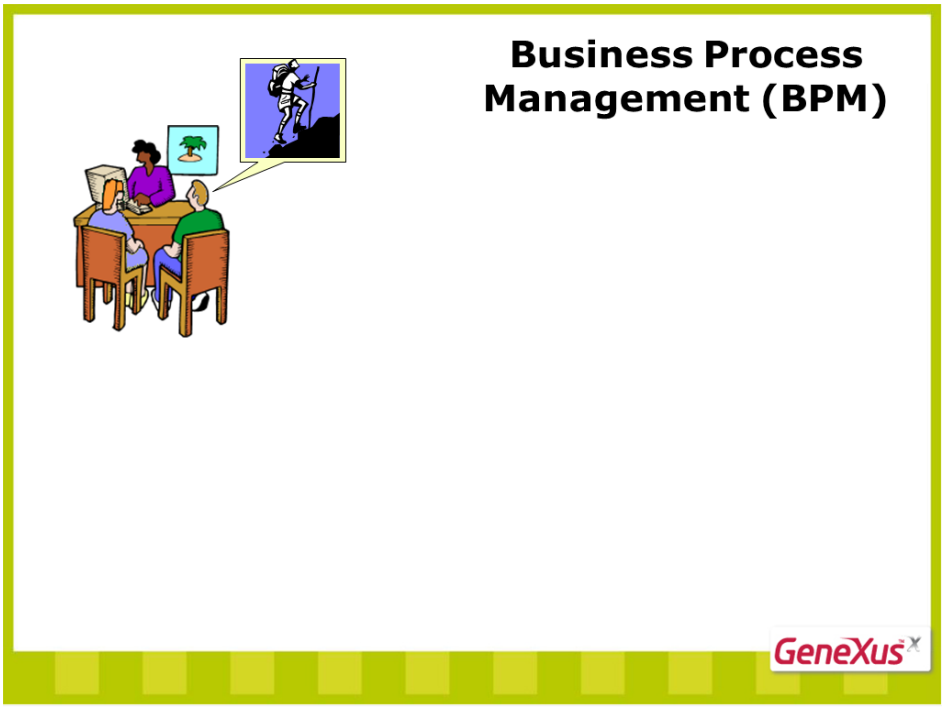
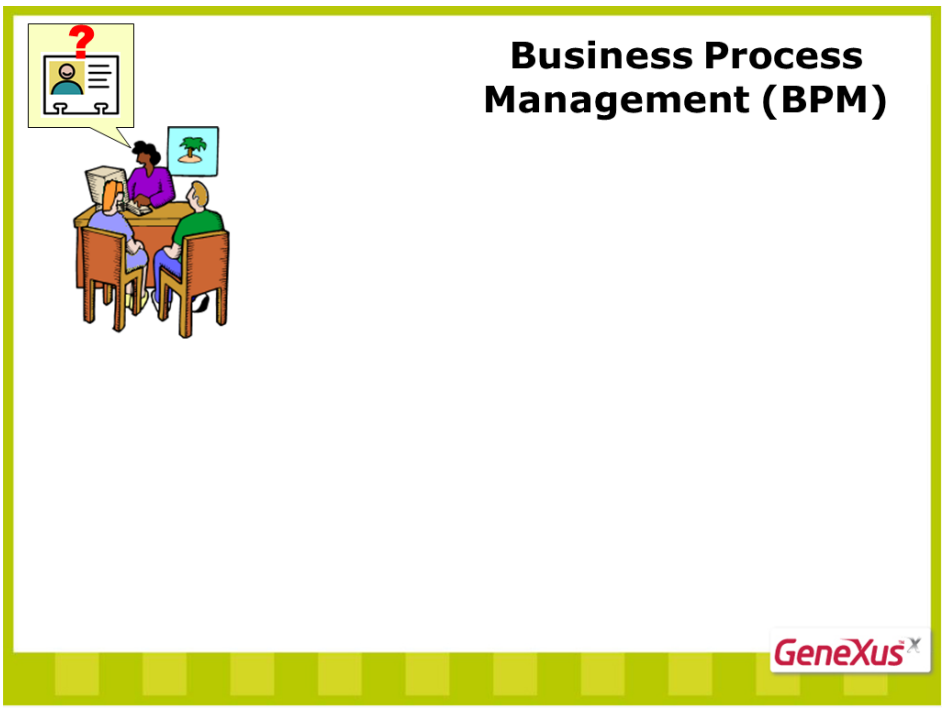
BPM (Business Process Management)

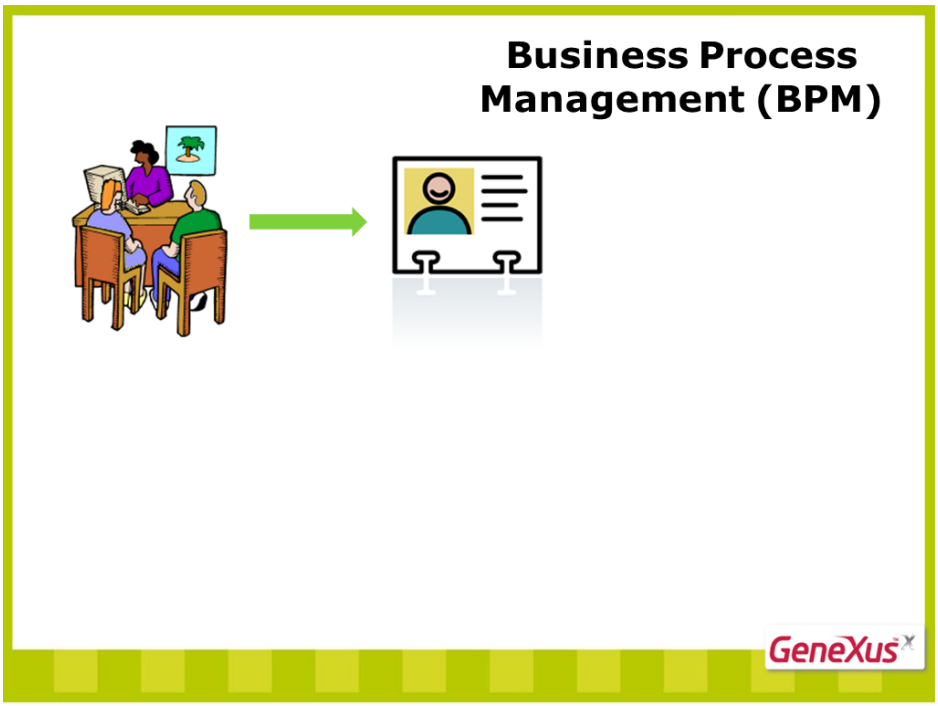
En la agencia de viajes, hay trámites que requieren de que varias tareas se realicen unas a continuación de otras, en un cierto orden, por ejemplo, cuando se desea hacer una reserva de un paquete turístico para una determinada atracción.



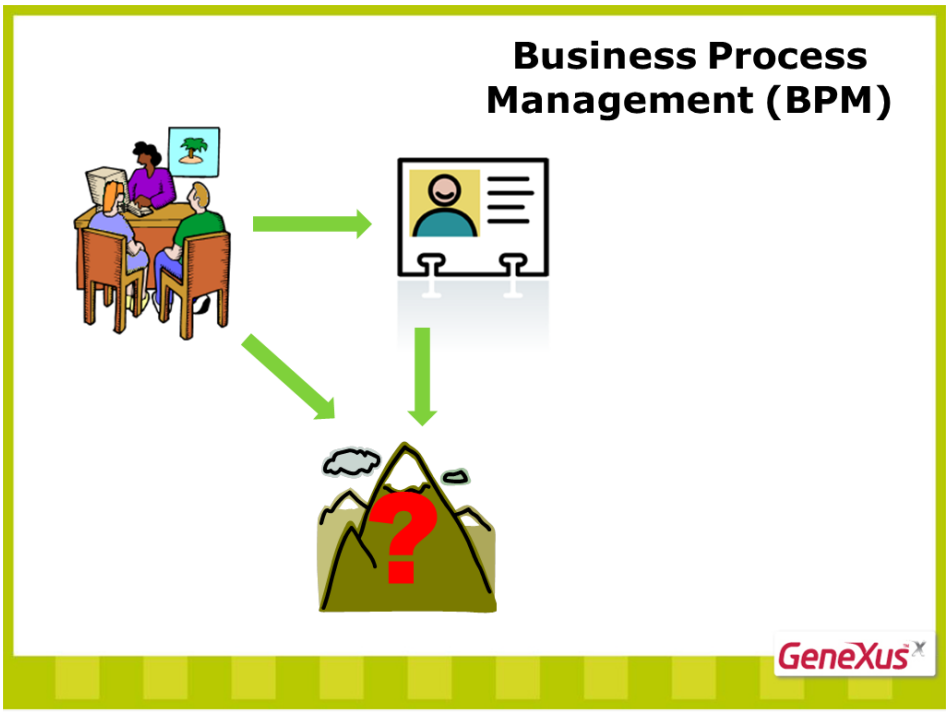
Cuando se ingresa la reserva, hay que verificar que la persona que solicite el paquete sea un cliente de la agencia



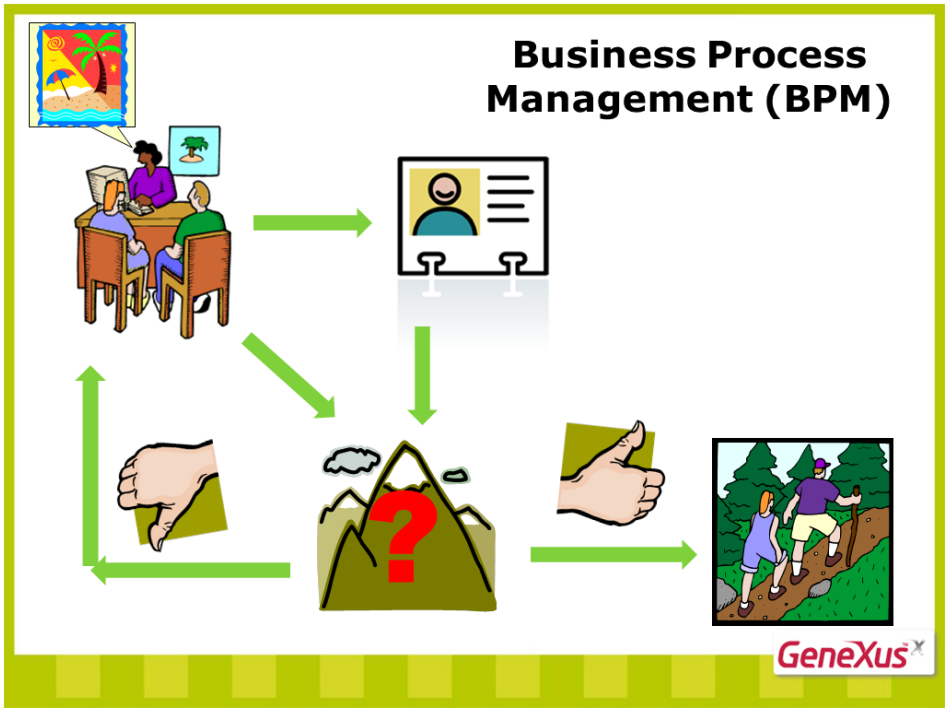
Y en caso de que no lo sea, se deberá ingresar como cliente.



Una vez completo este paso, es necesario verificar que haya paquetes disponibles para la cantidad de personas que deseen realizar el viaje.



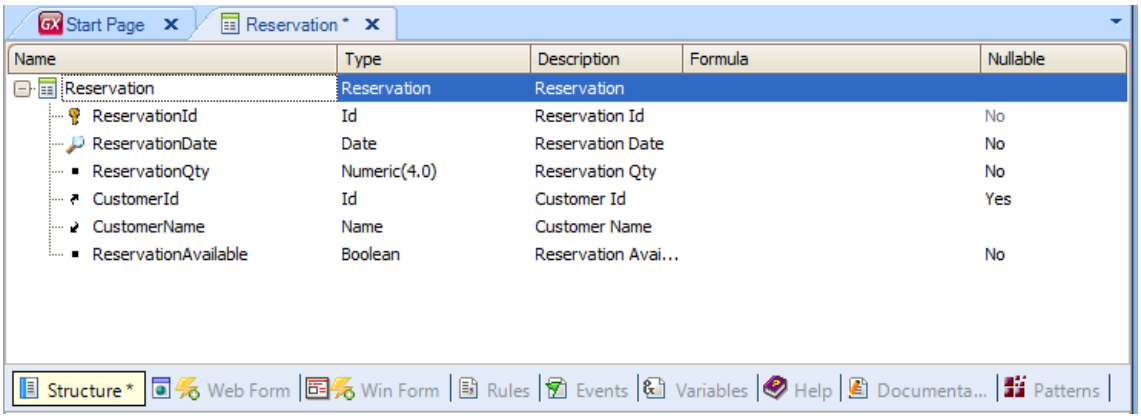
Si hay lugares disponibles, la reserva será asignada y en caso contrario, se le debe ofrecer al pasajero otro paquete.



