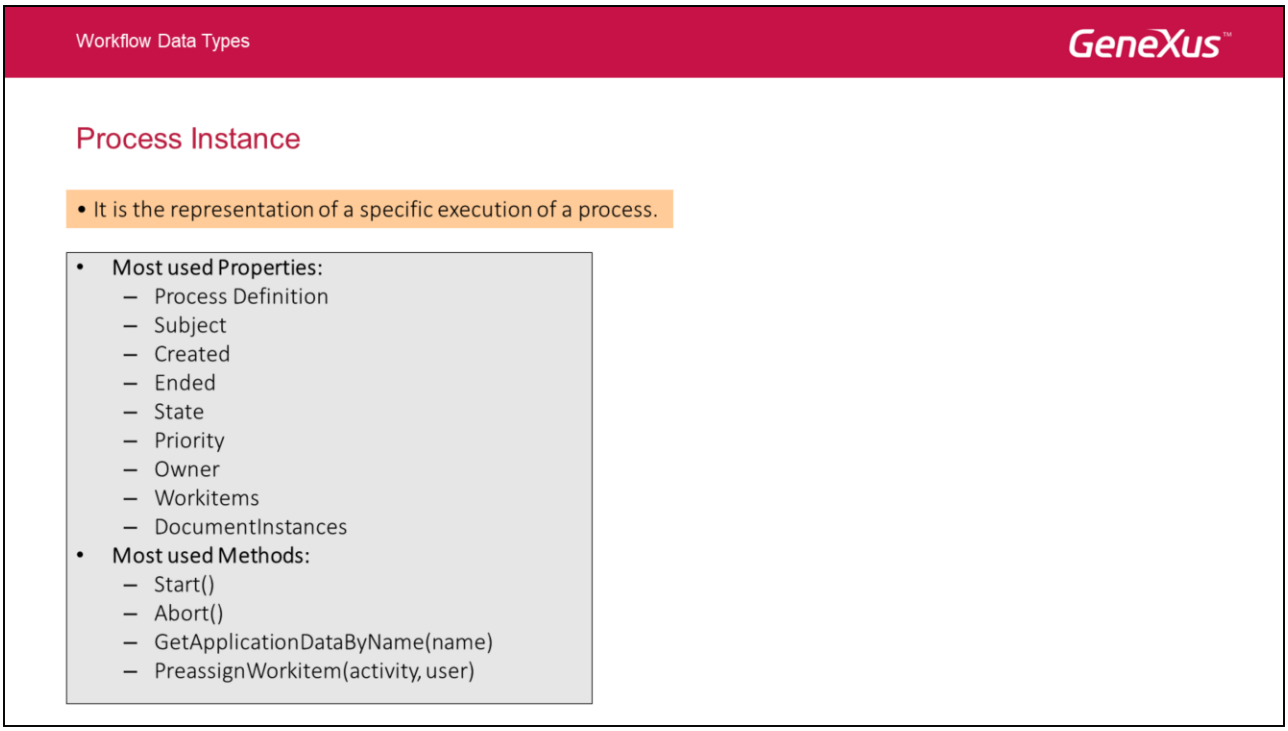
Mostra o nome da atividade.

Application

Mostra o nome do aplicativo associado à atividade.

Roles

Mostra uma coleção com os papéis que têm permissão para executar tal atividade.



O tipo Process Instance é utilizado para representar as instâncias de uma definição de processo.

Cada instância de processo representa uma execução de uma definição de processo, ou seja, se tivermos, por exemplo, um diagrama de processo “Processar Compra”, as diferentes instâncias poderiam ser “Processar Compra No. 1”, “Processar Compra No. 2” etc.

As propriedades mais importantes são:

Process Definition

Mostra a definição de processo na qual está baseada esta instância.

Subject

Permite acessar o assunto, uma propriedade que se utiliza para descrever a instância do processo.

Esta propriedade é importante porque essa descrição deve ser gerada pela aplicação.

Created

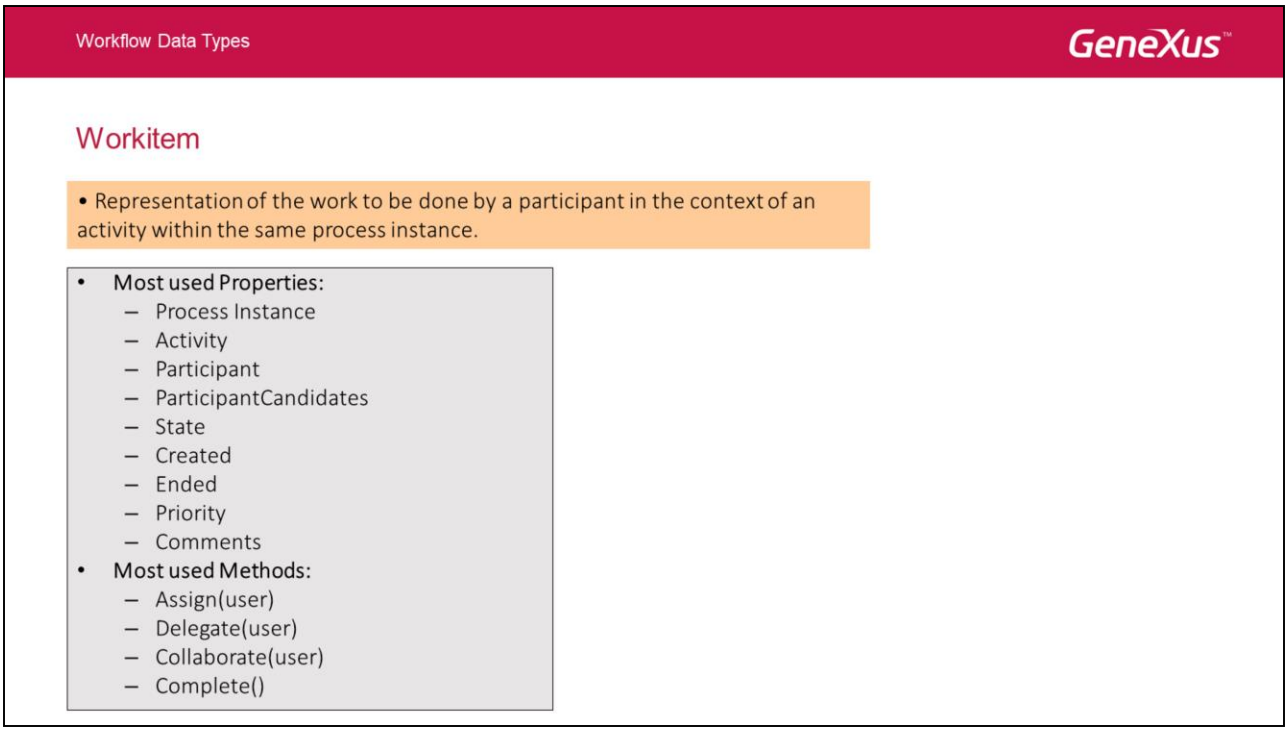
Mostra a data de criação.

Ended

Mostra a data de finalização.

State

Permite conhecer o estado da instância de processo (ver Enumerado WorkflowProcessInstanceState)



Um workitem representa o trabalho a ser feito por um participante no contexto de uma atividade dentro de uma instância do processo.

Dentre as propriedades mais importantes, temos:

ProcessInstance

É a instância de processo a que pertence.

Activity

Mostra a atividade que o workitem representa.

Participant

Usuário a quem o workitem foi atribuído (pode não existir).

ParticipantCandidates

Mostra uma coleção com os possíveis candidatos para executá-lo (os candidatos são determinados pelo motor com base em papéis e unidade organizacionais atribuídos os usuários)

State

Mostra o estado do workitem (ver enumerado WorkflowWorkitemState)

Created

Mostra a data de criação do workitem.

GeneXus

Workflow Context

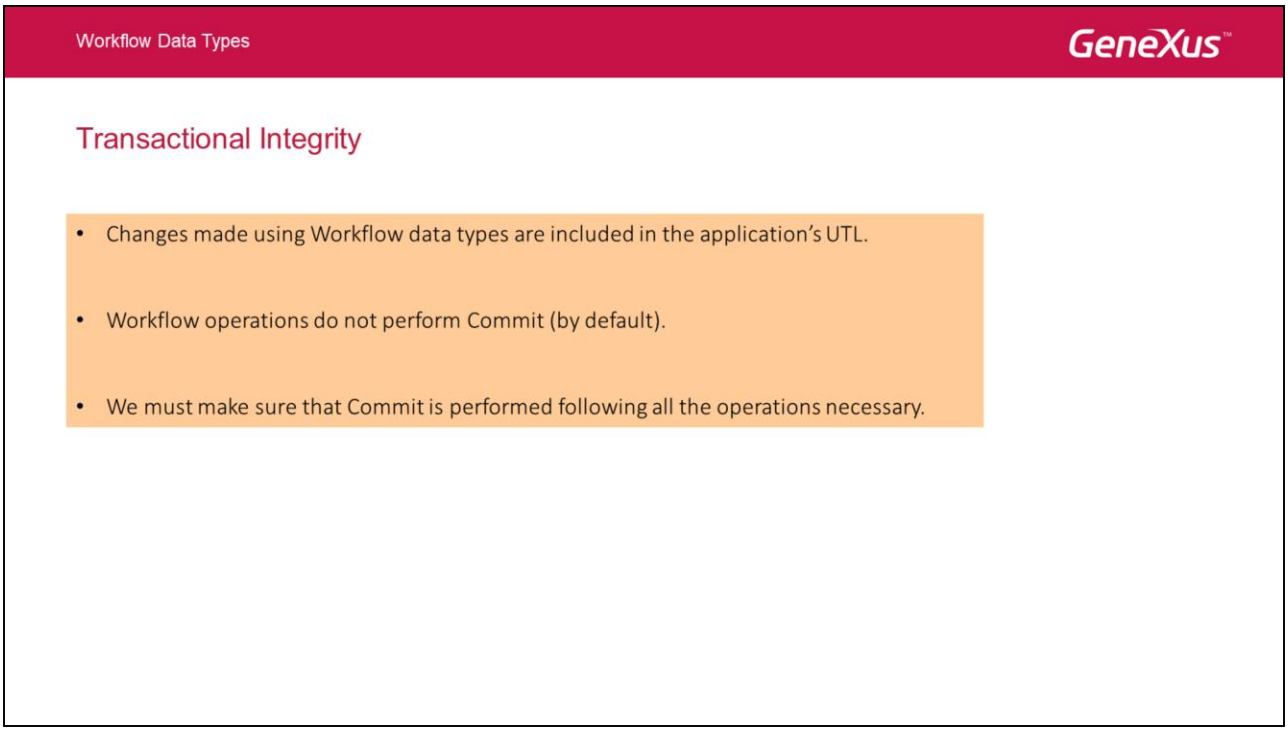
- Allows access to context information in executing an application associated with an activity
- It is automatically instanced
- Properties are:
  - ProcessDefinition
  - ProcessInstance
  - Workitem

Com o nome indica, o tipo de dados Context permite obter a informação de contexto de uma aplicação associada a uma atividade.

A informação de contexto é formada pelo workitem, pela instância do processo e pela definição do processo onde a aplicação é executada.

Eles podem ser acessados mediante as propriedades correspondentes (Workitem, ProcessInstance, ProcessDefinition).

Vale esclarecer que o contexto se instancia automaticamente se a aplicação associada à atividade (seja do tipo User ou Script) for um objeto GeneXus que faça parte da mesma KB onde se encontra o digrama de processos.

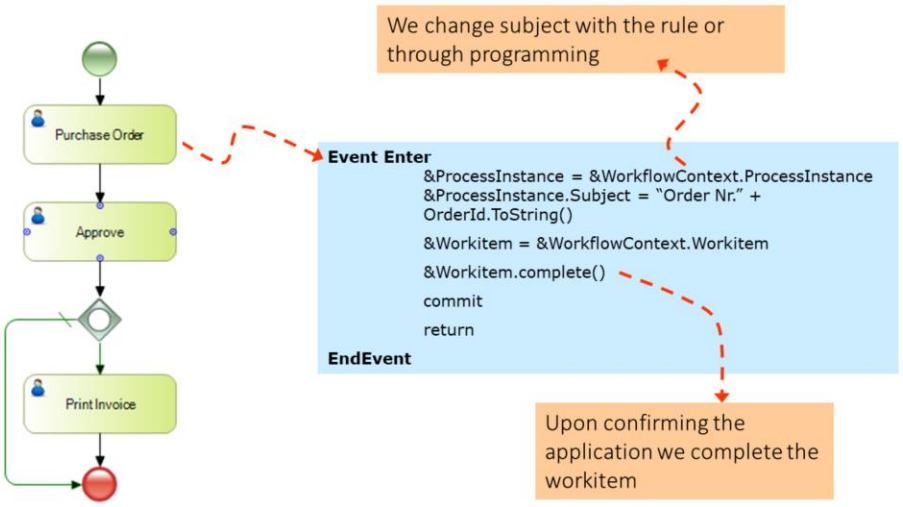


As alterações realizadas mediante o uso dos tipos de dados workflow estão compreendidas dentro da Unidade de Trabalho Lógica da aplicação.

Isso significa que tais alterações serão “commiteadas” quando a aplicação o fizer (que GeneXus o faça ou que o usuário o indique explicitamente mediante o comando commit).

As operações de workflow não fazem commit, portanto, é preciso garantir que sempre ao utilizar os tipos de dados workflow se defina bem a UTL e se faça o commit no fim.

Example



Veremos agora alguns casos concretos de uso dos tipos de dados Workflow.

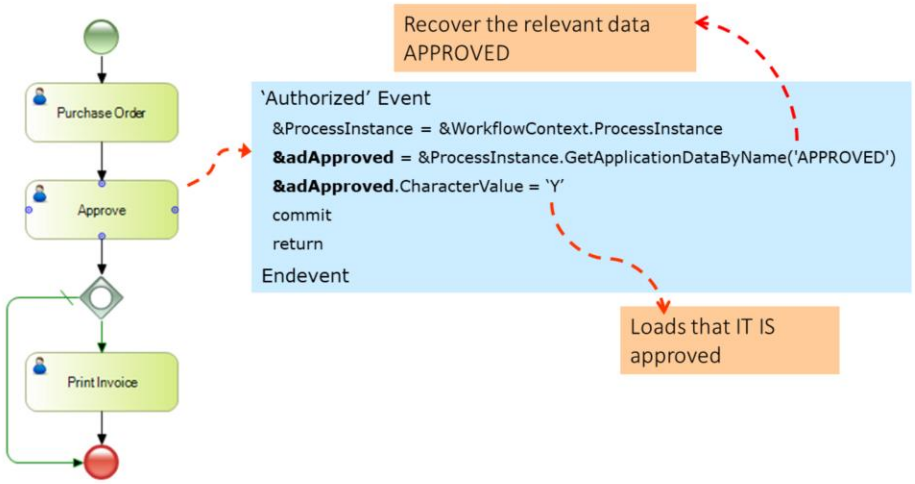
Suponhamos que temos um diagrama que modela um processo de compras.

O processo começa com uma atividade de usuário chamada “Purchse Order”, na qual se gera a informação da ordem de compra. Essa atividade pode estar associada a uma transação ou a um webpanel, o que é indiferente para o exemplo que estamos vendo.

Imaginemos que queremos que, assim que se confirme a ordem, se modifique o assunto da instância de processo com o número da ordem. Para isso, recuperamos primeiramente a instância do processo utilizando o contexto e depois utilizamos a propriedade Subejct para modificar o assunto.

Da mesma forma, se quiséssemos que, uma vez que se confirme a ordem, desse por finalizado o workitem e recuperasse o tipo utilizando o contexto, invocaríamos o método Complete.

Example



Continuando com o exemplo anterior:

Imaginemos que depois da atividade “Purchase Order”, temos uma atividade de usuário chamada “Approve”, na qual um supervisor terá a possibilidade de autorizar ou rejeitar essa ordem de compra. Além disso, definimos uma variável chamada “APPROVED” para deixar registrado o resultado dessa decisão.

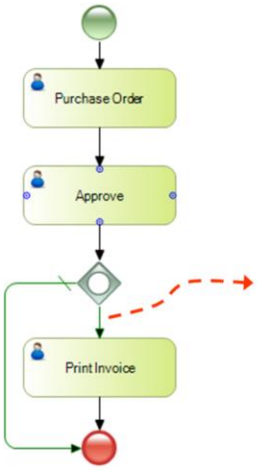
Na aplicação associada a essa atividade, para poder modificar essa variável, devemos seguir os seguintes passos:

Primeiramente, devemos recuperar a instância de processo utilizando o contexto. Depois, sobre a instância, utilizamos o método GetApplicationDataByName passando como parâmetro o nome da variável que queremos recuperar (“APPROVED”). Recuperada a variável, utilizamos a propriedade CharacterValue para atribuir o resultado da ação (se autorizou ou não).

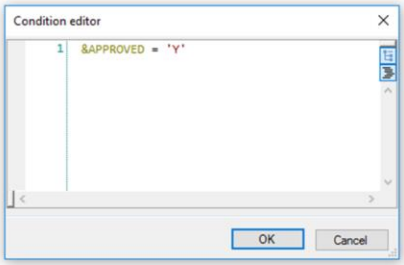
Nesse caso, a propriedade para modificar o valor é a CharacterValue, porque definimos a variável “APPROVED” do mesmo tipo. Se tivéssemos definido de tipo de numérico, então para modificar ou recuperar o valor, deveríamos utilizar a propriedade NumericValue.

GeneXus

Example

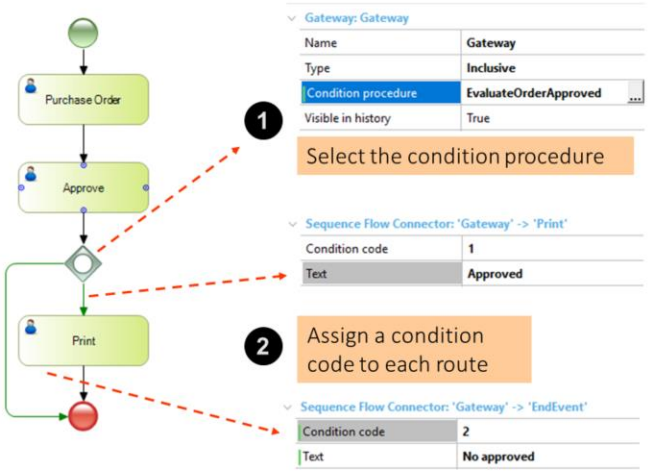


Recover the relevant data APPROVED using a gateway with a condition rule



Continuando com o exemplo anterior…

Using a condition assessment procedure



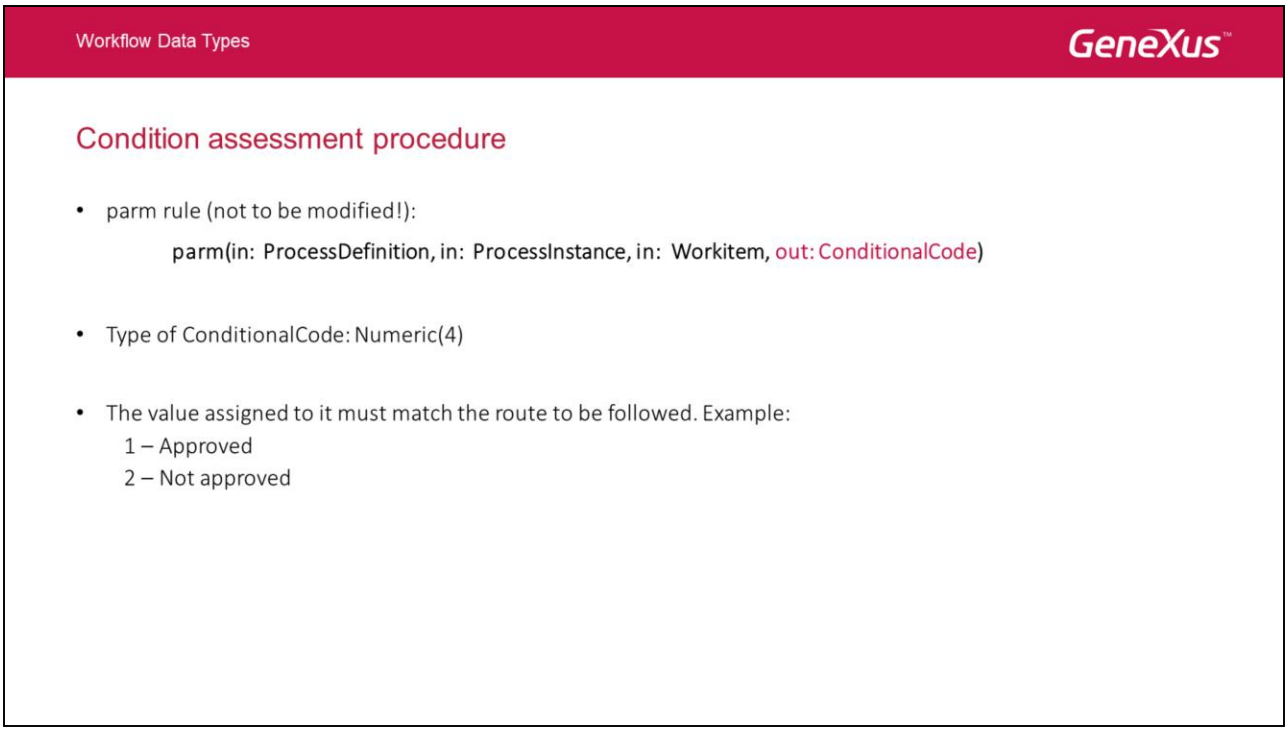
Seguindo com o exemplo. Suponhamos que depois da atividade de aprovação, existe uma comporta com a qual se quer modelar se se aprova a ordem, se passa a uma atividade na qual se deve imprimir a fatura ou, em caso contrário, dê o processo como finalizado.

Poderíamos configurar a comporta utilizando diretamente um regra de condição que compare o valor da variável “APPROVED” para saber que caminho tomar (como vimos anteriormente na modelagem de processos). Neste caso, porém, vamos optar por associar um procedimento de condição.

Utilizar um procedimento para avaliar uma condição nos dá maior potência, já que não estamos limitados a utilizar variável e/ou atributos de transações associadas ao diagrama, em vez disso, temos a possibilidade de navegar por qualquer atributo da KB e modelar uma condição muito mais complexa que o editor de regras permite.

O procedimento para utilizar um procedimento de avaliação de condição é o seguinte:

1. Nas propriedades da comporta, selecionamos o procedimento de condição;
2. Nas propriedades de cada uma das arestas que partem da comporta, atribuímos um código de condição (é um número) e opcionalmente um texto que o descreva.



Vejamos agora como se deve definir o procedimento de condição.

Para começar, o procedimento deve ter uma regra “parm” específica.

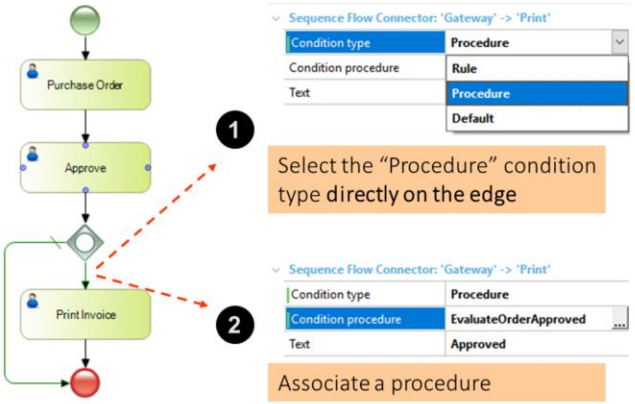
O procedimento deve receber os seguintes parâmetros:

1. Uma variável do tipo ProcessDefinition (entrada)
2. Uma variável do tipo ProcessInstance (de entrada)
3. Uma variável do tipo Workitem (de entrada também)
4. Finalmente e, a mais importante, uma variável de tipo numérico de 4 de saída, que é onde mostra o resultado da condição.

As primeiras variáveis são as de contexto e podem ser utilizadas pelo procedimento para recuperar a informação relevante do processo.

A variável de saída é o código que vamos mostrar e é o que o motor de workflow utilizará para comparar com os valores associados aos códigos das arestas que partem da comporta para decidir por qual a atividade continuar.

Using a condition assessment procedure Alternative Option



Uma alternativa para os procedimentos de condição é atribuir o procedimento diretamente na aresta e não na comporta.

Os passos são os seguintes:

1. Nas propriedades da aresta, no tipo de condição, selecionamos “Procedure”;
2. Na propriedade “Conditional procedure”, selecionamos o procedimento.

O procedimento de condição deve respeitar a mesma regra parm do caso anterior, com a ressalva de que o código de condição a retornar está limitado aos valores um e zero. Se se mostra um, quer dizer que se deve continuar por essa aresta. Se se mostra zero, significa que não se deve seguir por esse caminho.

GeneXus

Advanced Types

- **Server** – Data type: WorkflowServer
- **OrganizationalModel** – Data type: WorkflowOrganizationalModel

Vejamos agora os tipos de dados workflow mais avançados. Em particular, veremos os tipos de dados Server e OrganizationalModel.

GeneXus

Server

- Connection to the WF engine
- Access point in the types hierarchy

- Most used Methods:
  - Connect(user, password)
  - Disconnect()

O tipo de dados Server é o ponto de entrada na hierarquia dos tipos de dados Workflow, ou seja, aquele que nos permite acessar qualquer outro tipo de dado workflow.

O primeiro passo para poder utilizá-lo é invocar o método connect, que é o que nos vai permitir estabelecer uma conexão com o motor de Workflow de GXflow.

Estabelecida a conexão, podemos acessar o resto de seus métodos e demais propriedades.

Finalizada a operação, podemos, opcionalmente, invocar o método Disconnect para liberar a conexão.

Geralmente, o tipo de dado Server dispõe de dois tipos de operações ou métodos.

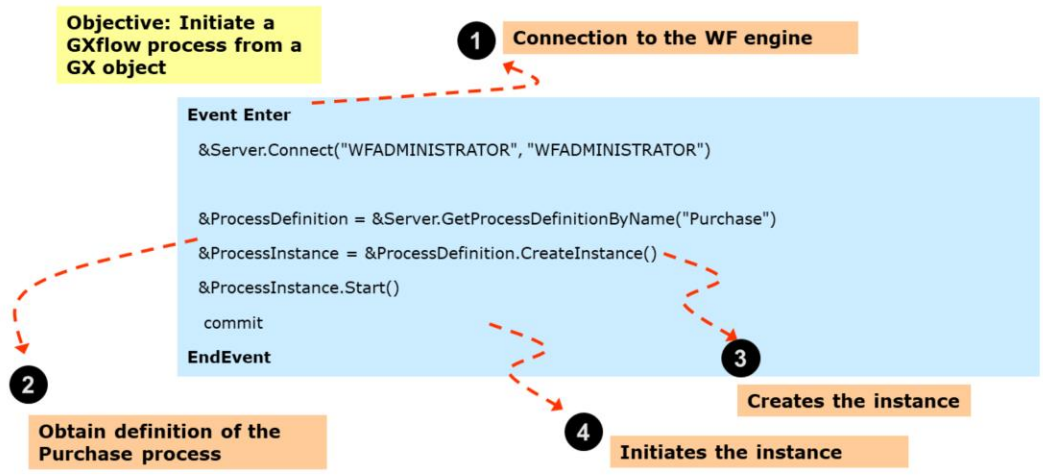
Algumas delas tem o prefixo “Get” e permitem-nos obter um tipo em particular a partir de um identificador ou de um nome.

O outro tipo de operações são as que tem o prefixo “List” e permitem-nos obter uma coleção de variáveis de determinado tipo que cumpre com determinados critérios de busca.

Os critérios de busca especificam-se com um parâmetro de tipo Filter.

Também existe a possibilidade de mostrar a coleção em uma determinada ordem. Para isso, dispõe-se de um método que replica o original com o literal OrderBy. Nesse caso, o método, além do filtro, recebe uma variável (ver enumerado WorkflowOrder) com a ordem em que se quer obter o resultado.

Example: Creating a process 'outside' the Inbox



Vejamos um exemplo. O objetivo é criar e iniciar a execução de um processo de workflow a partir de um objeto GeneXus.

Para isso, devemos seguir a seguinte sequência de passos:

1. Estabelecer uma conexão com o motor de wf com o método Connect;
2. Obter, a partir do nome, a definição de processo que queremos instanciar utilizando o método GetProcessDefinitionByName;
3. Criar uma nova instância dessa definição utilizando o método CreateInstance;
4. Iniciar a execução da nova instância criada (e que é retornada no método anterior se não há erro) utilizando o método Start.

GeneXus

Organizational Model

- Organizational Model with users, roles, etc.

- Most used Methods:
  - ListUsers
  - ListRoles
  - ListOrganizationalUnitDefinitions
  - ListOrganizationalUnits
  - GetUserByName
  - AddUser
  - AddRole
  - AddOrganizationalUnitDefinition
  - AddOrganizationalUnit

O tipo de dados OrganizationalModel permite acessar a informação da estrutura organizacional da empresa, isso é, as unidades organizacionais, os papéis funcionais, os usuários e suas relações.

Semelhante ao tipo de dados Server, existem métodos com prefixo Get que servem para recuperar um tipo por seu identificador ou nome, e métodos com prefixo List que nos permitem obter uma coleção de Usuários, Papéis, UnidadesOrganizacionais etc.

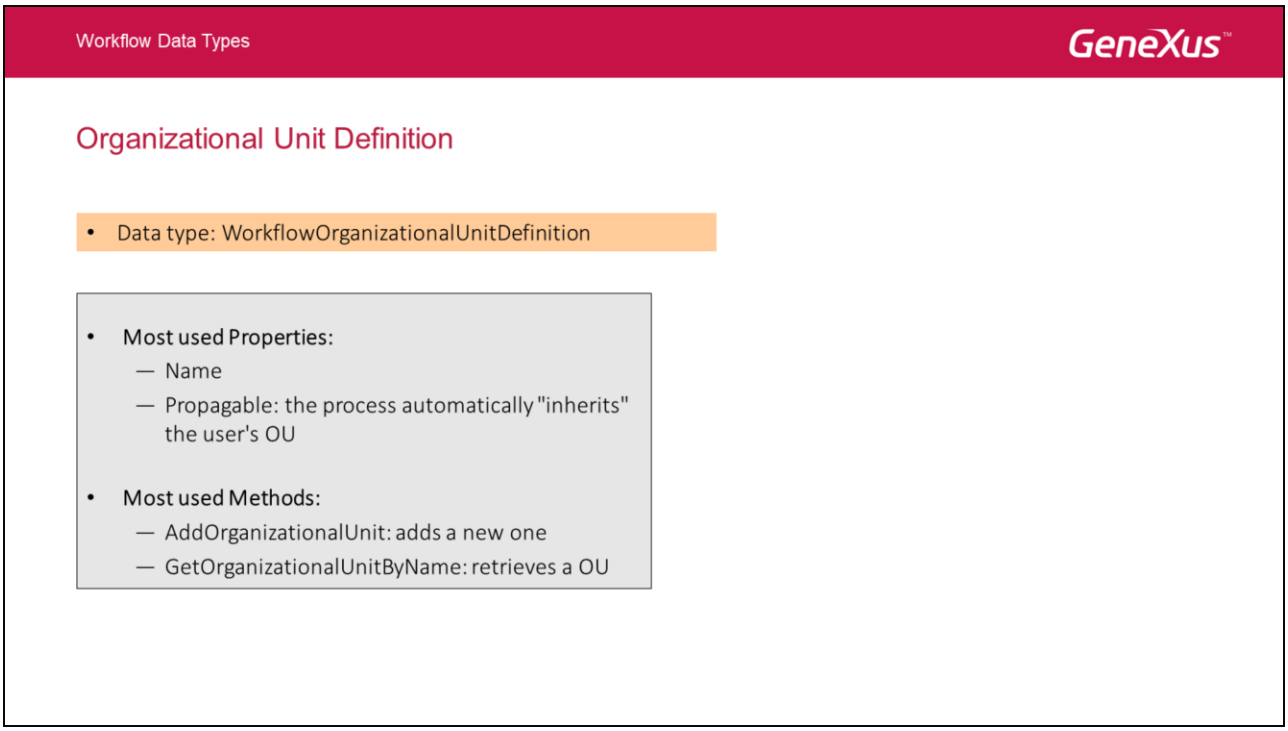
Também existem métodos com prefixos Add e Remove que nos permitem acrescentar ou eliminar entidades.

GeneXus

Organizational Unit

- Data type: WorkflowOrganizationalUnit

- Most used Properties:
  - Name
  - OrganizationalUnitDefinition: definition
- Most used Methods:
  - AddUser/RemoveUser: user management



Seguindo com exemplo proposto das Filias, criemos as Unidades Organizacionais.

1 – É preciso conectar-se ao servidor com um usuário e senha que tenham permissão, neste caso, usaremos o administrador;

```
&WorkflowServer.Connect('WFADMINISTRATOR','WFADMINISTRATOR')
```

2 – Obtém-se o modelo organizacional, para isso utiliza-se o método **GetOrganizationalModel**.

```
&WorkflowOrganizationalModel = &WorkflowServer.GetOrganizationalModel()
```

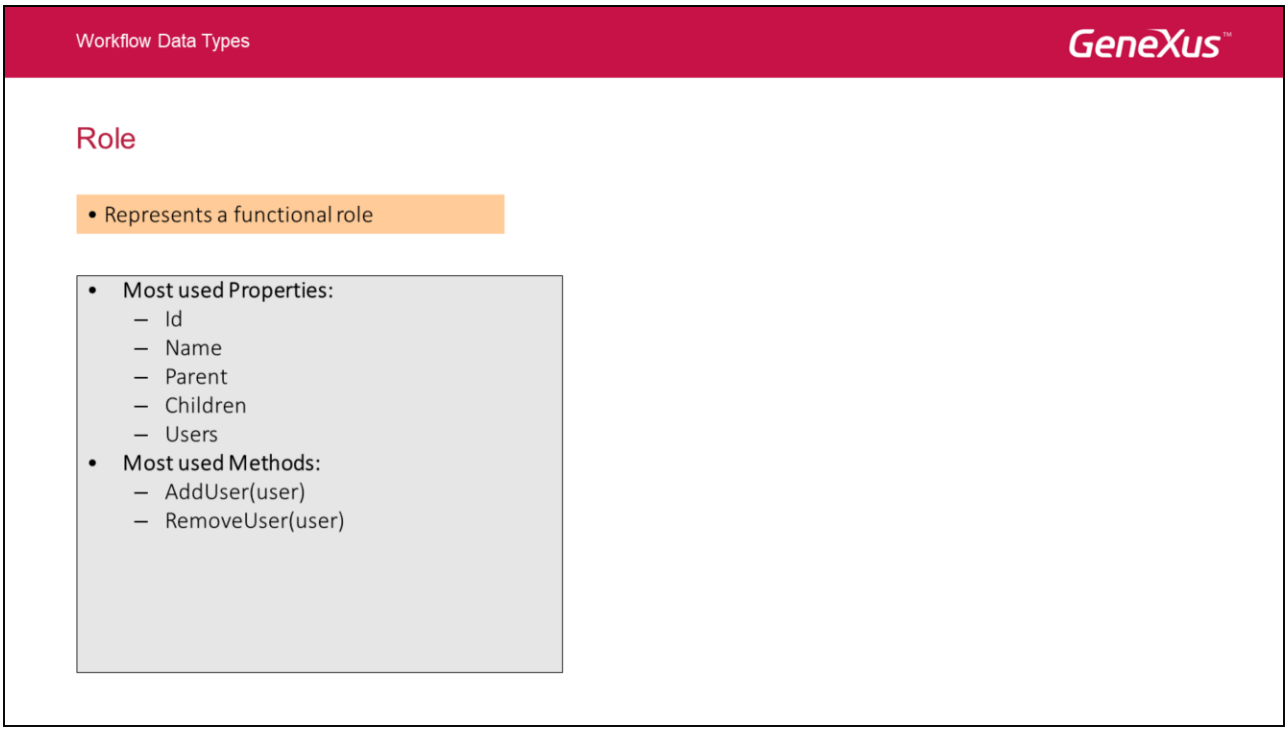
3 - Acrescenta-se como Definição de Unidade Organizacional *Office*, para isso utiliza-se o método **AddOrganizationalUnitDefinition**.

```
&WorkflowOrganizationalModel.AddOrganizationalUnitDefinition('Office', 'Offices', True)
```

4 - Acrescentam-se as unidades organizacionais *Red*, *Green* e *Yellow*. Para isso, utiliza-se o método **AddOrganizationalUnit**.

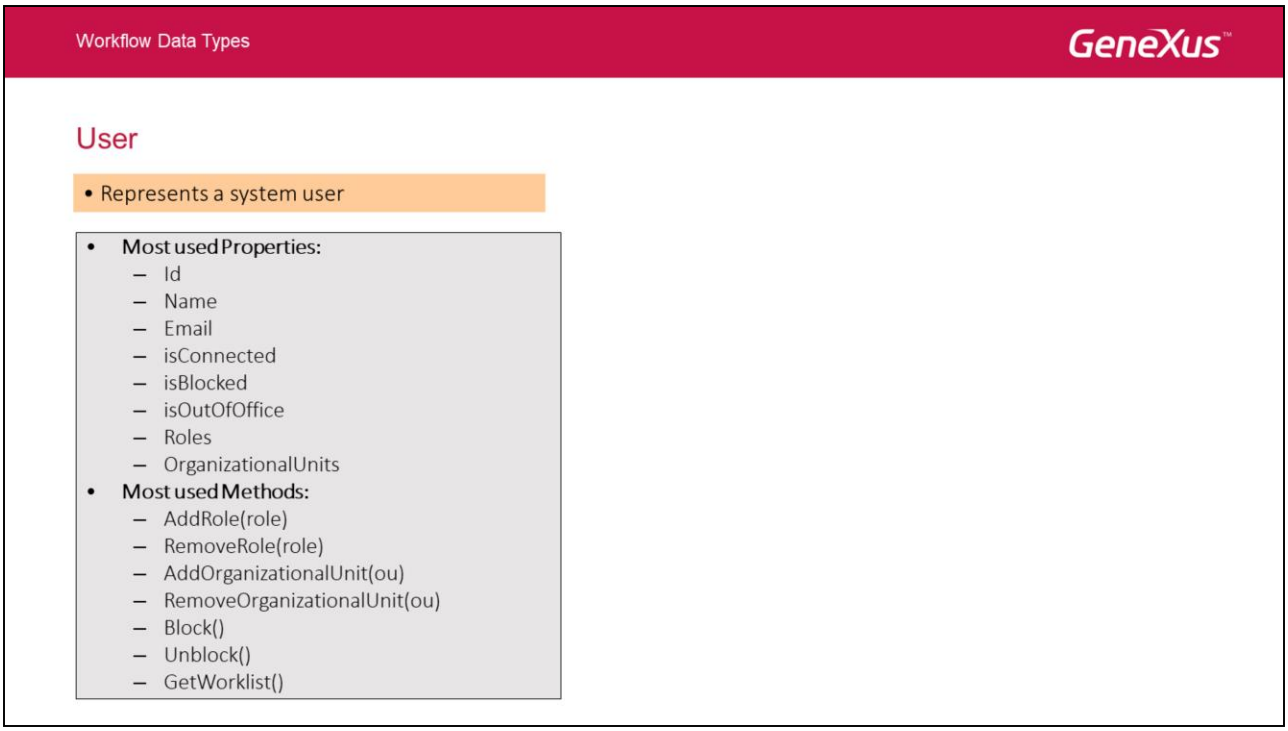
```
&WorkflowOrganizationalModel.AddOrganizationalUnit('Office','Red', 'Office Red')
&WorkflowOrganizationalModel.AddOrganizationalUnit('Office','Green', 'Office Green')
&WorkflowOrganizationalModel.AddOrganizationalUnit('Office','Yellow', 'Office Yellow')
```

Nota: em todos esses casos, foram utilizadas variáveis que estão baseadas nos tipos de dados que possuem o mesmo nome que elas.



O tipo de dado Role representa um papel funcional. Os papéis funcionais são os que associamos às atividades para definir permissões sobre elas.

Mediante o tipo de dados, podemos acessar as propriedades de um papel (identificador, nome, quem é o papel pai, os papéis filhos, os usuários associados etc) e também dispomos de métodos para acrescentar ou tirar usuários.



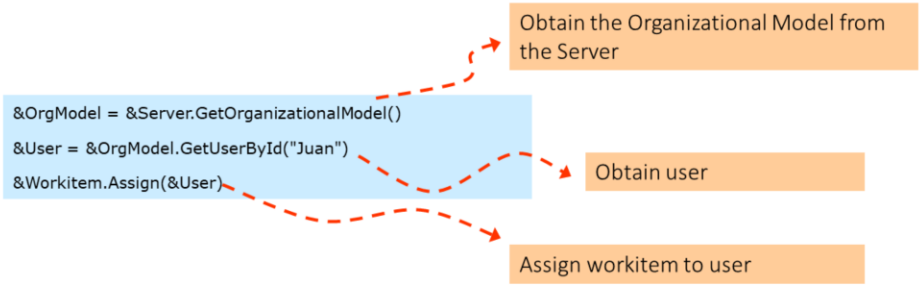
O tipo de dado User permite-nos acessar a informação dos usuários do sistema de workflow.

Dispomos de propriedades para saber: o identificador, o nome, o endereço de e-mail, se está conectado, se está bloqueado, se está de licença etc.

Também dispomos de métodos para atribuir ou deixar de atribuir papéis ou unidades organizacionais, bloqueá-lo etc.

Temos também um método GetWorklist que nos permite obter uma coleção com todos os workitems ativos e atribuídos ou aos quais usuário é candidato (ou seja, todos os workitems que o usuário poder ver em sua caixa de entrada)

Example: Assigning a workitem to a user



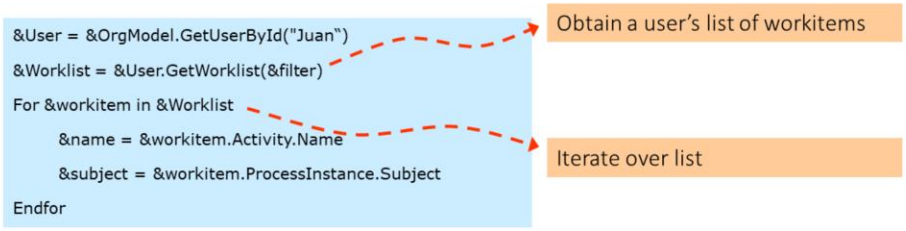
Vejamos um exemplo. Queremos atribuir um workitem a um usuário.

Pra isso, devemos seguir os seguintes passos:

1. Primeiramente, obtemos o modelo organizacional a partir do tipo Server utilizando o método GetOrganizationalModel;
2. Utilizando o método GetUserById, recuperamos o usuário cujo identificador coincide com o desejado (João);
3. Utilizamos o método Assign sobre o workitem que queremos atribuir (e que recuperamos previamente com alguns dos mecanismos já vistos) passando como parâmetro o usuário do passo anterior.

GeneXus

Example: Go over a user's pending task list



Exemplo seguinte. Queremos recorrer à lista de tarefas pendentes do usuário (isso pode servir para fazer alguma lista, criar uma caixa de entrada customizada etc).

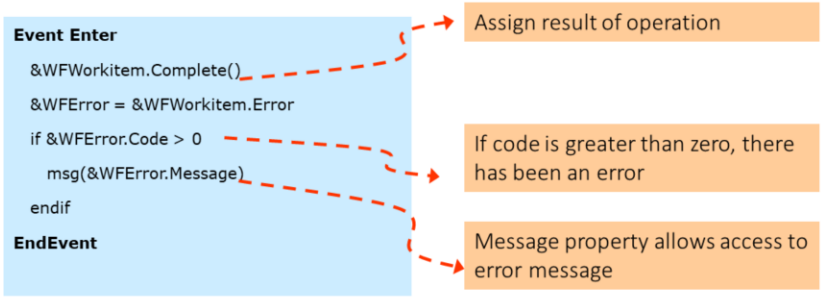
Em primeiro lugar, obtemos o usuário assim como fizemos no caso anterior.

Posteriormente, utilizamos o método que, como havíamos já dito, mostra uma coleção com todos os workitems ativos e atribuídos ao usuário (ou os que usuário é candidato).

Uma vez que obtemos a coleção, procedemos da mesma forma como em qualquer coleção de GeneXus. Por exemplo, podemos repetir sobre a mesma utilizando o construtor For &var in &collection.

Dentro da repetição, podemos obter, por exemplo, o nome da atividade associada ao workitem, o assunto da instância de processo a que pertence etc.

Error Management



Um tipo de dado que temos evitado até o momento por uma questão de simplificação, mas que é muito importante, é o tipo de dado Error.

Este tipo de dado é utilizado para comunicar os diferentes erros que podem ocorrer quando se está interagindo com o motor de workflow.

Alguns tipos de dados têm a propriedade error que permite que quando se realiza um operação (ou seja, invoca-se um método do tipo) a propriedade error se inicia com o resultado da operação.

O tipo de dado Error tem duas propriedades:

Code: que mostra o código de erro.

Message: que mostra a mensagem de erro.

Vejamos um exemplo de como esse tipo de dado é utilizado para recuperar os diferentes erros que podem ocorrer.

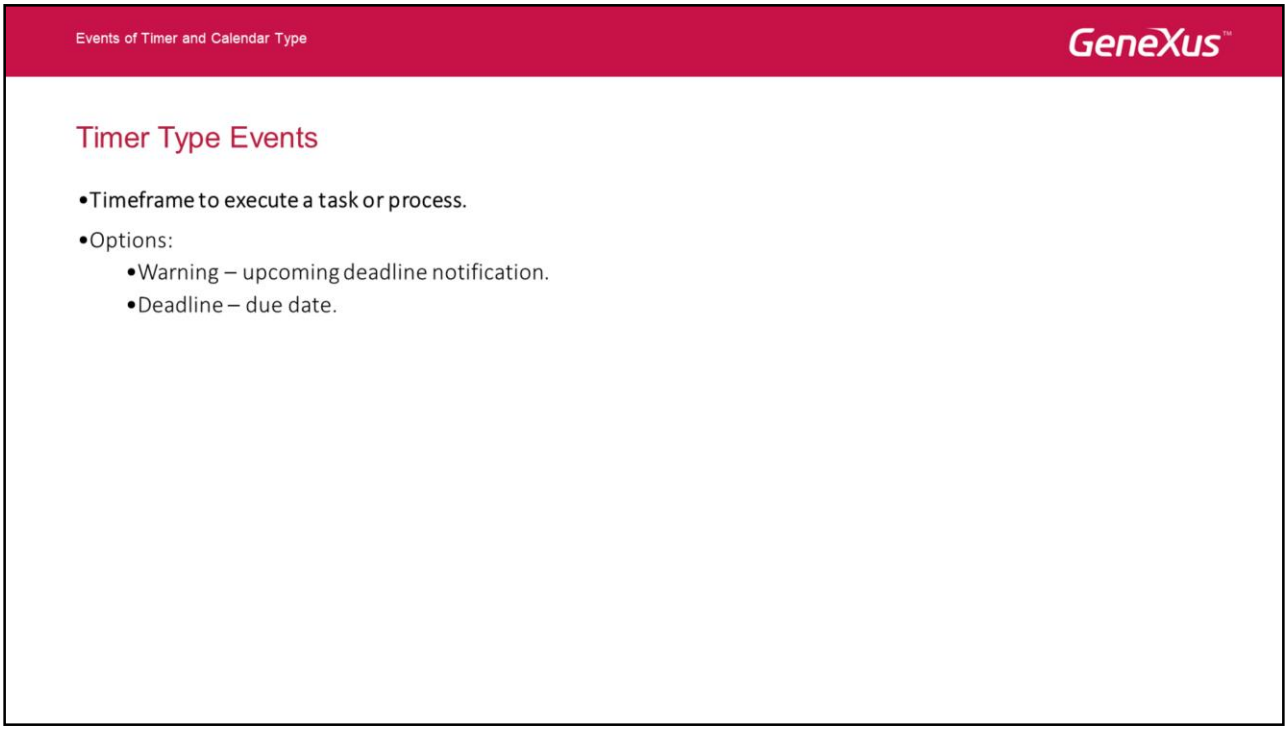
Suponhamos que desejamos completar um workitem. Como vimos anteriormente, para isso, devemos invocar o método Complete sobre o workitem correspondente.

Imediatamente depois que invocamos o método, utilizamos a propriedade Error sobre o tipo Workitem para recuperar o erro.

Depois, perguntamos pelo valor da propriedade Code sobre o tipo Error, que nos indicará se houve erro ou não. Se a propriedade mostrar zero, a operação realizou-se com erro. Se, em vez disso, o código mostrar um valor maior que zero, então houve um erro e consultando a propriedade Message, podemos obter a mensagem de erro correspondente.

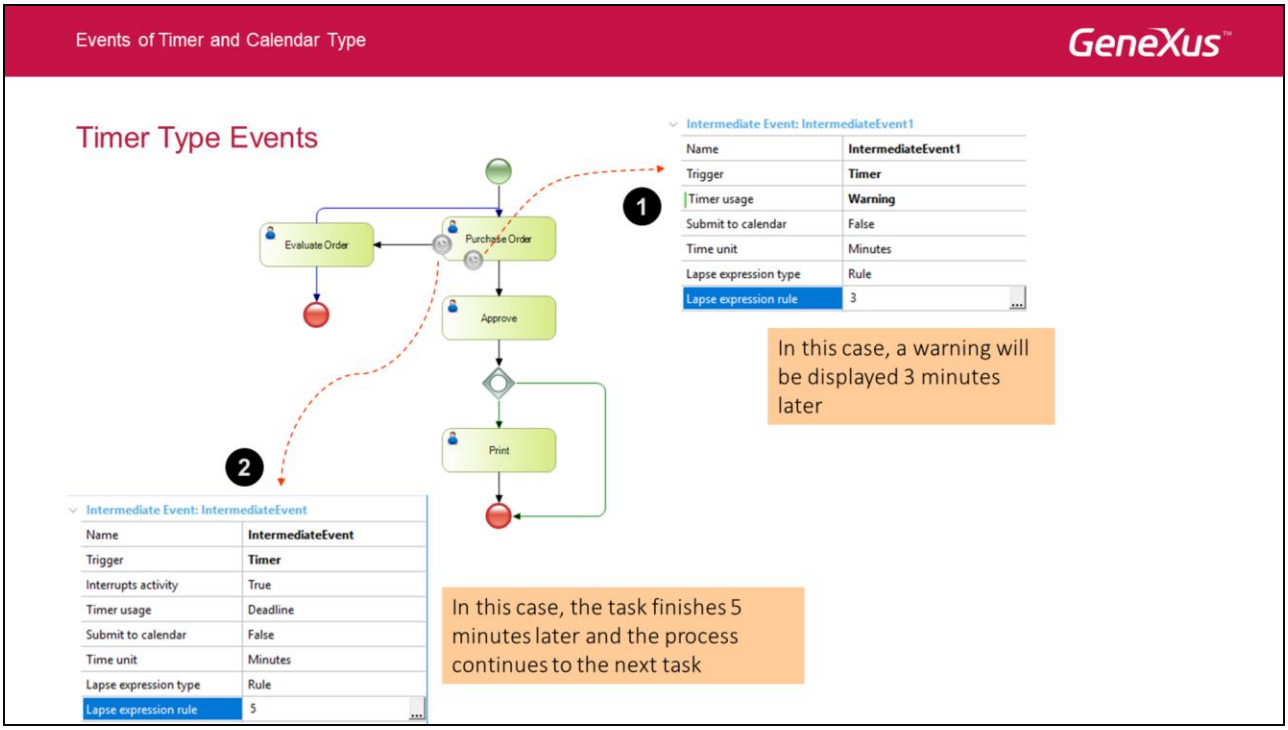
Module 6

Events of Timer and Calendar Type



Em alguns casos, queremos indicar que uma tarefa ou um processo tem um tempo limite de execução. Para indicar isso, são utilizados eventos intermediários do tipo Timer.

Dentro de um evento intermediário, é do tipo Deadline ou tipo Warning. Caso seja do tipo Deadline, indica-se qual é a tarefa que se deve seguir.



Vejamos um exemplo de como definir um Warning e um Deadline.

Por exemplo, a tarefa Purchase Order tem que ser finalizada antes dos 5 minutos, mas aos 3 minutos é preciso indicar um warning.

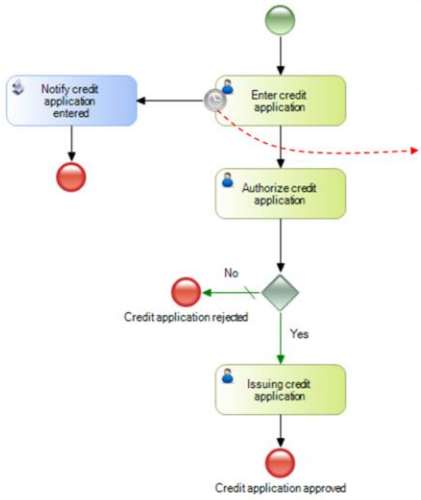
O warning é indicado na caixa de entrada na coluna que o calendário aparece.

Quando for para cumprir a deadline, a tarefa Purchase Order finaliza e a tarefa Evaluate Order é executada.

GeneXus

Time Monitoring

- It is also possible for the attached timer event to not interrupt the task.

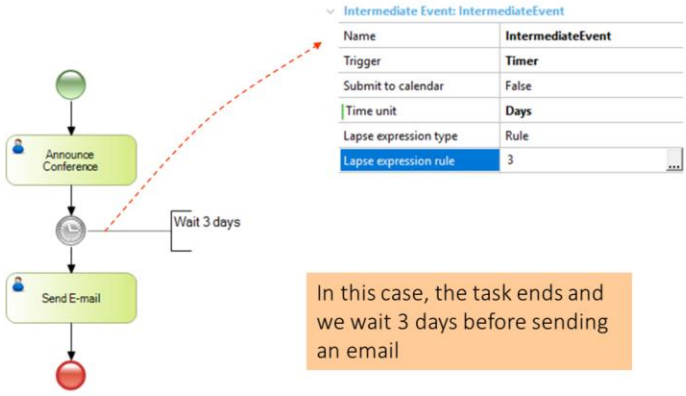


Intermediate Event: IntermediateEvent

Name	**IntermediateEvent**
Trigger	**Timer**
Interrupts activity	**False**
Timer usage	Deadline
Submit to calendar	False
Time unit	Minutes
Lapse expression type	Rule
Lapse expression rule	30

GeneXus

Delays



Como se define um delay em um diagrama?

Um delay é modelado inserindo um evento do tipo timer entre atividades.

Pode representar uma data e uma hora específicas (por exemplo: esperar até 15 de abril, às 17h), ou uma data relativa (esperar seis dias) ou data relativa repetitiva (esperar até a próxima segunda-feira, às 8h).

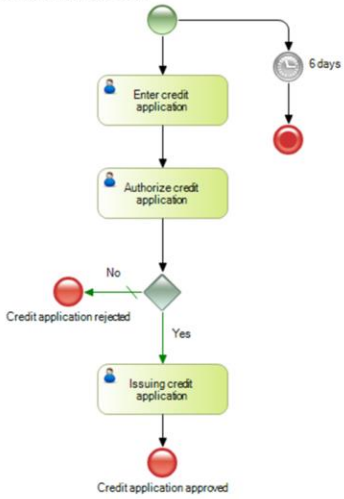
Quando o timer gera o alerta, o processo continua seu fluxo.

O elemento prévio ao delay pode ser uma atividade, um gateway ou outro evento intermediário.

GeneXus

Process Deadline

- The total duration of a process must be demarcated.



GeneXus

Calendar

- It contains the working days and times.
- Timer type events can be adjusted to the calendar.
- Available in tasks and processes.
- Options:
  - No calendar: the actual time is used.
  - Calendar: it is adjusted to the selected calendar

Contém os dias e horas hábeis de trabalho.

Os eventos do tipo timer podem se ajustar ao calendário.

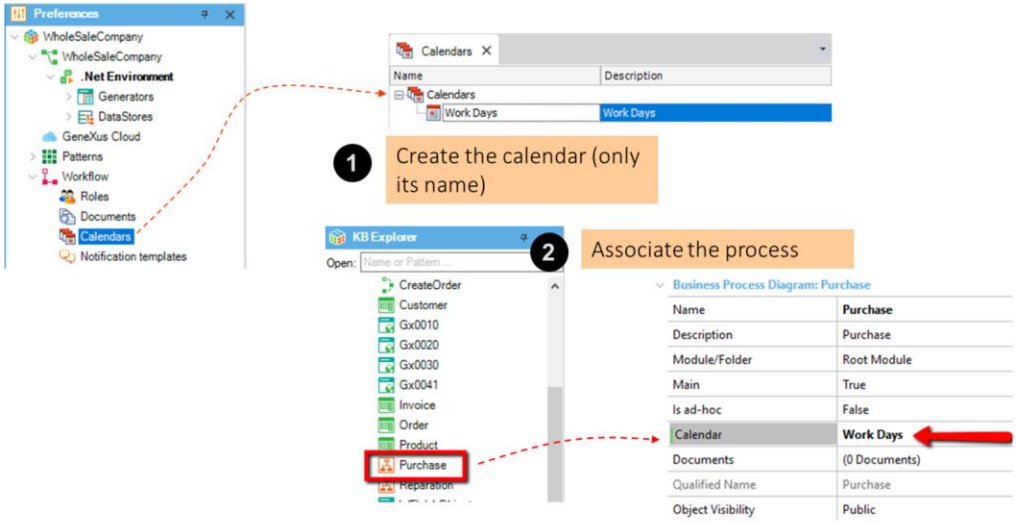
Disponível em tarefas e processos.

Opções:

No calendar – usa o tempo real.

Calendário – ajusta-se ao calendário selecionado.

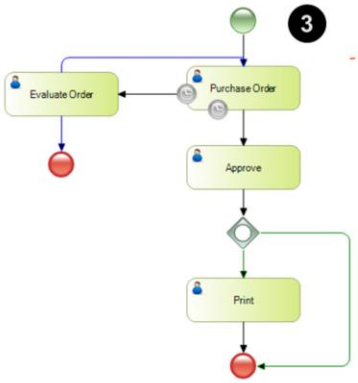
Calendar Configuration (I)



GeneXus

Calendar Configuration (II)

In activities we can configure the process calendar or use another one



Intermediate Event: IntermediateEvent

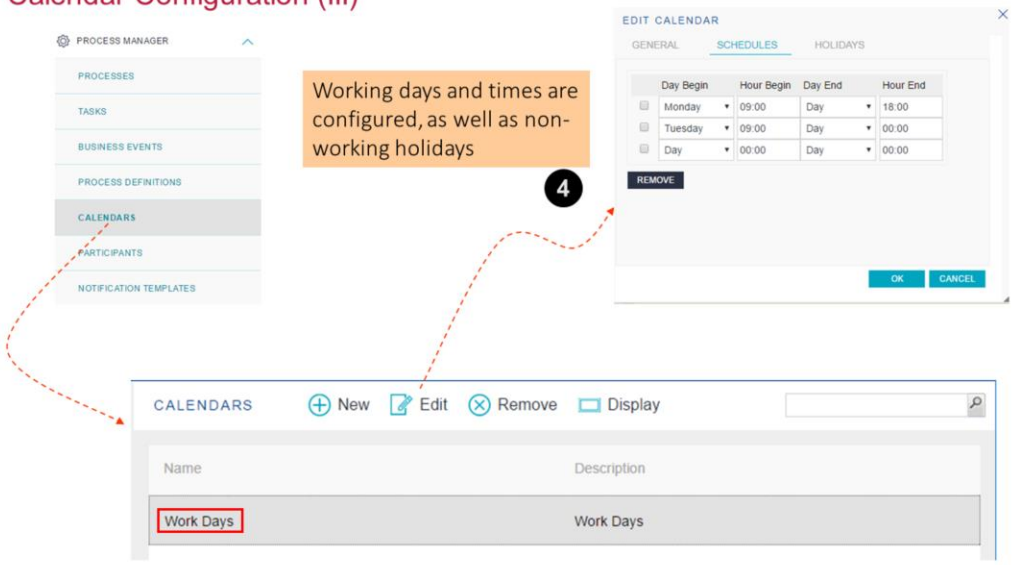
Name	**IntermediateEvent**
Trigger	**Timer**
Interrupts activity	True
Timer usage	Deadline
Submit to calendar	**True**
Calendar	Work Days
Time unit	Minutes
Lapse expression type	Rule
Lapse expression rule	5

Time unit used to calculate the deadline: days, hours, minutes o seconds

Adjust deadlines to the calendar (True) or use actual time (False)

GeneXus

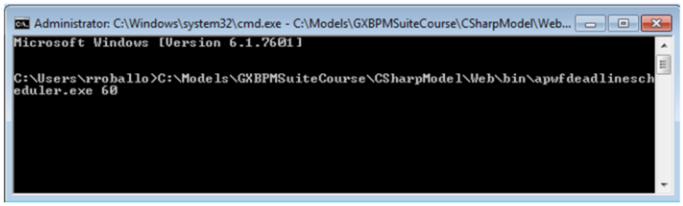
Calendar Configuration (III)

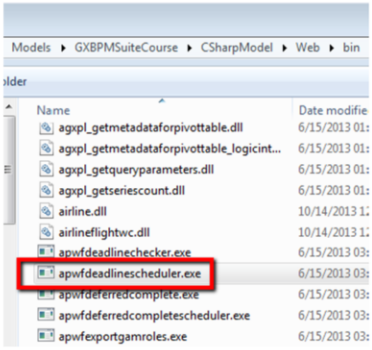


GeneXus

Calendar Configuration (IV)

- Utilities to check deadlines.
- Executed via command line.
  - WFDeadlineScheduler:
    - Executed on the background
    - The parameter is the elapsed time, in seconds, between each execution.
    - <WebApp>\web\bin\apwfdeadlinescheduler.exe 60
  - WFDeadlineChecker:
    - It is executed once and finished.





Útil para checar os deadlines.

Executa por linha de comando.

WFDeadlineScheduler:

Executa em segundo plano

O parâmetro é o lapso em segundos, entra cada execução.

<WebApp>\web\bin\apwfdeadlinescheduler.exe 60

WFDeadlineChecker:

Executa uma vez e termina.

GeneXus

Calendar related Workflow Data Types

- WorkflowProcessInstance and WorkflowWorkitem

- Most used Properties:
  - WarningTime – Warning date and time.
  - DeadlineTime – Deadline date and time.

Both properties are read-write

- WorkflowCalendar

- Most used Properties:
  - Name
- Most used Methods:
  - IsWorkday – indicates if a date corresponds to a working day.
  - FirstWorkDay – returns the date of the first working day of the month.
  - LastWorkdayInMonth – returns the last working day of the month.

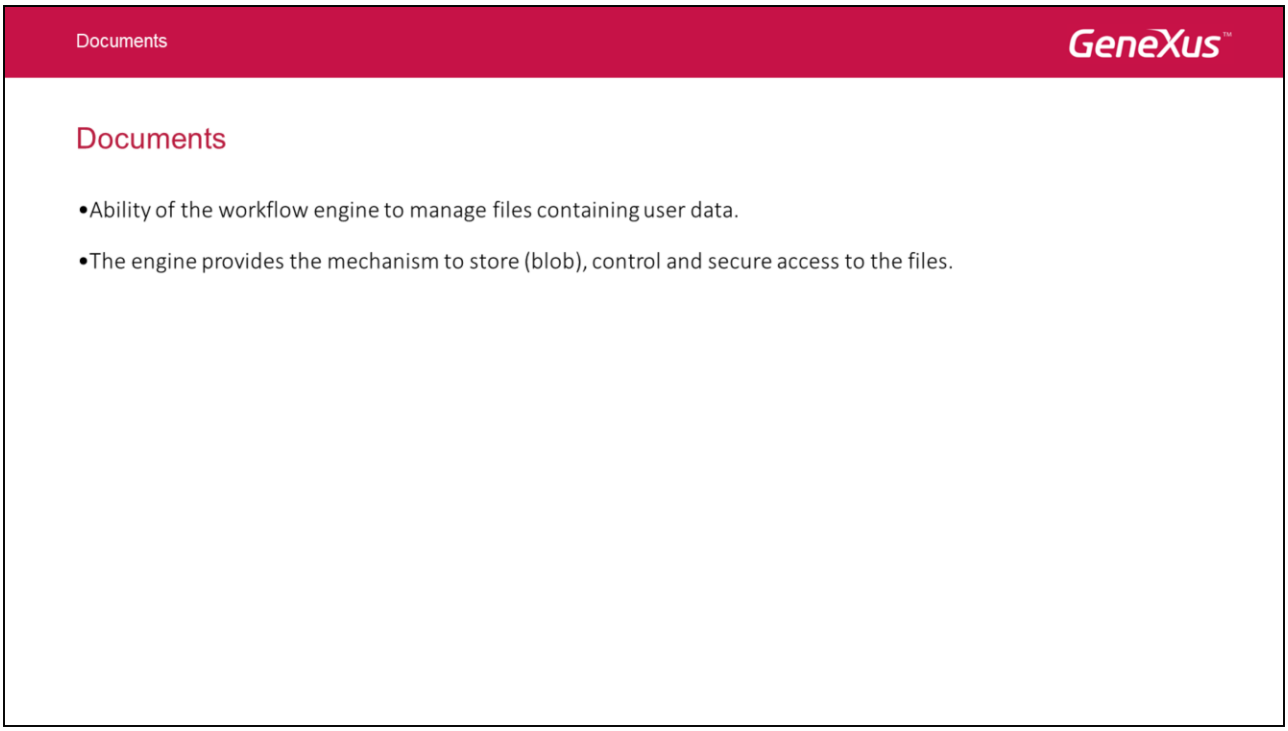
Los tipos de datos WorkflowProcessInstance y el WorkflowWorkitem manejan el warningtime y el deadlinetime través de propiedades. Por más información:

http://wiki.genexus.com/commwiki/servlet/wiki?17771 y
http://wiki.genexus.com/commwiki/servlet/wiki?11731

Por mas información sobre el tipo de datos Workflow Calendar
http://wiki.genexus.com/commwiki/servlet/wiki?11609

Module 7

Documents



Capacidade do motor de workflow para manejar arquivos com informação de usuário. O motor fornece o mecanismo de armazenamento (blob), controle e segurança de acesso aos arquivos

Documents

Types of Documents:

- Microsoft Word, Excel, Power Point
- PDF
- Open Office: Text, Spreadsheet, Presentation
- StarOffice: StarWriter, StarCalc, StarImpress
- Images: JPG, BMP, GIF
- Text
- XML
- ZIP
- Other

É possível definir de que tipo serão os documentos que farão parte do processo, permitindo restringir que o usuário a publique arquivos desse tipo. Caso não se especifique, o tipo ficará como Other, permitindo publicar arquivos de qualquer tipo.

GeneXus

Templates

Document template:

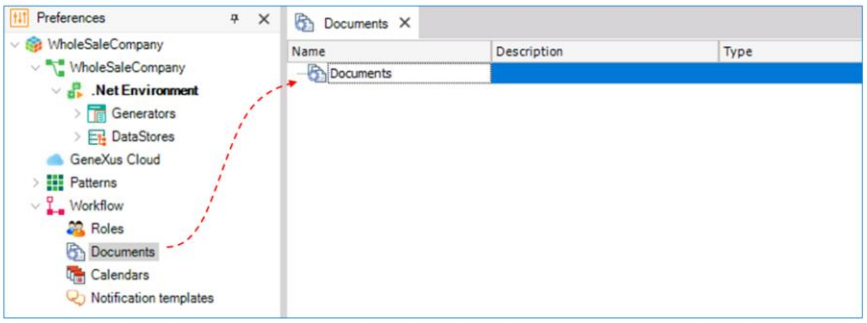
- Documents can be based on templates that define their basic structure.

- Template Property:
  - Defined by the user (associated file).
  - Empty.

Os documentos podem estar baseados em templates, modelos que definem sua estrutura básica. Para isso, utiliza-se a propriedade Template, que indica o nome do modelo que esse documento utiliza.

Devem-se criar templates como os demais arquivos associado à KB, para fazê-lo, deve-se ir à opção Folder View – Files. Os documentos criados nessa opção ficarão disponíveis para serem associados como modelos dos documentos manjedos pelo GXflow.

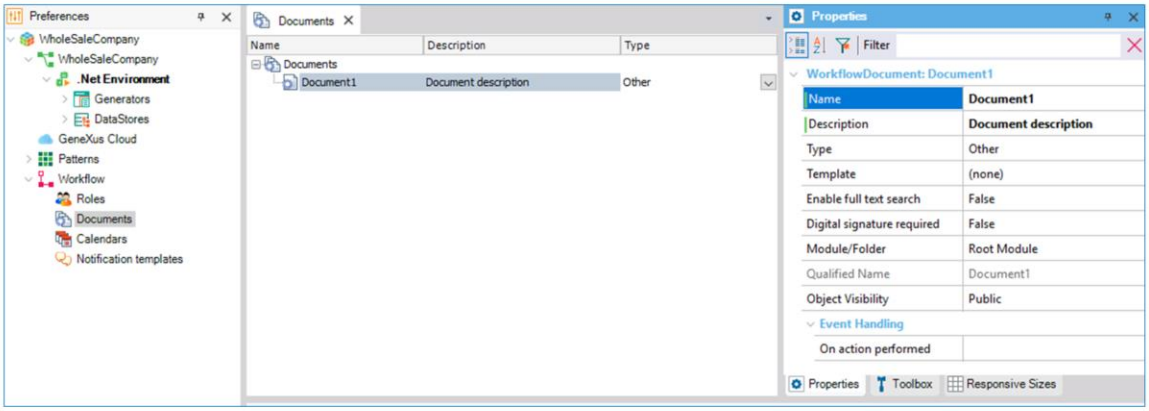
Definition in GeneXus



Para utilizar documentos em diagramas de processos, primeiro é preciso **defini-los na base de conhecimento (KB**). Para isso, é preciso ir a **Preferences – Workflow –Documents** e pressionar a tecla **Enter** para acrescentar novos documentos.

No momento de definir o documento, é preciso indicar o nome, uma descrição e, opcionalmente, o tipo do documento.

Definition in GeneXus



As seguintes propriedades também podem ser configuradas:

Name: Nome do documento

Description: Descrição do documento

Type: Tipo do documento

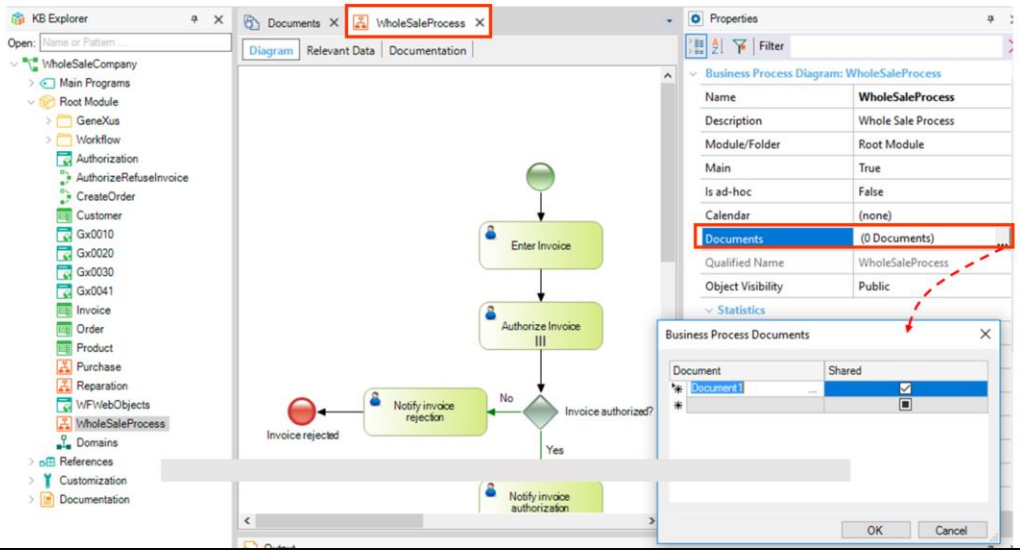
Template: Permite associar um modelo.

Enable Full Text Search: Permite habilitar as buscas sobre o conteúdo do documento.

Digital Signature Require: Permite indicar que o documento deve ser assinado digitalmente.

Event Handling - On action performed: Permite associar um manejador de eventos (um procedimento) para que seja executado quando se realiza uma ação sobre o documento.

Documents and Diagrams

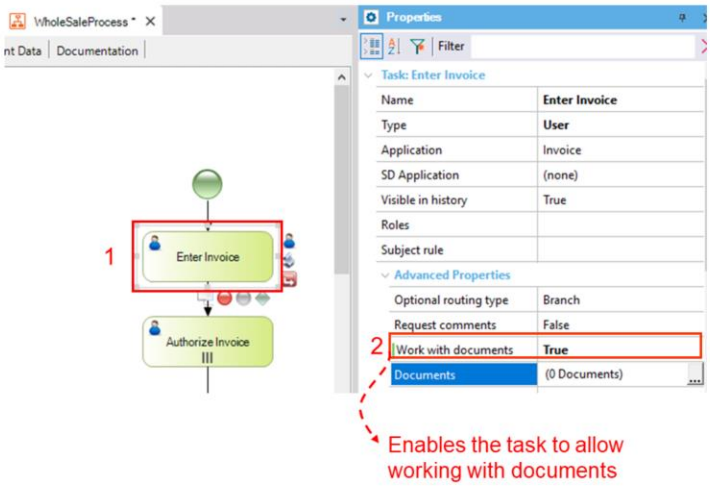


Os documentos podem ser associados aos diagramas, ao serem associados a esse nível, os documentos estarão disponíveis para todas as tarefas do processo.

Pode-se indicar depois o papel que pode trabalhar com esse documento, e os usuário que tiverem esse papel poderão trabalhar com eles.

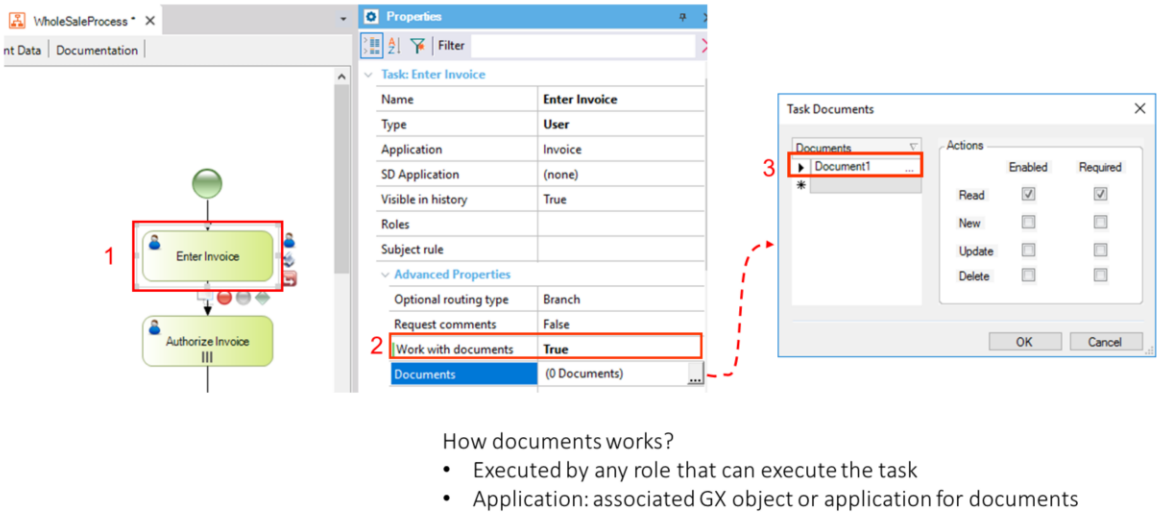
GeneXus

Documents and Tasks



Os documentos podem ser associados a cada tarefa; para tanto, é preciso marcar a propriedade Work With Documents em True, e feito isso, é habilitada a propriedade Documents, onde são indicados quais são os documentos associados à tarefa.

Documents and Tasks



How documents works?

- Executed by any role that can execute the task
- Application: associated GX object or application for documents

Ao marcar a propriedade Work With em True, habilita-se a propriedade Documents, onde são selecionados os diferentes documentos associados à tarefa.

Além de indicar quais são os documentos, é preciso indicar quais são as ações que podem ser realizadas sobre eles.

As ações que podem executadas sobre os documentos são:

- Read -> Permite-se a leitura do documento
- New -> Permite-se a criação de um novo documento
- Update -> Permite-se a modificação do documento
- Delete -> Permite-se a exclusão do documento

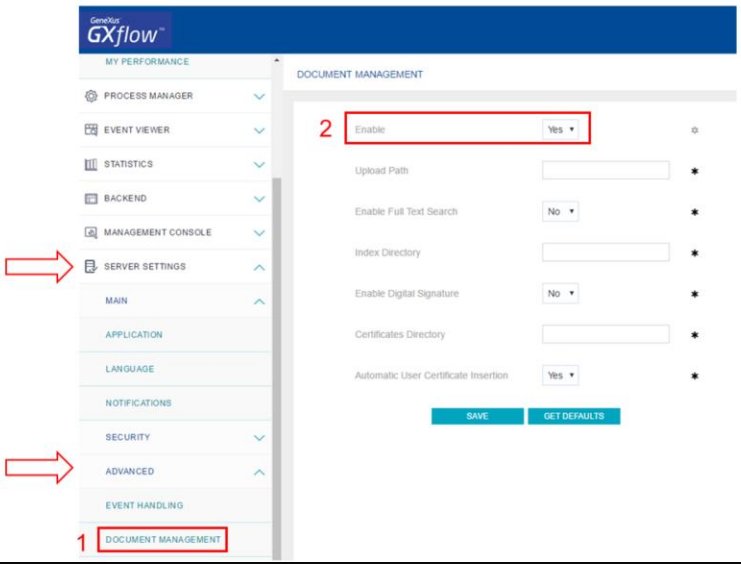
Além de indicar que ação pode se realizar sobre o documento, é preciso indicar se elas são Obrigatórias (Required) ou Disponíveis (Enabled).