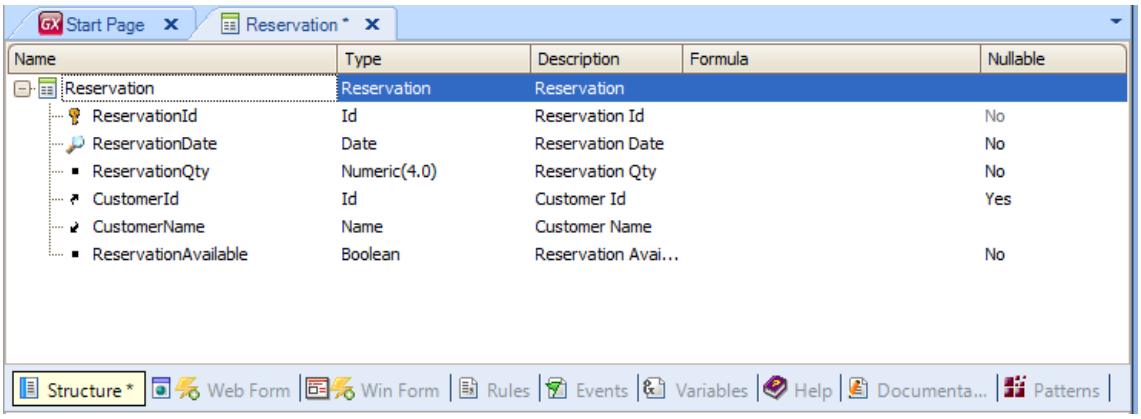
Una secuencia de pasos como la que acabamos de ver, es lo que se denomina proceso de negocio

En GeneXus podemos definir procesos de negocios como éste. Vayamos a hacerlo!

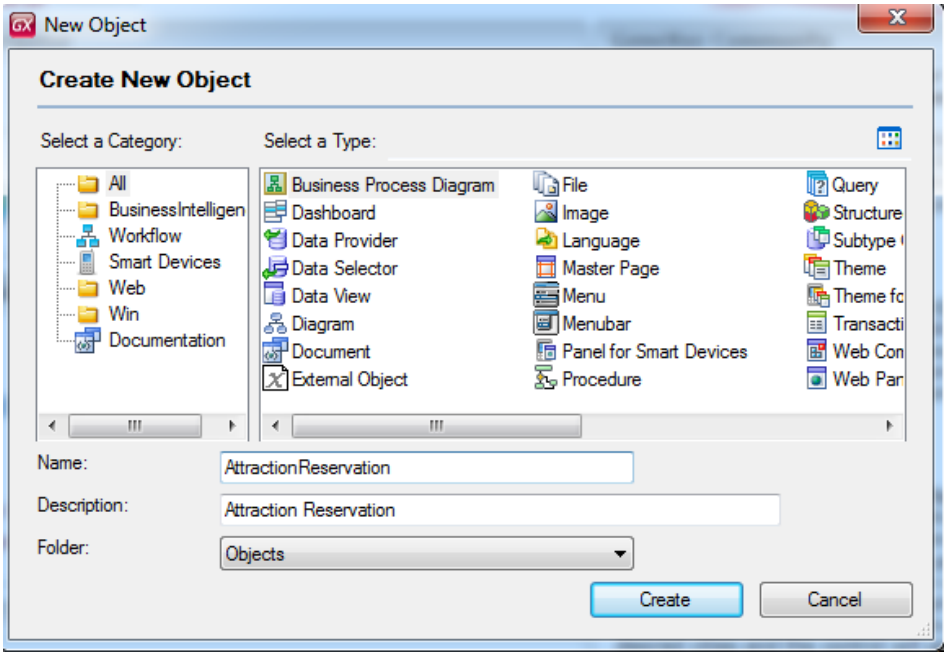
Como necesitamos registrar las reservas de paquetes turísticos, lo que hicimos previamente fue crear una transacción de reservas, con atributos para almacenar el identificador, fecha de la reserva, cantidad de personas que usarán el paquete, Identificador y nombre del cliente de la reserva y ReservationAvailable para marcar si el paquete turístico está disponible o no.



Para el atributo CustomerId, seleccionamos su columna Nullable con el valor Yes, para indicar que al momento de ingresar una reserva, podemos no tener aún al identificador de cliente, de la persona que está adquiriendo la reserva. …Salvamos…



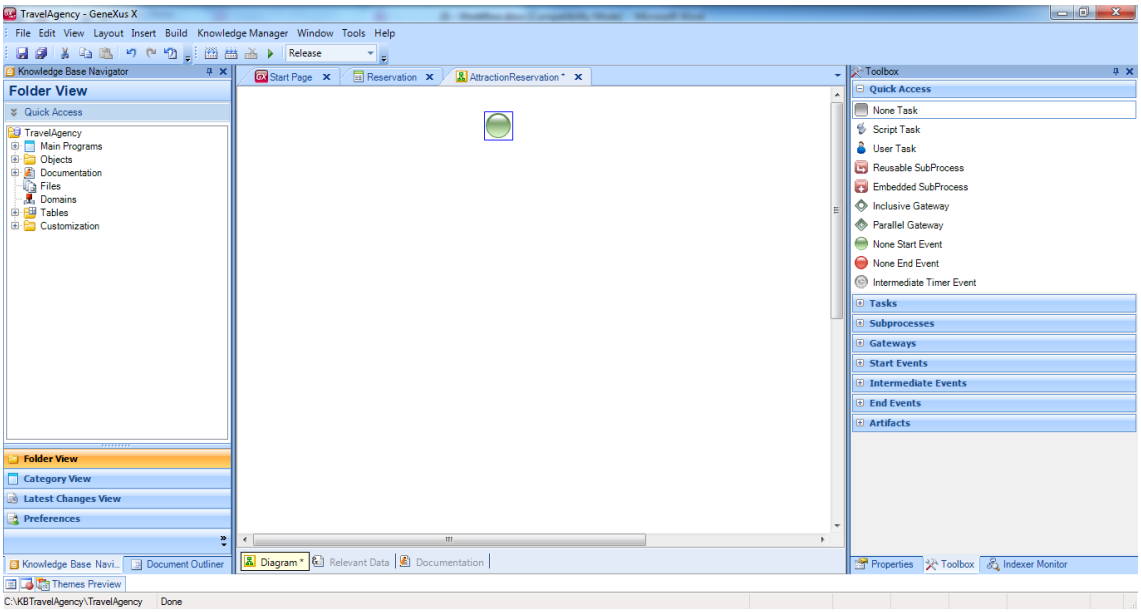
Ahora vamos a crear un objeto del tipo diagrama de procesos de negocios. Hacemos …File…New… y elegimos un objeto del tipo Business ProcessDiagram. Lo llamamos AttractionReservation.



Vemos una página en blanco del diagrama que vamos a construir.

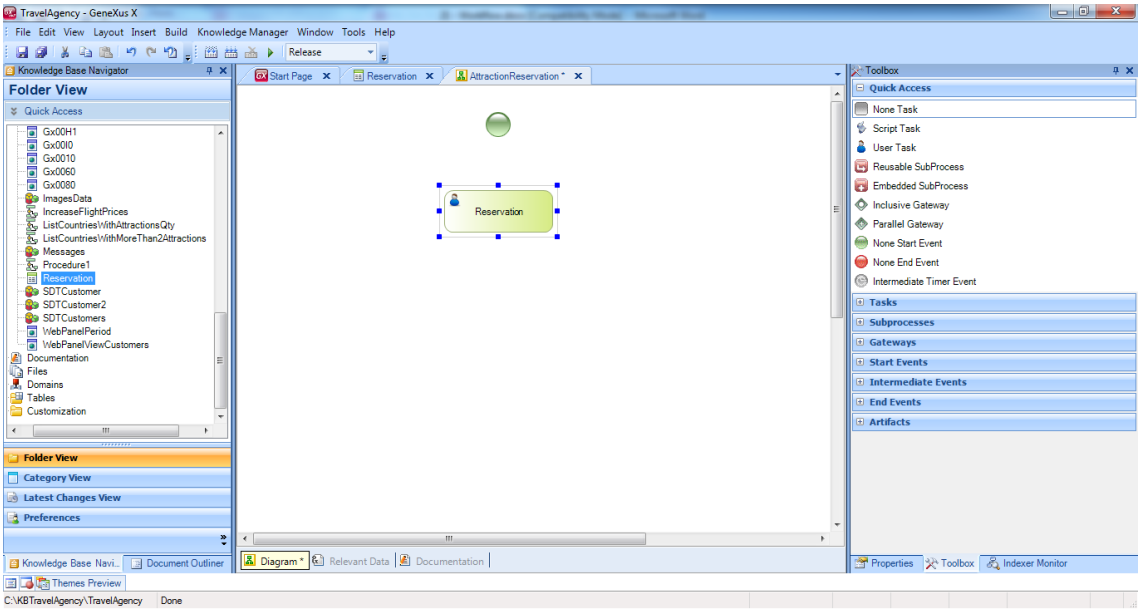
Si abrimos la toolbox, vemos que hay una gran cantidad de símbolos que podemos usar.

Para marcar el comienzo de nuestro proceso de negocios, arrastremos un símbolo de NoneStartEvent.

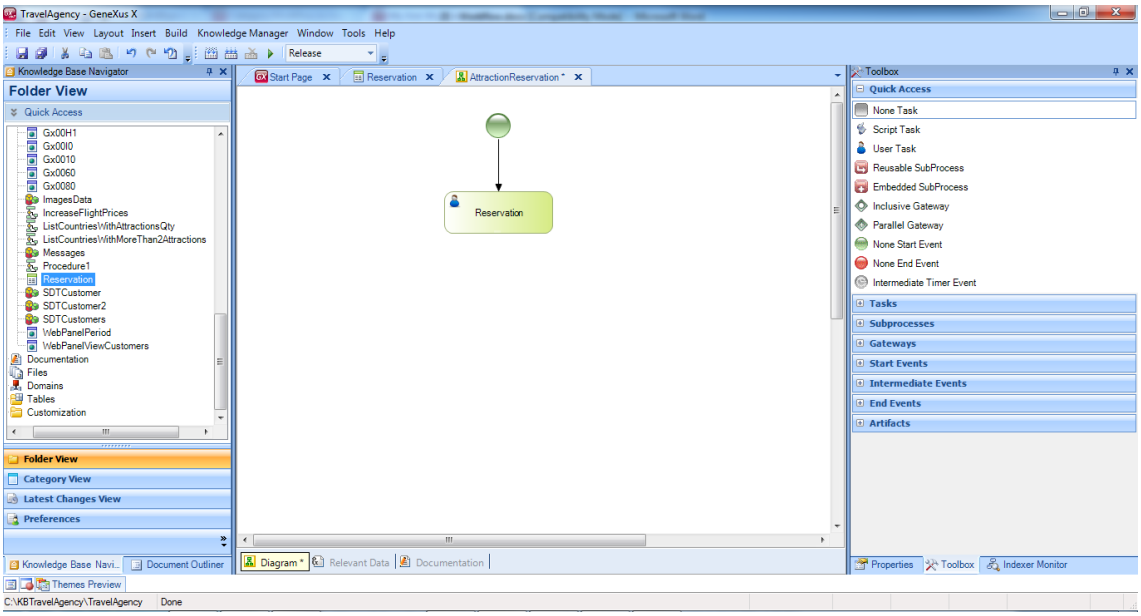


Como la primera actividad que vamos a realizar es ingresar una reserva, ubicamos a la transacción Reservation en la ventana de Folder View y la arrastramos hacia el diagrama.