Como funciona o manejo dos documentos em GXflow?

O documento associado a uma tarefa pode ser manejado por qualquer um dos papéis que podem executar a tarefa.

Se a tarefa não tem um programa GeneXus associado, então executa-se uma aplicação para trabalhar com os documentos.

Desktop Documents



Quais são as configurações que é preciso fazer em GXflow para poder manejar com os documentos?

Configuração na KB GeneXus:

Propriedades do gerador -> Client server information -> Blob local storage path: diretório onde são armazenados temporariamente os documentos blob durante a supressão ou inclusão no banco de dados.

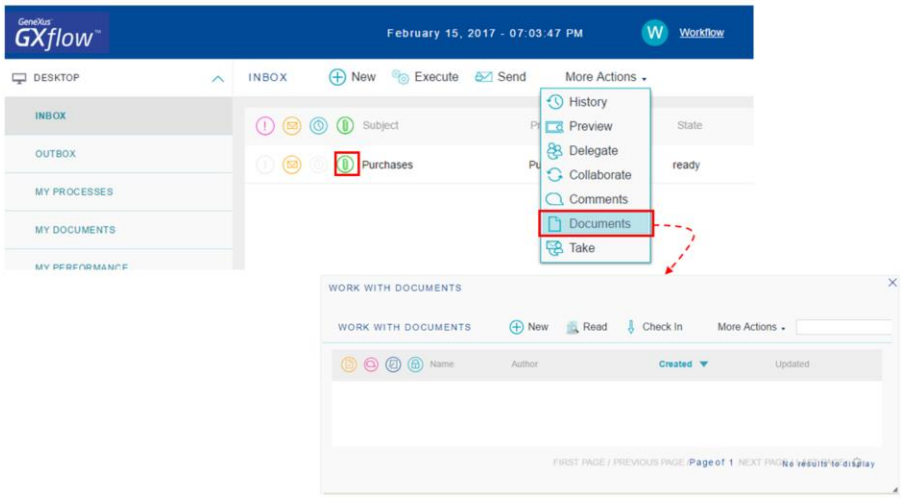
Configuração na área de trabalho:

En Settings -> Server –> Advanced -> Document Management

Habilitado: Habilita ou não o manejo de documentos.

Rota de publicação: diretório para publicar os documentos no servidor web.

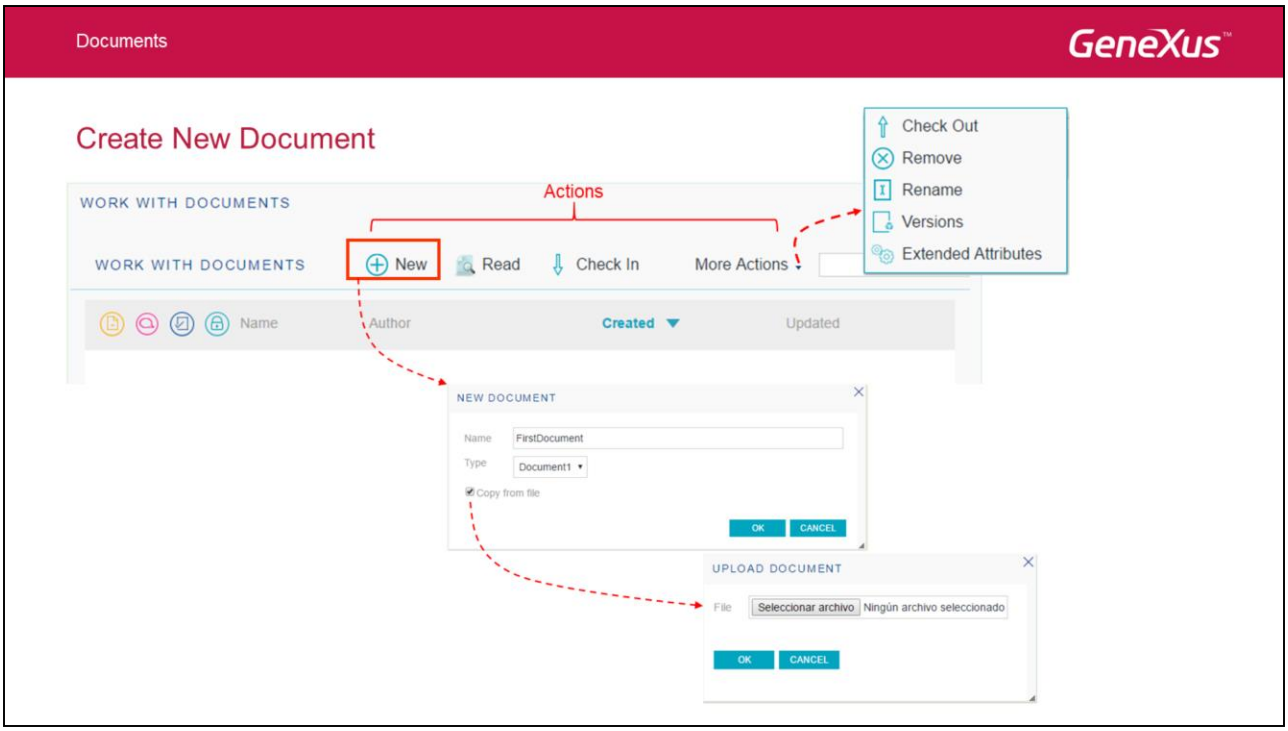
Desktop Documents



Para mostrar que uma tarefa está associada a documento, indica-se com a seguinte imagem.

Caso a tarefa esteja composta por uma aplicação e por documentos, para acessá-los é preciso ir à opção **More Actions - Documents**, onde se abre o Work With Documents.

Caso a tarefa não esteja associada a nenhuma documento, abre-se automaticamente o Work With Documents.

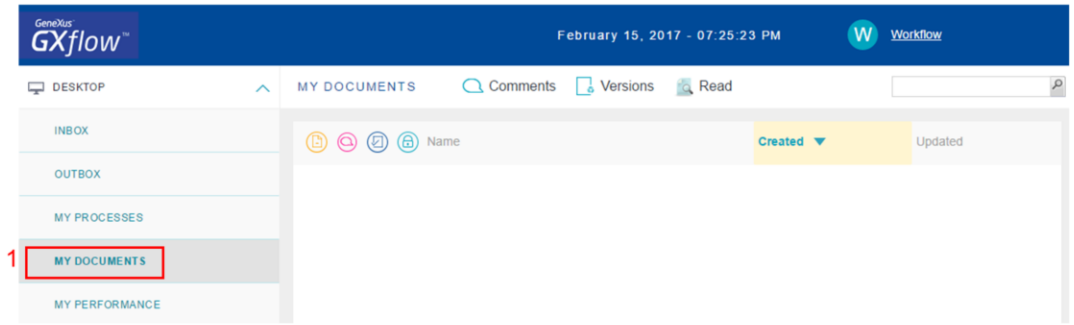


As ações disponíveis vão depender das permissões que um usuário tiver sobre os documentos.

As opções que se mostram no slide são New, Read, Check In e Check Out.

No caso de New, abre-se um nova janela na qual é preciso registrar o nome do documento e seu tipo. No momento de criar o documento, é possível criá-lo copiando outro documento existente no PC.

My Documents



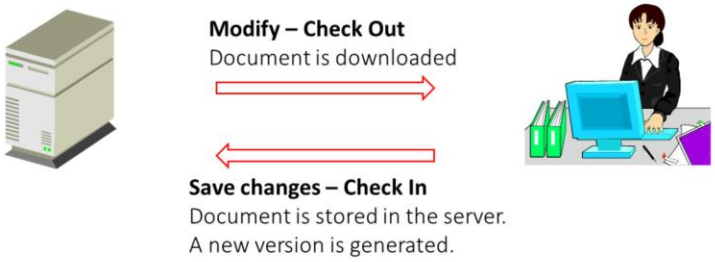
Sob a opção **Desktop – My Documents**, todos os documentos com os quais o usuário trabalhou podem ser vistos.

Selecionando um documento em especial, pode-se acrescentar comentários a ele, ler sua última versão ou ir a versões nas quasis podem ser visualizadas cada uma das versões de cada documento.

GeneXus

Desktop Documents

- Document repository: The engine maintains a document repository.
- How it works:



O motor de GXflow tem um repositório de documentos que funciona da seguinte maneira:

- Para modificar um documento, é preciso fazer um **Check Out**, isso faz com que se baixe o documento.

- O usuário trabalha com o documento e quando já finalizou as alterações tem que fazer um **Check In**, com o qual se publica o documento no servidor.

- Ao realizar um **Check In** do documento, cria-se uma nova versão dele.

GeneXus

Desktop Documents

Document repository:

- Only the first user who performs the Check Out operation can perform the Check In.
- Other users can read the document (Check Out) but they cannot perform the Check In operation.
- The administrator can cancel the Check Out.
- A new version is created with every Check In.
- The changes made between Check Out and Check In are NOT recorded by the Workflow engine.

Quando um mesmo documento é compartilhado por vários usuários, apenas o primeiro que fizer o Check Out pode fazer o Check In.

Durante esse processo, o documento está bloqueado, mas os usuários poderão ler (Read) ou baixar o documento (Check Out).

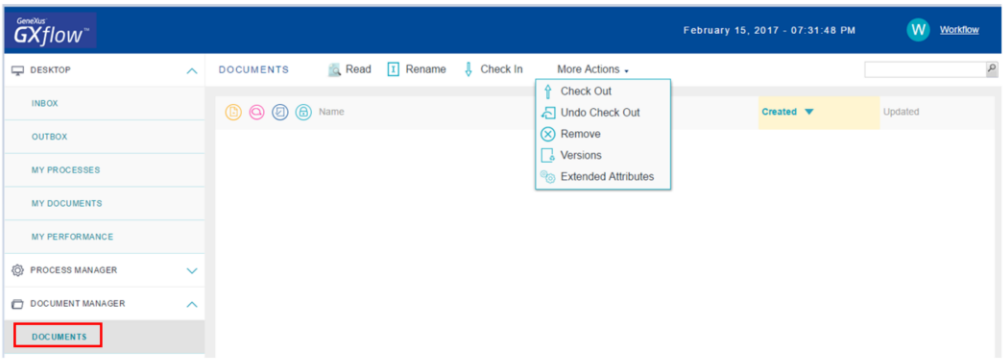
Somente o usuário administrador tem permissões suficientes para cancelar o Check Out de um documento.

A cada vez que se faz um Check In, cria-se uma nova versão do documento.

As alterações realizadas entre o Check Out e Check In NÃO são registradas pelo motor de Workflow.

GeneXus

Document Administrator

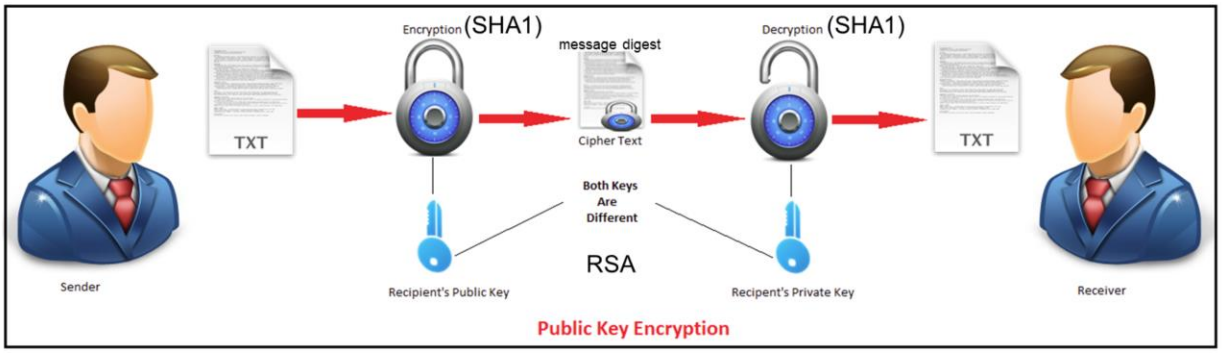


Para poder realizar as tarefas de administração dos documentos, o administrador tem que ir à opção: **Document Manager – Documents.**

Aí se verá a lista de documentos e o administrador poderá realizar várias operações, como:

Desfazer um check out, renomear um arquivo ou eliminá-lo.

Document Digital Signature



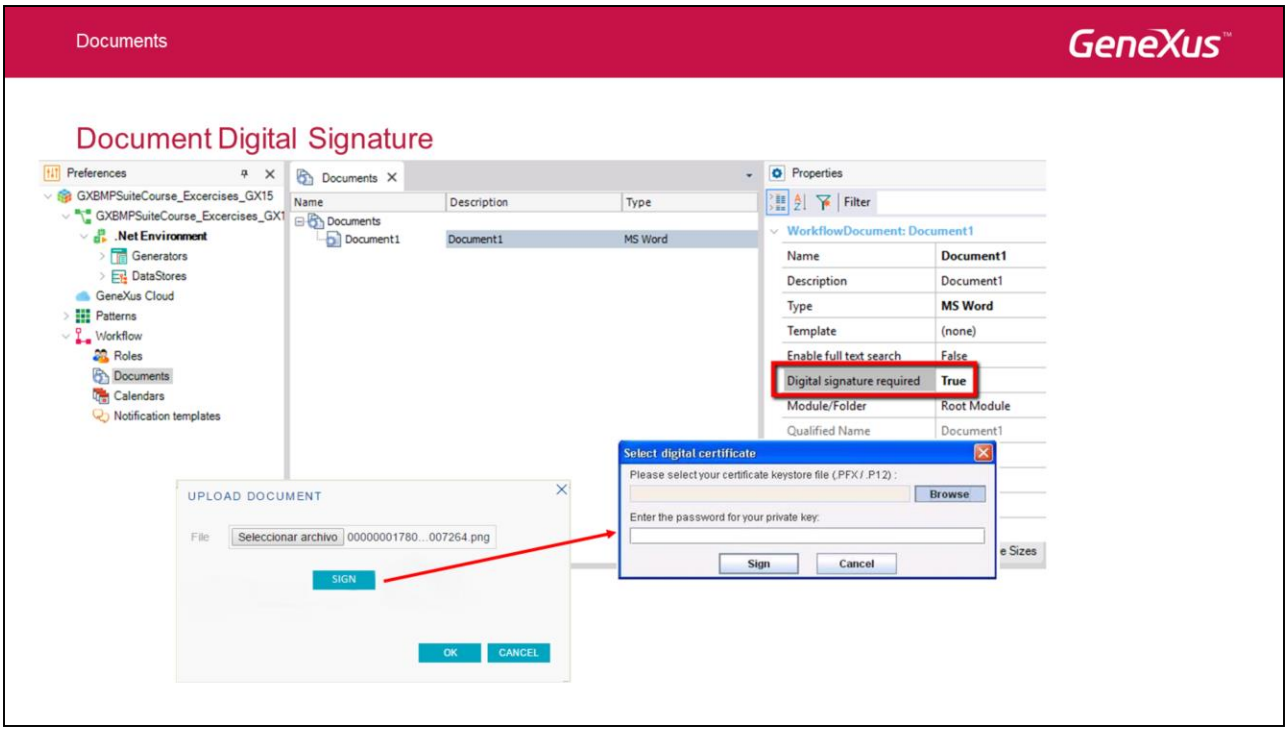
Permite assinar um documento ao postá-lo no servidor.

A assinatura permite:

- Verificar a integridade do documento (que não foi alterado).
- Assegurar a veracidade do documento (comprovar que um usuário é quem efetivamente o assinou).

Tecnologia utilizada:

- Algoritmo RSA para as assinaturas.
- SHA1 – Para calcular a assinatura



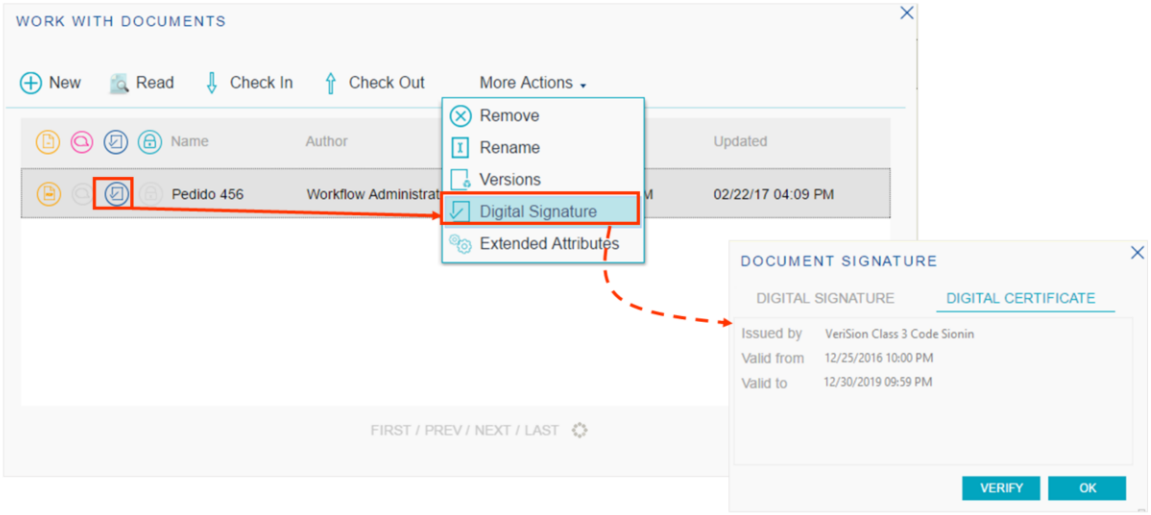
Antes de salvar o documento no Banco de Dados, verifica-se:

Se a assinatura é válida (o documento não foi alterado),

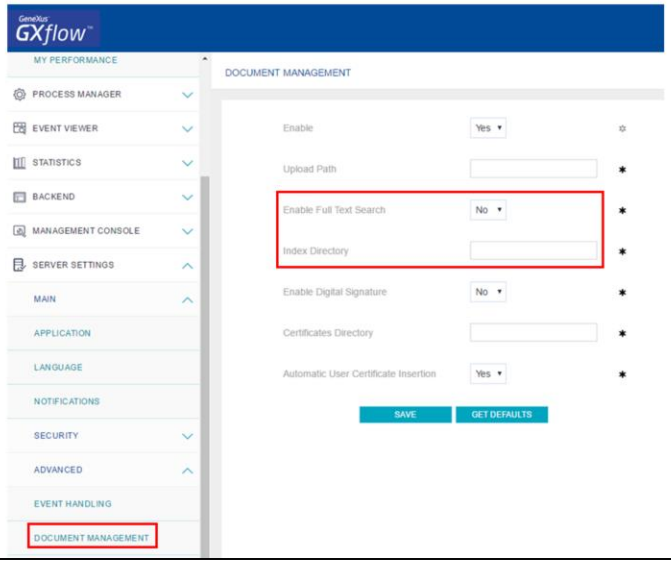
Se o certificado do usuário é válido (não expirou e não revogado)

Se o certificado é de confiança (emitido por alguma autoridade certificadora de confiança).

Document Digital Signature



Document Searches



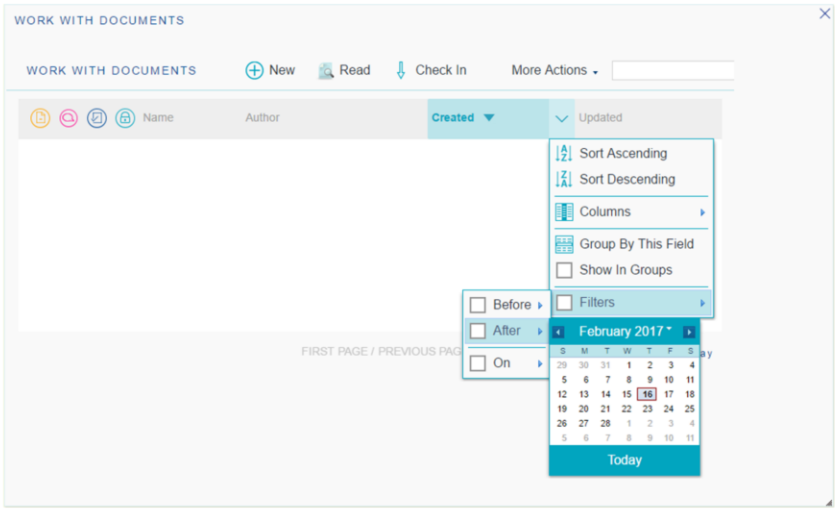
Para configurar as buscas sobre o conteúdo dos documentos, é preciso ir a **Settings – Server – Advanced - Document Management**

Aí, é preciso configurar as propriedades:

- **Full Text Search**: Permite habilitar o mecanismo de busca.
- **Index Directory**: Permite indicar o diretório onde se armazenará o índice de palavras-chave.

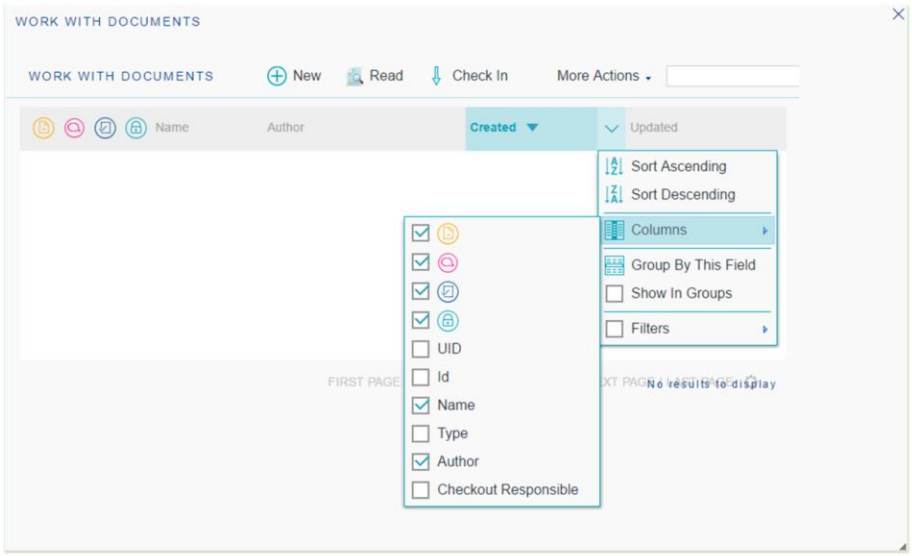
GeneXus

Document Searches



GeneXus

Document Searches



Filtros nas aplicações de Documentos:

Tipo de documento.
Autor – usuário que criou o documento.
Conteúdo – palavras que se encontram dentro do documento.
Filtros de Data.
Responsável pelo check out.
Assinado digitalmente.
Comentários.
Se o documento está ou não em uso

GeneXus

Workflow Data Types

- WorkflowDocumentRepository: it represents the document repository

- Most used Methods:
  - ListDocumentDefinitions (filter): returns document definitions.
  - ListDocumentDefinitions (filter): returns document instances.

GeneXus

Workflow Data Types

- WorkflowDocumentDefinition: represents a document definition

- Most used Properties:
  - Id
  - Name
- Most used Methods:
  - CreateInstance(workitem, name) – Creates a new document instance associated with the work item
  - CreateInstanceFromFile(workitem, name, file) – Uses a file to create the document instance

GeneXus

Workflow Data Types

- WorkflowDocumentInstance: represents a document instance

- Most used Properties:
  - Name
  - Version – Document version
  - CheckedOutBy – User who downloaded the document
- Most used Methods:
  - Read (user, workitem, targetDirectory) – Opens the document to read it; it is temporarily downloaded to the directory.
  - CheckIn (user, workitem, file) – Adds the document to the repository and enables it again to be downloaded.
  - CheckOut (workitem, targetDirectory) – Downloads and blocks the repository document, and allows making changes to it.
  - UndoCheckOut (user) – Undo the blocking

GeneXus 15

Module 8

Statistics

Neste Modulo, poderemos analisar as diferentes estatísticas incorporadas ao Cliente de GXflow.

Statistics

- Real-time monitoring of system statistics within the GXflow Client.

O usuário administrador tem a possibilidade de monitorar em tempo real diferentes estatísticas geradas no sistema.

As estatísticas são:

- Análise de Processos.
- Análise de tarefas.
- Performance de Processos.
- Performance de Tarefas.
- Performance de Equipes.

A seção Análise de processos inclui os seguintes gráficos:

- Atividade do processo:

Subdivide-se em 2 gráficos:

1. Atividade de processos: Permite ver o estado do processo, a quantidade de instâncias criadas, concluídas e interrompidas em um período selecionado.

2. Tendência de Instância Ativa: Permite ver a tendência da instância ativa para um processo em um determinado período.

- Tempo do ciclo

1. Estado do Processo: Mostra a porcentagem das instâncias de processos fechadas que terminaram a tempo e aquelas que demoraram mais que o previsto.

2. Duração do processo: Permite analisar a duração dos processos mostrando a duração média vs. Duração estimada das instâncias completas para o processo nos períodos selecionados.

- Ranking de processos

Compara o número de instância de processo criadas para um processo num determinado período.

GeneXus

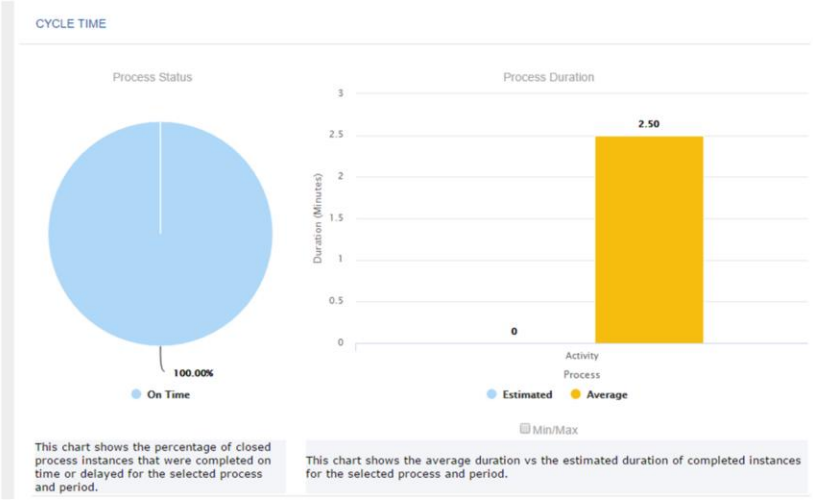
Process Analysis

- Process Activity

GeneXus

Process Analysis

- Cycle time



GeneXus

Process Analysis

- Process Ranking

This chart compares the number of created instances for each process in the selected period.

Como no caso dos processos, há um indicador para ver o tempo do ciclo, mas, desta vez, em relação às tarefas, o que nos permite um nível granularidade mais fino para fazer uma análise.

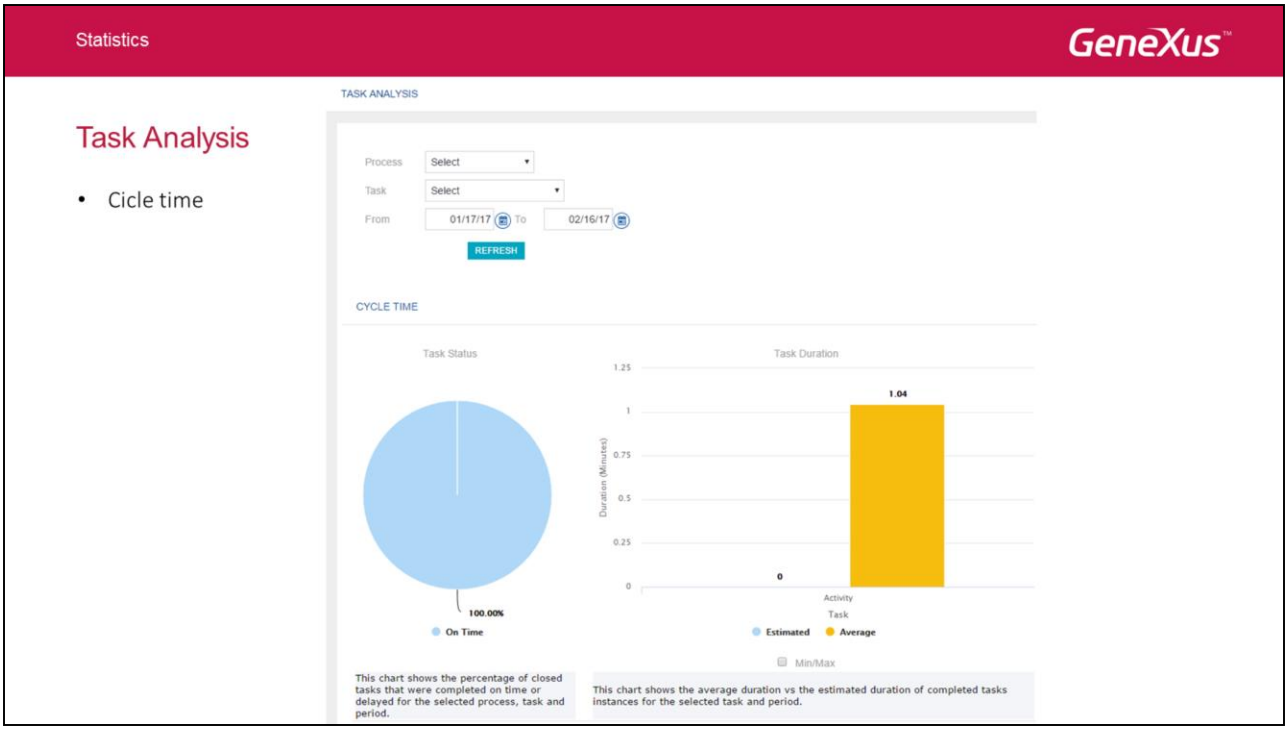
- Tempo do ciclo:

Pode ser analisado em dois gráficos:

1. Estado da tarefa: Mostra a porcentagem das tarefas fechadas, concluídas a tempo ou demoradas segundo o processo, a tarefa e o período selecionado.
2. Duração da tarefa: Mostra a duração média em relação à duração estimada das instâncias das tarefas concluídas, segundo um período selecionado.

- Ranking de tarefas:

Compara a quantidade de instâncias criadas para cada tarefa para o processo num período selecionado.



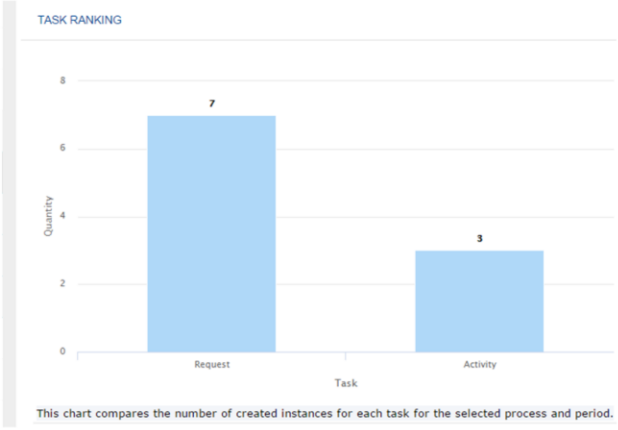
A análise das tarefas pode ser feita com base nos seguintes filtros:

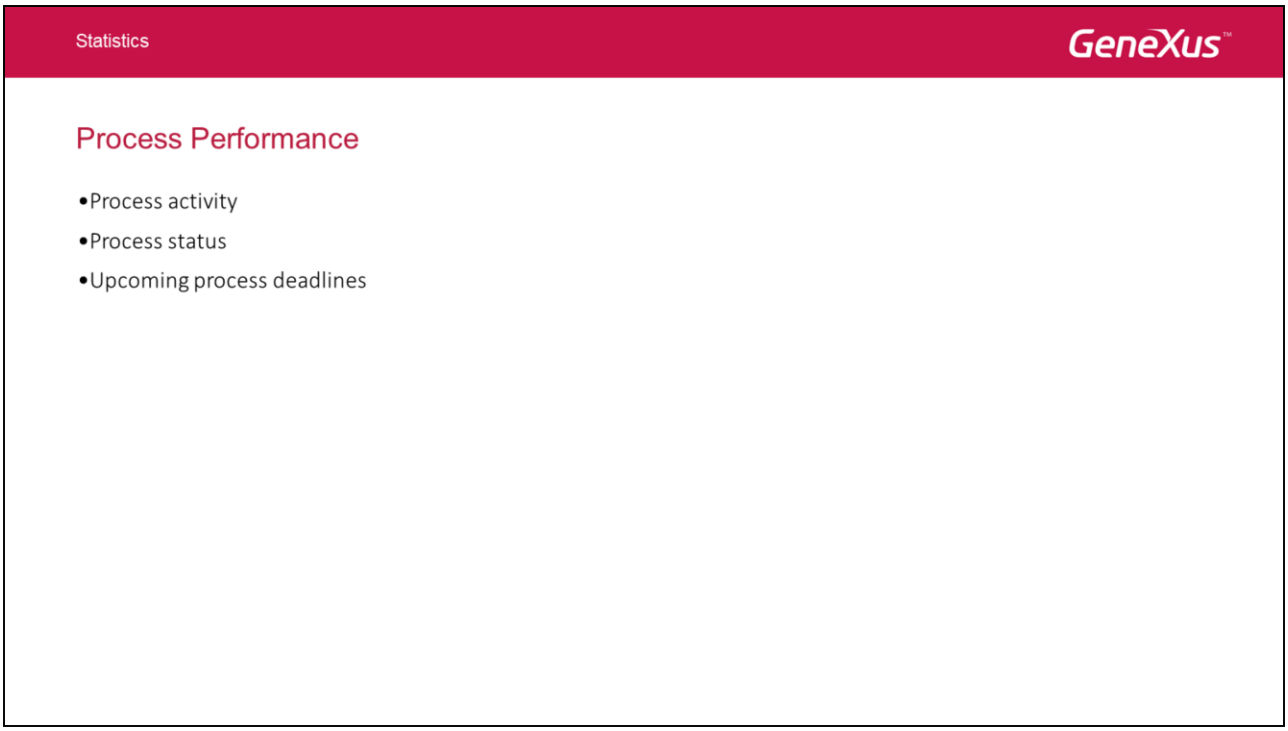
- Processo
- Tarefa
- Categorias de datas

GeneXus

Task Analysis

- Task ranking





Esta seção permite-nos ver o estado atual dos diferentes processos em execução, podemos identificar se estão dentro do tempo e forma ou se estão atrasados.

Os gráficos são:

- Atividade do processo:

Este gráfico compara a tendência da instância ativa nos últimos dias para o processo selecionado.

- Estado do processo:

Possui dois gráficos.

1. Estado do Processo Geral: Mostra a porcentagem de instâncias de processo ativos que estão a tempo, em situação de risco ou atrasadas.
2. Detalhe do Estado do Processo: Mostra a quantidade de instâncias de processos ativos que estão a tempo, em situação de risco ou atrasadas para cada processo.

- Próximos vencimentos do processo:

Este gráfico mostra quando as instâncias de processos estão próximas do vencimento.

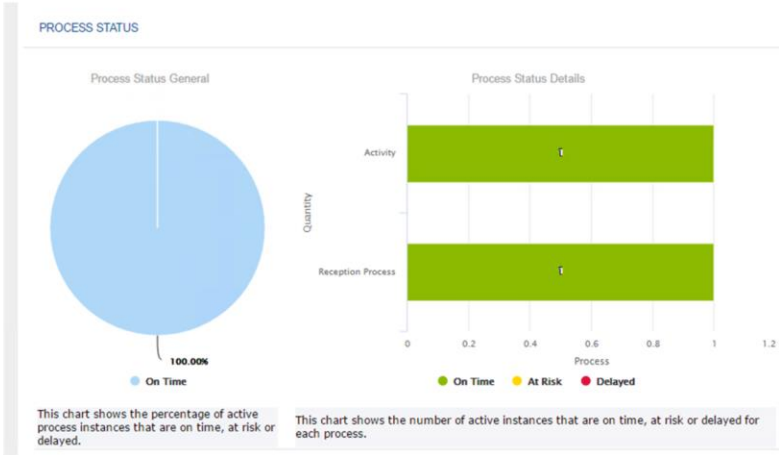
Process Performance

- Process Activity

GeneXus

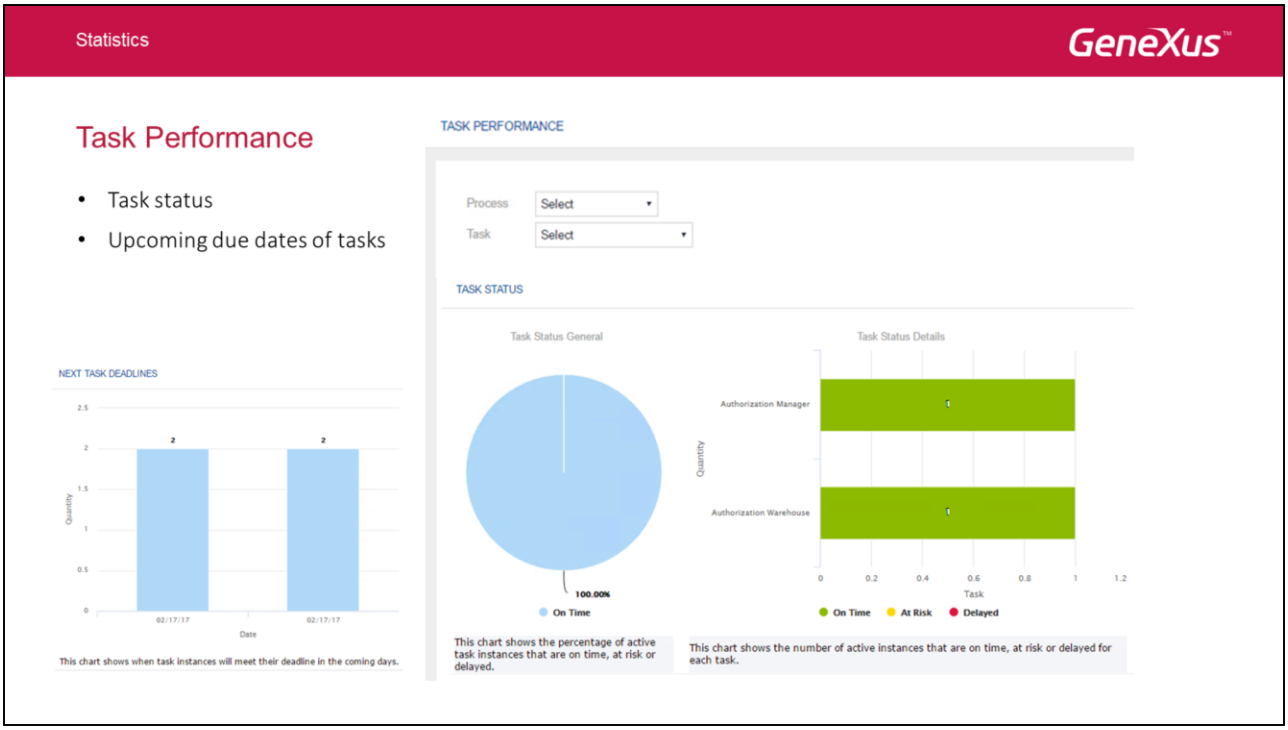
Process Performance

- Process status



Process Performance

- Next process deadlines



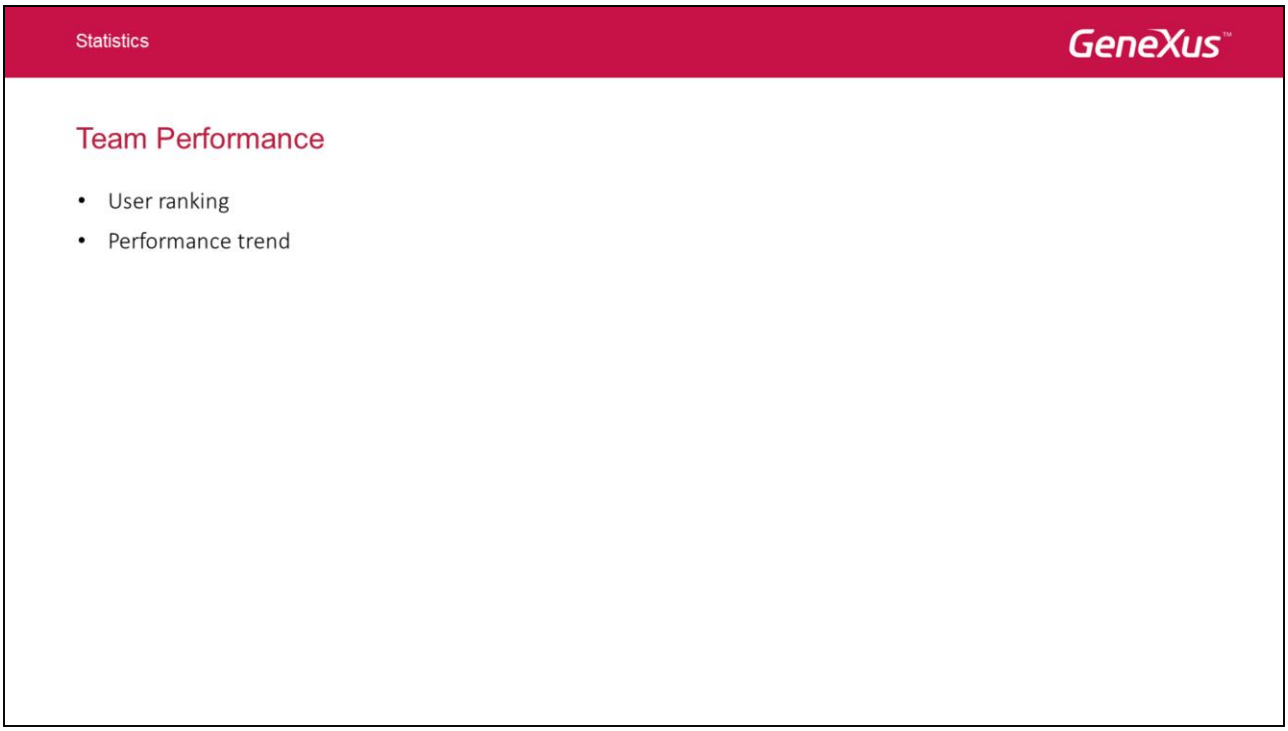
Assim como a seção de performance de processo, é possível saber o estado em que as tarefas se encontram.

- Estado da tarefa:

1. Estado geral da tarefa: Mostra a porcentagem de tarefas ativas que estão a tempo, em situação de risco ou atrasadas.
2. Detalhe do estado da tarefa: Quantidade de tarefas ativas que estão a tempo, em situação de risco ou atrasadas para cada tarefa.

- Próximos vencimentos da tarefa:

Este gráfico mostra quando as tarefas vencerão nos próximos dias.



Permite analisar a produtividade dos diferentes usuários da organização, tendo a possibilidade de comparar com base em um papel um processo ou tarefa determinada.

- Ranking de usuários:

Compara a quantidade de tarefas concluídas ao longo dos últimos dias por cada usuário e processo selecionado. Mostram-se apenas os usuários que têm ao menos uma tarefa completa no período selecionado.

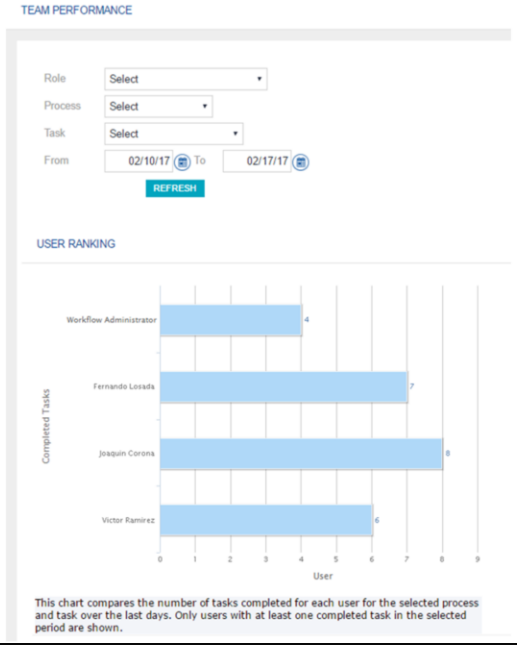
- Tendência do rendimento:

Esse gráfico compara a tendência de rendimento ao longo dos últimos dias entre os membros de um papel selecionado para o processo e a tarefa selecionados.

GeneXus

Team Performance

- User ranking



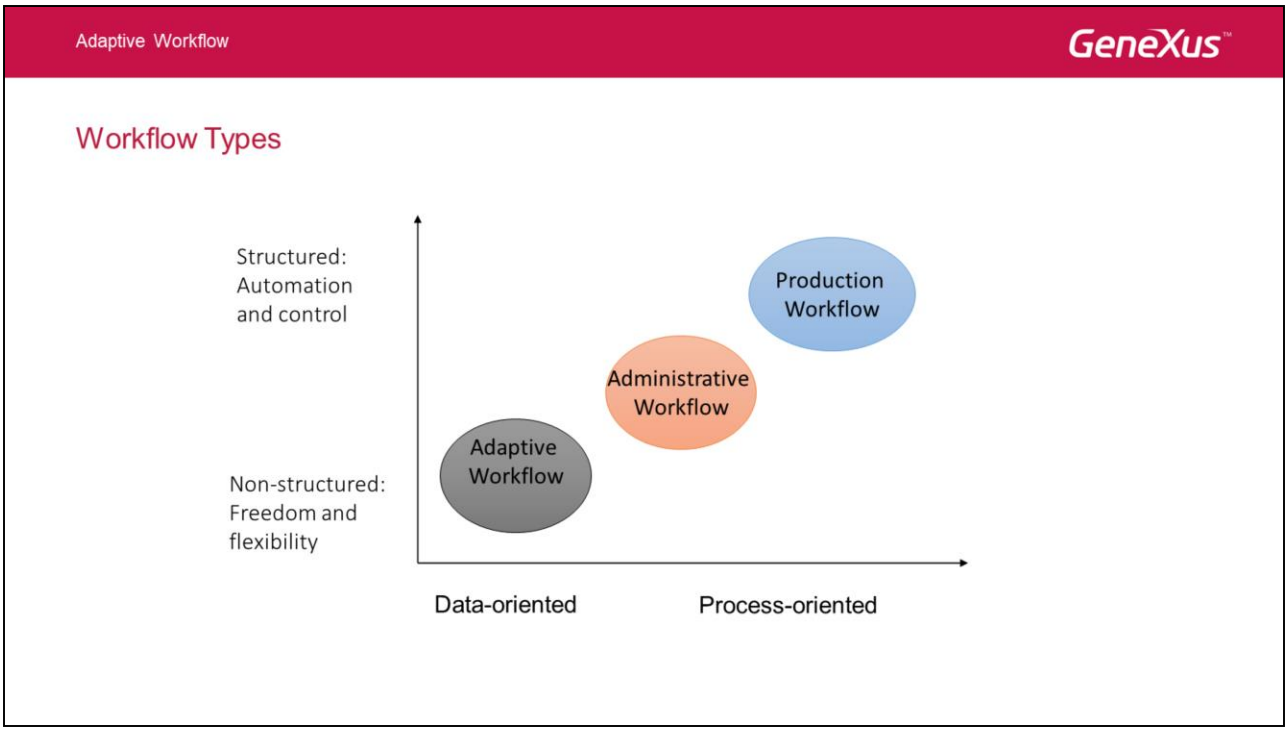
GeneXus

Team Performance

- Performance trend

Module 9

Adaptive Workflow



Tipos de Workflow

Que tipo de Workflow deveríamos usar? Isso depende do que quisermos fazer. Em companhias muito grandes são utilizados diferentes produtos de Workflow, de distintos fornecedores. Para ajudar a entender melhor o mercado, propomos uma segmentação dos distintos tipos de Workflow.

Workflow de Produção

O objetivo dos Workflows de Produção é administrar um grande número de tarefas semelhantes e otimizar a produtividade. Isso é feito automatizando as atividades tanto quanto seja possível, até o ponto em que a interação humana é requerida somente para cuidar das exceções.

Os eventos que requerem a contribuição das pessoas são minimizados. Esse tipo de Workflows é otimizado para atender a elevados níveis de qualidade e exatidão, executando tarefas bastante repetitivas; pode manejar processos muito complexos e estar intrinsecamente integrado a sistemas existentes. A tendência é embeber o componente de Workflow em grandes aplicativos nos quais seu papel é atuar como um motor de regras.

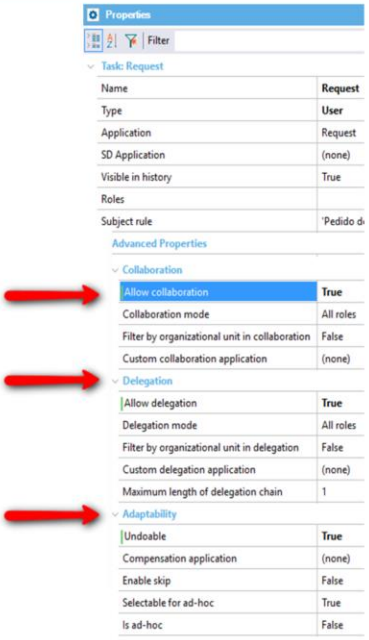
Workflow Administrativo

A funcionalidade mais importante de um sistema de Workflow administrativo é a facilidade para definir o processo. Habitualmente, há muitas instâncias sendo executadas onde há um envolvimento de um grande número de empregados. A definição de processos geralmente é feita usando formulários, e se a definição é muito complexa para o formulário, deve-se utilizar outro produto. Flexibilidade é mais importante que produtividade; esses sistemas manejam uma ou duas ordens a menos de magnitude no número de instâncias por hora comparando-as com os sistemas de Workflows de Produção.

GeneXus

Adaptability with GXflow

- Collaboration
- Delegation
- Undo a task
- Skip a task
- Ad-Hoc Processes



Como vimos em Modulos anteriores, GXflow permite modelar processos de negócios bem estruturados e orientados a processos mediante o uso de tarefas batch, também permite um híbrido entre processos estruturados e não estruturados e orientados a processos ou dados que se realizam mediante tarefas interativas e batch.

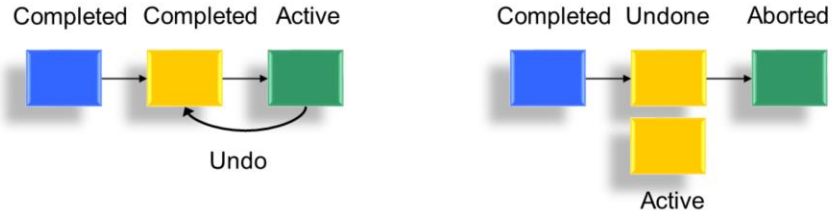
A seguir, são descritas algumas das funcionalidades que nos permitem modelar de forma adaptativa e flexível os processos de negócio com GXflow.

Undo a Task

- Undoability: ability to undo a task
- Compensation application: procedure executed upon aborting the process instance or undoing the task.

Adaptability

Undoable	True
Compensation application	(none)
Enable skip	False
Selectable for ad-hoc	True
Is ad-hoc	False



Esse padrão de Workflow permite-nos, no tempo de execução, voltar atrás algo em que já se fez e começar de novo o processo do ponto até onde se voltou.

Para as tarefas do tipo “User Task” ou “Script Task”, pode-se configurar a propriedade Undoable com o valor True. Isso permite que do “Manejador de processos” um usuário com o papel de Manager ou Administrador da opção Tarefas possa desfazer a execução daquelas tarefas que se completaram e cujas sucessoras não se completaram ainda. Para isso, é necessário ativar essa propriedade para aquelas tarefas que queremos que seja permitido fazer a ação e Desfazê-la.

Quando se configura essa propriedade com o valor True, habilita-se a propriedade “Compensation application”, que pode ser associada a um procedimento GeneXus que se executará no momento em que se faça a ação de Desfazer a tarefa. Eventualmente, esse procedimento pode chegar a recompor os dados que foram modificados quando se executou tal tarefa.

Esse procedimento deve ter os seguintes parâmetros:

parm(&WorkflowProcessDefinition, &WorkflowProcessInstance, &WorkflowWorkItem);

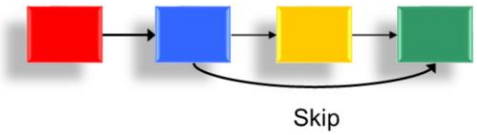
Cada um desses parâmetros estão definidos com um mesmo tipo cujo nome é igual ao nome das variáveis.

GeneXus

Skip a Task

Enable Skip

- It allows **skipping** the activity without executing it and continuing to the next one.



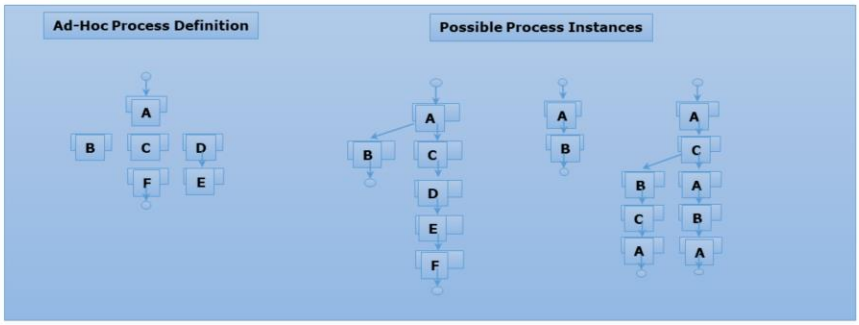
A propriedade Enable Skip está habilitada somente para as tarefas do tipo User, ou seja, as tarefas interativas.

O usuário com o papel Manager Administrator pode chegar a evitar a execução da tarefa. Para isso, da opção “Manejador de processos → Tarefas” é preciso selecionar a tarefa que se quer pular e pressionar a opção “Pular”. Isso deixa a tarefa em estado “pulado”, e o fluxo continua como se tivesse sido definido depois da tarefa pulada.

GeneXus

Ad-Hoc Processes

- Non-structured process whose Process Definition can be **adapted at runtime**.
- The user selects the next step

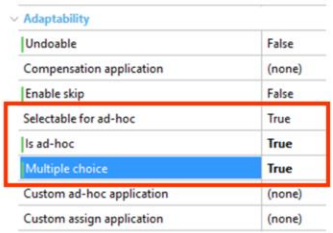


Os processos Ad-Hoc são aqueles nos quais há determinados fluxos que não podem ser determinados em sua etapa de modelagem, em vez disso, o fluxo pode variar dependendo do contexto em cada execução, e é usuário quem decide que caminho tomar e quem é selecionado para realizar a tarefa.

Teremos, portanto, uma definição das possíveis tarefas a realizar no processo, em alguns casos poderemos determinar até determinados fluxos, mas em outros casos não. No momento em que se vão gerando instâncias, a própria execução de tal processo vai definindo novos fluxos, dependendo das decisões que se tomam em cada passo. Para o caso mais extremo, poderíamos chegar a ter tantas definições do processo quanto instâncias que tenham sido executadas.

Ad-Hoc Task

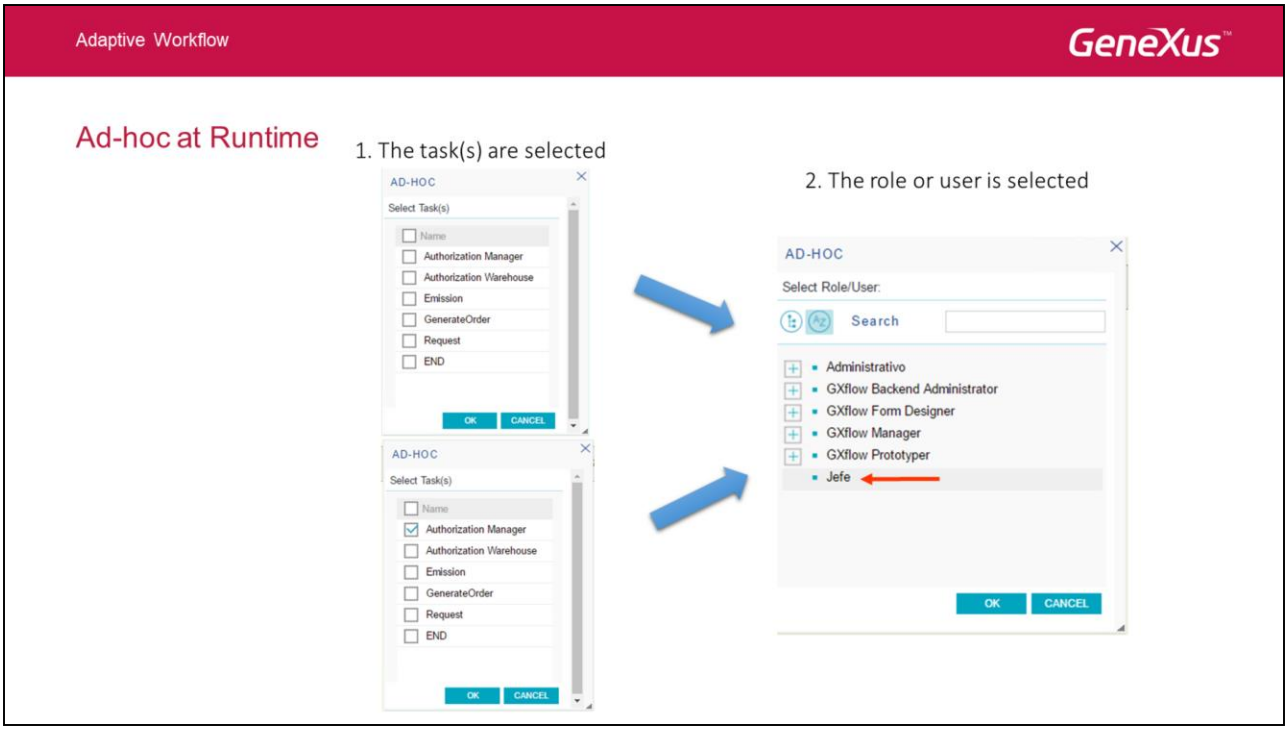
- Is Ad-Hoc: Indicates that the following task will be selected from a list of possible successor tasks.
- Multiple Choice: More than one successor task can be selected.
- Selectable for Ad-Hoc: The task will be available in the list of successor tasks of Ad-Hoc tasks.



A propriedade “Is Ad-Hoc” está disponível apenas para as tarefas interativas. Quando se configura essa propriedade com o valor True, estamos indicando que uma vez que se dê por finalizada tal tarefa, o usuário deverá selecionar a ou as tarefas sucessoras e também atribuí-las ao papel ou usuário que deseje. Isso permite sair da estrutura do fluxo que foi definido previamente, quando o processo foi modelado.

Pode-se selecionar quais tarefas dentro do processo queremos habilitar para essa seleção Ad-hoc com a propriedade “Selectable for Ad-Hoc”, dessa forma, pode-se evitar que o usuário possa pular de um passo a outro quando não tem sentido.

No que refere às propriedades do processo, existe uma propriedade Ad-Hoc que indica que todas as tarefas são Ad-Hoc, independentemente da propriedade da tarefa.

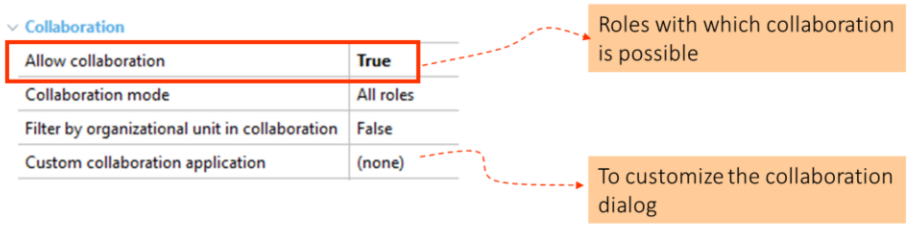


Uma vez que se pressione a opção Enviar sobre uma tarefa que a propriedade Is Ad-Hoc marcada com o valor True, mostra-se a caixa de diálogo para selecionar qual o quais serão as tarefas seguintes. Essa caixa de diálogo permite selecionar apenas uma tarefa ou mais de uma, dependendo se a propriedade Multiple Choice está configurada como o valor True.

Essas caixas de diálogo podem ser personalizadas utilizando as propriedades “Custom Ad-hoc application” e “ Custom Assign application” , assim, o desenvolvedor GeneXus conta com a possibilidade de personalizar a caixa de diálogo de seleção de tarefas e papéis/ usuário, em cujo caso, através do uso do tipo de dados de Workflow, terá que realizar a criação e atribuição dos Workitems que correspondam.

Collaboration

- Allow several users to work in parallel on the same task
- Task coordinator
  - Defines the list of collaborators
  - Is the only one who has the necessary permissions to complete the task



Há casos nos quais é preciso que vários usuários colaborem para realizar determinada tarefa, como quando é necessário realizar um relatório técnico em que vários colaboradores contribuem com suas ideias.

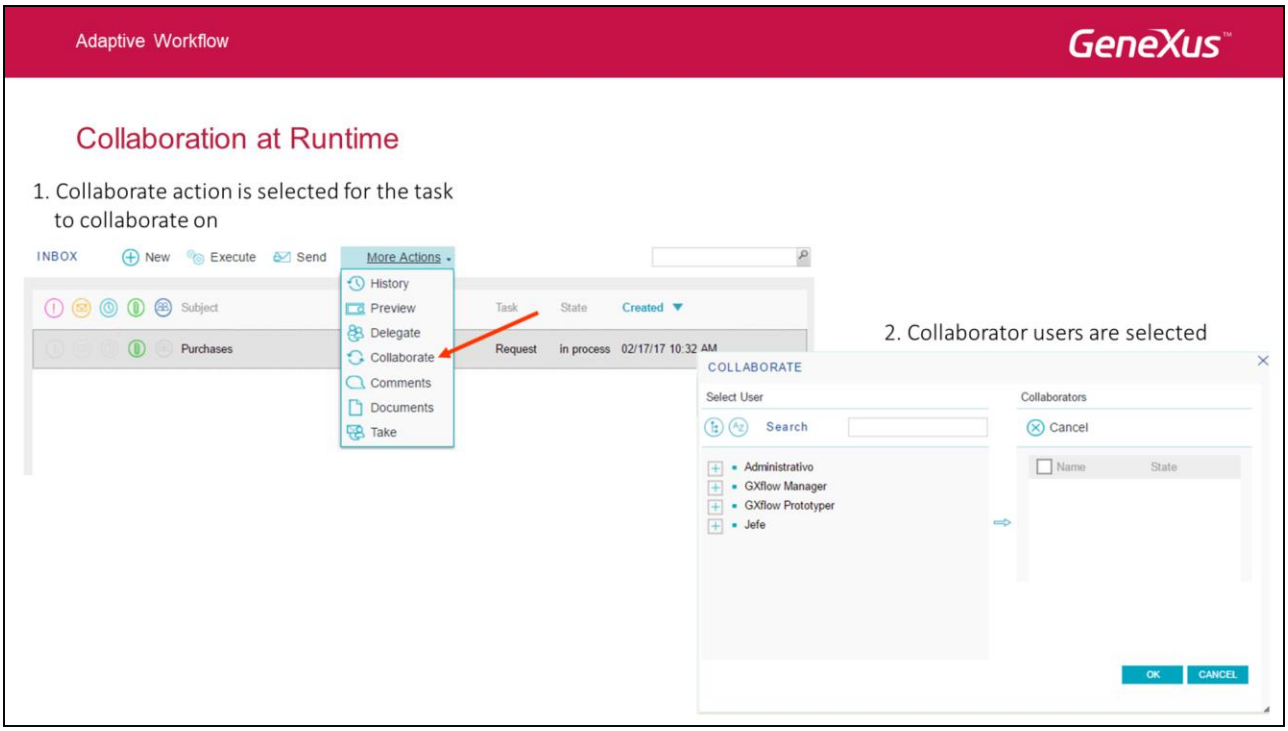
Para habilitar a possibilidade de que um usuário a quem uma tarefa foi atribuída possa pedir colaboração a outros usuários, é preciso configurar a propriedade Allow colaboration em True.

A propriedade Collaboration mode permite selecionar quem poderá colaborar, se usuários de qualquer papel (All Roles), usuários que tem seus papéis atribuídos à tarefa (Task Roles) ou usuários com determinados papéis (Select Roles).

O usuário tem a possibilidade de escolher na execução os que irão colaborar em tal tarefa e, em qualquer momento, pode cancelar a colaboração deles. O usuário que seleciona quem colaborará converte-se em coordenador da tarefa e só quando este dê a tarefa por terminada o fluxo seguirá adiante.

A propriedade “Filter by Organizational Unite in collaboration” permite indicar se os usuários com os quais se quer colaborar devem estar na mesma Unidade Organizacional que o usuário que está selecionando os colaboradores ou não. O valor True indica que se quer aplicar esse filtro.

A propriedade “Custom collaboration application” permite selecionar um webpanel que será invocado quando o usuário selecionar a ação “Colaborar” sobre a tarefa. Isso dá a possibilidade de personalizar a tela de colaboração e o desenvolvedor deve usar o Tipo de dados de Workflow para realizar a atribuição correspondente.



Quando uma tarefa tem a opção de Colaborar habilitada, o usuário a quem a tarefa é atribuída pode selecionar a ação Colaborar. Depois, lhe é apresentada a tela onde se pode selecionar os colaboradores.

Esse slide mostra, à esquerda, um quadro com os papéis e usuários, ao clicar duas vezes sobre o usuário, ele é acrescentado à lista de colaboradores que se mostra à direita. Caso se clique duas vezes sobre um papel, todos os usuários do papel selecionado são acrescentados à lista de colaboradores. Para cada um dos colaboradores, gera-se uma instância da mesma tarefa. O controle de concorrência sobre os dados do aplicativo manipulados em tal tarefa deve ser controlado pelo próprio aplicativo.

Da lista de colaboradores, pode-se selecionar um ou mais usuários e pressionar o botão Cancelar, o que permite cancelar a colaboração de determinados usuários.

Delegation

- Delegate
  - Send the task to another user
- Delegate with return
  - When the task is completed, it returns to the original user

A delegação é uma ferramenta útil quando queremos delegar a outro usuário a autoridade para fazer uma tarefa que nos foi encomendada.

Para habilitar a possibilidade de que em um usuário delegue a outros uma tarefa que a ele foi atribuída, é necessário configurar a propriedade Allow delegation em True.

A propriedade Delegation mode permite selecionar a quem poderá delegar, se aos usuários de qualquer papel (All Roles), aos usuários com os papéis aos quais a tarefa foi atribuída (Task Roles) ou os usuários com determinados papéis (Selected Roles).

O usuário tem então a possibilidade de escolher na execução a quem delegar tal tarefa e também pode especificar se quer que a tarefa regresse depois de concluída pelo usuário a quem a tarefa foi atribuída.

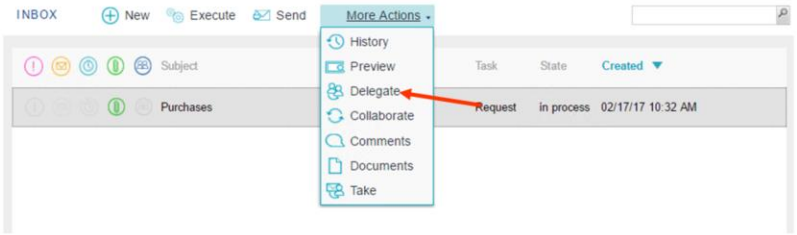
A propriedade “Filter by Organizational Unit in delegation” permite indicar se os usuários aos quais é possível delegar tarefas devem estar na mesma Unidade Organizacional que o usuário que está selecionando a quem vai delegar. O valor True indica que se quer aplicar esse filtro.

A propriedade “Custom delegation application” permite personalizar a caixa de diálogo invocada quando o usuário selecionar a ação “Delegar”. O desenvolvedor GeneXus permitirá a delegação correspondente utilizando os tipos de dados de Workflow.

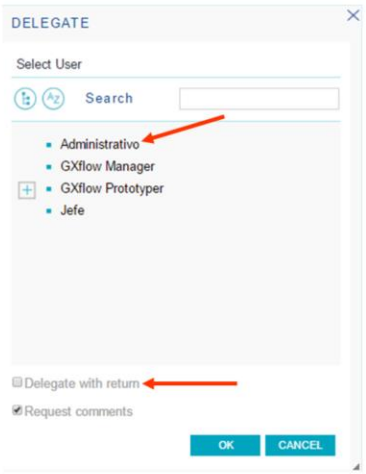
A propriedade “Maximum length of delegation chain” indica quantas vezes se deixará delegar a tarefa.

Delegation at Runtime

1. Delegate action is selected for the task to delegate



2. Select role or users



Quando uma tarefa tem a opção de Delegar habilitada, o usuário a quem a tarefa é atribuída pode selecionar a ação Delegar. Em seguida, a tela em que se pode selecionar a quem quer delegar a tarefa é apresentada.

Este slide mostra, à esquerda, um quadro com os papéis e usuários. Clicando-se duas vezes sobre um papel, se está delegando a todos os usuários dentro desse papel, isso é, a todos eles aparecerá a tarefa delegada e o primeiro que a tome poderá executá-la. Ao se clicar sobre um usuário, então se estará selecionando esse usuário em particular para delegar a tarefa.

A opção “Delegar com retorno” assegura ao usuário que está delegando que depois de concluída pelo usuário a quem se delegou, a tarefa voltará a ele.

A opção “Exigir comentários” permite que, ao pressionar OK nessa tela, se passe a outra em que o usuário pode acrescentar comentários que ajudem a entender o porque da delegação ou que forneça outra informação relacionada à tarefa.

Module 10

Monitoring Events

GeneXus

Events

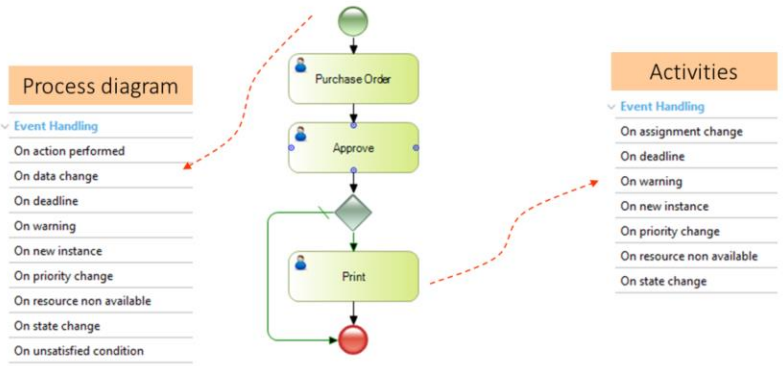
- During the operation of the Workflow engine, the following **internal events** take place:
  - Creation of a new work item
  - Creation of a process instance, etc.
- The event handling mechanism allows **running** a user-defined GeneXus procedure when the event takes place.

Esses eventos não devem ser confundidos com os eventos declarados nos diagramas (os “business events”).

GeneXus

Events

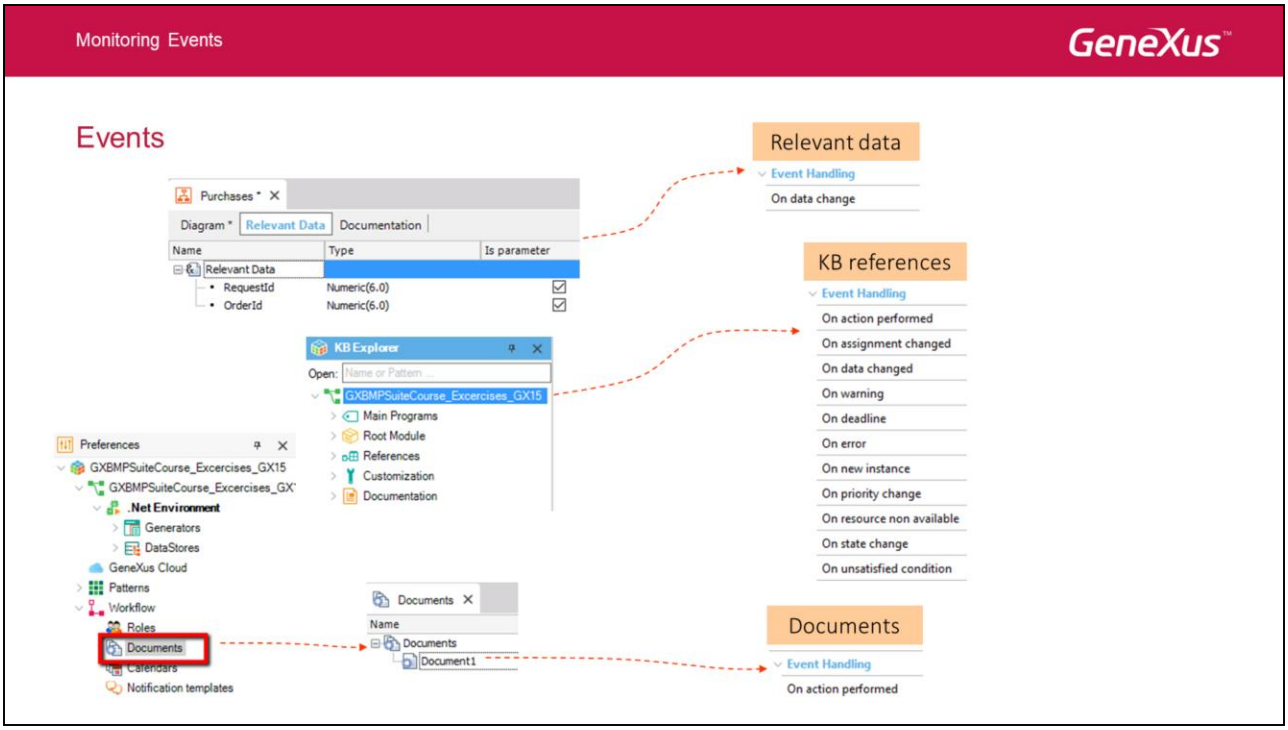
- Event handlers can be configured from different places:



Um procedimento manejador de eventos pode ser declarado em diferentes níveis:

- Nível de processo: declaram-se nas propriedades do business process diagram. Neste nível, o manejador invocará sempre que o evento estiver relacionado com o processo em questão. Por exemplo, se o manejador se associa ao evento “On new instance”, ele será invocado a cada vez que se criar um nova instância do processo e também a cada vez que criar um novo workitem de qualquer tarefa do processo.

- Nível de atividade: declaram-se nas propriedade das atividades. Neste nível, o manejador será invocado sempre que o evento estiver relacionado com a atividade em questão. Por exemplo, se o manejador se associar ao evento “On new instance”, ele será invocado somente quando se criar um workitem da atividade a qual está associado.



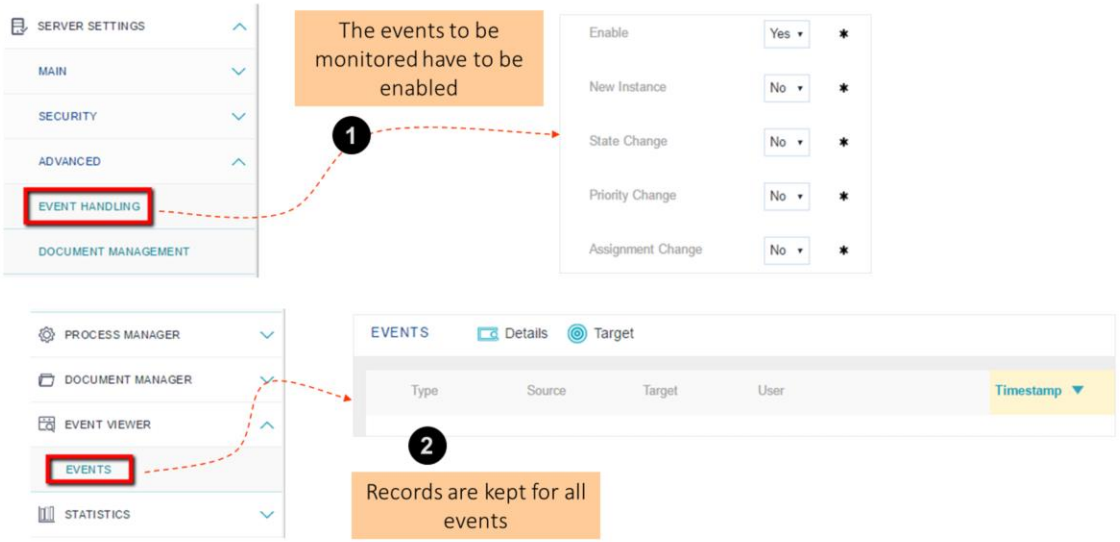
-Nível de dado relevante: declaram-se nas propriedades do dado relevante. Neste nível, têm-se apenas o evento “On data change”, que ocorre a cada vez que se modificar o valor de um dado relevante.

- Nível de documento: declaram-se nas propriedades do documento. Neste nível, há somente o evento “On action performed”, que ocorre a cada vez que realiza uma ação sobre o documento (criação, leitura, atualização ou exclusão).

- Nível environment: o manejador será invocado a cada vez que ocorrer o evento ao qual está associado sem importar se se originou em uma entidade particular (instância do processo, worktiem etc).

GeneXus

Events



GeneXus

Events

Event	Occurrence	Level
NewInstanceEvent	A new instance has been created	Process or task
DataChangeEvent	The value of a relevant data item has been changed	Relevant data item
StateChangeEvent	An instance status has been changed	Process or task
PriorityChangeEvent	An instance priority has been changed	Process or task
AssignmentChangeEvent	The assignment of an instance has been changed	Task

GeneXus

Events

Event	Occurrence	Level
WarningEvent	Instance warning	Process or task
DeadlineEvent	Instance deadline	Process or task
ErrorEvent	An error has occurred	All
ActionPerformedEvent	An action has been performed over a process document	Documents

Workflow Data Types

- **EventRepository**: Keeps a history log of all events occurred.
  - ListEvents(filter, events, error): Lists the events in the repository.