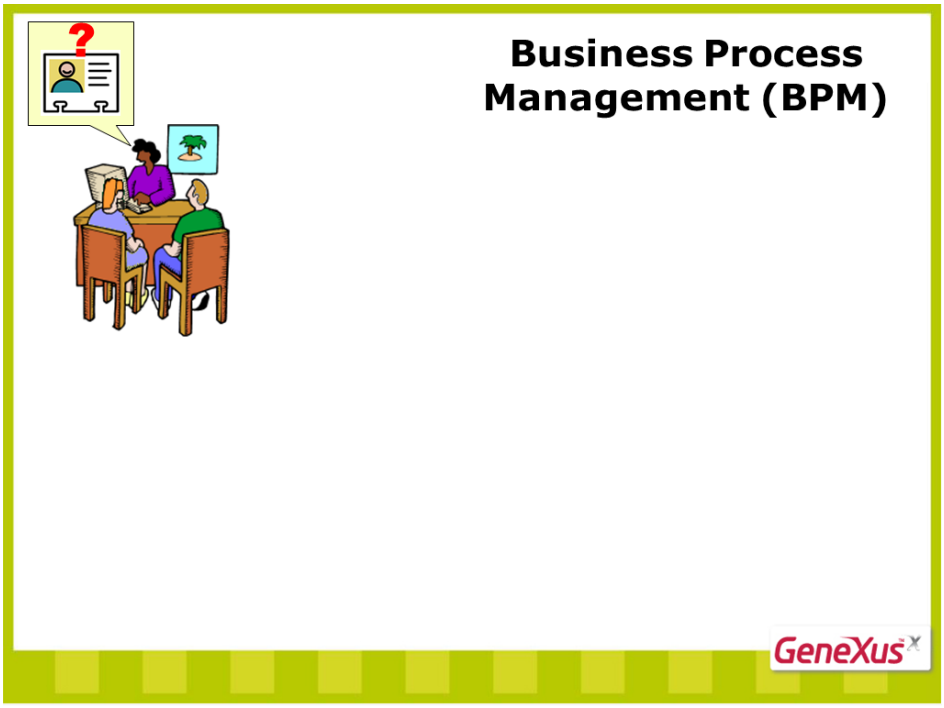
Vemos que se creó un rectángulo verde con el nombre Reservation y que tiene una figura humana en el extremo superior izquierdo. Esto nos indica que se creó una **actividad del tipo interactiva**, ya que la transacción permite que un usuario ingrese datos mediante la misma.



Para conectar el nodo de Start con la transacción, hacemos clic en la parte inferior del círculo verde y sin soltar, arrastramos hasta que la punta de la flecha toque el borde superior del rectángulo de la transacción.

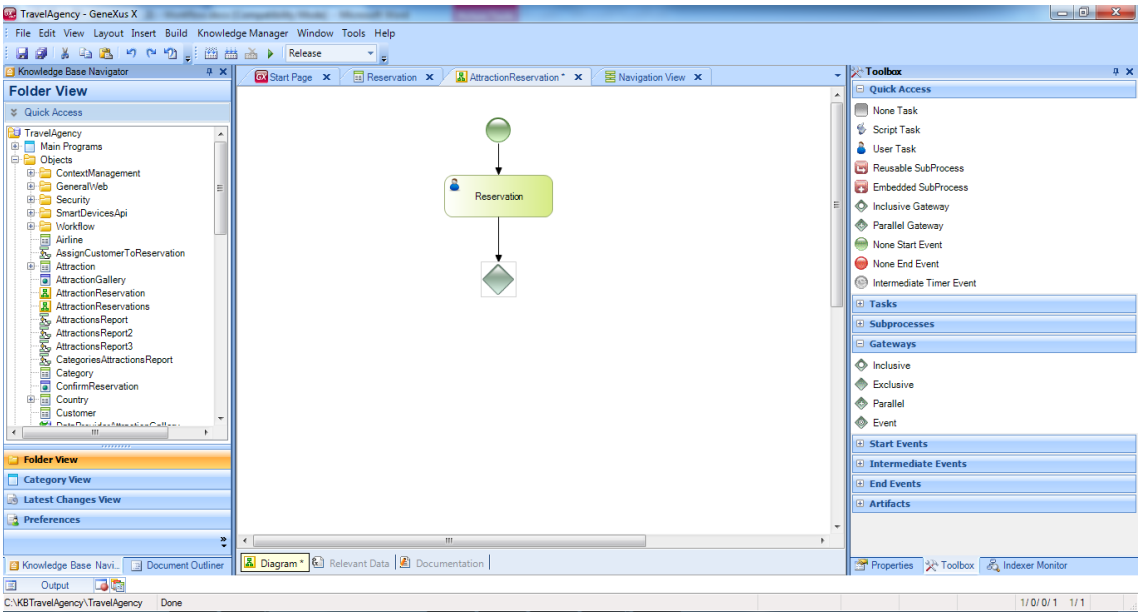


Según el proceso que siguen en la agencia de viajes, el sistema debe controlar si la persona que contrata la reserva, ya es cliente de la empresa o no.



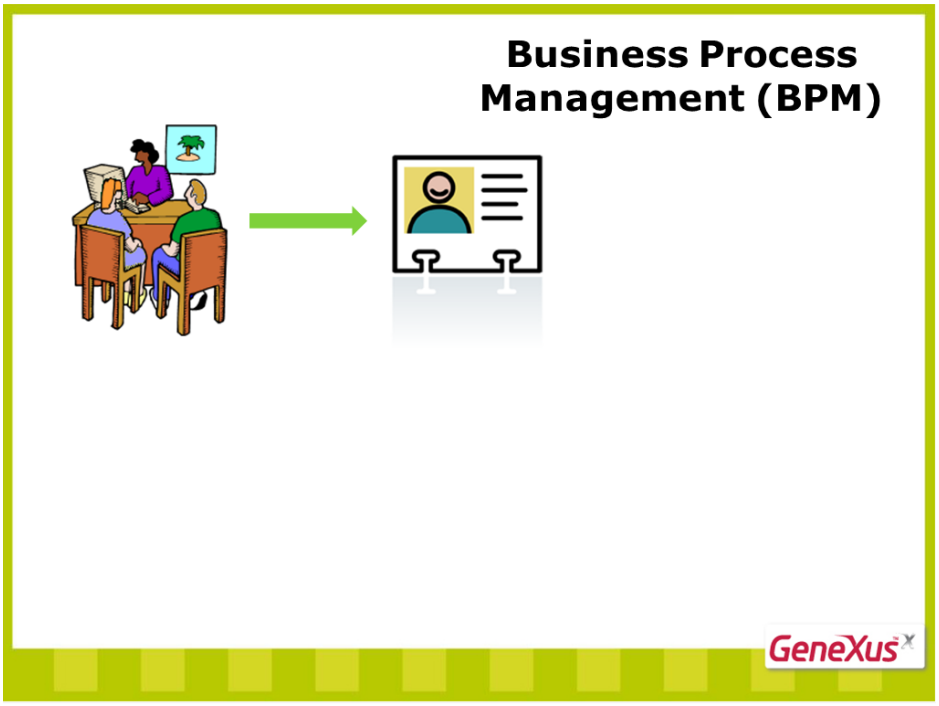
Para representar una **decisión** en el diagrama, vamos a la toolbar, hacemos clic sobre Gateways, arrastramos un nodo **Exclusive Gateway** sobre el diagrama y lo unimos desde la transacción Reservation.

Este tipo de nodo evalúa una condición y dependiendo del resultado, hace que el flujo siga hacia abajo (que sería el curso normal) o hacia la derecha del símbolo (que sería el curso alternativo).

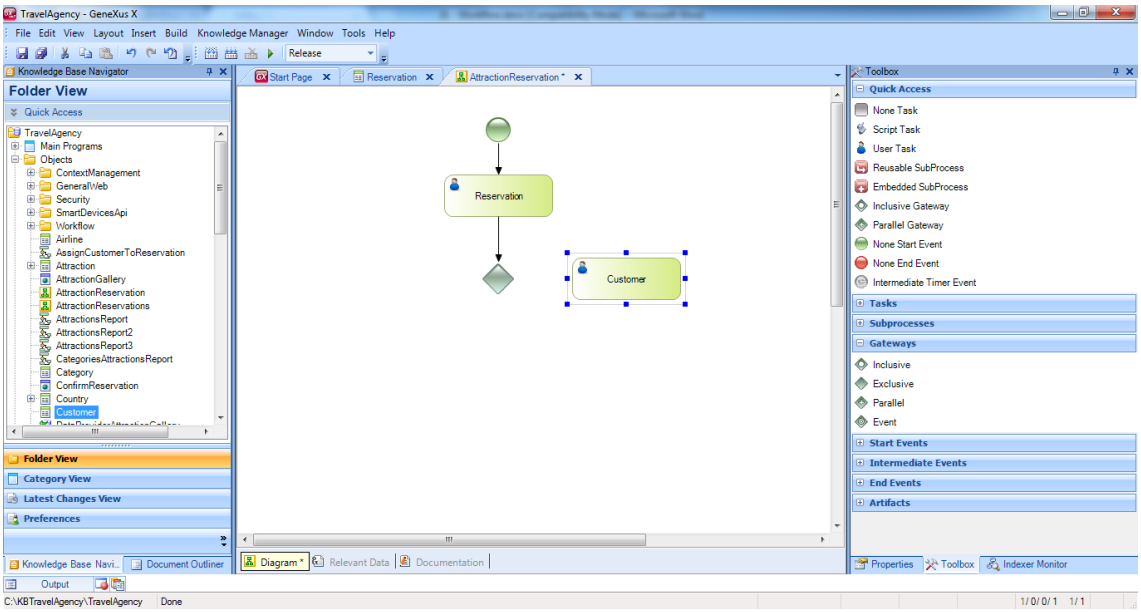


En nuestro ejemplo debemos definir la condición del Gateway que hará que el flujo siga el curso normal o el flujo alternativo, para agregar un cliente.

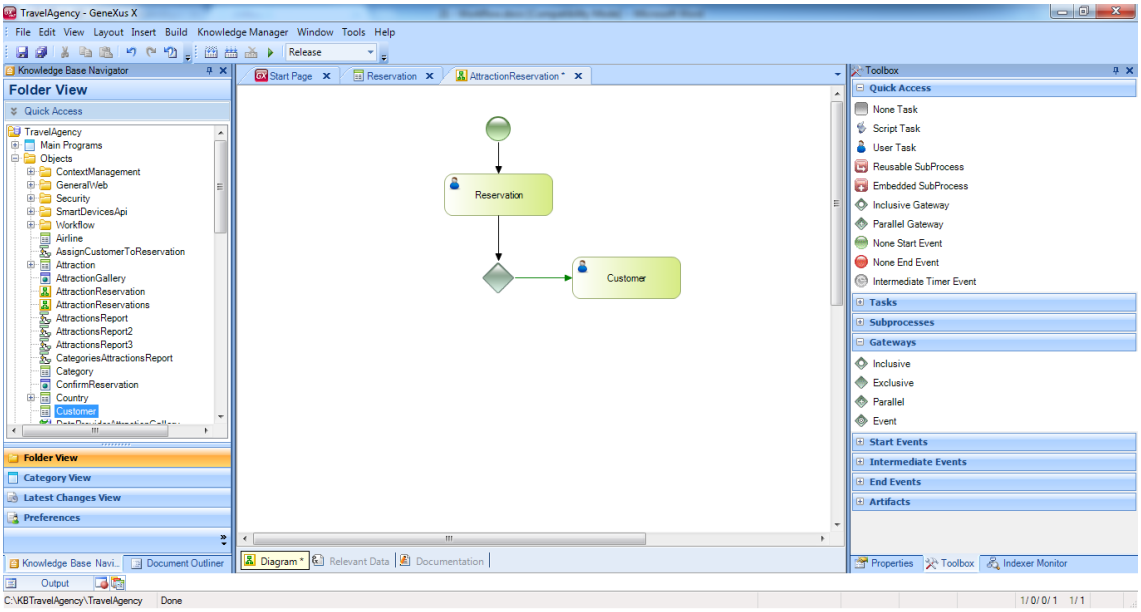
Es decir, tenemos que hacer que si la persona que desea hacer la reserva del paquete turístico **no es** cliente de la empresa, se invoque a la transacción Customer, para agregarlo como cliente.