- **Event**: Represents a generic event.
  - TimeStamp: Creation date and time.
  - Type: Type of event.
  - User: User who caused the event.

GeneXus

Workflow Data Types

- **StateChangeEvent**: Change of status.
  - OldState: Previous status.
  - NewState: New status.
- **PriorityChangeEvent**: Change of priority.
  - OldPriority: Previous priority.
  - NewPriority: New priority.
- **DataChangeEvent**: Change of relevant data item.
  - DataName: Relevant data.
  - OldValue: Previous value.
  - NewValue: New value.
- **ActionPerformedEvent**: Execution of an action.
  - ActionPerformed: Performed action.
  - Context: Associated work item.

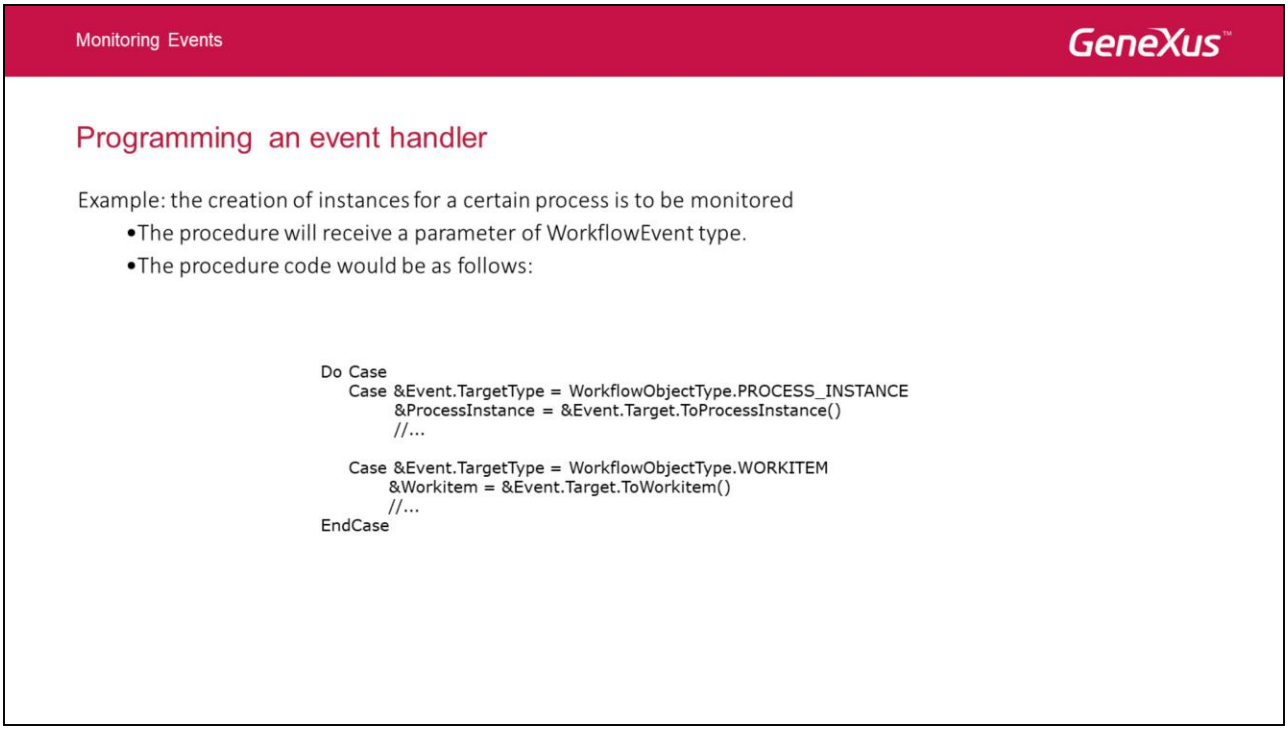
O tipo de dados base para trabalhar com eventos é o WorkflowEvent, os outros tipos estendem a este tipo base acrescentando métodos e propriedades específicas para o evento. Quando se quer trabalhar com um evento de alteração de um dado relevante, por exemplo, o tipo WorkflowDataChangeEvent acrescenta propriedades que permitem saber qual era o valor do dado relevante antes do evento (OldValue) e qual é o novo valor (NewValue).

Programming an event handler

- The procedure must have a parm rule of this type:

  Parm(in:&Event);

- &Event will be of WorkflowEvent type or one of its extensions, as applicable.



Quando se escreve um manejador de processos, é necessário considerar os distintos casos nos quais se invocará este manejador. Em particular, para o caso do exemplo, se associarmos ao manejador um processo, o mesmo se invocará a cada vez que:

- Sejam criadas instâncias do processo.
- Sejam criados workitems das tarefas do processo.

Para distinguir esses casos, o tipo WorkflowEvent fornece a propriedade que indica qual é o tipo da entidade de workflow sobre a qual ocorreu o evento. Os valores dessa propriedade estão baseados no domínio WorkflowObjectType e, no exemplo, são utilizados apenas os valores PROCESS_INSTANCE e WORKITEM, já que são as duas possibilidades existentes para o tipo de evento que se está manejando.

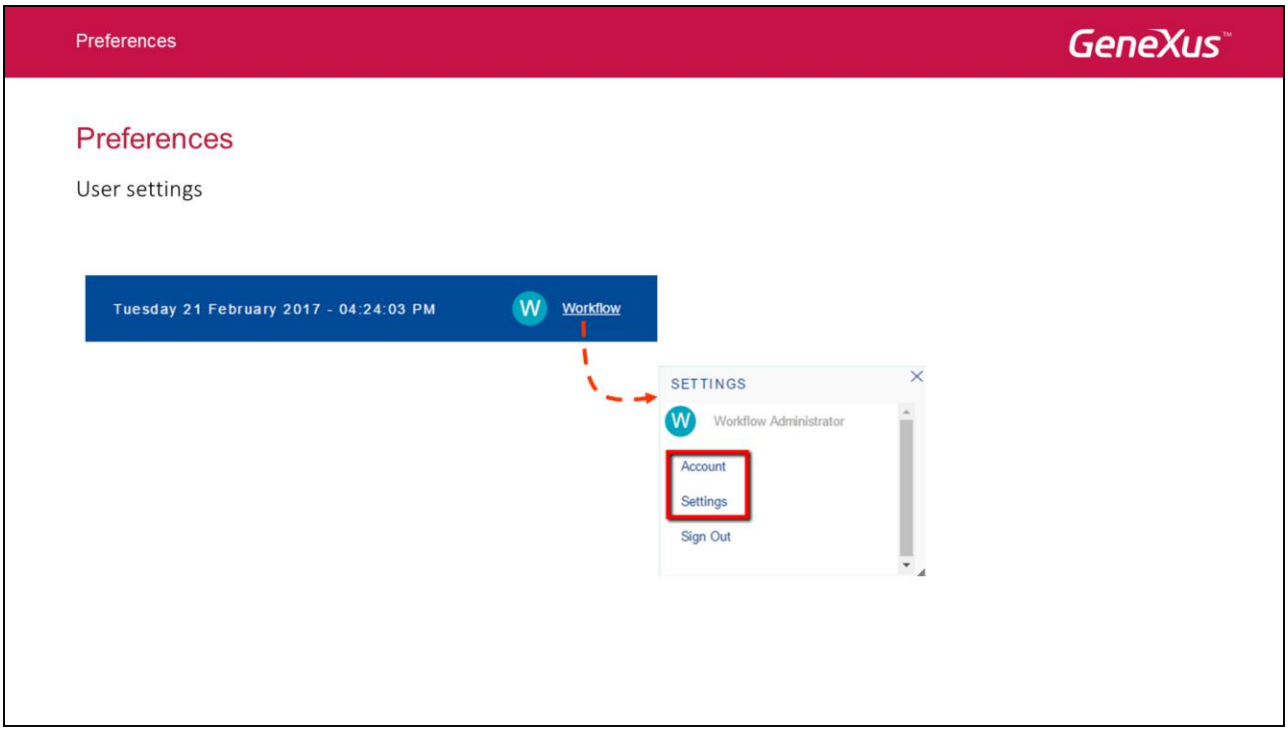
Outra propriedade importante do tipo de dados WorkflowEvent é a propriedade Target, que representa genericamente a entidade de workflow sobre a qual ocorre o evento. Uma vez que se sabe qual é o tipo concreto da entidade, é possível converter o valor da propriedade Target a seu valor concreto. Por exemplo, se a entidade é uma instância de processo:

&ProcessInstance = &Event.Target.ToProcessInstance() // &ProcessInstance é do tipo WorkflowProcessInstance

Logo, todos os métodos que o tipo WorkflowProcessInstance fornece para operar sobre a instância de processo podem ser utilizados.

Module 11

Preferences



Preferências de Usuário

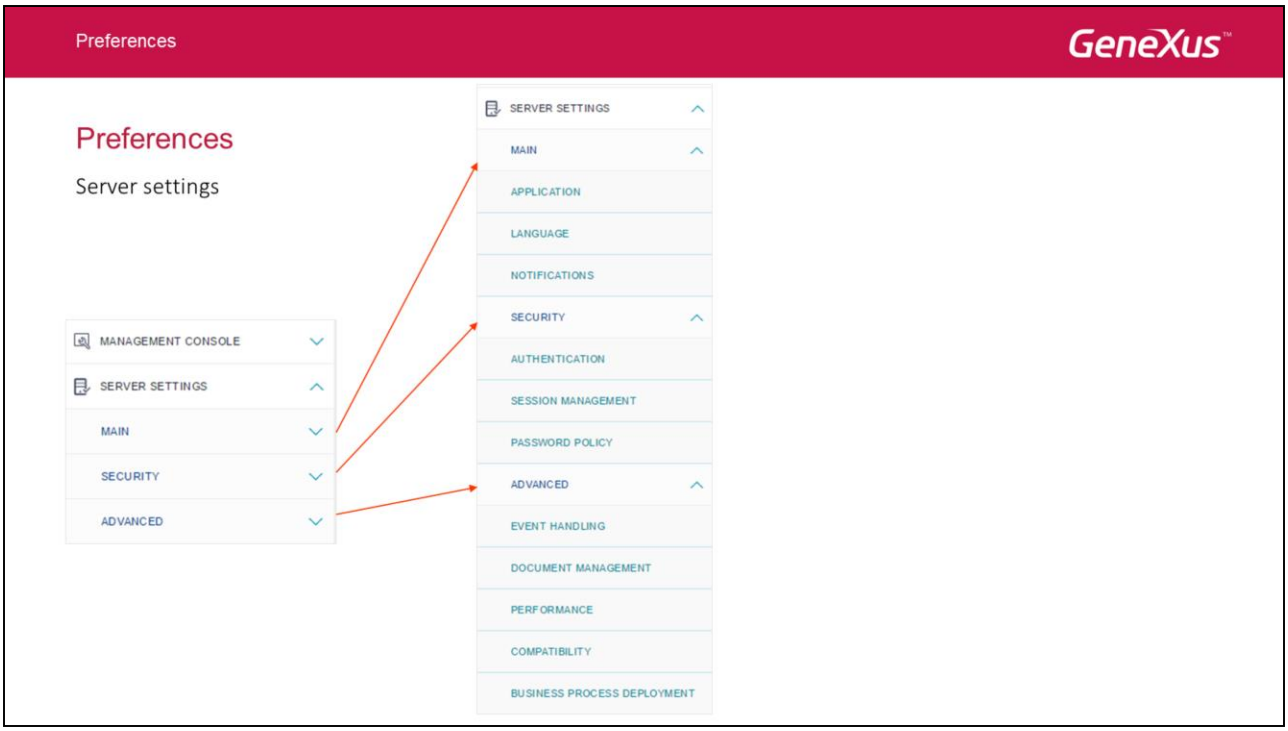
São preferências que cada um dos usuários do sistema pode ver

Conta

Permite ao usuário configurar informações de sua conta (nome, endereço de e-mail, se está de licença)

Configuração

Permite configurar aspectos da interface gráfica (quantidade de linhas por página, número de botões, habilitar/ desabilitar duplo clique, habilitar/ desabilitar recepção de notificações, atualizar e executar automaticamente a caixa de entrada)



Preferências do Servidor (mudei de lugar)

São preferências globais do servidor wf. Só podem ser vistas/modificadas por um usuário com o papel de administrador.

Aplicativo

Conjunto de preferências que se aplicam no tempo de execução (executar as tarefas em um nova janela, ajustar automaticamente o tamanho dos formulários ou definir um tamanho padrão, encriptar parâmetros da url)

Linguagem

Permite configurar o idioma pré-determinado do aplicativo Cliente (embora cada usuário possa escolher depois seu idioma de forma independente)

Notificações

Permite configurar a conta de e-mail para o envio das notificações. Requere-se de um servidor SMTP.

Autenticação

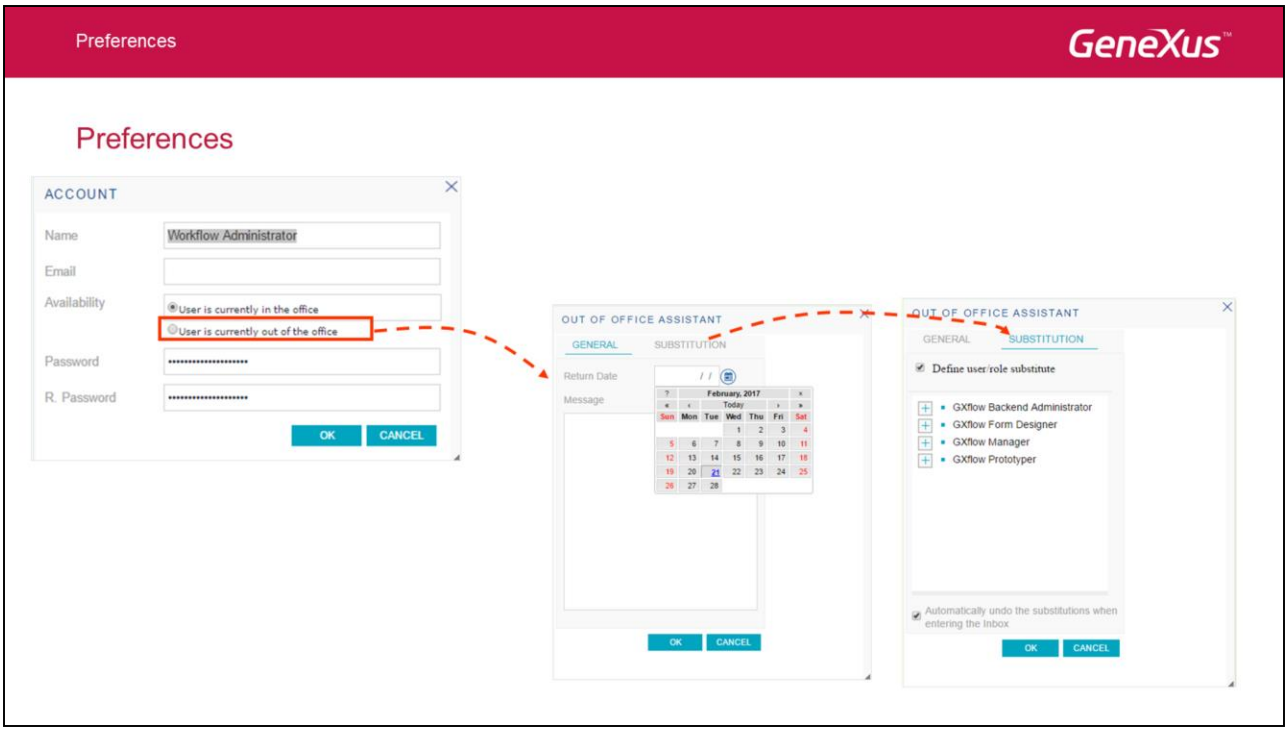
Esquema que se utilizará para autenticar os usuários, seus valores são: GXflow, Windows, Externo e GAM, o valor default é GXflow.

Manejo de Sessões

Permite configurar a duração das sessões dos usuários.

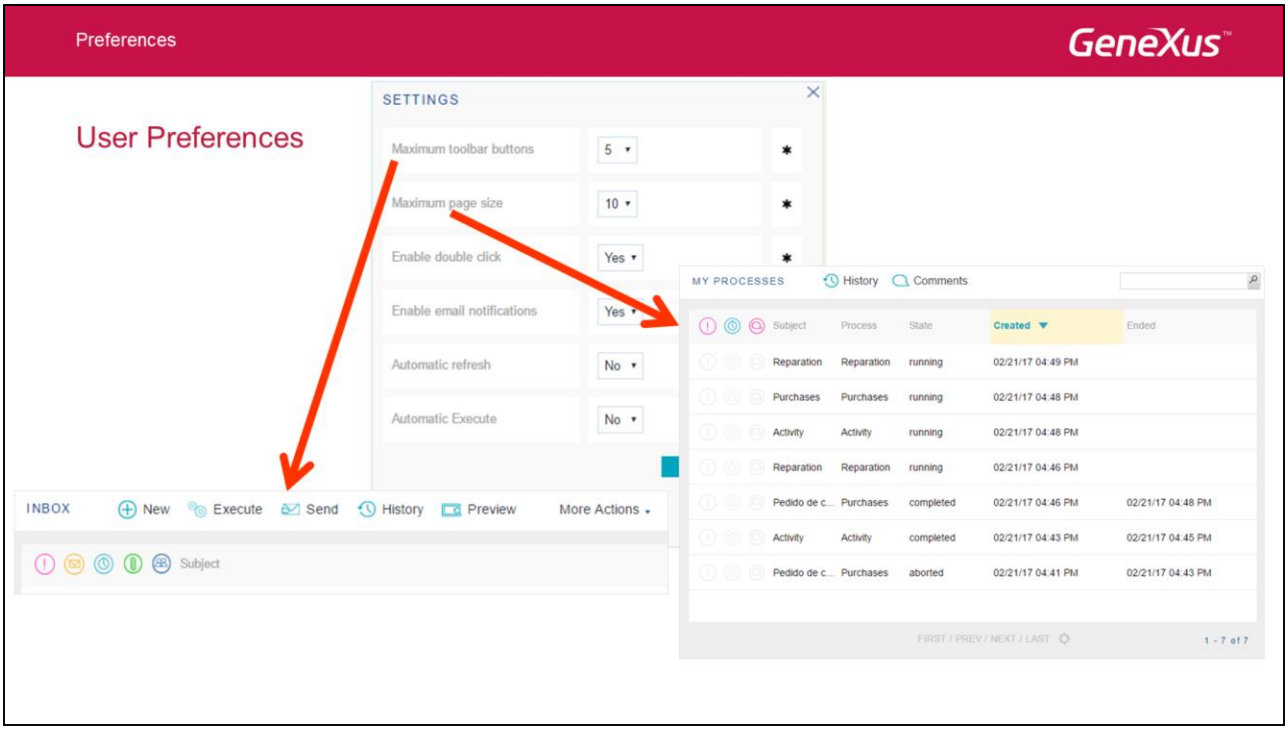
Política de senhas

Permite configurar o vencimento das senhas e regras de GXflow (duração, mínimo de caracteres, uso de maiúsculas e minúsculas, caracteres numéricos etc) ou customizar suas próprias regras.



Si elegimos que el usuario está fuera de la oficina y presionamos OK, se abre el asistente del Out of Office.

Podemos elegir la fecha de regreso, el mensaje que se desea que aparezca y qué usuario será el sustituto.



As preferências por usuário são:

Quantidade máxima de botões na barra de ferramentas

Define a quantidade de botões da barra de ferramentas a mostrar. Os valores são de 1 a 10, incluindo o botão Mais ações.

Linhas por página

Indica o número de linhas que serão mostradas por página.

Habilitar duplo clique

Permite habilitar ou não a execução de uma tarefa mediante duplo clique sobre sua descrição na caixa de entrada. Seus valores são: True: habilita a execução da tarefa (valor default) ou False:não habilita a execução da tarefa.

Habilitar notificações por e-mail

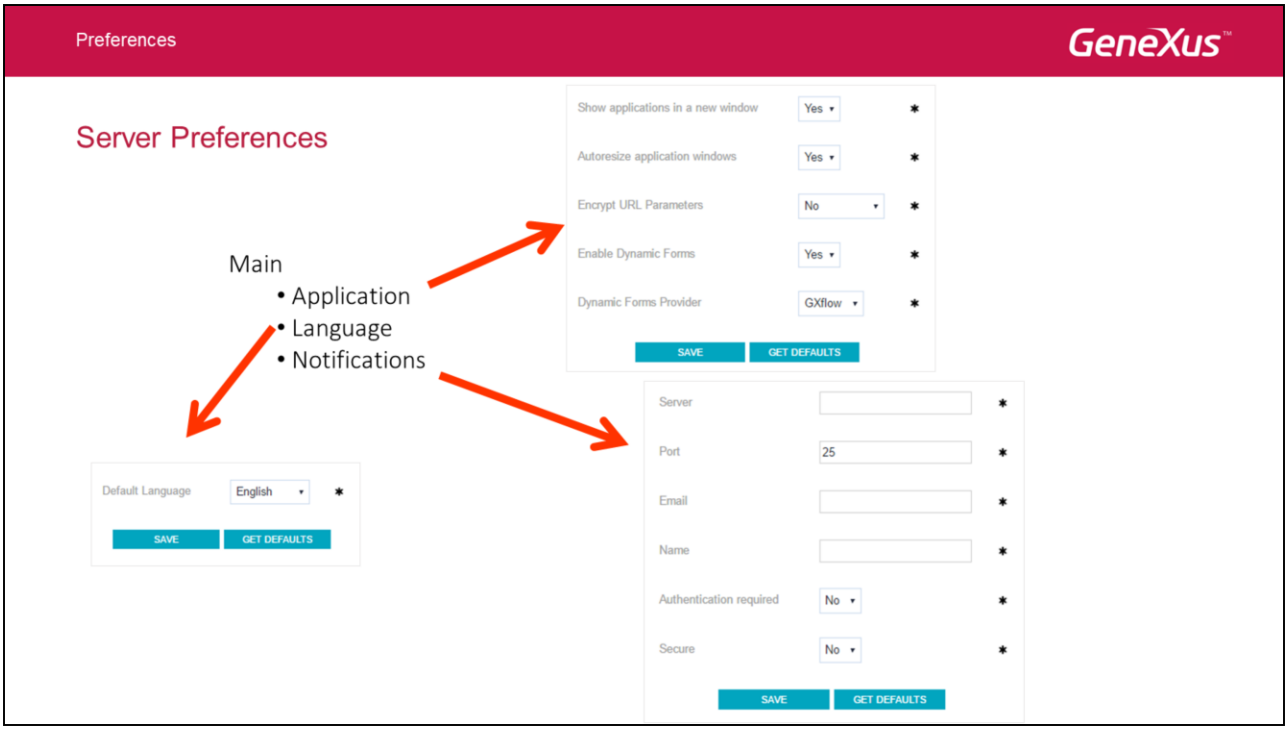
Permite configurar a conta de e-mail para o envio das notificações. Requere-se de um servidor SMTP.

Atualizar automaticamente

Permite que a caixa de entrada se atualize automaticamente. Ao ativar essa propriedade, pode-se escolher o tempo de espera para atualizar. Os valores possíveis são 30, 60 ou 90 segundos e 2, 3 ou 5 minutos.

Executar automaticamente

Quando um usuário inicia um nova instância do processo, a tarefa inicial é atribuída e executada para tal usuário de forma automática caso essa propriedade esteja habilitada.



Preferências Main:

1. Application

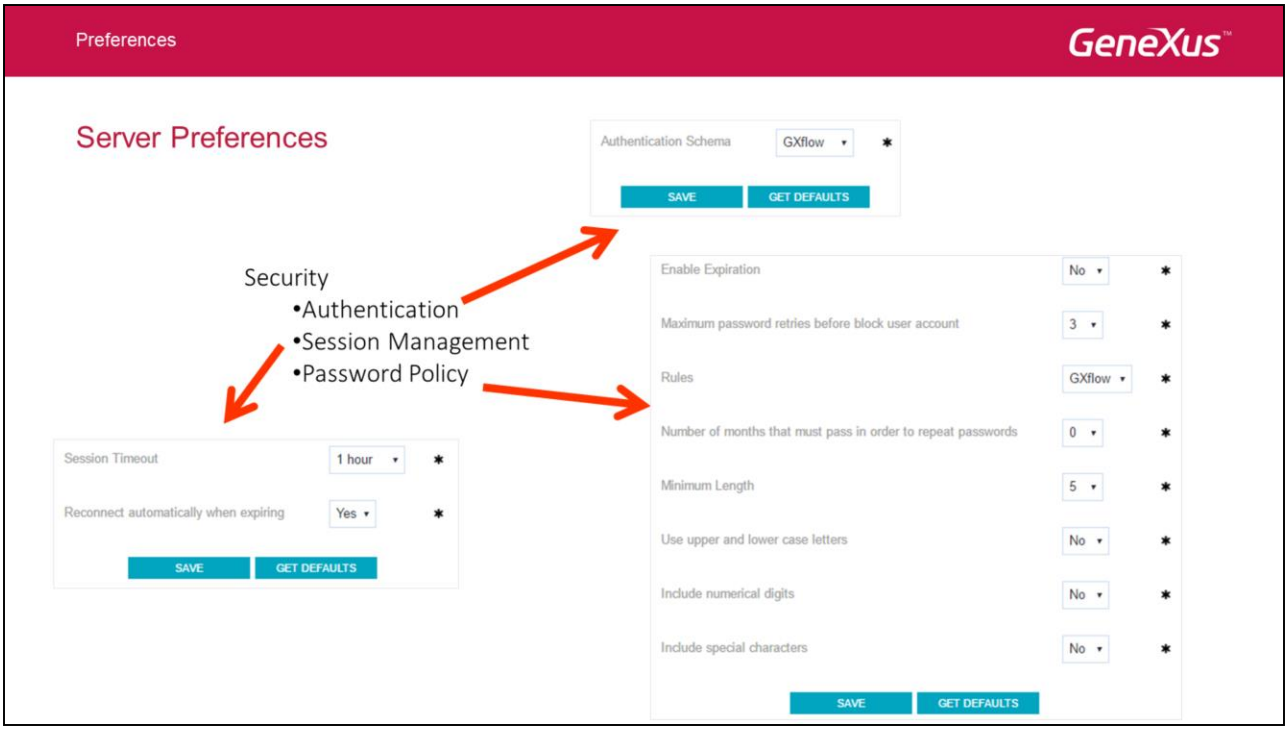
É dada a possibilidade de configurar que a tarefa se execute em uma nova janela, ajuste automaticamente seu tamanho e possa encriptar os parâmetros da URL.

2. Language

Permite configurar o idioma pré-determinado do aplicativo Cliente.

3. Notifications

Configuram-se os dados do servidor SMTP para que o envio de e-mails seja permitido.



Preferências do Servidor

1. Authentication

Os esquemas para autenticar os usuários são: GXflow, Windows e Externo, o valor default é GXflow.

GXflow: Esquema padrão de autenticação. A verificação realiza-se com base nos usuários definidos no cliente.

Windows: Utiliza-se o usuário do Sistema Operativo Windows.

Externo: Cria-se uma aplicativo personalizado para verificar os usuários.

2. Session Management

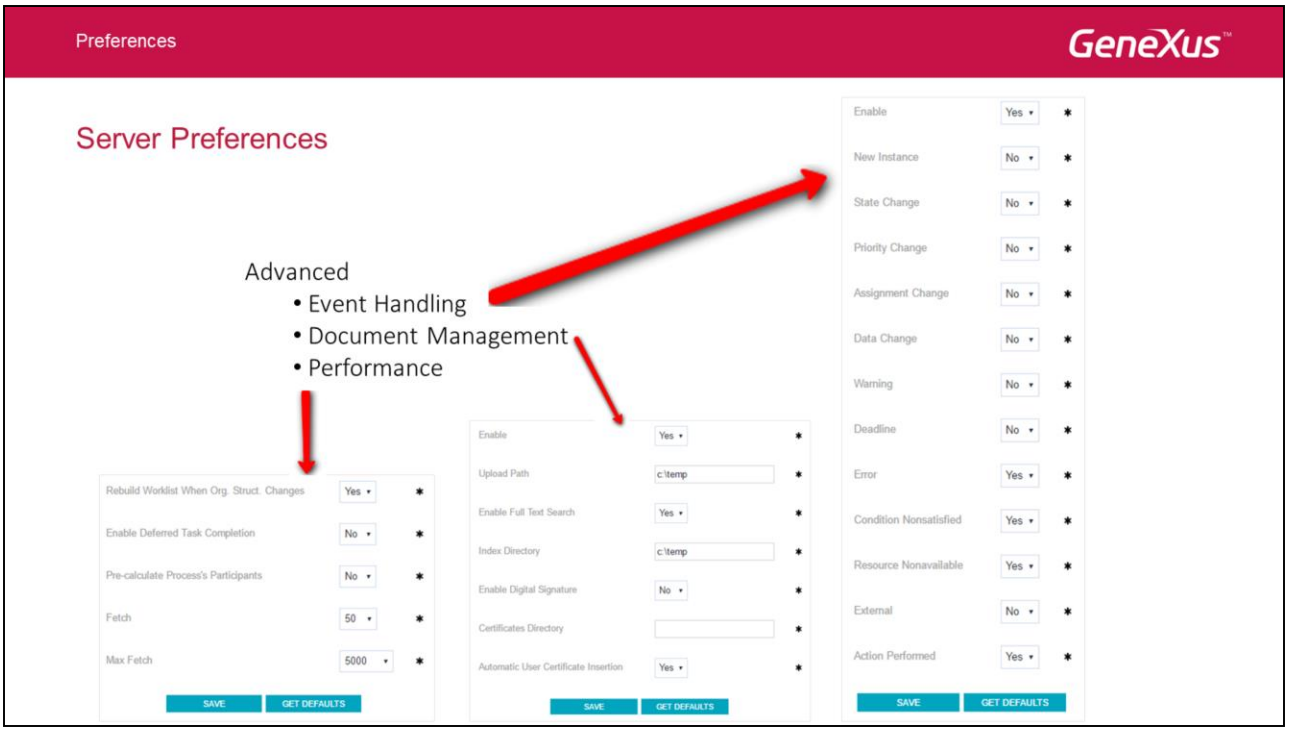
Permite configurar a duração das sessões dos usuários. Os usuários a configurar são:

Tempo de espera da sessão: Permite especificar a duração da sessão de um usuário conectado. Pode ser 1, 2, 8 ou 24 horas.

Voltar a conectar automaticamente quando expira: Permite especificar que quando a sessão do usuário registrado expira, se conecte automaticamente para evitar uma nova autenticação.

3. Password Policy

Permite configurar o vencimento das senhas e regras de GXflow (duração, mínimo de caracteres, uso de maiúsculas e minúsculas, caracteres numéricos etc) ou customizar suas próprias regras.



Avançado:

Event Handling

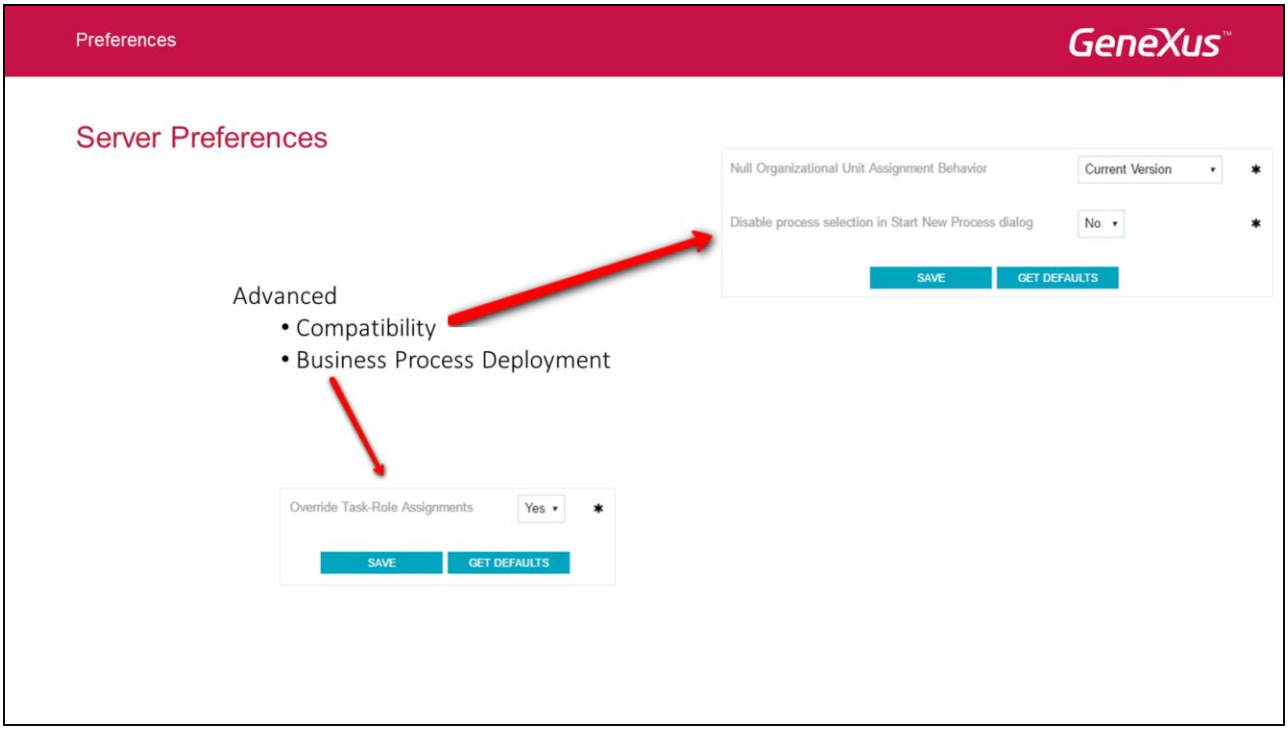
Permite configurar o nível de granularidade na geração de eventos de auditoria. Pode-se habilitar/ desabilitar num nível geral ou para cada tipo de evento.

Manejo de Documentos

Permite habilitar e configurar o manejo de documentos. Pode-se configurar a rota de publicação, habilitar/ desabilitar a busca full-text e o diretório onde se armazenará o índice, habilitar o uso da assinatura digital etc.

Performance

Um conjunto de preferências que permite modificar o momento em que o motor wf efetua determinadas ações que afetam a performance da aplicação. A configuração pré-determinada é a que consideramos melhor, tomando como referência o quociente Usabilidade/Performance. No entanto, em determinados cenários, pode ser necessária sua reconfiguração. Por exemplo, se num cenário real a relação entre usuários, papéis e unidades organizacionais é muito dinâmica (muda permanentemente), provavelmente tenha que mudar para que não se reconstrua a worklist diante dessas alterações e agendar uma reconstrução geral a cada x tempo.



Compatibilidade

Permite manter a compatibilidade com bases de conhecimento migradas de GXflow 9.0 U1.

Valores:

Current Version: O usuário **não** pode ver as instâncias de processos com unidade organizacionais não atribuídas a ele.

Version 9.0 and prior: O usuário poder ver as instâncias de processos com unidade organizacionais não atribuídas a ele.

Automaticamente, o valor é **Current Version**.

GeneXus 15

Module 12

Deployment

Neste Modulo, são descritos os passos necessários para implantar um aplicativo baseado em processos de negócios.

GeneXus

Deployment

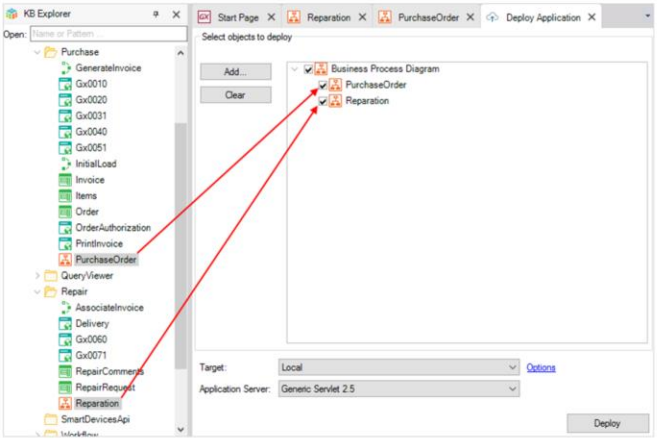
The complete process consists of deployment of three components:

- The binary files of the business process diagrams and their dependencys
- The database files and metadata of process definitions
- The protection

El proceso completo consiste de tres components: los archivos binaries, la base de datos / metadata y la protección.

Deployment of binaries application files

1) Build > Deploy application
2) Drag every Business Process Diagram from KBExplorer. Dependencys are automatically included.
3) Set your preferred options and press Deploy



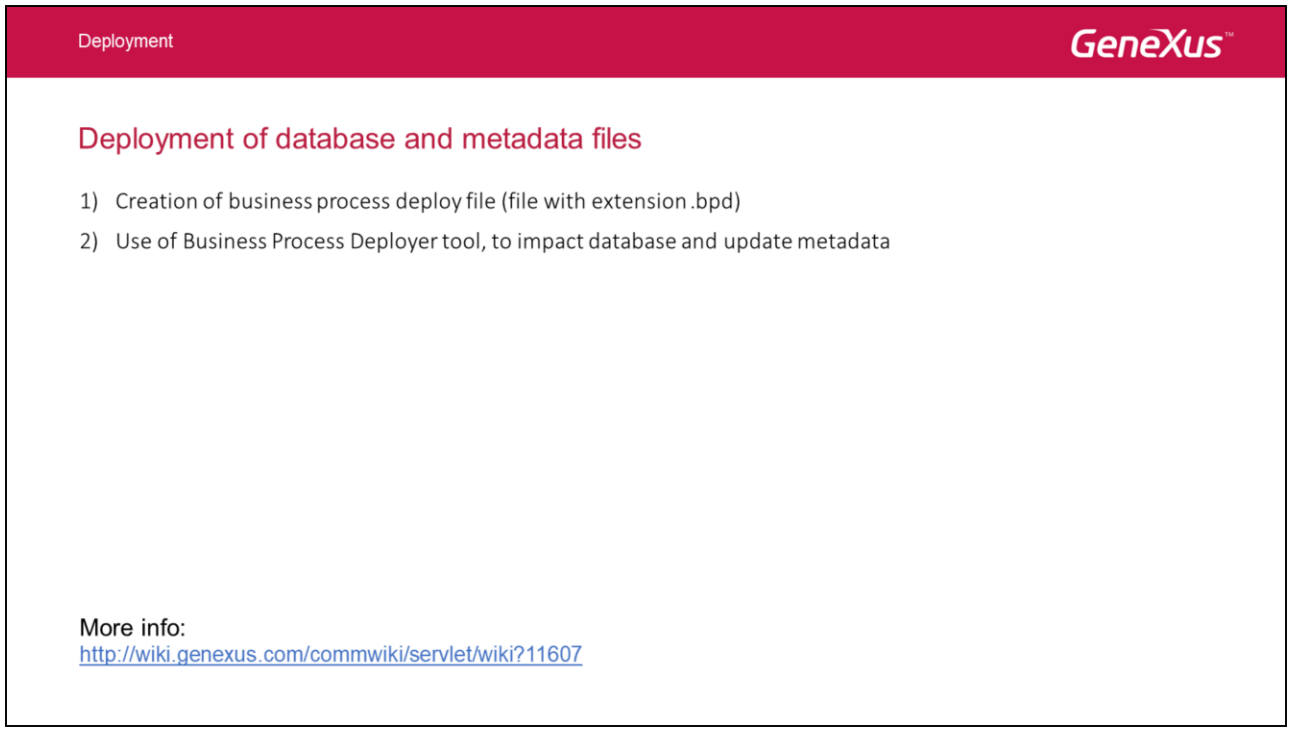
More info:
http://wiki.genexus.com/commwiki/servlet/wiki?32092

En esta etapa se distribuyen los archivos binarios de los diagramas de procesos y de los objetos GeneXus asociados a los mismos.

GeneXus prove un mecanismo muy simple para empaquetar los archivos generados y desplegar la aplicación.

Se siguen los siguientes pasos:

1) Build > Deploy application. Esto desplegará una ventana contextual en el IDE
2) Arrastrar cada objeto Business Process Diagram desde el KBExplorer y soltarlo en la aventana de “objetos a desplegar”. No es necesario incluir cada dependencia del diagram (como WebPanels o procedimientos), ya que GeneXus los incluirá automáticamente sin mostrarlo en la pantalla.
3) Setear las opciones que se prefieran y presionar Deploy



Nesta etapa se disponibilizan as definições do banco de dados e os metadados dos processos que se modelam em GeneXus para que possam ser executados pelo motor de workflow.

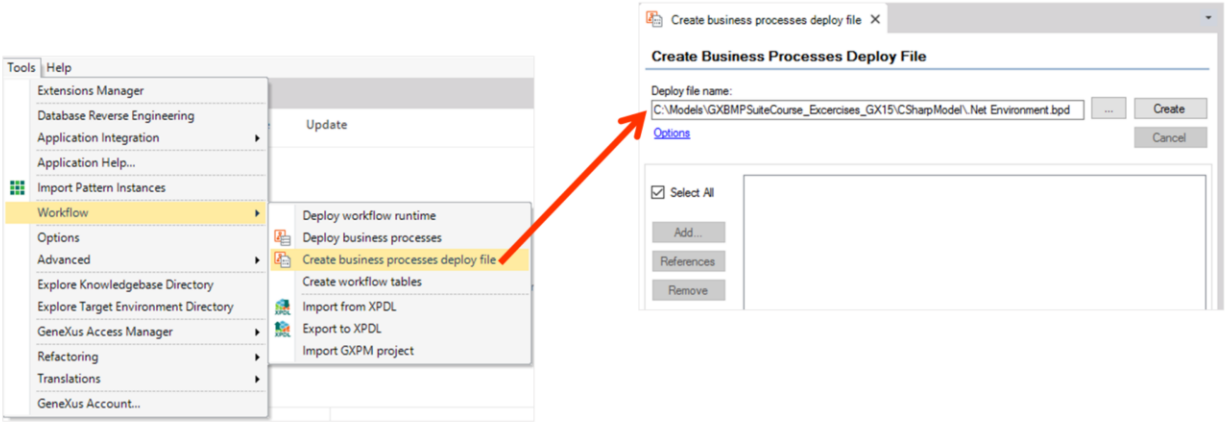
É importante esclarecer que essa operação pode exigir realizar reorganizações no banco de dados. Na primeira vez que se fizer um deploy de processos, serão criadas as tabelas de workflow. Nos seguintes deploys, também pode ser necessário executar reorganizações de upgrade quando ao utilizar um novo upgrade ou versão de Gxflow. Todas as organizações se executarão automaticamente solicitando a autorização do usuário previamente.

La herramienta Business Process Deployer hace posible el impacto de la base de datos de producción incluyendo a los nuevos procesos de negocios o modificando los existentes.

Los procesos que están corriendo mientras se hace el deploy no serán afectados, solo los procesos creados luego del impacto reflejarán los cambios.

Creation of Business Process Deployment file

1) Tools Menu > Workflow > Create business process deploy file (file with extension .bpd)



A primeira coisa que se deve criar é o arquivo de deploy de processos de GeneXus mediante a opção "Tools->Workflow-> Create business processes deploy file". O resultado dessa operação é um arquivo que contém o seguinte:

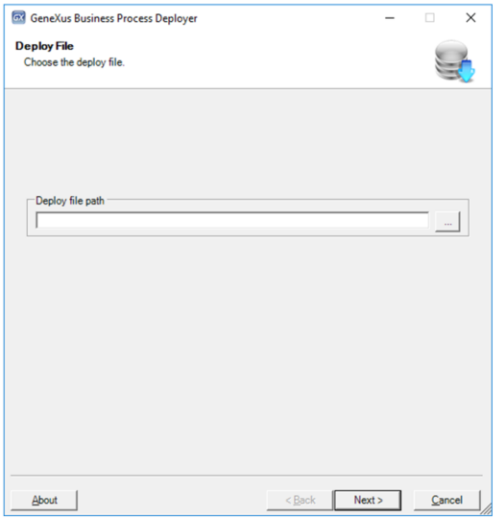
- A definição dos processos modelados em GeneXus.
- Todas as organizações que poderão ser exigidas antes de realizar o deploy de processos (criação de tabelas e reorganizações de upgrade). Essas reorganizações já estão geradas e compiladas para a plataforma com a qual se está trabalhando em GeneXus. Isso faz com que o arquivo de deploy seja da plataforma com a qual se está trabalhando. Se, por exemplo, este arquivo é criado para o ambiente .NET SQLServer, não poderá ser utilizado para realizar o deploy em uma base de dados MySQL.

Use of Business Process Deployer to impact database and update metadata

2) Open the BPDeployer and import the generated file.

Business Process Deployer:

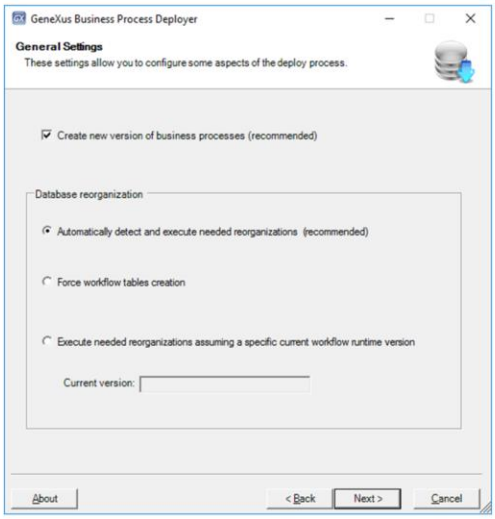
- Tool to perform process deployment.
- Uses the deployment file created in GeneXus as input.
- No licenses are required.



Uma vez que o arquivo de deploy de processos está criado para executar o deploy no banco de dados de produção, utiliza-se o GeneXus Business Process Deployer. Essa ferramenta está incluída no pacote de instalação do ambiente de produção de Gxflow e não requer licenças. Tal ferramenta pode ser executada no modo wizard (automaticamente) ou por linha de comandos.

Use of Business Process Deployer to impact database and update metadata

- Create new version of business processes
- Automatically detect and execute necessary reorganizations



Selecionado o arquivo de deploy, devem ser configuradas as seguintes opções:

Create new version of business processes (recommended)

Esta opção permite estabelecer que os processos incluídos no arquivo de deploy serão levados em conta a partir de novas instâncias de processo. As instâncias de processo que estão atualmente em execução seguirão baseando-se na definição de processo a partir da qual foram criadas.

Automatically detect and execute needed reorganizations (recommended)

O arquivo de deploy gerado em GeneXus contém a versão com a qual foram criados esses processos, e o banco de dados de produção tem sua própria versão. Essa opção permite especificar que automaticamente sejam realizadas reorganizações para migrar o banco de dados de produção para a versão indicada nos diagramas.

Force workflow tables creation

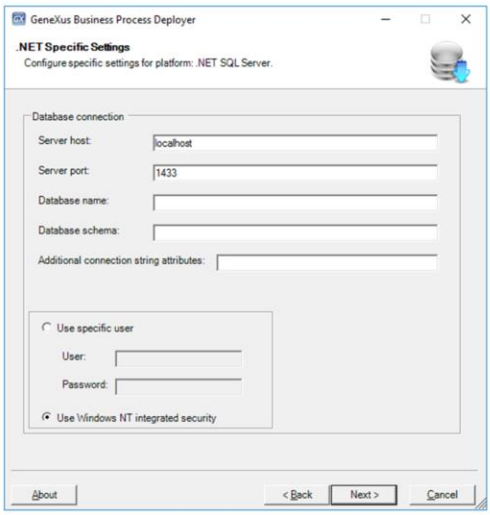
Essa opção permite forçar a criação das tabelas de workflow.

Execute needed reorganizations assuming a specific current workflow runtime version

Esta opção permite especificar determinada versão, dessa forma, ignora-se a versão do banco de dados de produção.

Use of Business Process Deployer to impact database and update metadata

- Configuring the production database

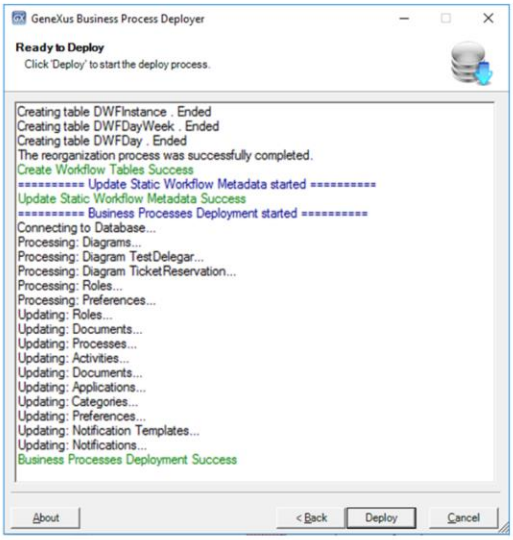


O passo seguinte é configurar o banco de dados de produção.

GeneXus

Use of Business Process Deployer to impact database and update metadata

- Reorganizing the database
- Process Deployment



Por último, fazemos o deploy dos processos e, na primeira vez, serão criadas as tabelas de workflow. Depois, à medida que nossos processos sejam atualizados para novas versões de GeneXus, serão realizadas reorganizações para atualizar as tabelas de workflow no banco de dados de produção.

GeneXus

Protection settings (for Windows platform)

1. Install **GXflow Production Environment**.
2. Go to Start > All Programs > GXflow GeneXus 15 > License Manager and install licenses.
3. Copy the file "protect.ini" file that resides in C:/Program Files (x86)/GeneXus/GXflow/GXflowGx15/LicMgr to:
   - NET: application´s bin directory (replace the existing one).
   - Java: Tomcat´s lib directory (by default in C:/Program Files/Apache Software Foundation/Tomcat X.Y/lib)
   - Java: JBoss's bin directory or the folder associated with the startup process.
4. Install GeneXus Protection Server.
5. (Only for Java) Copy and extract this file to the Tomcat´s lib directory (by default in C:/Program Files/Apache Software Foundation/Tomcat X.Y/lib).
6. Go to Start > All Programs > GeneXus 15 Business Process Deployer and deploy the processes.

For other platforms, go to :
http://wiki.genexus.com/commwiki/servlet/wiki?19848

More info about this subject:
http://wiki.genexus.com/commwiki/servlet/wiki?11607

Cuando la aplicación está pronta para ser desplegada en el servidor, la protección debe ser actualizada con ella.

En esta sección se dan los pasos cuando la plataforma es Windows. Si usan otra plataforma como Linux o IBM iSeries (AS400), sigan los links en pantalla.

GeneXus

GXflow Production Environment

It consists of installing the following:

- GXflow License Manager
- Business Process Diagram Deployer

Steps:

1. Download the setup file from Download Center
2. Execute *Setup.exe* file at production server.
3. Choose the target path for the application and click "Install" to initialize the process.
4. Once the application is installed, close the dialog.
   It is recommendable to restart your computer once installation process has finished

More info:
http://wiki.genexus.com/commwiki/servlet/wiki?19849

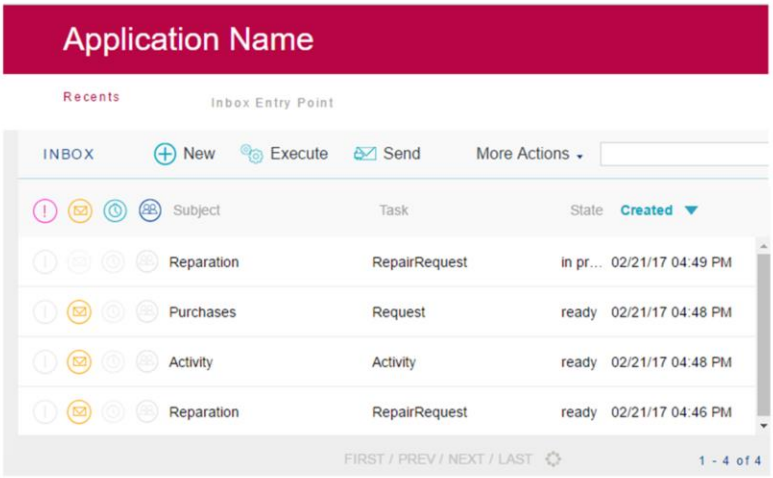
El ambiente de producción de GXflow comprende todo lo necesario para poder ejecutar una aplicación que utilice procesos de negocio.

Esencialmente está compuesto por el License manager de GXflow que permite administrar las licencias de los usuarios que se conectan o acceden al cliente GXflow y el Business Process Deployer que permite impactar la base de datos con los cambios en los procesos.

Module 13

GXflow Client Integration & Customization

GXflow Entry Point

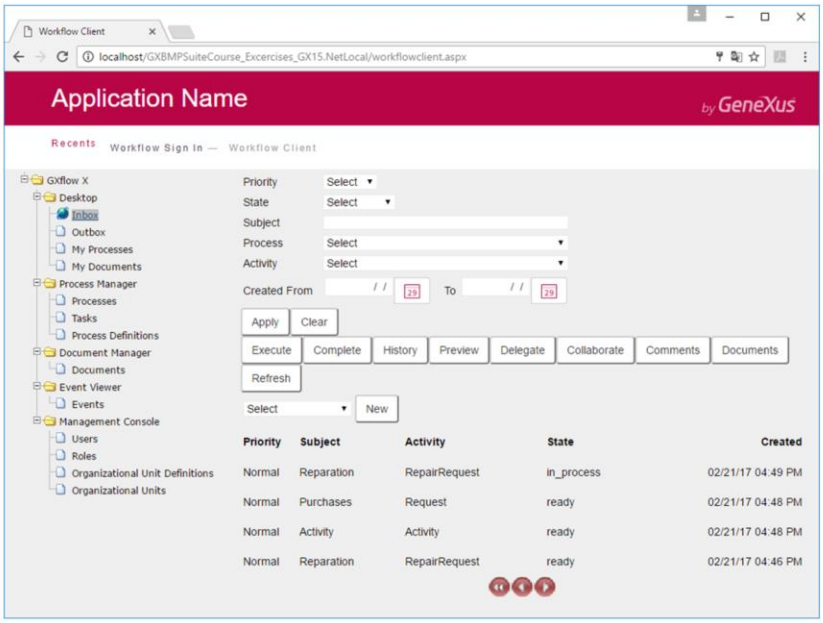


More info: http://wiki.genexus.com/commwiki/servlet/wiki?10721

GXflow Entry Point é um **user control** que é distribuído com GeneXus (<diretório de GeneXus>/Packages/Gxpm/Extra/GXflowEntryPoint) e é instalado igualmente ao resto dos users controls do GeneXus.

Permite utilizar os distintos componentes do cliente GXflow (Caixa de Entrada, Caixa de Saída, Processos, etc.) como em aplicativos separados, por exemplo em um web panel como é mostrado na imagem.

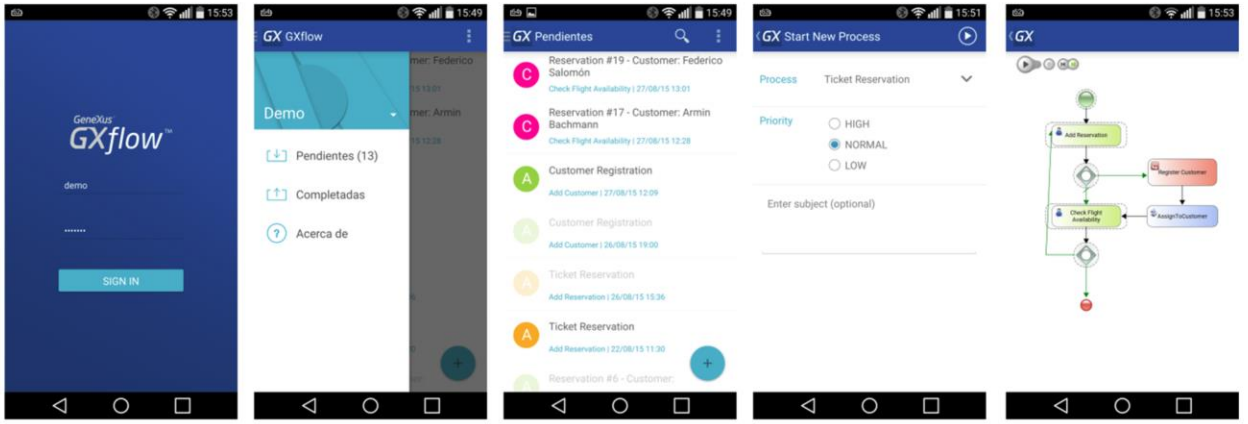
GXflow Custom Client



GXflow Custom Client é um conjunto de objetos (xpz) que utilizam a API e implementam o cliente GXflow. Este xpz é aberto, o usuário pode consolidá-lo e modificar o cliente, desde o ponto de vista estético como do ponto de vista do comportamento.

É distribuído com GeneXus (<diretório de GeneXus>/Packages/Gxpm/Extra/CustomClient).

GXflow SD Client

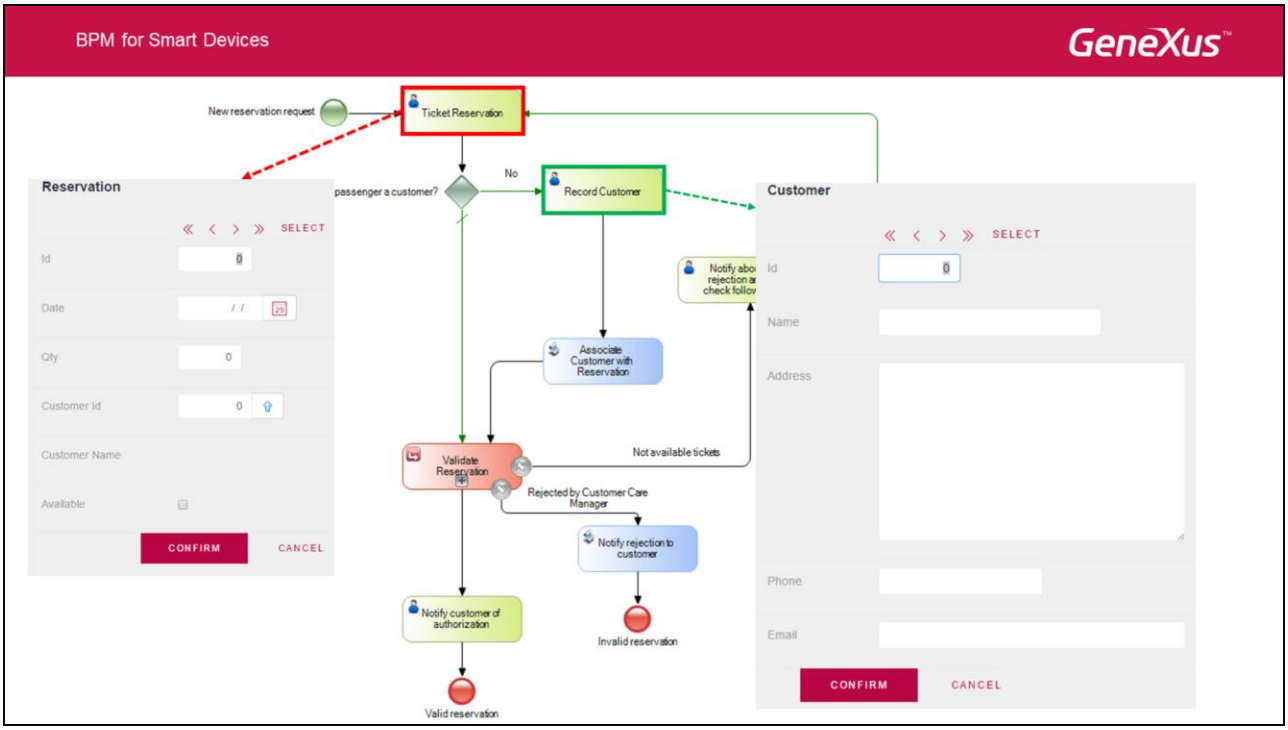


GXflow Smart Devices Client como o GXflow Custom Client é um conjunto de objetos, construídos utilizando a API do GXflow que o usuário pode consolidar em sua base de conhecimento. Isto permite incorporar na aplicação a possibilidade de executar processos e tarefas através de um dispositivo móvel.

É distribuído com GeneXus (<diretório de GeneXus>/Packages/Gxpm/Extra/GXflowClientSD).

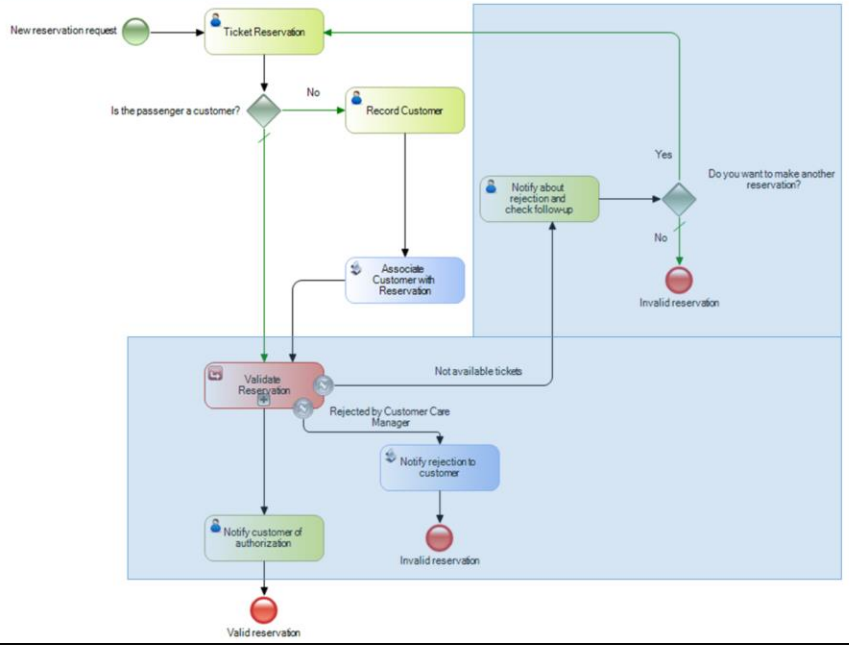
Module 14

BPM for Smart Devices



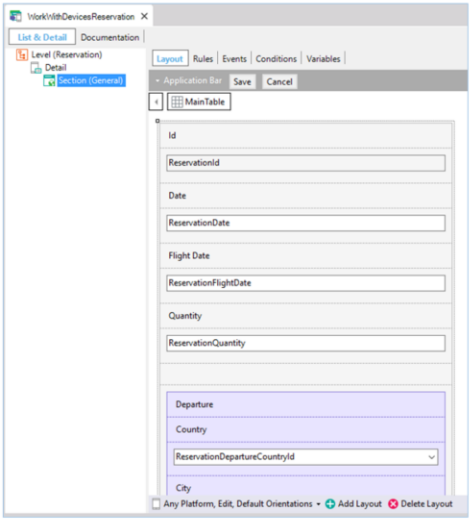
Até agora temos associado objetos GeneXus que executavam em uma página web, as tarefas interativas dos diagramas de processos.

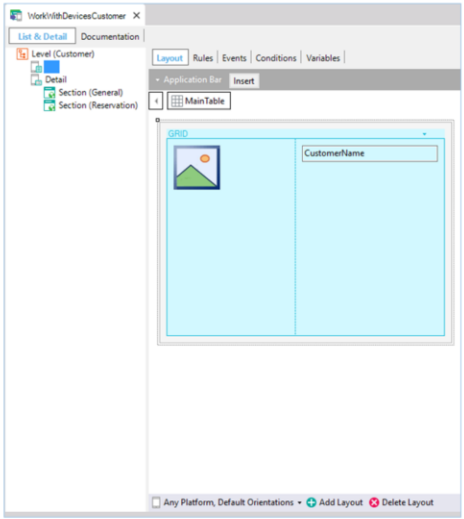
Mas também é possível associar as tarefas de usuário, objetos GeneXus que executem em um Smart Device. Por exemplo, suponhamos que o processo de adicionar uma reserva na agência de viagens, queremos executar desde um dispositivo móvel.



Vamos simplificar o processo de reserva de passagens que havíamos desenhado. Removemos o subprocesso de validação e alguns símbolos que não usaremos. Para especificar a disponibilidade da reserva utilizaremos a transação Reservation e seu atributo ReservationAvailable.

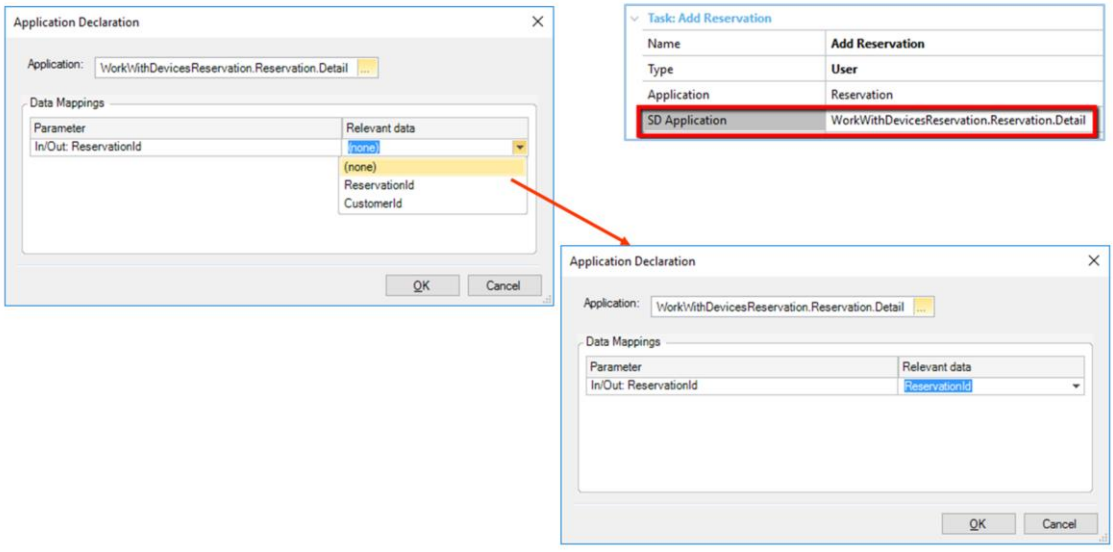
Association of objects WW for SD





O primeiro que vamos fazer agora é aplicar o padrão Work With Smart Devices as transações Reservation e Customer, que são os envolvidos no processo. Para isso, vamos ao Folder View, selecionamos ambas transações, clicamos com o botão direito, escolhemos Apply Pattern e Work With for Smart Devices.

Association of object WorkWithDevicesReservation to the Add Reservation task

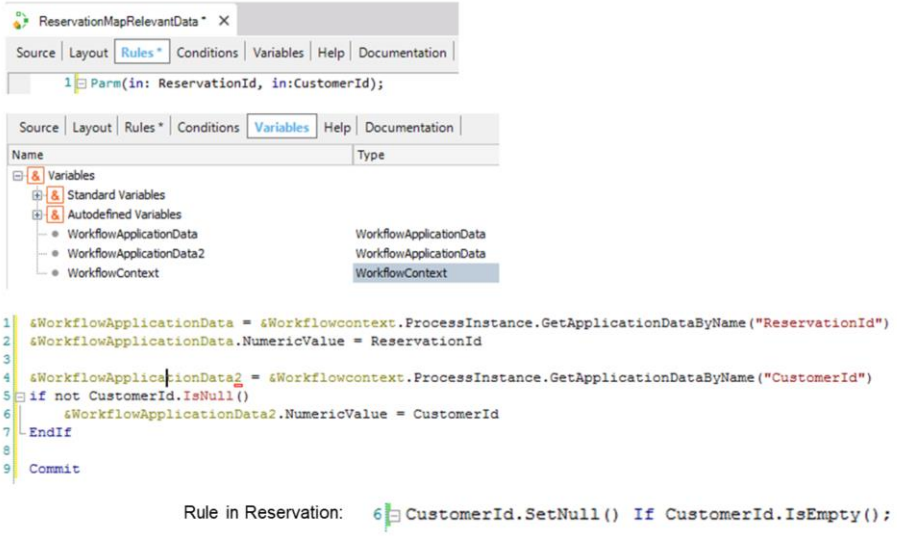


Agora associaremos a tarefa TicketReservation, a aplicação SD que gerou o padrão.

Vamos as propriedades da tarefa e na propriedade SD Application pressionamos o botão e escolhemos WorkWithDevicesReservation.

Pressionamos a guia e associamos o dado relevante ReservationId.

Creation of procedure to associate relevant data of Reservation



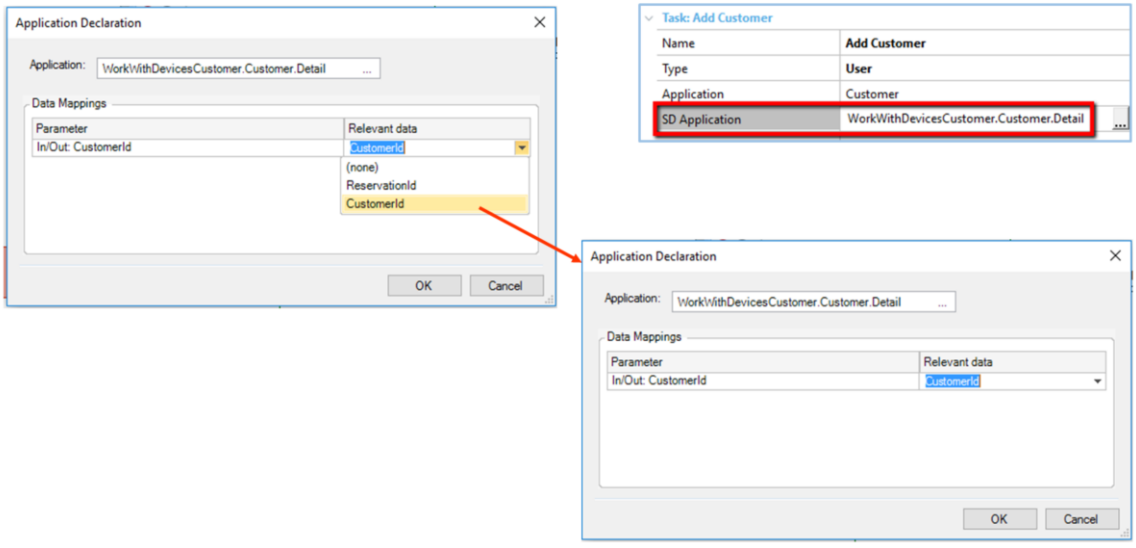
Os dados relevantes não se mapeiam automaticamente entre o diagrama e o objeto SD. Para fazer esta associação devemos contar com um procedimento

ReservationMapRelevantData, para fazer o mapeamento, através da API de Workflow.

Não devemos esquecer de incluir o Commit, quando trabalhamos com tipos de dados Workflow.

Como perguntamos pelo valor isNull() de CustomerId, vemos que adicionamos a seguinte regra a transação Reservation.

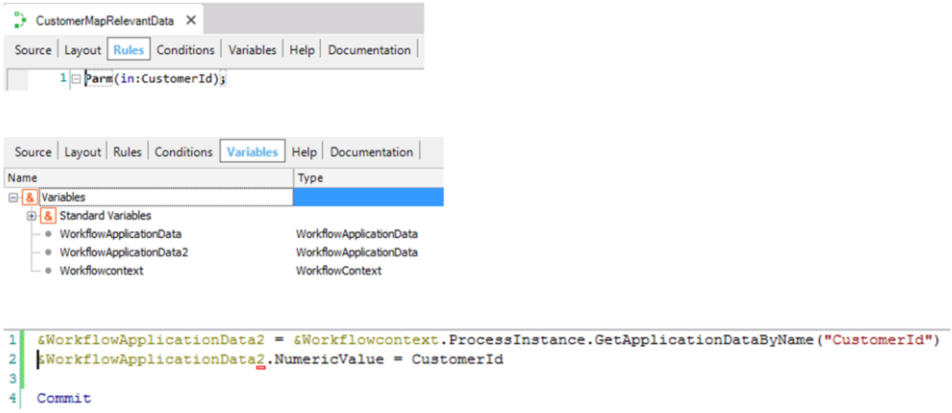
Association of object WorkWithDevicesCustomer to the Add Customer task



Vamos as propriedades da tarefa e na propriedade SD Application pressionamos o botão e escolhemos WorkWithDevicesCustomer.

Pressionamos a guia e associamos o dado relevante CustomerId.

Creation of procedure to associate relevant data of Customer



De forma similar ao que aconteceu com as reservas, temos que contar com um procedimento *CustomerMapRelevantData*.

Não devemos esquecer de incluir o Commit, quando trabalhamos com tipos de dados Workflow.

Inclusion of invocations to mapping procedures in Save events of WW for SD

In WorkWithDevicesReservation:

```
Event 'Save'
    Composite
        SDActions.Save()
        ReservationMapRelevantData.Call(ReservationId,CustomerId)
        return
    EndComposite
EndEvent
```

In WorkWithDevicesCustomer:

```
Event 'Save'
    Composite
        SDActions.Save()
        CustomerMapRelevantData.Call(CustomerId)
        return
    EndComposite
EndEvent
```

Tasks prior to executing the app

1) Import and set up **GXflow para Smart Devices client**

http://wiki.genexus.com/commwiki/servlet/hwiki?HowTo%3A+Configuring+GXflow+Client+For+Smart+Devices

2) Invoke each SD object used in our process diagram

We do it on the WorflowSDClient dashboard:

```
Event 'DummyCalls'
        WorkWithDevicesReservation.Reservation.Detail(1)
        WorkWithDevicesReservation.Reservation.List()
        WorkWithDevicesCustomer.Customer.Detail(1)
        WorkWithDevicesCustomer.Customer.List()
EndEvent
```

Para poder executar o diagrama de processos que construímos, devemos importar e configurar o **cliente GXflow para Smart Devices.**

Uma vez importado e configurado o cliente GXflow para SD, é necessário que tenhamos uma invocação a cada objeto SD usado em nosso diagrama de processos.

Para isso, devemos agregar código ao dashboard WorkflowSDClient.

Execution on a mobile device

Pressionamos F5 é vemos que o emulador de Android executou automaticamente mostrando a tela do login.

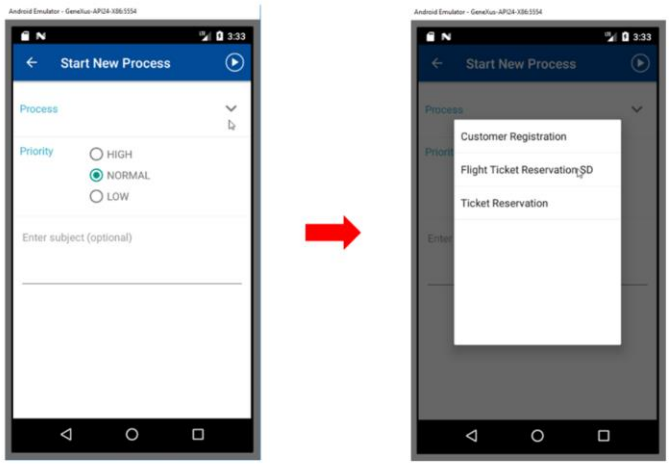
Vamos logar com o usuário administrador de workflow, assim que adicionamos como usuário WFADMISTRATOR e o mesmo como password.

Ao entrar vemos uma tela que nos mostra a bandeja de entrada

Execution on a mobile device

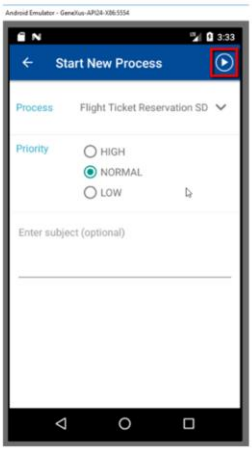
Selecionamos a bandeja de entrada e pressionamos o botão de '+' para instanciar um processo.

Execution on a mobile device



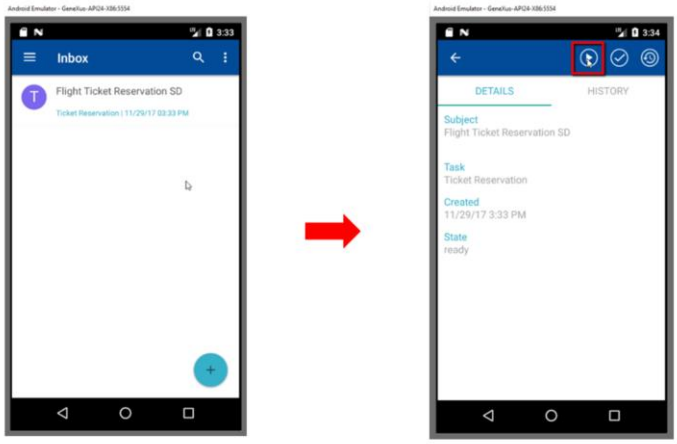
Agora selecionamos o processo FlightTicketReservationSD.

Execution on a mobile device



Vemos que abre a janela do processo. Pressionamos o botão de Start para iniciar o mesmo.

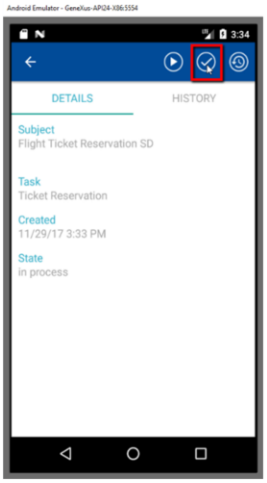
Execution on a mobile device



Agora vemos que o processo tem iniciado e que temos a tarefa TicketReservation pendente para executar.