Para comenzar, arrastramos a la transacción Customer al diagrama…



y la conectamos desde el nodo Exclusive Gateway…

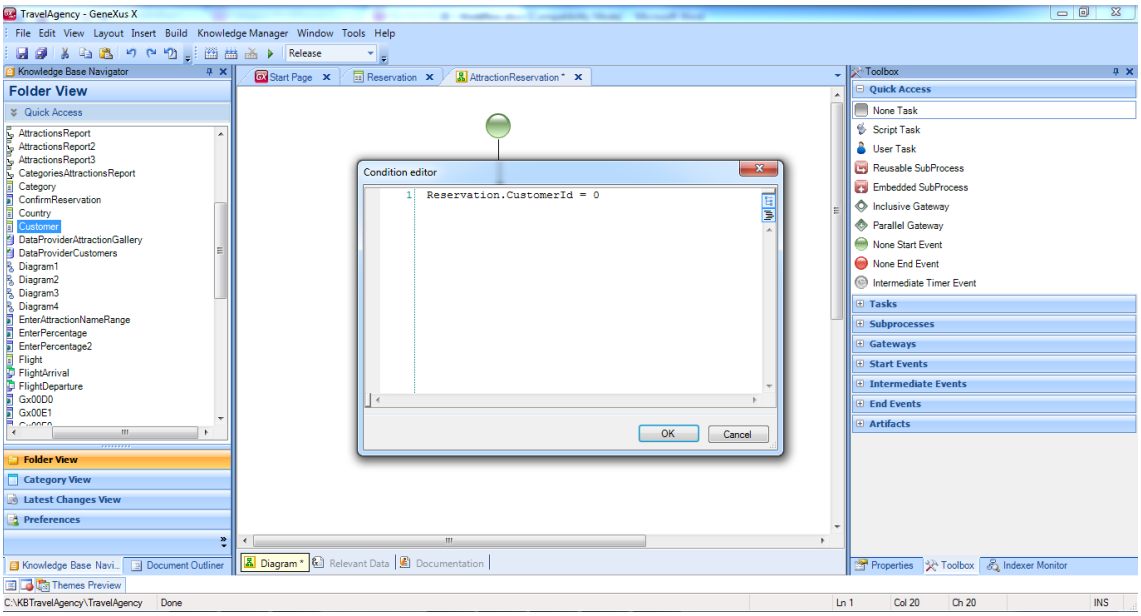


Para completar la definición de la decisión, tenemos que agregar la condición que hará que partiendo del nodo de decisión, el flujo del proceso siga hacia la derecha invocando a la transacción Customer, o siga hacia abajo hacia la siguiente tarea.

El camino hacia la derecha sería en nuestro caso el flujo alternativo, mientras que el camino hacia abajo sería al flujo habitual, es decir, cuando el que solicita la reserva ya es cliente de la agencia de viajes.

Para definir la condición que provoque la bifurcación, hacemos doble clic en la flecha verde que une el Gateway con la transacción Customer y vemos que se abre una ventana del **editor de condiciones**.

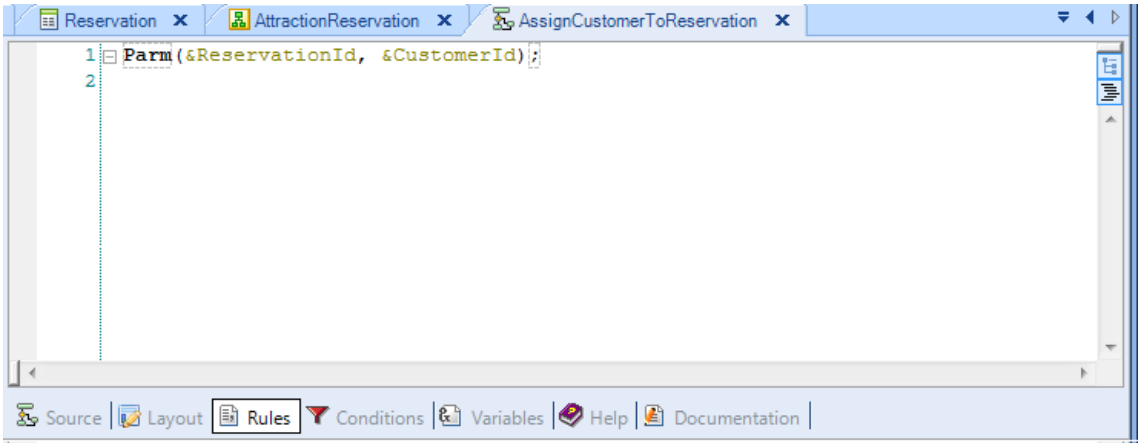
Escribimos la siguiente expresión: Reservation.CustomerId = 0, para indicar que el flujo debe tomar ese camino si cuando se ingresó la reserva, el atributo CustomerId se dejó con valor 0.



Si se sigue este flujo alternativo, se abrirá la transacción de Customer para que agreguemos a la persona de la reserva, como cliente.

Una vez que confirmamos el ingreso, debemos asociar el cliente recién creado a la reserva.

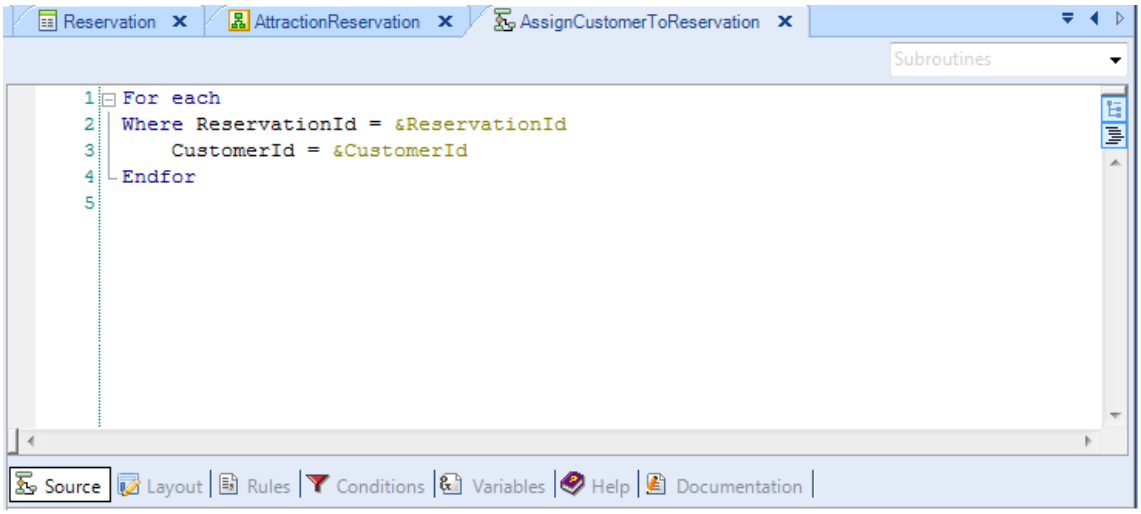
Para eso, debemos crear un procedimiento… le llamamos AssignCustomerToReservation y en la sección de reglas escribimos una regla parm, con los parámetros &ReservationId y &CustomerId…



Estas variables almacenan al identificador de la reserva que creamos al principio y al identificador del cliente que queremos asociarle a dicha reserva.

En la source escribimos…For each… WhereReservationIdigual … a la variable ReservationId que recibimos por parámetro. Luego asignamos a CustomerId el valor de la variable &CustomerId y cerramos el For Each.

De esta forma le asignamos el cliente recién cread, a la reserva ingresada previamente.

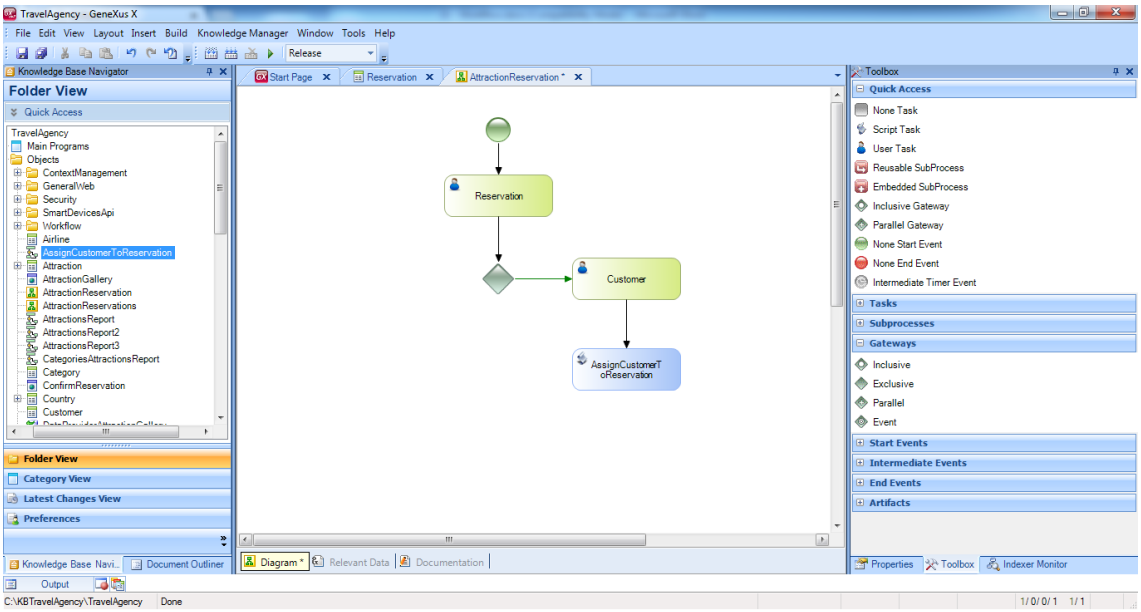


Salvamos y volvemos al diagrama.

Ahora arrastramos el procedimiento recién creado hacia el diagrama. Vemos que se nos creó un rectángulo color azul, que significa que la tarea que se creó es del tipo **no interactiva,** también llamada tipo **batch**.

Esto es así porque el procedimiento se ejecuta sin intervención de una persona, es decir que no hay interacción con el usuarioingresando datos, como en el caso de una transacción.

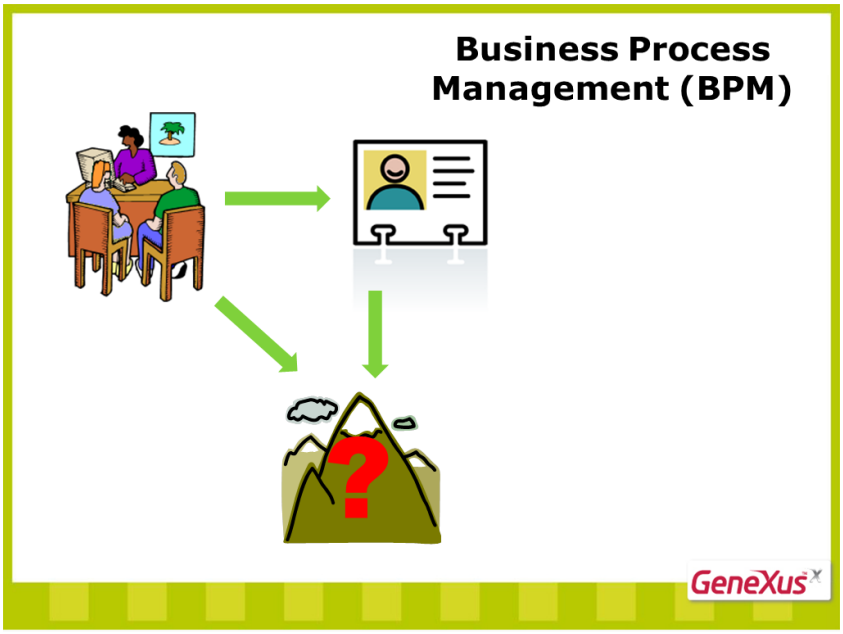
Por último, conectamos al procedimiento AssignCustomerToReservation desde la transacción Customer….



Muy bien… hasta ahora hemos descripto las tareas que se realizarán si se ejecuta el flujo alternativo, es decir, si hay que agregar un cliente nuevo y asignarlo a la reserva.

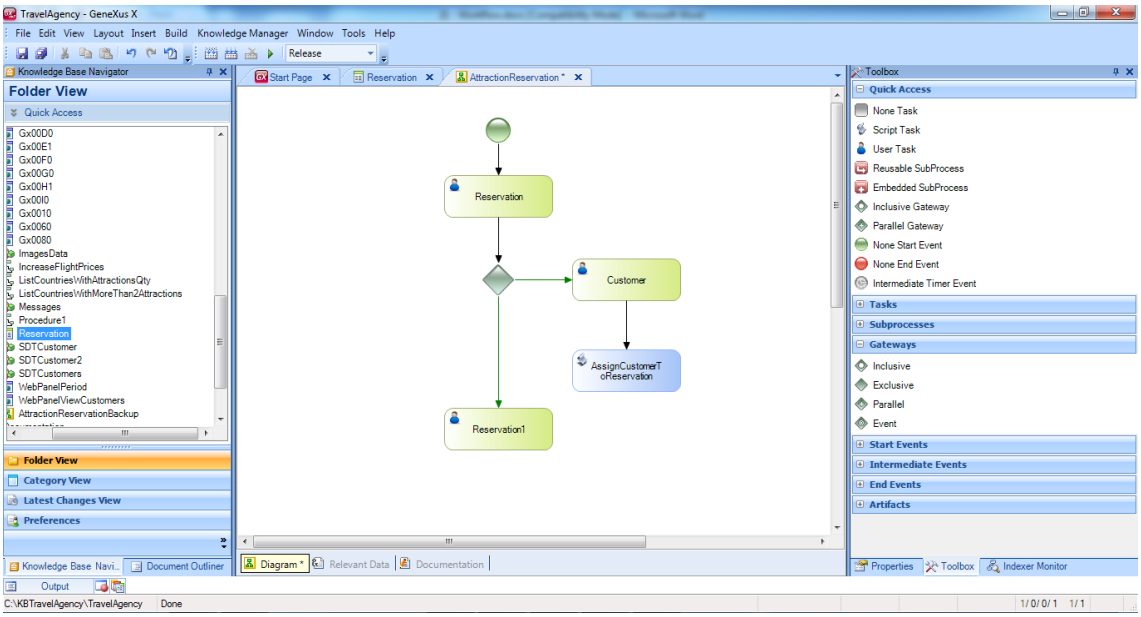
Ahora agregamos la actividad que se realizará si el flujo es el habitual, es decir, si no es necesario invocar a la transacción cliente.