Para executar a tarefa, clicamos sobre a tarefa e pressionamos o botão da data para iniciá-la.

Execution on a mobile device

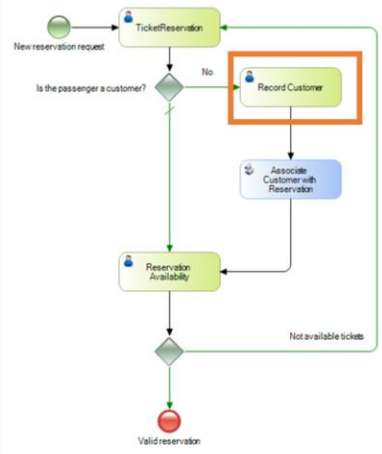


Vemos que abre o objeto SD para trabalhar com reservas, para que adicionemos uma reserva. Vamos adicionar uma reserva nova, deixamos o Id sem especificar já que é autonumerado e deixamos o identificador do cliente vazio.

Agora pressionamos o botão de Confirmar.

Para finalizar a tarefa, escolhemos Completar.

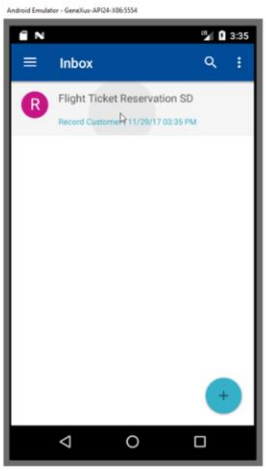
Execution on a mobile device

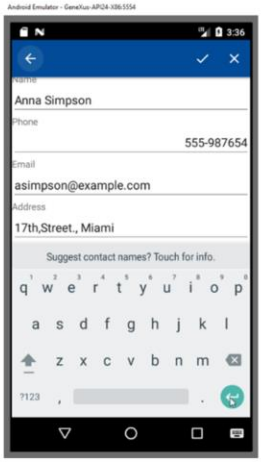


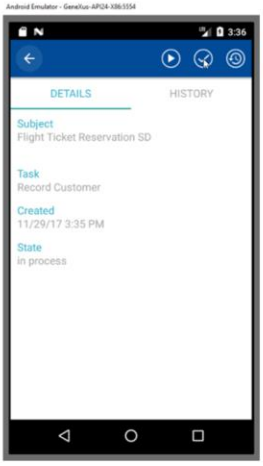
Vemos que abre a bandeja de entrada e agora a tarefa pendente de execução é RecordCustomer.

Como deixamos o cliente sem adicionar, o motor de workflow avaliou a condição do exclusive Gateway “Is the passenger a customer?” e determinou que a seguinte tarefa será RecordCustomer, a qual invocará o objeto SD Trabalhar com Clientes, para que adicionemos o cliente.

Execution on a mobile device

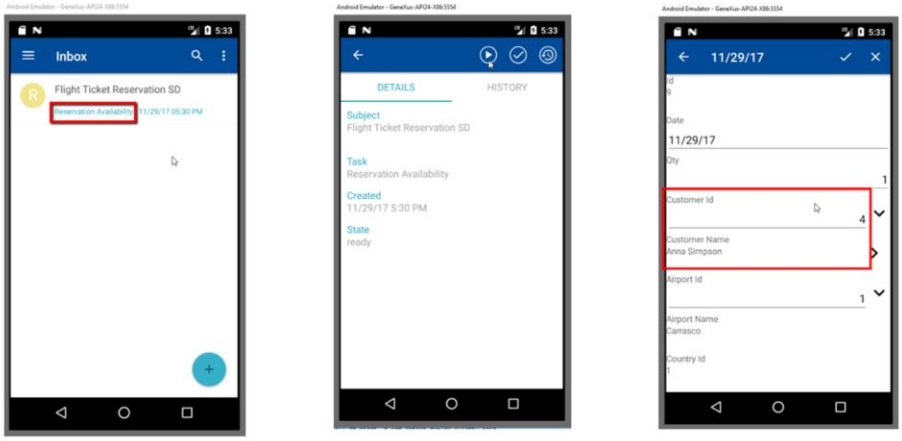






Executamos a tarefa Record Customer…Adicionamos os dados do cliente e pressionamos Confirmar. Para finalizar, completamos a tarefa Record Customer.

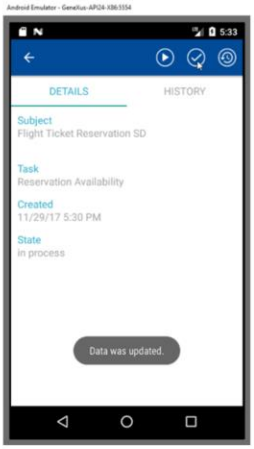
Execution on a mobile device



Agora a próxima tarefa que aparece como pendente é ReservationAvailability. Clicamos sobre a mesma para abri-la e a continuação a executamos.

Observamos que o cliente foi com sucesso atributo a reserva, dado que executou corretamente o procedimento associado a tarefa batch AssociateCustomerToReservation.

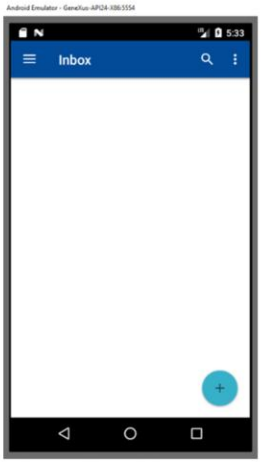
Execution on a mobile device

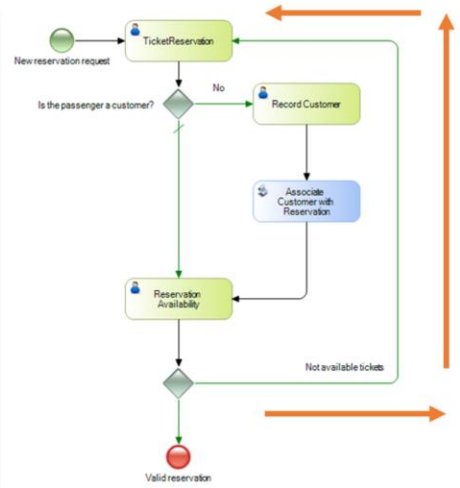


Vamos marcar que a reserva está disponível e pressionamos Confirmar.

Para finalizar, completamos a tarefa ReservationAvailability.

Execution on a mobile device





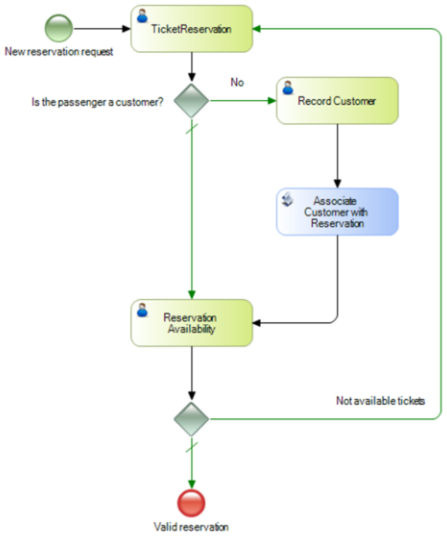
Agora a bandeja de entrada não mostra mais tarefa pendentes o que implica que finalizou a execução do processo de reserva da agência de viagens.

Se tivéssemos marcado que a reserva não estava disponível, tivesse executado novamente o objeto SD WorkWithDevicesReservation, para que adicionarmos uma reserva nova.

BPM for Smart Devices

GeneXus

Smart Devices: https://training.genexus.com/smart-devices-en/mobile-applications-with-genexus-15-course-en?en



BPM Suite: https://wiki.genexus.com/commwiki/servlet/wiki?24723

Vimos assim como executar um diagrama de processo de negócios, no dispositivo móvel.

Em particular, temos associados as tarefas, os objetos gerados pelo padrão Work With for Smart Devices, mais também poderíamos ter associado um objeto SD criado por nós, como por exemplo um panel para DS.

Neste caso temos gerado a aplicação para Smart Devices em Android e executamos a mesma utilizando um emulador, mas é possível prototipar sobre um dispositivo físico e gerar aplicações para outras plataformas, como por exemplo, dispositivos iOS como Ipad ou Iphone.Para saber mais sobre aplicações Smart Devices, visite o link, e para conhecer mais possibilidades o suite BPM de GeneXus visite o seguinte link do wiki.

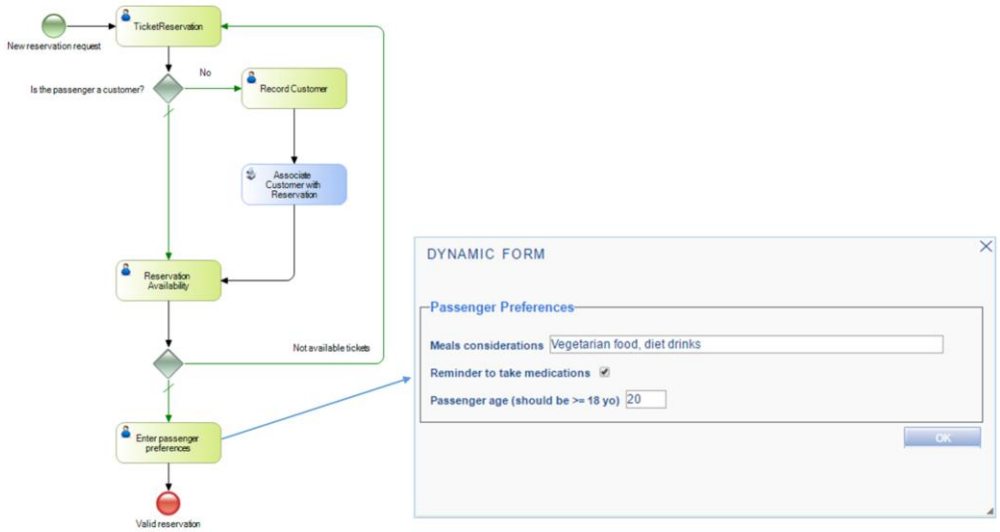
Module 15

Dynamic Forms

Em geral, se necessitamos executar uma tarefa interativa de um diagrama de processos de negócios, devemos contar em nossa KB com um objeto transação ou um webpanel, para associá-lo a tarefa.

Os formulários dinâmicos permitem criar telas de ingressos de dados diretamente desde o cliente do workflow e associá-los a tarefas interativas, em tempo de execução, sem necessidade de escrever nenhuma linha de código.

Dynamic Forms

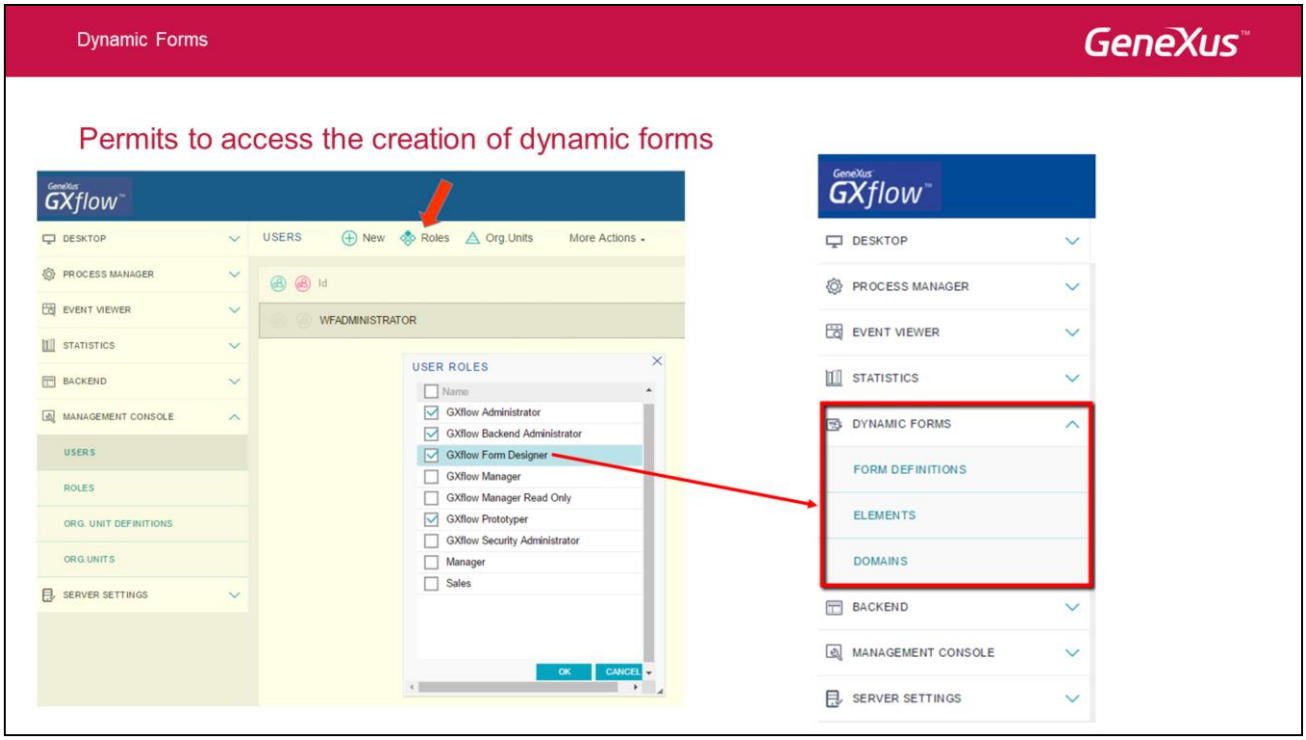


Um cenário típico do uso destes formulários é quando recebemos um requerimento novo de ingresso de dados e o sistema já está em produção. Em lugar de redesenhar a aplicação, podemos criar formulários dinâmicos e adaptá-los em tempo de execução para armazenar informação, sem ter que modificar a base de dados da aplicação.

Outro caso, quando se deseja guardar dados de um certo processo que não tínhamos previsto no workflow original, na qual devemos incluir informação que só é relevante a cada instância do processo. Os formulários dinâmicos permite-nos armazenar esta informação nas tabelas do workflow sem ter que contar com estruturas na tabela de dados para este tipo de informação.

No nosso exemplo, a agência de viagens deseja registrar algumas preferencias, dos clientes que tenham feito reservas de passagens..

Estes dados não os queremos armazenar na tabela de dados do sistema, porque podem depender das preferências do passageiro para cada voo em particular, pelo que solicitaremos esses dados mediante uma tarefa associada a um formulário dinâmico, no mesmo processo de reserva, quando confirmou a mesma.

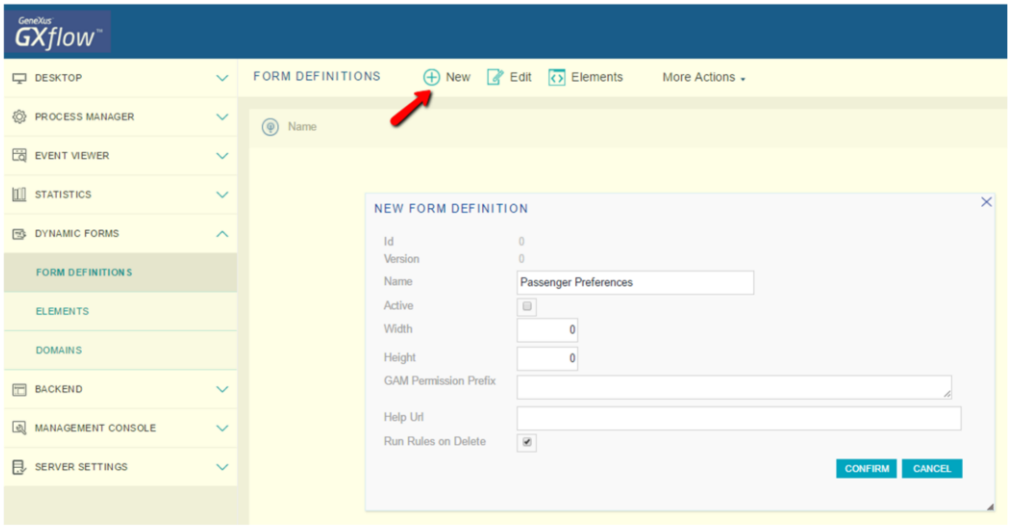


Para ter acesso a criação e modificação de formulários dinâmicos, devemos ter as permissões adequadas. Para isso, no cliente do Gxflow vai ao Management Console, seleciona Users e na janela de Usuários pressiona o botão Roles, marca Gxflow Form Designer e pressiona OK.

Agora fechamos a sessão e voltamos a fazer login como WFADMINISTRATOR.

Vemos que no Navegador, baixo estatísticas agora há um grupo chamado Dynamic Forms que tem 3 componentes: as **definições dos formulários** dinâmicos que fizemos, os **elementos** dos formulários definidos e os **domínios** que são utilizados na definição dos elementos dos formulários.

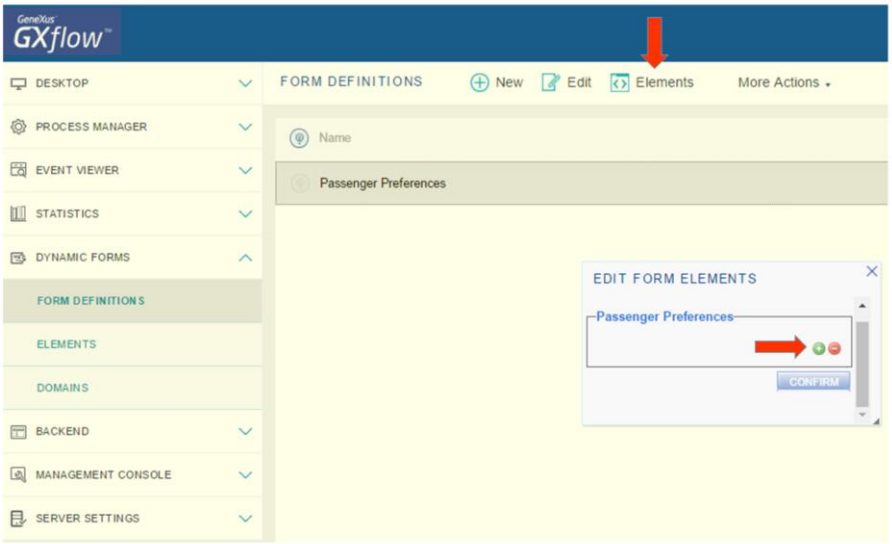
Creation of dynamic forms



Para criar un formulário dinâmico , Clicamos em Form Definitions e pressionamos o botão de New.

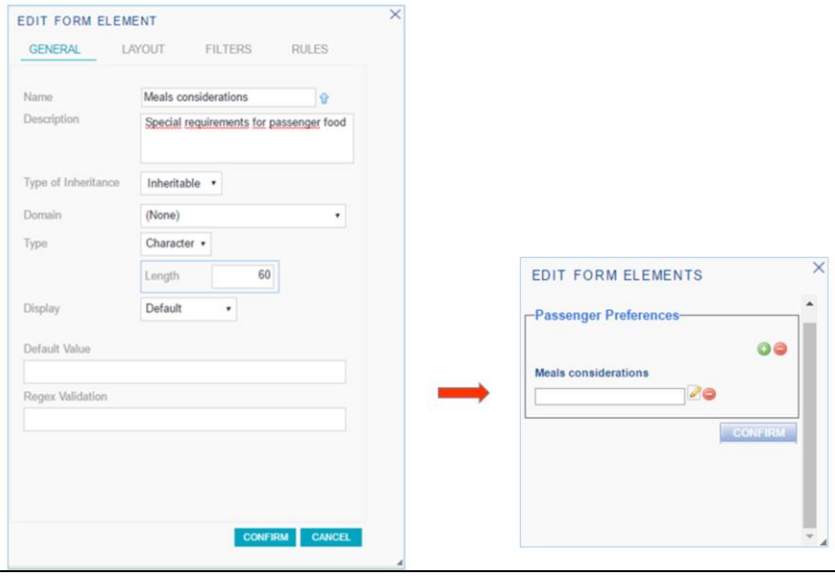
Damos o nome **Passenger Preferences** e pressionamos Confirmar.

Creation of dynamic form elements



Para definir os elementos que incluiremos dentro do formulário, selecionamos o formulário Passenger Prefrences e pressionamos o botão Elements. Vemos que abre a janela do formulário, movemos o mouse dentro do marco e pressionamos o botão de +.

Creation of a text element



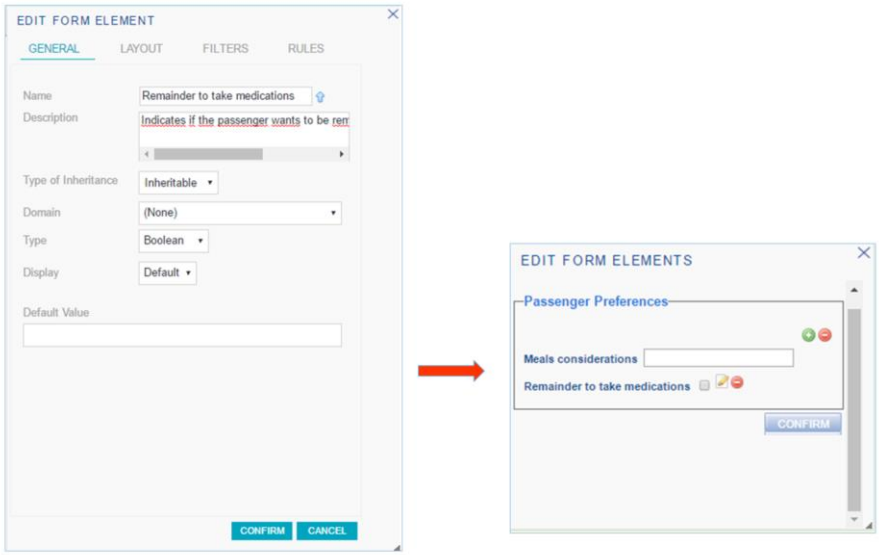
Abre-se um quadro de diálogo para adicionar o elemento.

Neste elemento armazenaremos as considerações que o passageiro tem a respeito a sua comida, assim que no nome adicionamos “Meals considerations” e na descrição coloquemos “Special requirements for passenger food”

Modificamos o tipo a Character e damos um espaço de 60 caracteres.

O resto dos campos deixamos com seu valor por padrão e pressionamos Confirmar.

Creation of a Boolean element

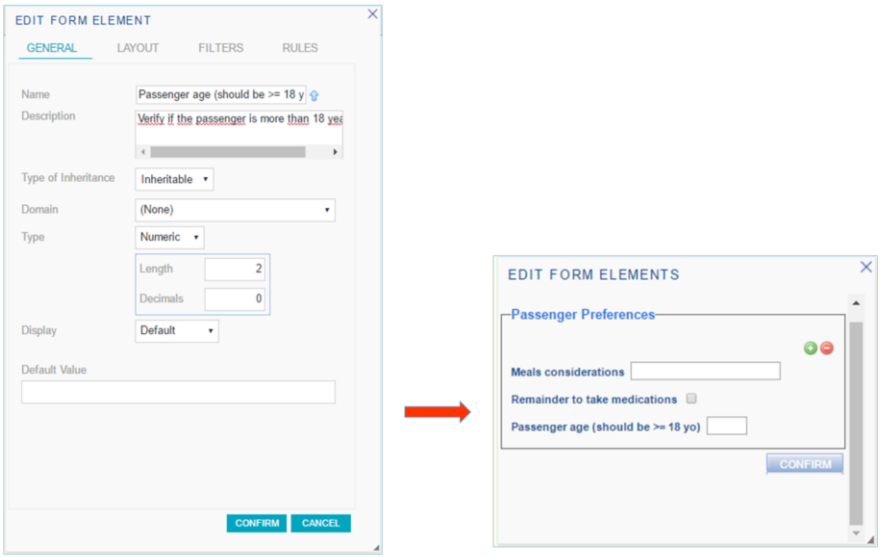


Pressionamos novamente o símbolo + e adicionaremos um novo elemento para registrar se o passageiro deseja que o lembrem que deve tomar certos medicamentos.

No nome escrevemos “Remainder to take medications” e na descrição “Indicates if the passenger wants to be remainded to take medicines”.

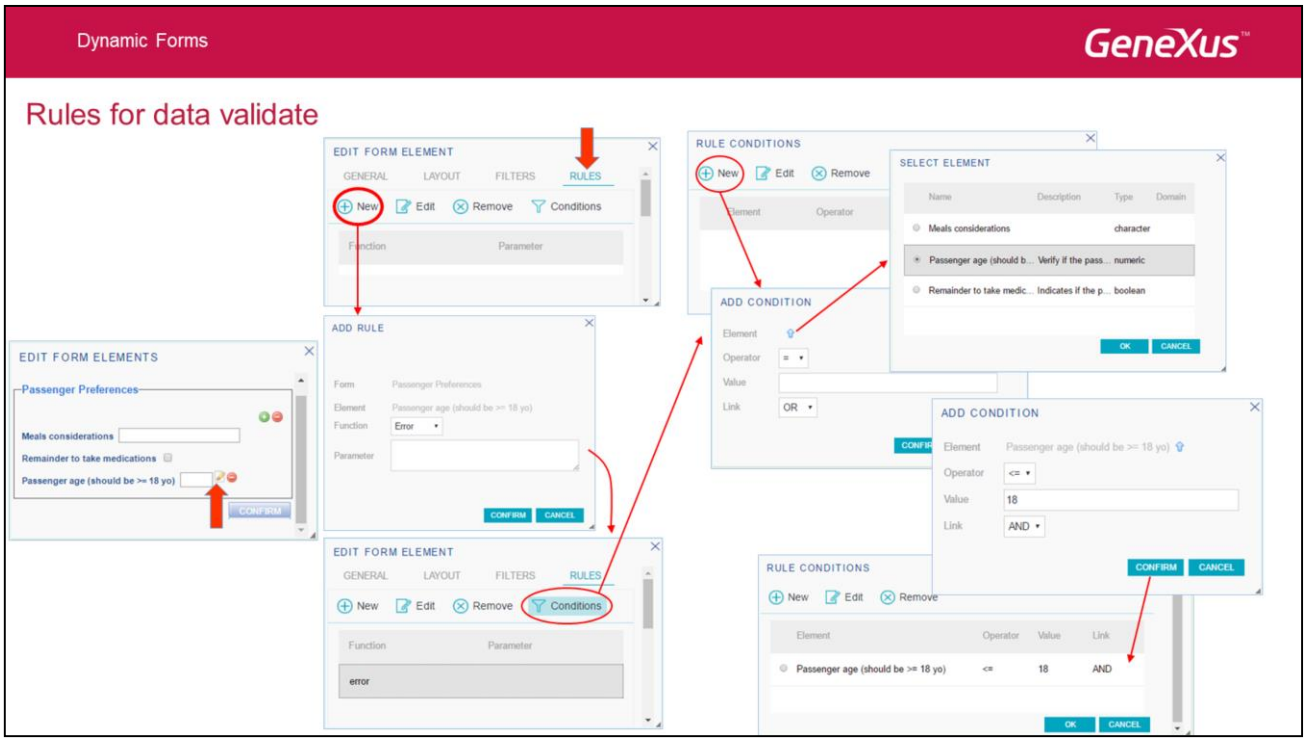
Deixamos o valor do tipo em Boolen e pressionamos Confirmar.

Creation of a numeric element



Por último criamos um elemento a mais para registrar a idade do passageiro.

Pressionamos o botão verde e coloquemos o nome “Passenger age – should be greather than 18” e na descrição “Verify if the passenger is more than 18 years old”. Selecionamos o tipo Numeric, longo 2 e sem decimais

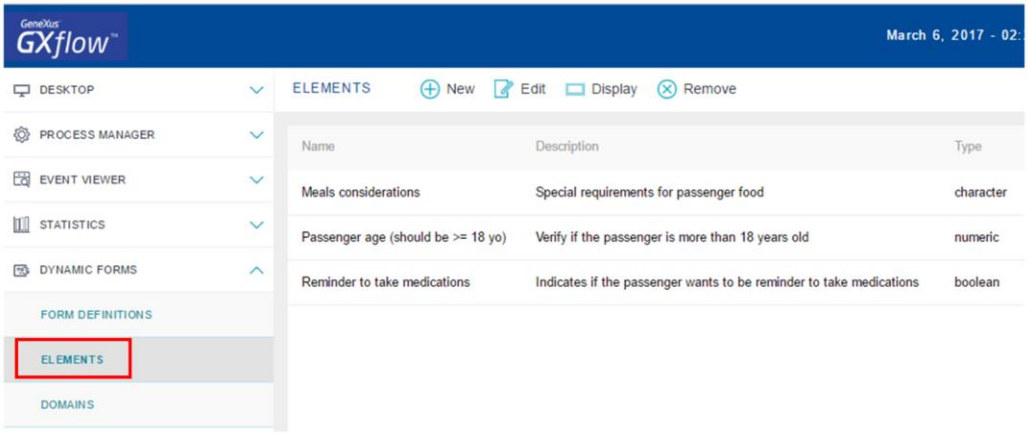


Agora vamos adicionar uma regra para que se tenha este controle automaticamente. Passamos o mouse sobre o elemento e escolhemos Editar. Vamos a aba Rules e pressionamos o botão de +. Selecionamos a função Erro e pressionamos Confirmar.

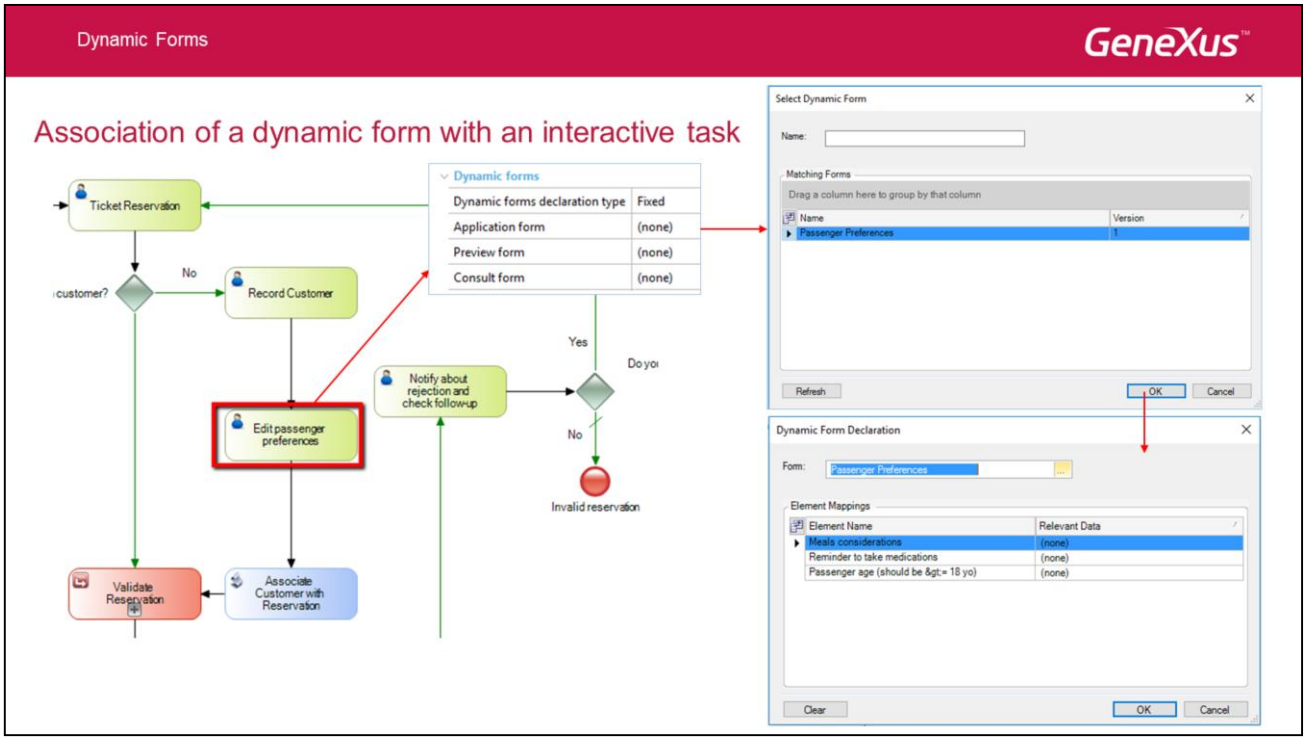
Vemos que criou uma regra com a descrição “error”. Selecionamos a mesma e pressionamos o botão de Conditions, vemos que abre a venda para adicionar de condições.

Selecionamos o elemento ao que queremos adicionar a regra de validação, no nosso caso “Passenger age” e pressionamos OK. Agora selecionamos o operador de menor ou igual que, como valor colocamos 18 e no link selecionamos AND. Pressionamos Confirmar.

Verifying form elements

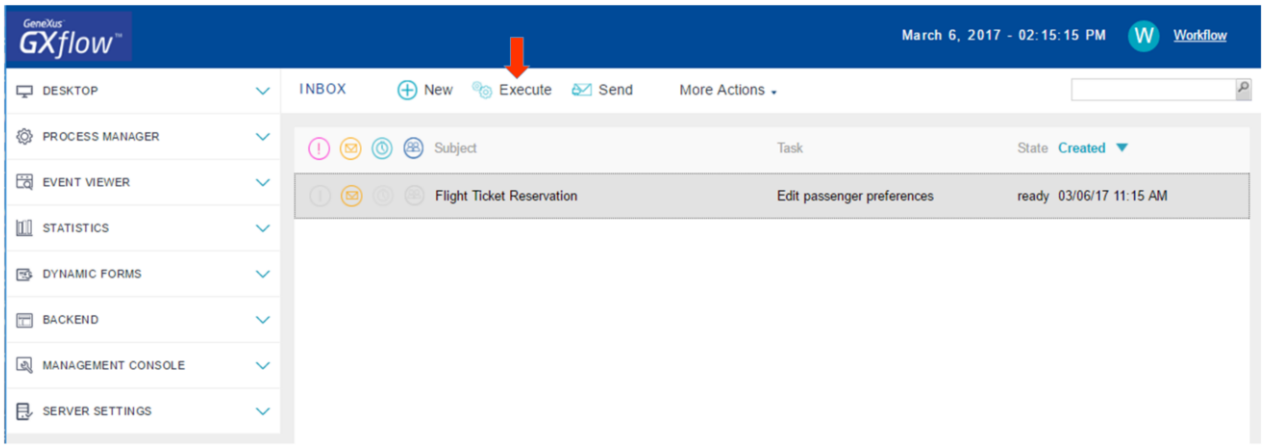


Na seção Elements, podemos ver as definições que acabamos de fazer.



Nas propriedades da tarefa vamos ao grupo Dynamic Forms e pressionamos o botão da propriedade Application form. Agora, pressionamos o botão em Forms, selecionamos o formulário Passenger Preferences e pressionamos OK.

Execution of a dynamic form



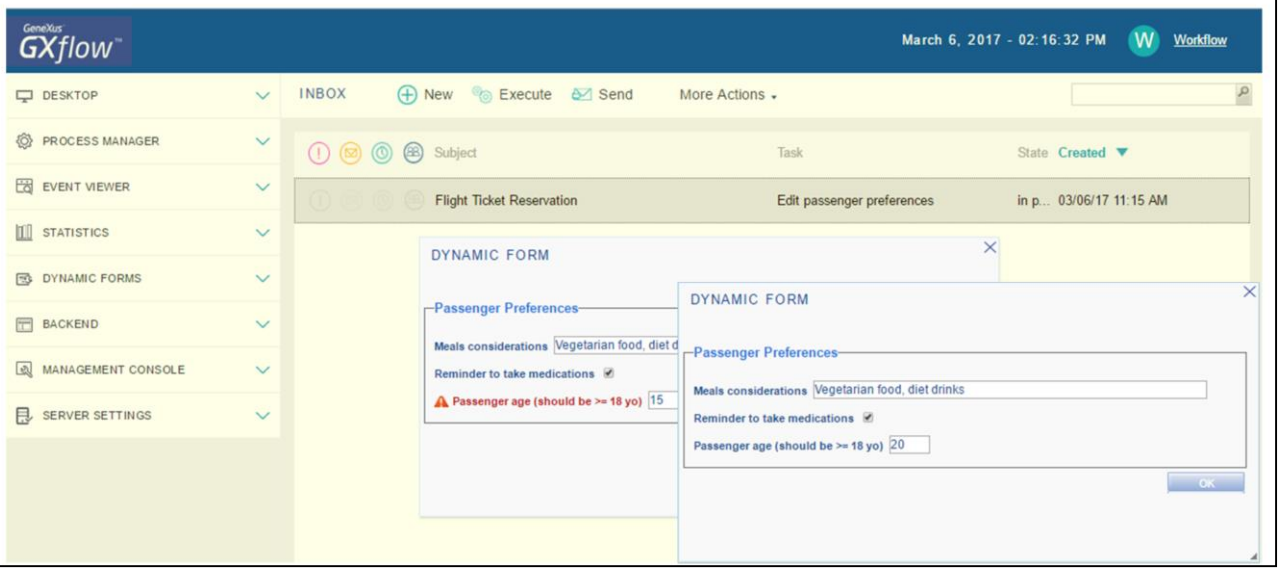
Vamos executar o diagrama, clicamos com o botão direito sobre o mesmo e escolhemos Run.

Vemos que abre o cliente de Gxflow e aparece como pendente a tarefa TicketReservation do processo FlightTicketReservationDynForm.

Selecionamos a tarefa e a executamos. Seguimos a execução, adiciomos uma reserva selecionando um cliente.... e agora colocamos na reserva como disponível.

Vemos que a próxima tarefa é **Enter passenger preferences** e a executamos.

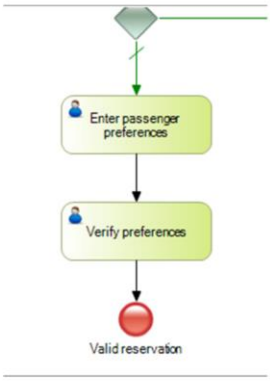
Execution of a dynamic form



Em Meals considerations adicionamos: “Vegerarian food, diet drinks” marcamos o lembrete para tomar medicamentos e adicionamos uma idade de 15. Pressionamos Confirmar.

Vemos que a regra é disparada, dando-nos um alerta que o valor adicionado para a idade não é valido, já que a mesma deve ser maior que 18. Se inserimos um valor maior que 18 (por exemplo, 20), pressione confirmar que o formulário seja fechado.