En este punto, el empleado de la agencia debería verificar la disponibilidad de la reserva (por ejemplo viendo si hay vuelos disponibles, si quedan cupos en el paquete solicitado para las fechas previstas, etc.) y finalmente indicarle al sistema si la reserva puede completarse o no.

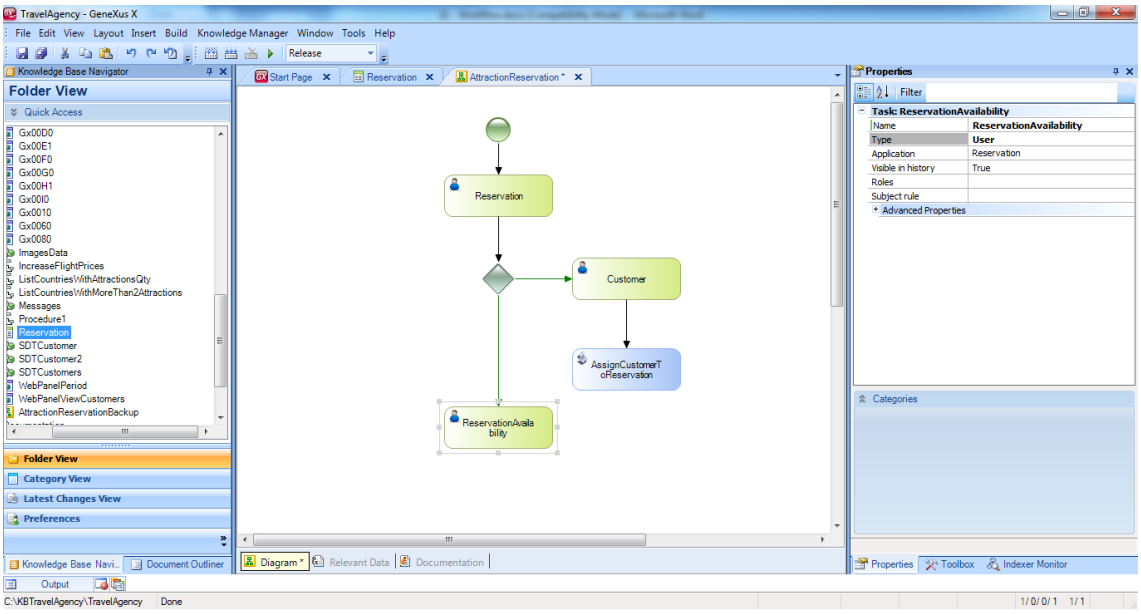


Para ingresar este dato, debemos contar con una pantalla donde podamos confirmar o cancelar la reserva. Para eso, podemos abrir nuevamente la transacción Reservation y marcar a la reserva como disponible o no, mediante el atributo ReservationAvailable.

Para esto, ubicamos a la transacción Reservation en la ventana de Folder View, la arrastramos hacia el diagrama y la conectamos desde el Gateway.

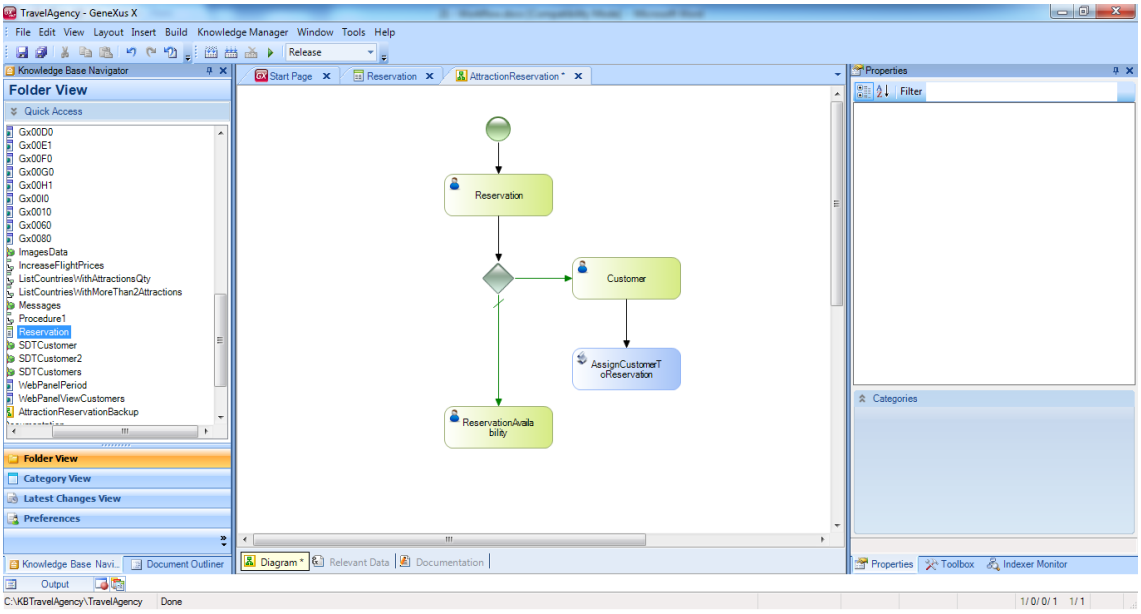


Vemos que se creó una actividad con el nombre **Reservation1**, así que presionamos F2 y cambiamos su nombre a ReservationAvailability. Notemos que aquí no le estamos cambiando el nombre a la transacción, sino a la actividad del diagrama que invoca a la transacción Reservation.

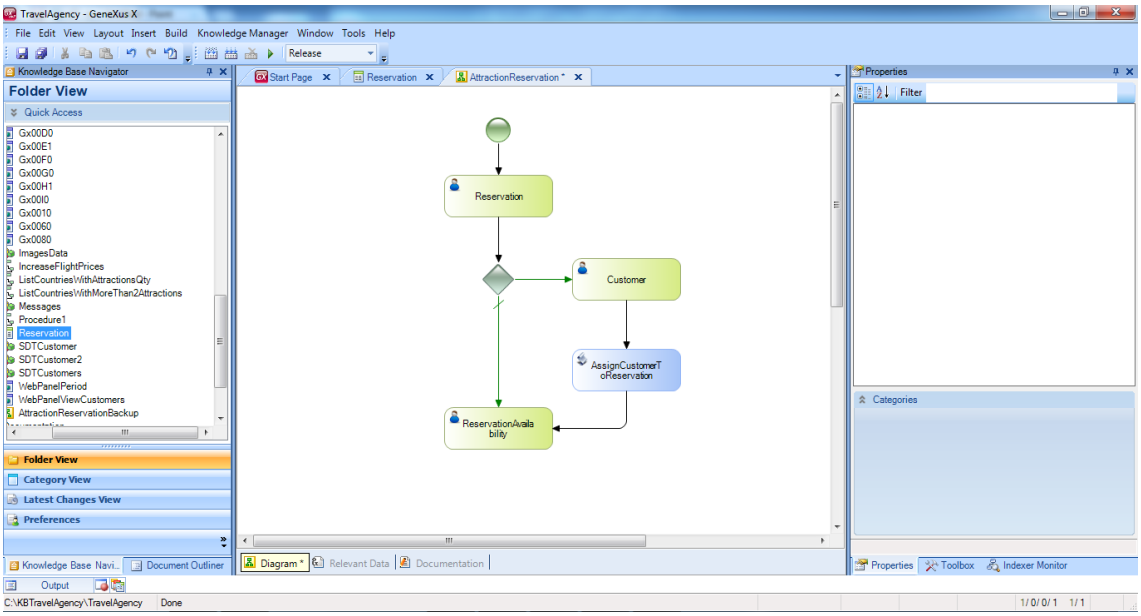


Cuando insertamos el Gateway que nos permitía tomar la decisión, dijimos que el flujo hacia la derecha que iba a la transacción Customer era el flujo alternativo y el flujo que saldría hacia abajo era el flujo normal, es decir cuando la persona que hacía la reserva ya era cliente de la agencia.

Para indicar que el flujo hacia abajo es el flujo normal, seleccionamos la conexión y en la ventana Propiedades ponemos su propiedad ConditionType en el valor Default. Observemos en el diagrama que el flujo quedó señalizado con una raya de color verde que cruza al mismo.

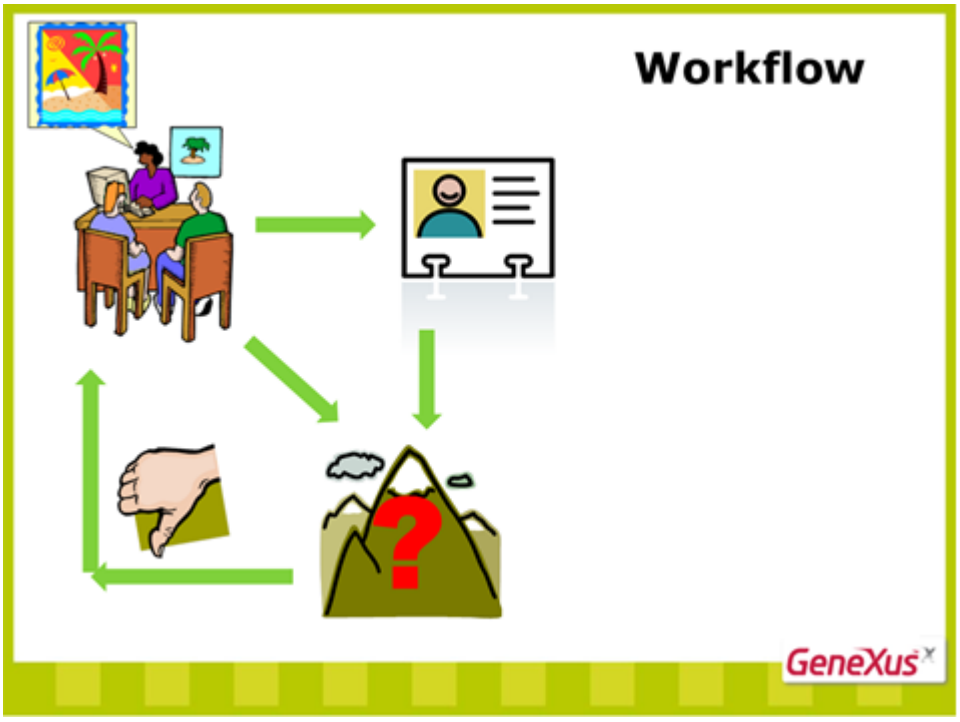


Aprovechamos ahora para conectar a la tarea AssigntCustomerToReservation a la tarea ReservationAvailability, ya que una vez completada la asignación del cliente, se debe continuar también con la verificación de la disponibilidad de la reserva.

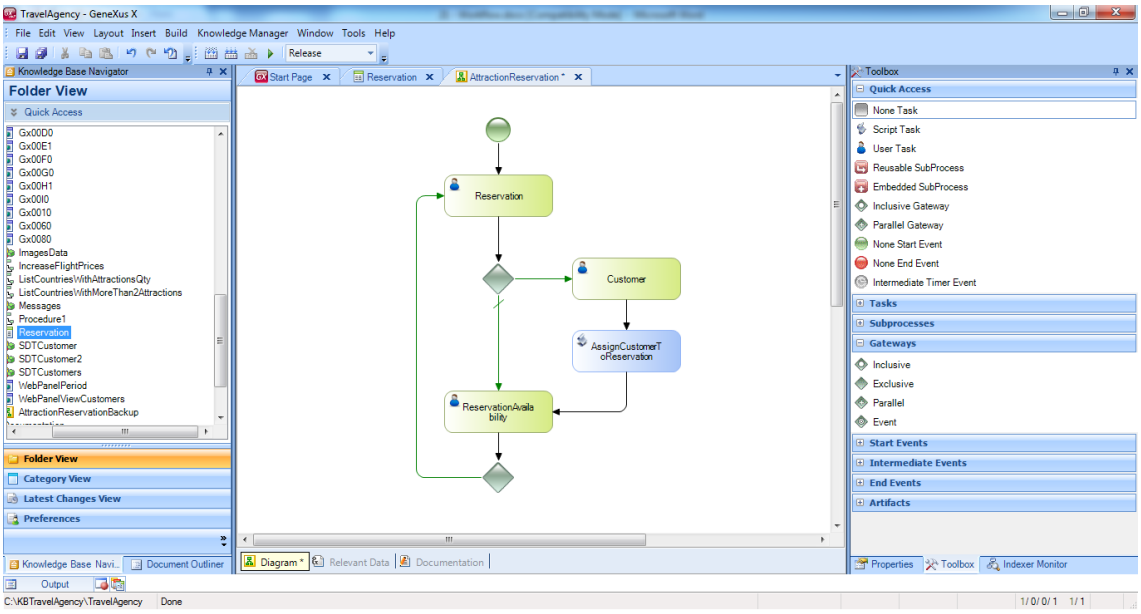


Continuemos con nuestro diagrama…

Una vez que finaliza la tarea ReservationAvailability, debemos evaluar el valor que le hayamos puesto al atributo ReservationAvailable. Si el check box se dejó desmarcado, significa que la reserva no estará disponible y se debe ofrecer al cliente un nuevo paquete para reservar.

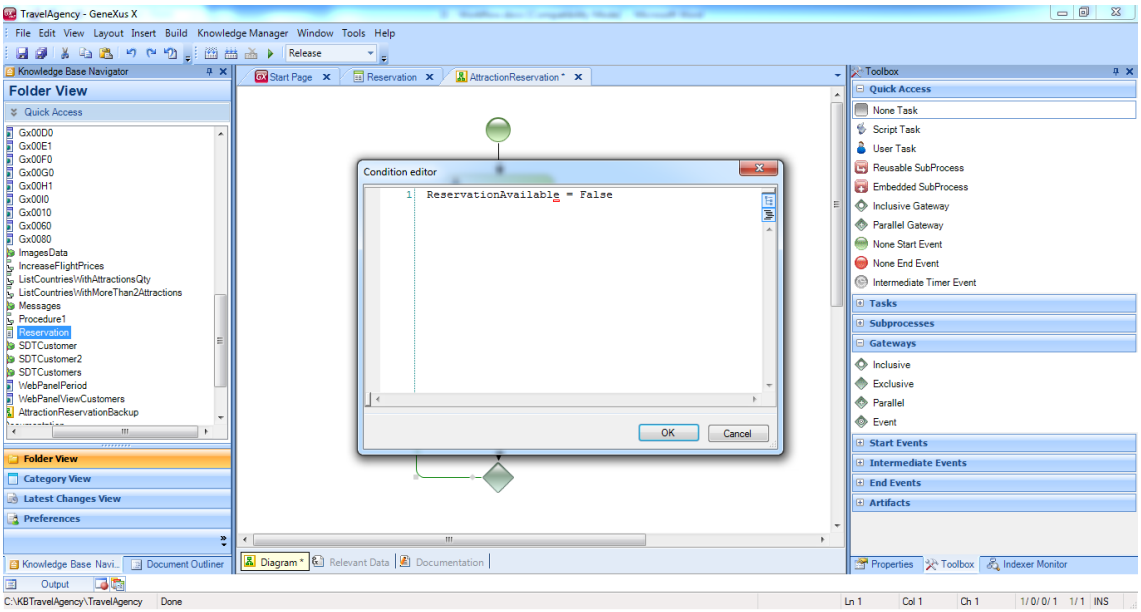


Para reflejar esto, insertamos desde la toolbar un Exclusive Gateway y lo conectamos desde la tarea ReservationAvailability. Luego conectamos el flujo alternativo del mismo (que lo dibujamos hacia la izquierda) a la tarea Reservation, para que se pueda agregar una reserva nueva.

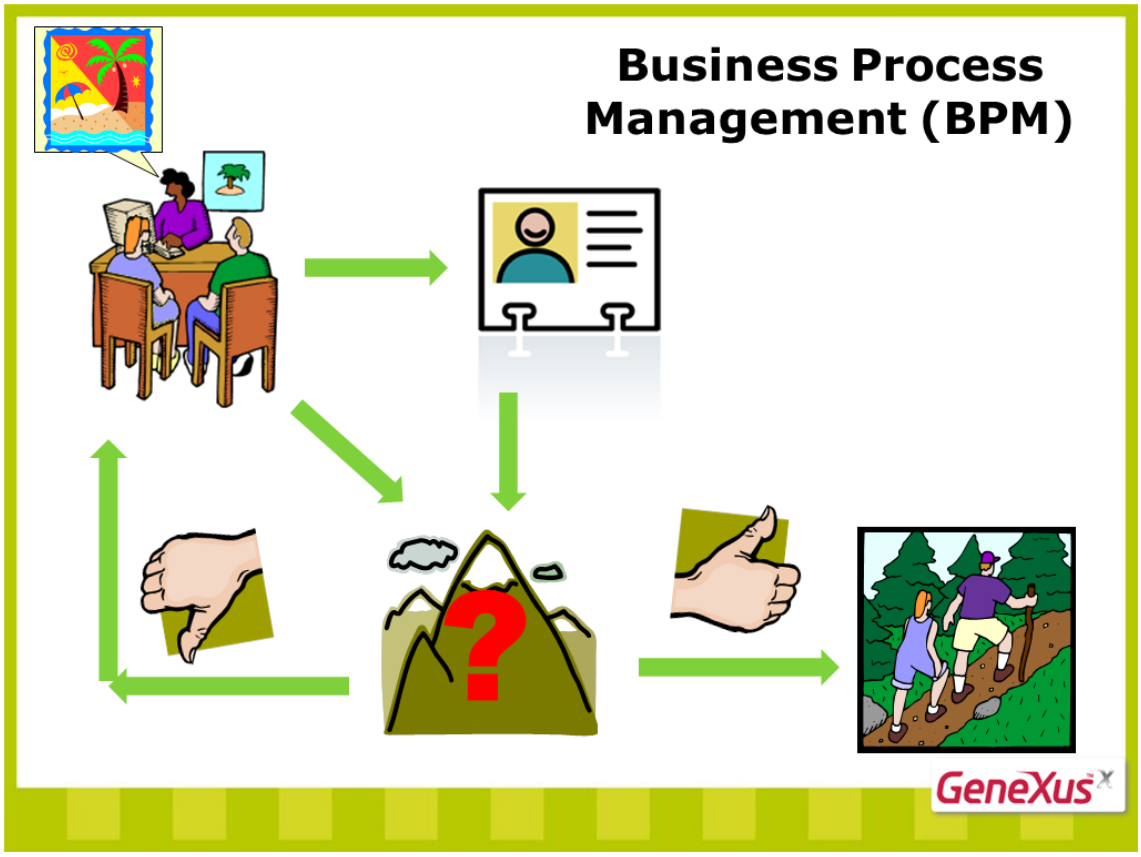


Bien. Ahora debemos ingresar la condición que debe darse para que el flujo alternativo se cumpla, es decir cuando la reserva no está disponible.

Para indicar esto en el diagrama, hacemos doble clic sobre la conexión que sale hacia la izquierda y escribimos ReservationAvailable=False.



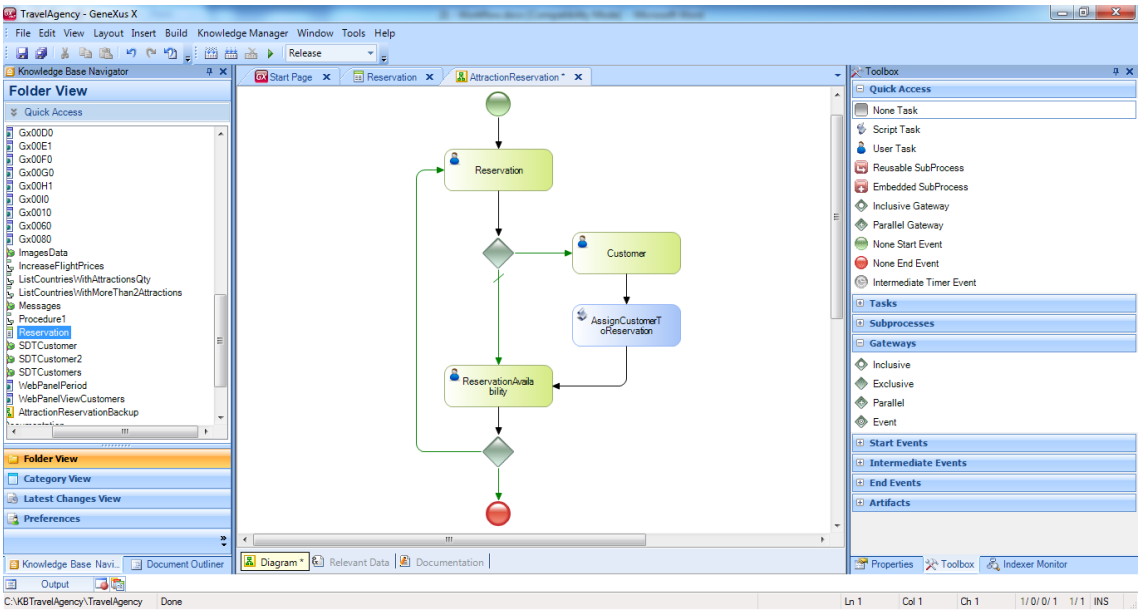
Muy bien, hasta aquí analizamos todo lo que sucede en el caso de que hubiera que ofrecer otra alternativa de viaje al cliente, ya que la reserva no se había podido confirmar. Lo que nos falta hacer ahora es tomar en cuenta el caso de que la reserva se confirme.



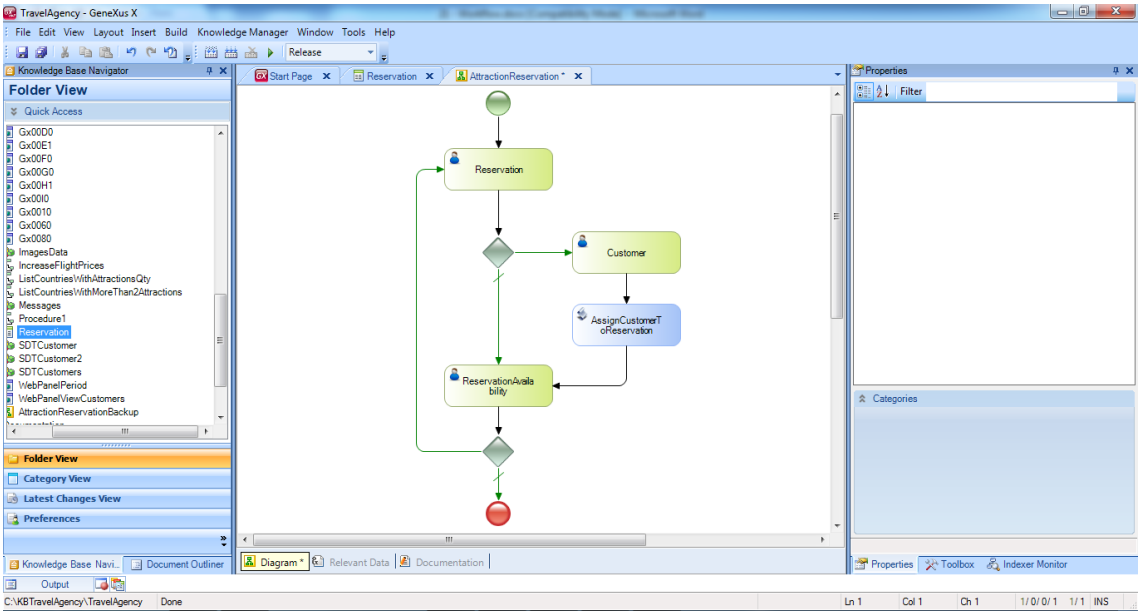
En este caso ya no habrá más tareas y deberá finalizar el proceso.

Para indicar que queremos terminar el diagrama, insertamos desde la toolbar un símbolo de NoneEndEvent y lo conectamos desde el Gateway.

Esta conexión hacia abajo es el flujo normal del Gateway, de modo que cuando la reserva se confirma, finalizará el proceso.