Verification of stored data



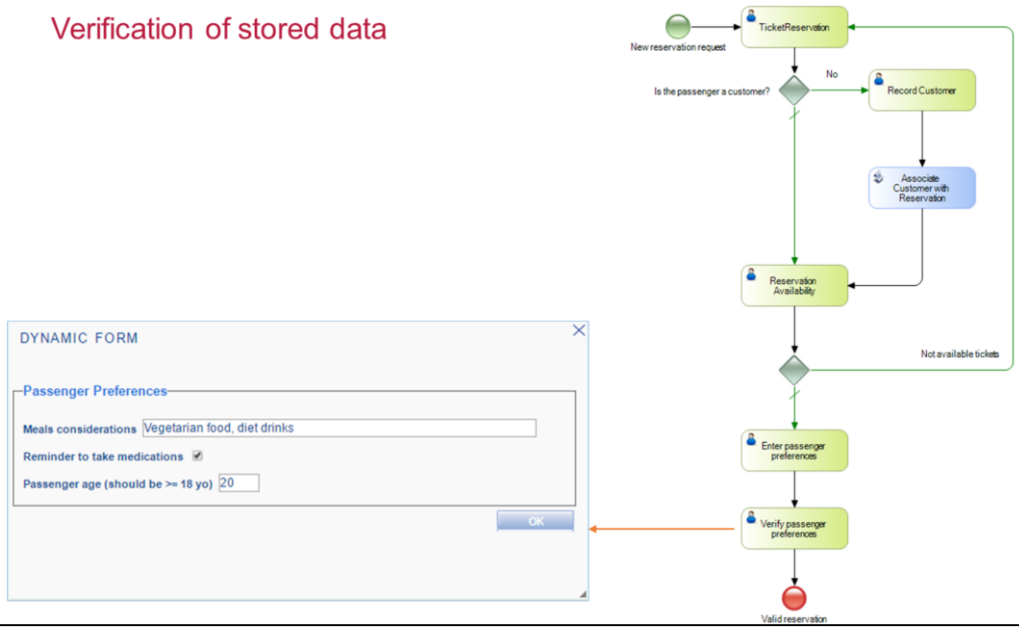
Task: Verify preferences	
Name	**Verify preferences**
Type	**User**
Application	(none)
SD Application	(none)
Visible in history	True
Roles	
Subject rule	
Advanced Properties	
Optional routing type	Branch
Request comments	False
Work with documents	False
Send e-mail	False
Strong synchronization	False
Skip predecessors	True
Enable deferred completion	True
Consult application	(none)
Consult SD application	(none)
Preview application	(none)
Preview SD application	(none)
Metadata	
Dynamic forms	
Dynamic forms declaration type	Fixed
Application form	(none)
Preview form	(none)
Consult form	(none)

Os dados que temos armazenado, ficaram guardados nas tabelas de workflow e estavam disponíveis dentro da instância do processo.

Em nosso exemplo, o processo justo termina depois de adicionar estes dados, mas se o processo continuar e teve outra tarefa associada ao mesmo formulário, o mesmo abrirá com os dados previamente adicionados.

Para testar isto modificamos o diagrama de processo e adicionamos uma tarefa nova entre a tarefa **Enter passenger preferences** e o None End Event, com o nome **Verify preferences**. Associamos a tarefa que recém criamos ao formulário dinâmico **Passenger preferences** e executamos o diagrama de processos

Verification of stored data



Si executamos a tarefa **Verify preferences**, observamos que ao abrir o formulário associado, todos os campos contém os dados que adicionamos previamente quando executou a tarefa para adicionar as preferências do usuário.

Os dados adicionados no formulário, ficaram armazenados na base de dados do Workflow e estavam disponíveis em toda a estância do processo.

More information about dynamic forms

http://wiki.genexus.com/commwiki/servlet/hwikibypageid?25809

Module 16

Independent DataStore for Workflow tables

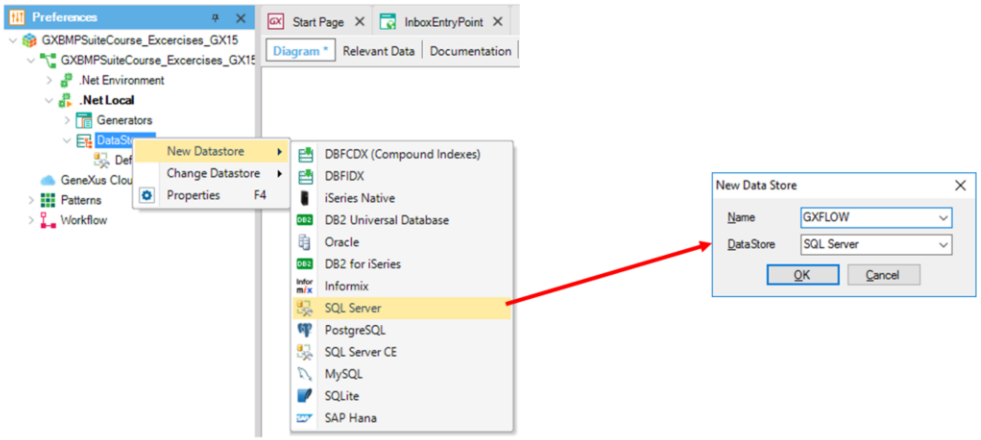
Independent DataStore for Workflow tables

Workflow Data Store

Por padrão, as tabelas de workflow são criadas no mesmo data store onde serão armazenadas as tabelas da aplicação.

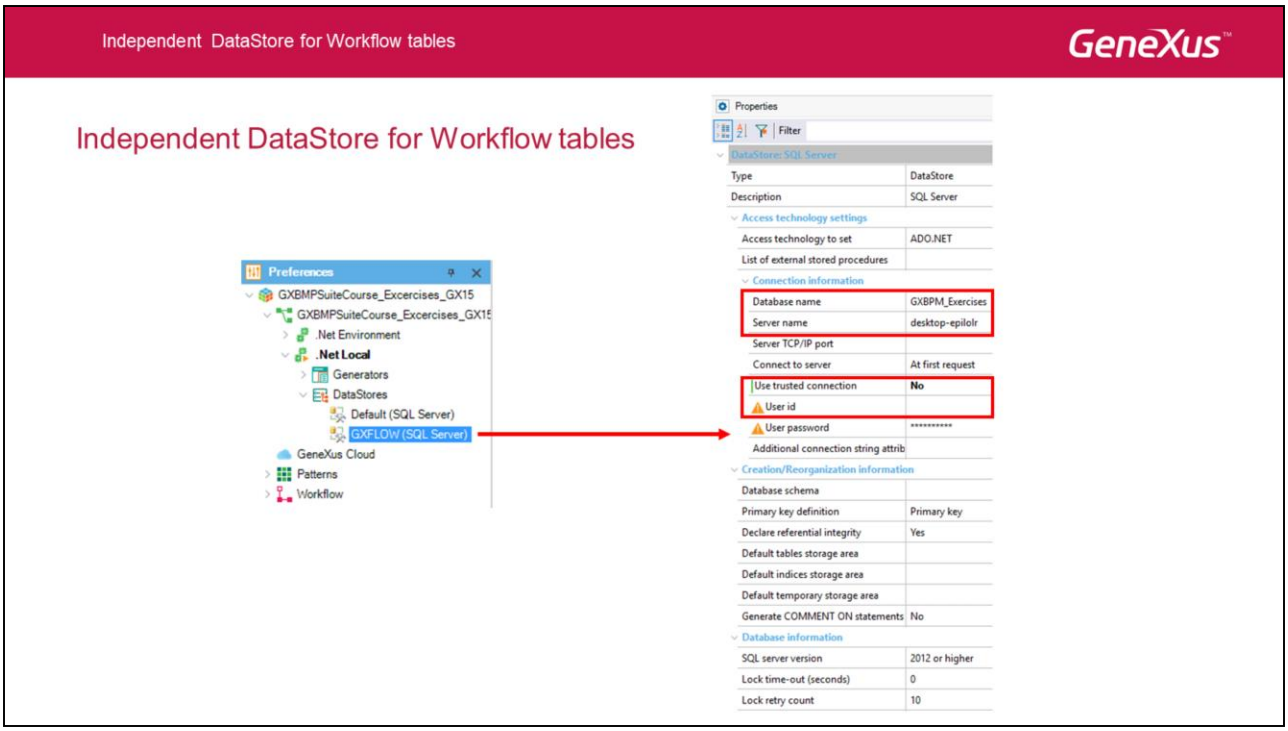
Mas é possível armazenar essas tabelas em um data store independente. Isto melhora a seguridade dos dados da aplicação e dos dados do workflow, já que é possível fixar restrições de acesso de forma independente para cada tipo de tabela.

Independent DataStore for Workflow tables



Primeiramente o tem que fazer é criar um data store novo e o mesmo deverá chamar-se GXFLOW.

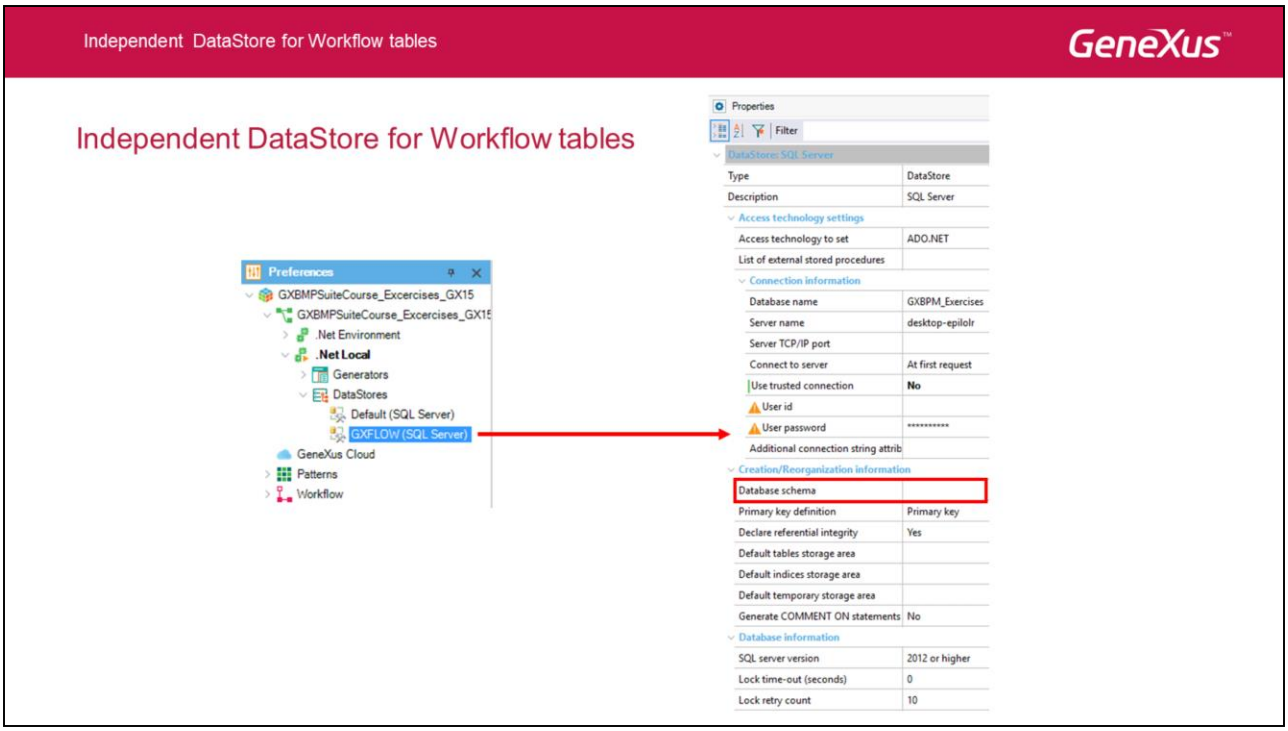
É importante respeitar este nome, de uma outra forma que o GeneXus não reconhecerá este como o data store onde deverá guardar as tabelas do workflow e as continuará guardando no default do data store.



Uma vez criado, é possível mudar o valor das propriedades Database name, Server name, User ID e User Password.

Isto permite que a base de dados onde se armazena as tabelas do workflow seja totalmente independente da base de dados da aplicação.

Neste caso, as operações sobre cada base de dados são independentes e não é possível integrar operações sobre diferentes bases de dados em uma mesma unidade de trabalho lógica (UTL)



Outra possibilidade é que a base de dados seja a mesma, mas as tabelas do workflow estarão atribuídas a um esquema diferente.

Neste caso não mudam-se os valores por padrão das propriedades de conexão (nome da base, servidor, usuário e password) e são atribuídos um valor diferente ao esquema.

Para atribuir as tabelas do workflow a um esquema diferente, são utilizados a propriedade Schema Name.

Como resultado, o motor do GxFlow realiza todas as operações no default data store e estas operações integram as mesmas unidades de trabalho lógicos que o resto da aplicação.

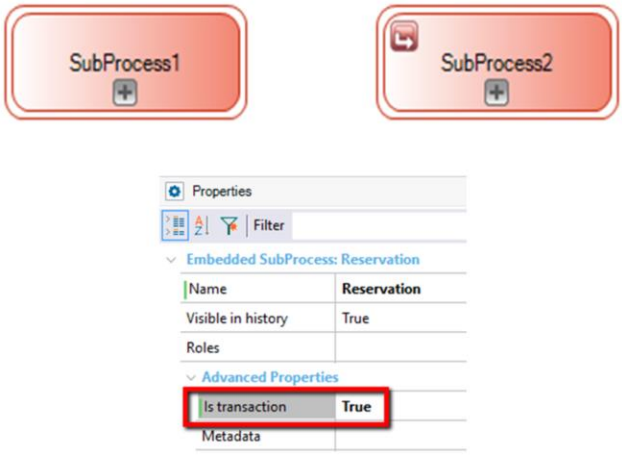
Module 17

Transactional Subprocesses

Existem processos em que é necessário coordenar várias atividades que precisam ser cumpridas com sucesso para que o fluxo do processo possa continuar e no caso de não se satisfazer satisfatoriamente, é necessário devolvê-los ao seu estado inicial.

Essas atividades integram o que chamamos de unidade de trabalho lógico (UTL), ou seja, uma unidade atômica que deve ser executada de forma indivisível. Uma UTL em um processo de negócios poderia durar dias ou mesmo semanas.

Transactional Subprocesses



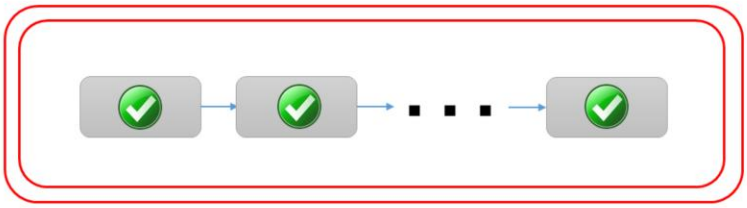
Para modelar esses cenários de negócios com transações, onde cada transação é composta por várias operações que devem ser executadas e todas elas corretamente encerradas, ou nenhuma delas, usamos os subprocessos transacionais.

Em BPMN eles são representados com uma borda dupla.

Em GeneXus, para definir um subprocesso como transacional, atribuímos a propriedade **Is transaction** com o valor True. Ao fazê-lo, veremos que a borda do símbolo do segmento muda e passa a ser representada por uma borda dupla.

Possible results in a transaction

1) All operations conclude satisfactorily



Um subprocesso transacional é concluído com sucesso quando as mudanças de dados no banco de dados terminam em um estado consistente. No caso de uma falha na execução de uma transação, é necessário iniciar ações que desfaçam as mudanças, levando os dados de volta ao estado que tinham antes dessas alterações.

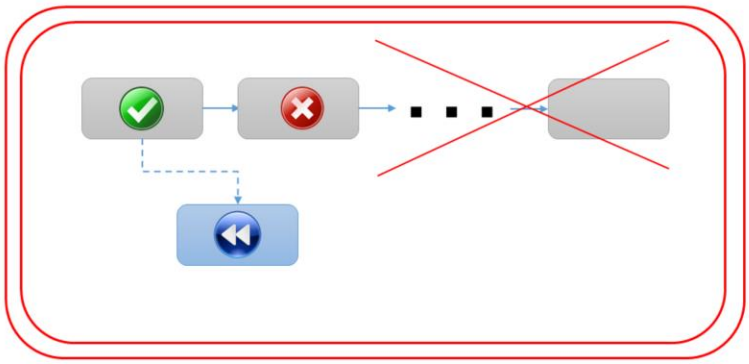
As transações, em um modelo de processo de negócios, podem ter três resultados possíveis :

Que todas as operações **finalizem satisfatoriamente**, fazendo com que o processo continue pelo fluxo normal.

GeneXus

Possible results in a transaction

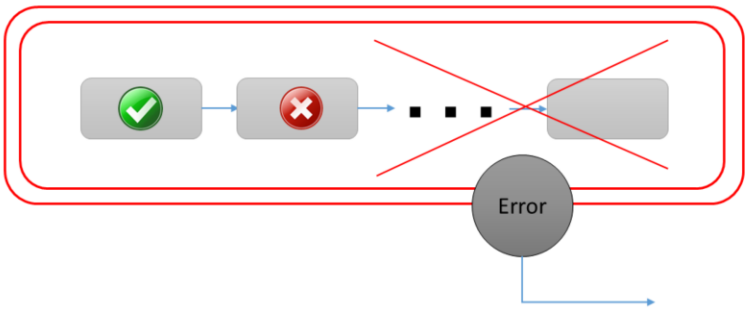
2) A failure occurs: we must revert tasks already concluded



Que **ocorra uma falha** e seja necessário reverter as atividades que já foram concluídas no subprocesso. Neste caso, certas tarefas de compensação devem ser executadas, responsáveis por deixar o processo no estado inicial em que era antes do subprocesso transacional ter sido iniciado.

Possible results in a transaction

3) An unexpected error occurs: no compensation takes place

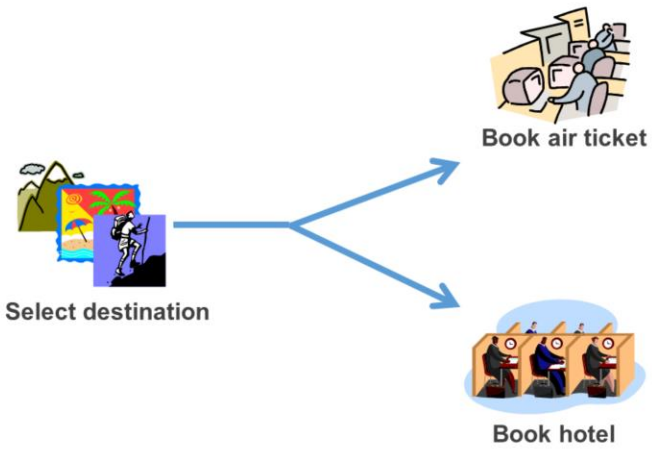


- Que **aconteça um erro inesperado**, caso em que as atividades do subprocesso são interrompidas e nenhuma atividade compensatória é executada. Nesse caso, o processo continua com a execução de um evento de erro intermediário.

Portanto, para modelar um subprocesso transacional, é necessário capturar eventos que respondam a essas situações, isto é, eventos de cancelamento de erros e processos devido a uma falha.

Veremos abaixo os conceitos mencionados até agora, com base em um exemplo.

Example: Coordinating a trip



Na agência de viagens em que trabalhamos, a coordenação de uma viagem pode ser um exemplo de subprocesso transacional, uma vez que, para ser completado corretamente, é necessário poder completar várias atividades, como a reserva de bilhetes aéreos, reserva de hotel, aluguel de carros e entrada para uma atração turística, entre outros.

Considere um processo de reserva dividido em duas atividades: reservar um voo e reservar um quarto de hotel.

Considere que ambas as atividades são executadas em uma única transação, ou seja, se o bilhete aéreo não for obtido e a reserva do hotel foi obtida, é necessário desfazer a reserva do hotel, e vice-versa, se o ingresso foi obtido, mas não o hotel , a reserva do ingresso deve ser cancelada.

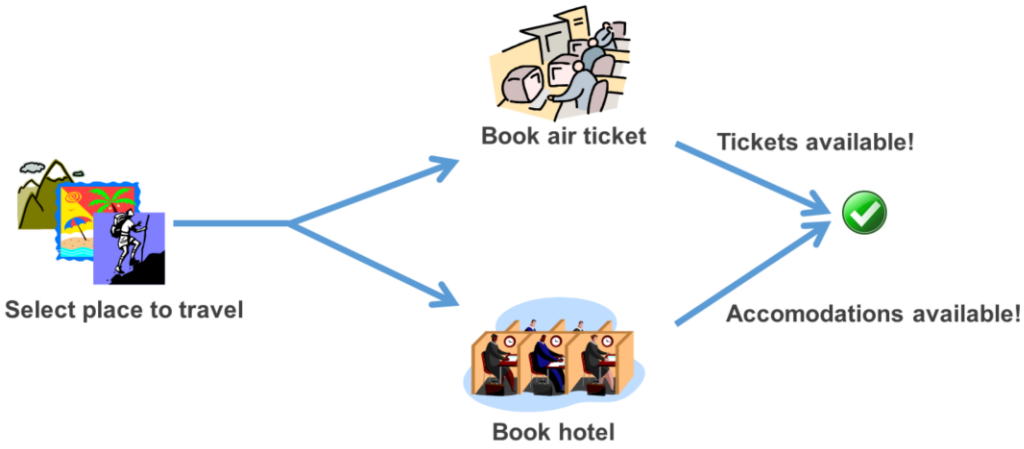
Isso implica que devemos considerar os três casos anteriormente vistos:

Que obtemos as duas reservas eo processo acaba normalmente.

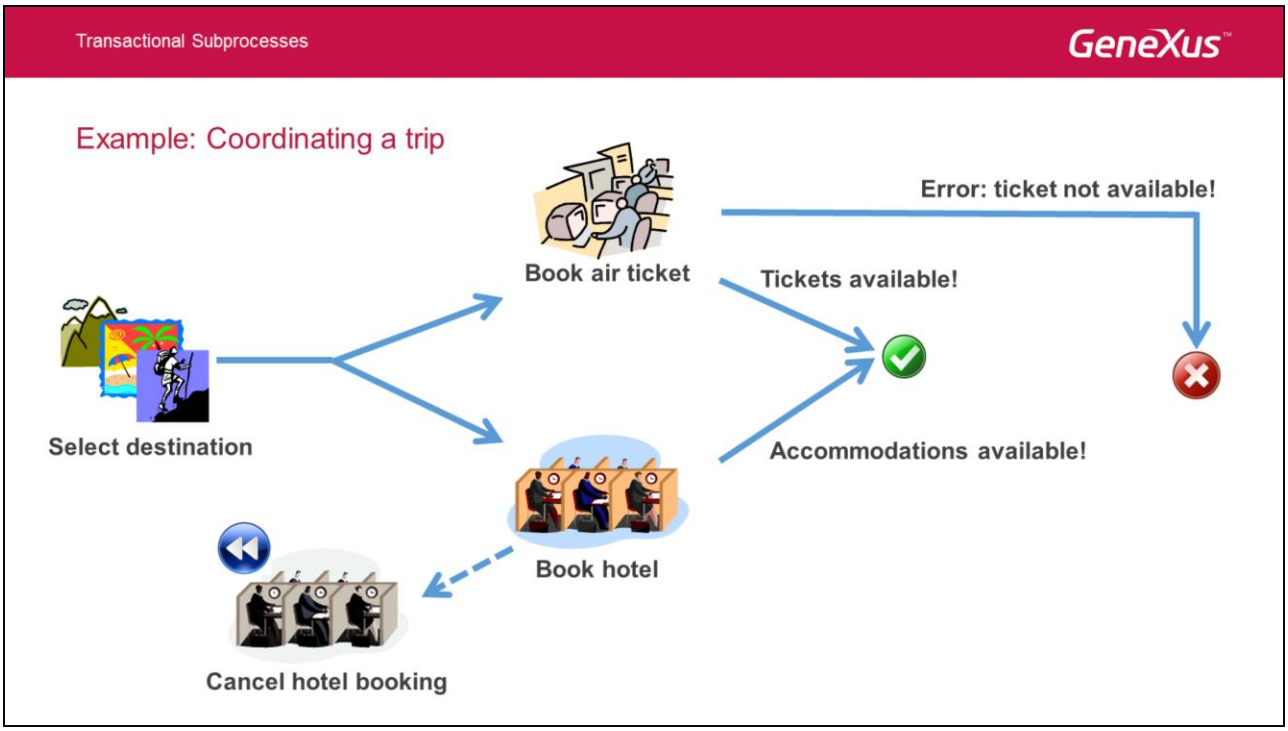
Se alguma das reservas falhar e o processo for cancelado, a reserva que foi obtida deve ser desfeita, uma vez que ambos ou nenhum devem ser obtidos.

Ocorre um erro inesperado e o processo termina.

Example: Coordinating a trip

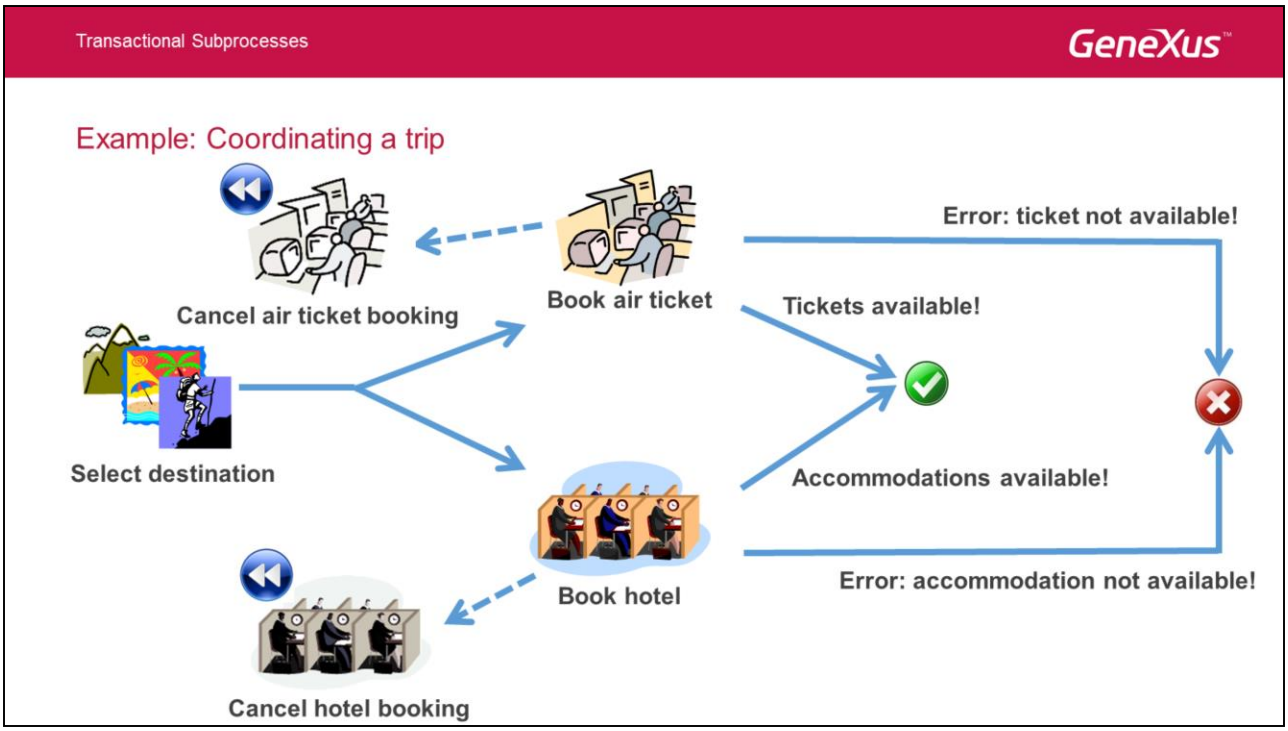


No caso de ambas as reservas serem obtidas de forma satisfatória, o processo acaba.



Mas se o bilhete aéreo falhar e a reserva do hotel for finalizada, o processo não pode ser concluído porque uma das atividades da transação não foi cumprida, portanto, é necessário desfazer as atividades que foram realizadas.

Neste caso, seria necessário executar um processo compensatório que desfaz a reserva do hotel, de modo que o processo de coordenação de viagens seja completamente desfeito.



Se, por outro lado, a reserva do bilhete aéreo foi bem sucedida e não foi possível conseguir um lugar no hotel, é necessário desfazer a reserva do bilhete.

As tarefas de cancelamento de reservas de hotéis ou bilhetes são conhecidas como atividades de compensação e muitas vezes são realizadas por um sistema externo.

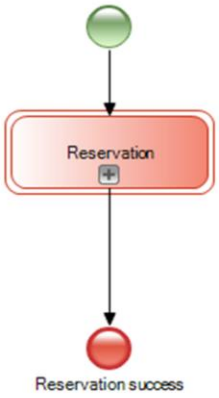
Até agora, vimos o caso em que o processo transacional termina com sucesso porque todas as atividades que o compõem terminaram corretamente. Ou, se houve uma falha em qualquer das atividades do processo de transação e essa falha produz um cancelamento do processo, sendo necessário desfazer as atividades que foram concluídas de forma que o processo seja revertido para o estado inicial.

O terceiro caso acontece quando ocorre um erro que não pode ser tratado pelo subprocesso e impede que ele continue, como quando um servidor não responde ou ocorre uma falha no sistema. Neste caso, as atividades do subprocesso são interrompidas sem compensação e é feita a reversão do banco de dados até o último commit antes do início do subprocesso.

No nosso exemplo, quando o processo de reserva não termina com sucesso, esta situação é detectada e o cliente é informado de que não é possível fazer a reserva.

Vamos ver como modelamos esse processo com GeneXus BPM Suite.

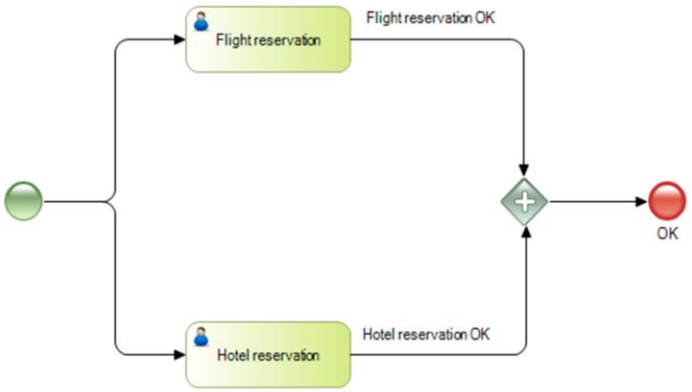
Modeling in GeneXus



Primeiro executamos a IDE de GeneXus ou o GeneXus Business Process Modeler. Depois, em nossa Knowledge Base, criamos um objeto Business Process Diagram.

Arrastamos um None Start Event, um símbolo de subproceso que conectaremos no Start Event, e um None End Event ao qual conectamos o subproceso. as propriedades do subprocesso, atribuiremos a propriedade **Is transaction** como **True.** Isso fará com que o símbolo do segmento mude e fique com bordas duplas, que é a forma como um subprocesso transacional é representado no BPMN.

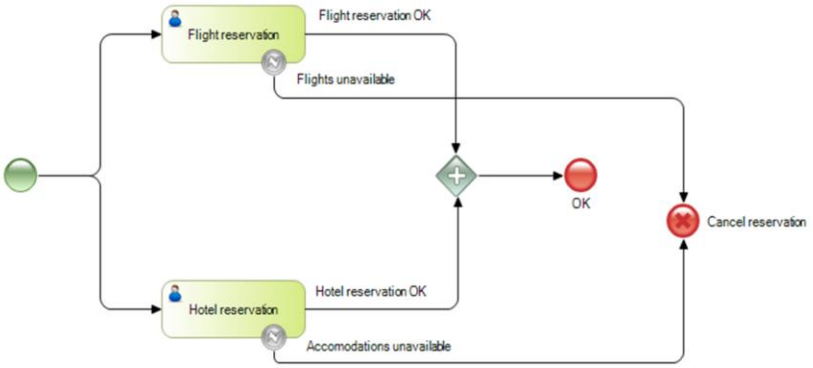
Modeling in GeneXus



Agora, abrimos o subprocesso, começamos a adicionar um None Start Event, depois adicionamos 2 tarefas interativas, uma para a reserva dos vôos e outra para a reserva do hotel, e conectemos no Start. Unimos a saida de ambos em um Parallel Gateway, e conectamos a saida deste em um None End Event.

Com isso, estamos modelando o caso em que ambas as tarefas terminam corretamente, caso em que o Parallel Gateway será capaz de sincronizar os dois fluxos que chegam e o fluxo de saída continuará até o End Event. No entanto, é necessário levar em consideração os outros casos, porque se uma das duas tarefas falhar, o fluxo não poderá continuar devido ao Parallel Gateway.

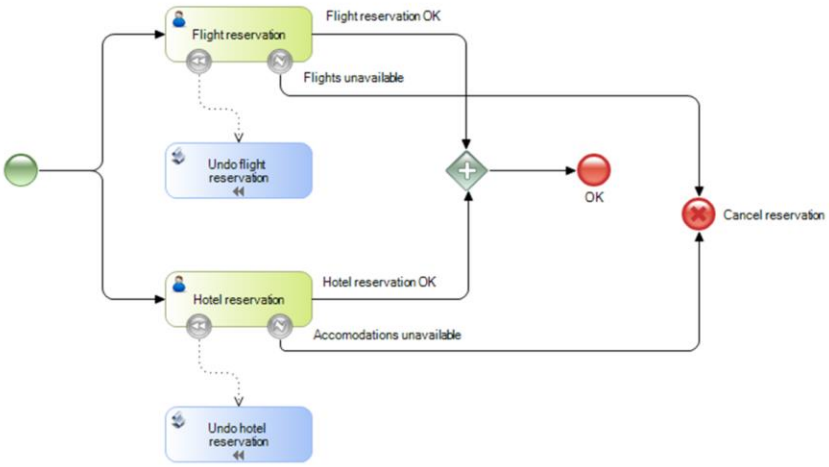
Modeling in GeneXus



Para levar em conta as possíveis falhas das reservas, adicionamos um Intermediate Error Event a cada tarefa e conectamos a saída dos mesmos a um Cancel End Event.

Este evento Cancel irá disparar quando uma das duas tarefas interativas falhar. O evento Cancel é um evento especial usado em processos transacionais, que é desencadeado **durante** a execução do subprocesso, em vez de ser disparado no final do processo.

Modeling in GeneXus



Quando a transação é cancelada, antes do controle passar para o processo pai, as **tarefas de compensação** são acionadas. Ou seja, todas as atividades que terminaram com sucesso até então devem ser desfeitas, executando as tarefas de compensação definidas para cada uma delas.

No nosso caso, devemos definir tarefas que desfaçam a reserva do bilhete aéreo ou a reserva do hotel, conforme o caso.

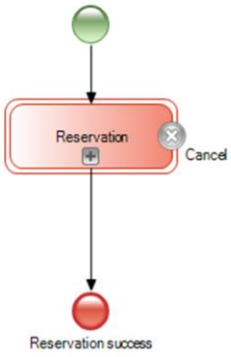
Para isso, adicionamos um Intermediate Cancel Event anexado à tarefa de reserva de vôo, adicionamos uma tarefa de script com o nome Undo flight reservation e a ligamos com o Intermediate Cancel Event.

Repetimos o mesmo para a tarefa de reservar um hotel, inserindo outro Intermediate Cancel Event e uma tarefa não interativa com o nome Undo hotel reservation, que conectamos ao Intermediate Cancel.

Observamos que, a partir do momento em que ligamos as tarefas de script com os Intermediate Cancel Event, o aspecto da conexão muda já que agora o conector é do tipo Association e aparece um símbolo "<<" acima das tarefas. Este tipo de conector é representado por uma linha pontilhada e permite estabelecer uma relação diferente daquela dos conectores de seqüência, uma vez que as atividades associadas não podem fazer parte de nenhuma seqüência de fluxo, ou seja, não podem ter conectores de seqüência de entrada ou saída.

GeneXus

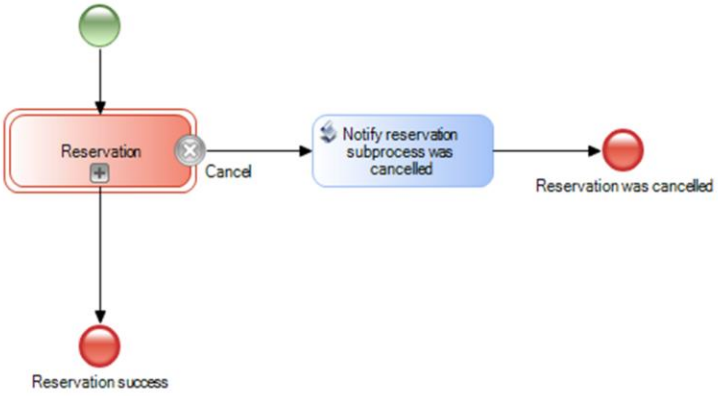
Modeling in GeneXus



Uma vez que a compensação da transação esteja completa, a execução no subprocesso é interrompida e um evento de cancelamento é desencadeado, redirecionando do subprocesso para o processo pai.

Para capturar este evento no processo pai, devemos anexar um Cancel Intermediate Event ao símbolo do subproceso.

Modeling in GeneXus



Ao receber o cancelamento no processo pai, informamos o usuário que o processo de reserva foi cancelado e terminamos o processo.

Como vimos anteriormente com o evento Error, o Cancel utiliza um mecanismo de throw-catch no Cancel End Event do subprocesso, através do Cancel Intermediate Event anexado ao subproceso, no diagrama pai.

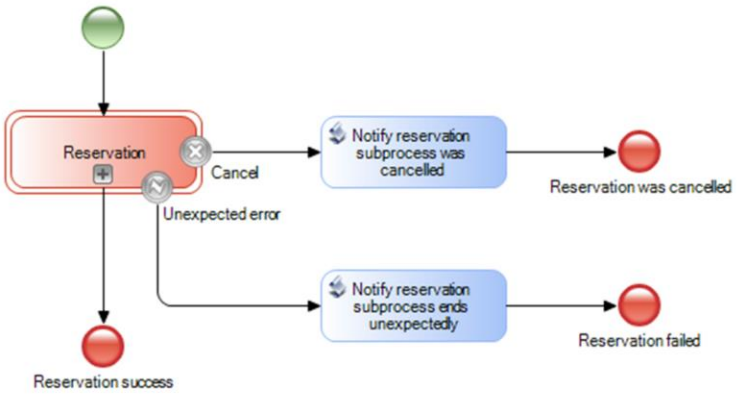
Este evento Cancel desencadeia automaticamente as tarefas de compensação que foram definidas no subprocesso, garantindo a integridade transacional da mesma.

Qualquer outro tipo de interrupção que o subprocesso sofra, como um evento de erro, a execução do mesmo é interrompida sem qualquer compensação.

Transactional Subprocesses

GeneXus

Modeling in GeneXus



http://wiki.genexus.com/commwiki/servlet/hwikibypageid?24922

Para fornecer um tratamento desse erro, inserimos um Intermediate Error Event anexado ao símbolo do subprocesso e, através de uma tarefa de enviamos um aviso de que o processo de reserva terminou de maneira inesperada.

Desta forma, vimos o funcionamento dos subprocessos transacionais e como são modelados no GeneXus.

Você pode encontrar mais informações no seguinte link:
http://wiki.genexus.com/commwiki/servlet/hwikibypageid?24922

Thank you!

Documentation:
genexus.com/gxflowdocs

GeneXus™ 15