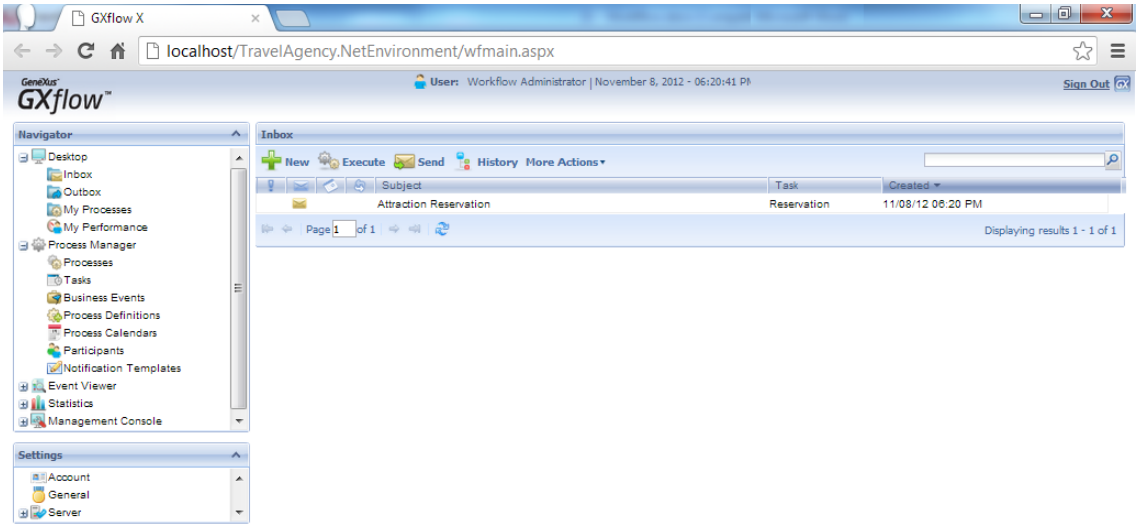
Para indicar esto, seleccionamos la conexión y en la ventana Propiedades ponemos su propiedad ConditionType en el valor Default.



Hasta aquí hemos finalizado el diagrama de nuestro proceso de negocio.

Para probar el funcionamiento del mismo, vamos a ejecutarlo. Sobre la solapa con el nombre del diagrama, damos botón derecho y elegimos Run…

En el análisis de impacto GeneXus nos informa que se va a crear la tabla Reservation, lo cual es lo esperado, así que procedemos a reorganizar y a continuar con la ejecución.

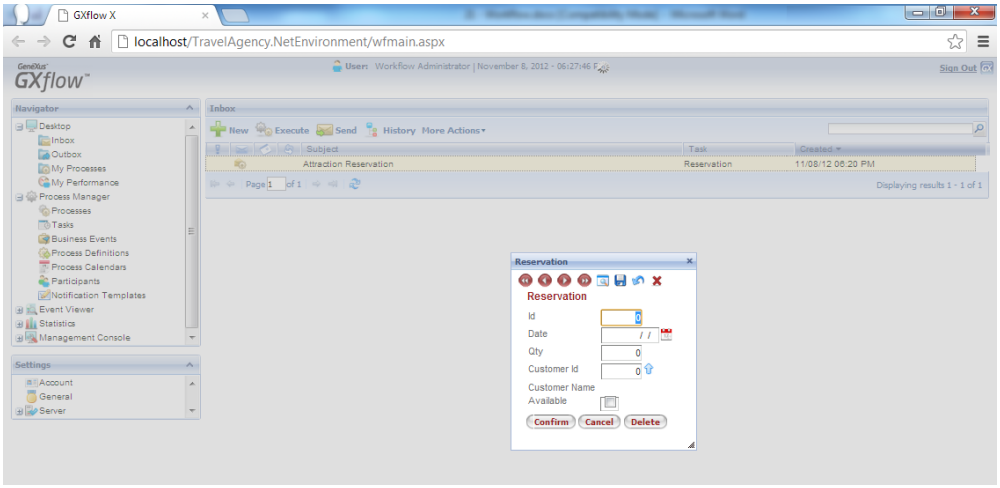


Vemos que se nos abre una pantalla con el Cliente GXflow, que se parece a una bandeja de entrada de e-mails, donde se muestra para cada usuario, las tareas que tiene pendiente de realizar, según el proceso definido.

Si observamos en esta bandeja de entrada, vemos un sobre cerrado y que en la columna Subject dice: AttractionReservation, indicándonos que tenemos definido ese proceso y en la columna Taks, tenemos pendiente la tarea Reservation.

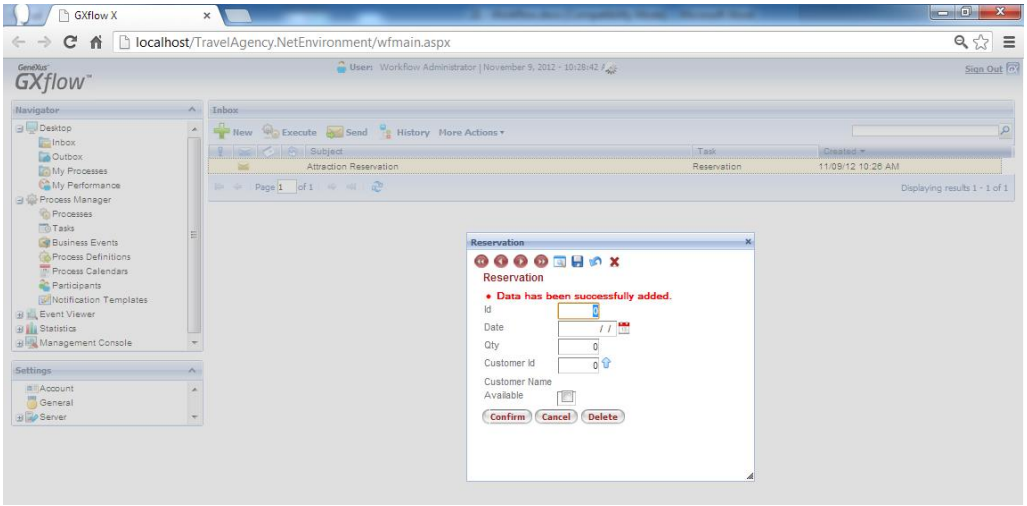
Para ejecutarla, seleccionamos la misma y presionamos el botón de Execute, o bien hacemos doble clic sobre la tarea…

Observamos que se abre la transacción Reservation, para que ingresemos la reserva.

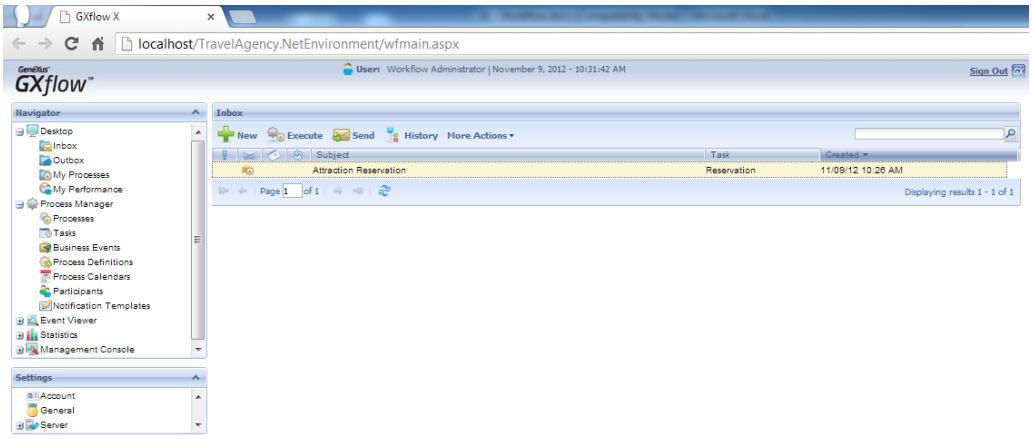


Dejamos el id en blanco porque está autonumerado, ingresamos la fecha del día….ponemos la cantidad de pasajeros en 2… y vamos a dejar el CustomerId en 0, porque la persona que desea hacer la reserva no es cliente aún de la empresa.

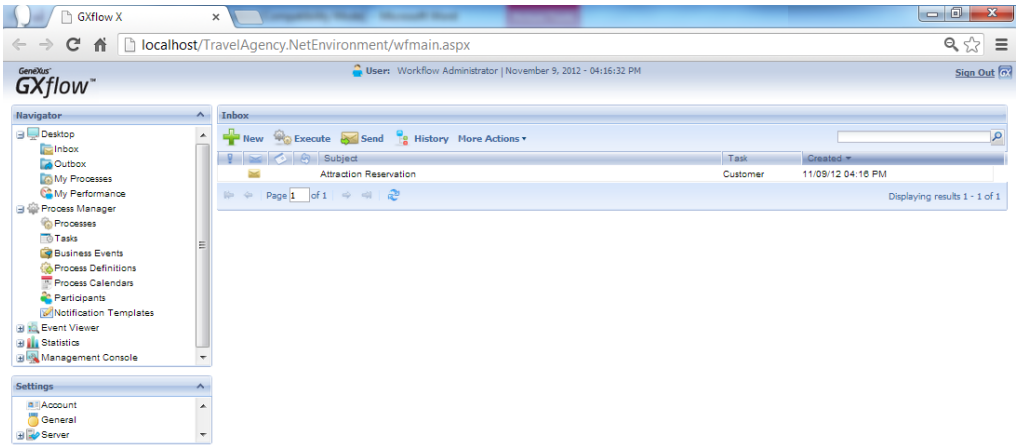
Presionamos Confirmar.



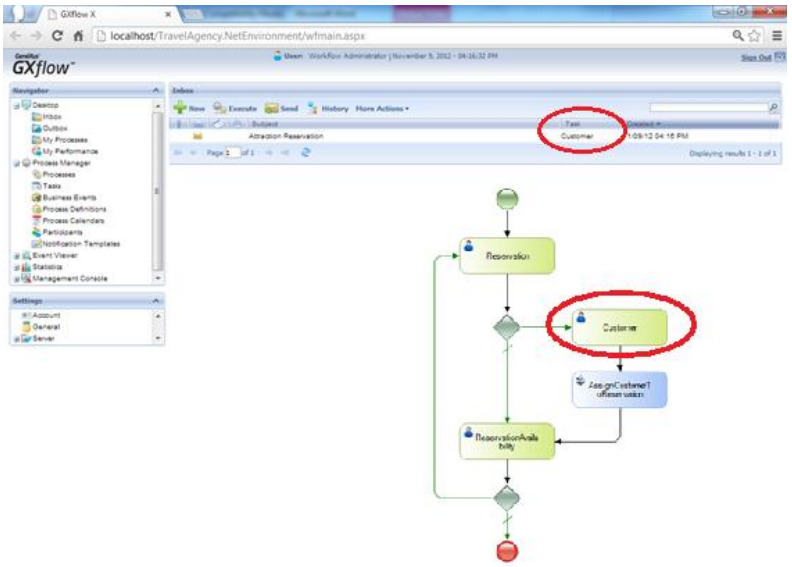
Vemos que GeneXus nos avisa que los datos se ingresaron correctamente, así que procedemos a cerrar la ventana con el símbolo de X.



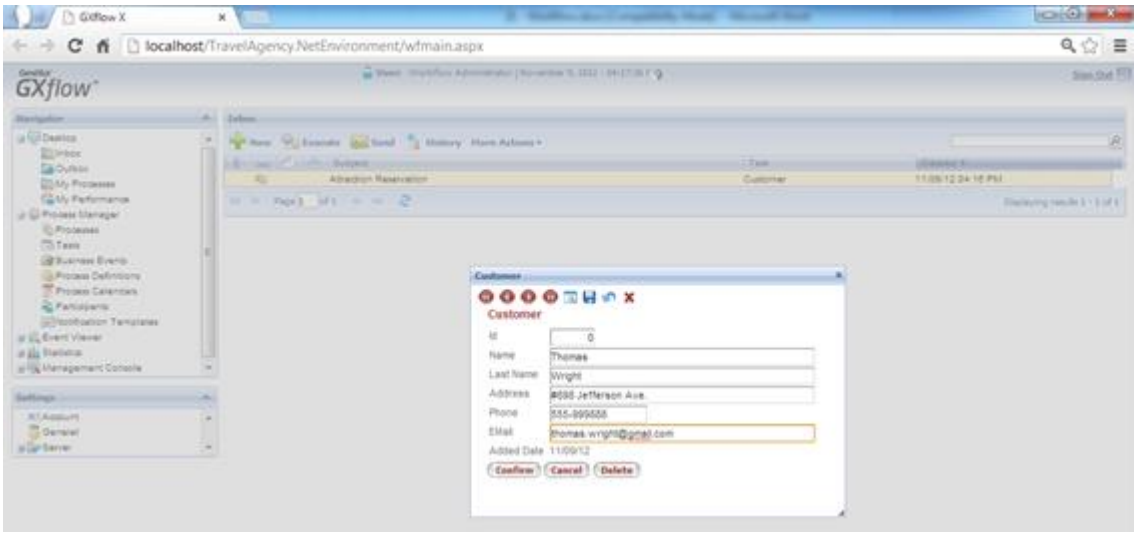
Ahora la tarea no tiene más un sobre cerrado, sino que el sobre está abierto, con un símbolo que muestra que la tarea ya fue ejecutada. Para pasar a la tarea siguiente presionamos Send.



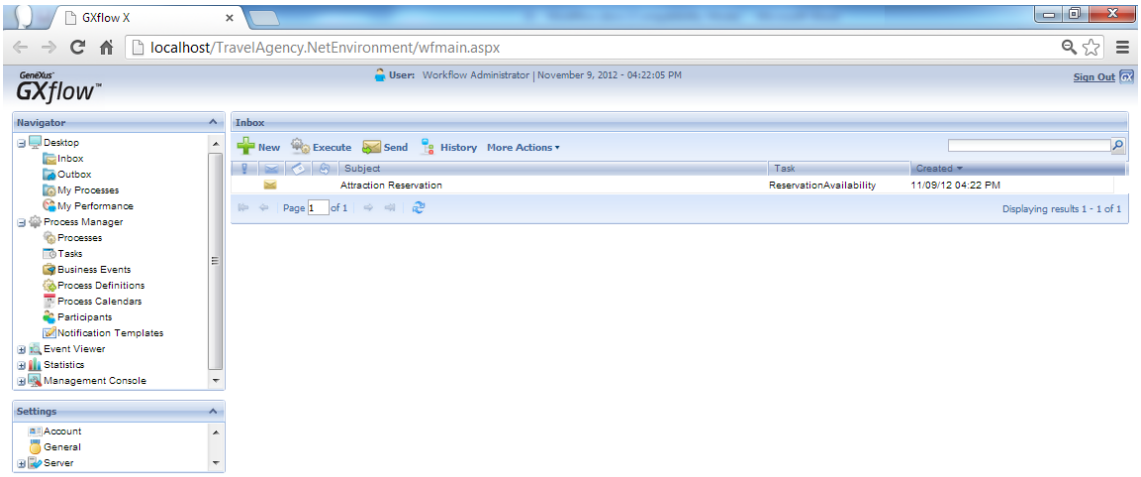
Ahora la tarea que tenemos pendiente de ejecución es la transacción **Customer**, dado que como no ingresamos identificador de cliente en la reserva, el flujo del diagrama seguirá hacia la derecha, abriendo la transacción Customer para que agreguemos a la persona, como cliente nuevo.



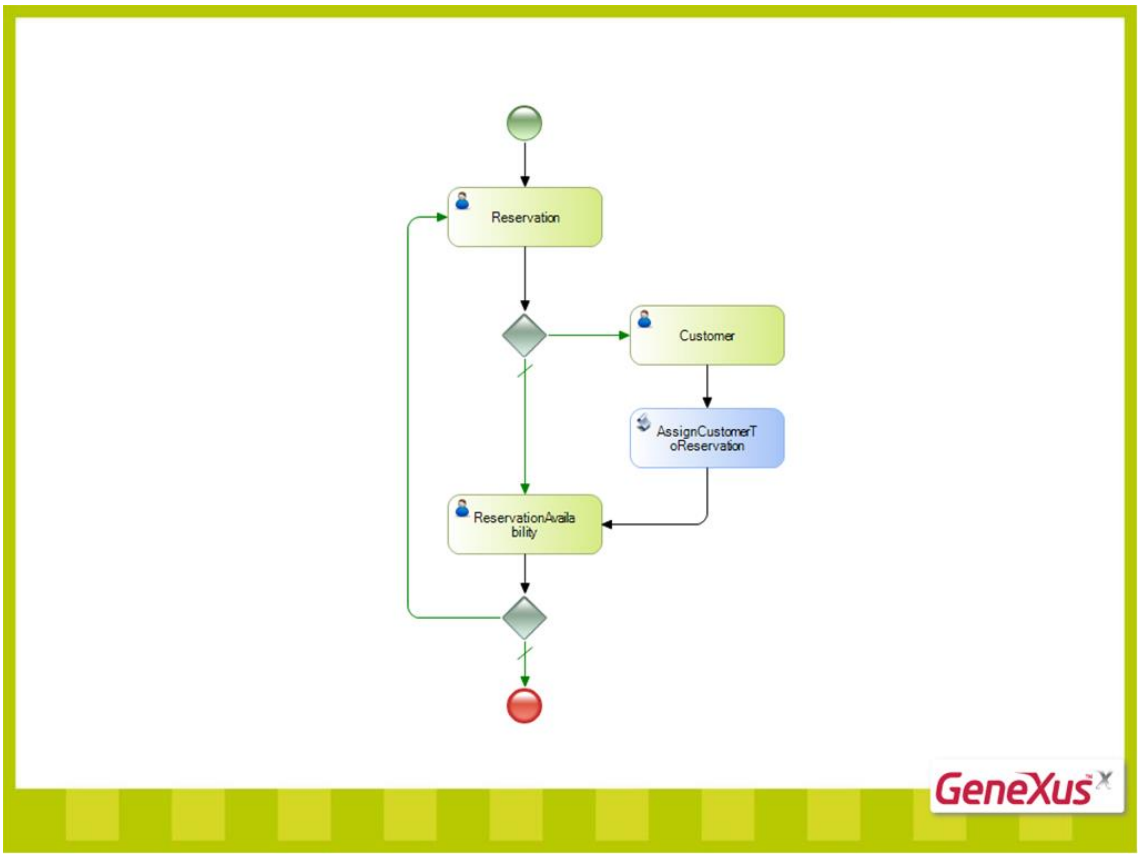
Ejecutamos la tarea, se abre la pantalla de la transacción Customer y procedemos a ingresar al cliente.



Confirmamos y cerramos la ventana…. Ahora presionamos el botón de Send para terminar la tarea y ejecutar la siguiente.

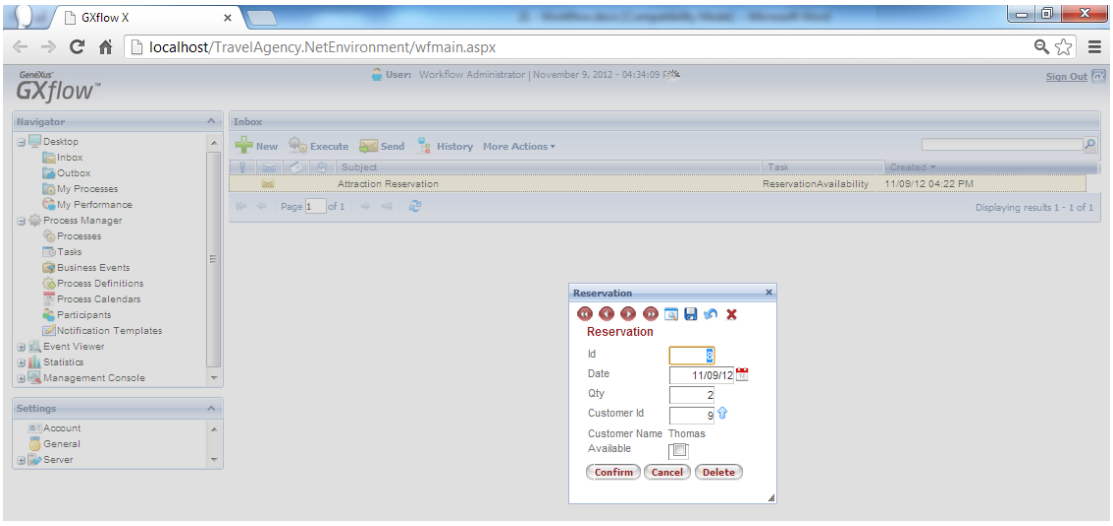


Vemos que la tarea próxima que tenemos es la llamada ReservationAvailability. Si recordamos el proceso, la tarea Customer invocaba a una tarea llamada AssignCustomerToReservation, que invocaba a un procedimiento para asignar el identificador del cliente recién ingresado a la reserva.

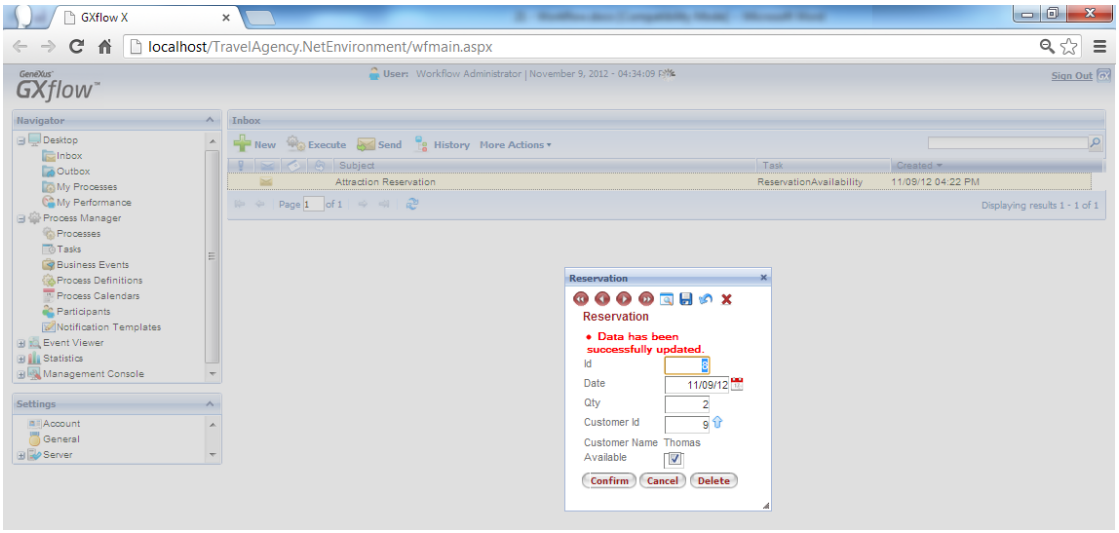


Como la tarea AssignCustomerToReservation es no interactiva, no se nos presentó en la bandeja de entrada cuando finalizamos la tarea Customer, sino que directamente se nos mostró la próxima tarea en la que tenemos que intervenir, es decir ReservationAvailabilty.

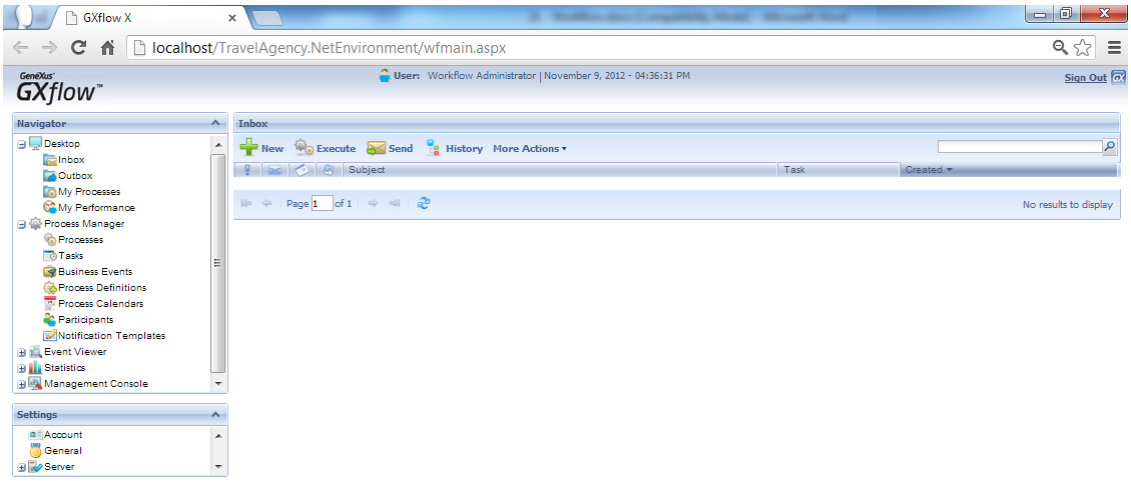
Vamos a volver a la ventana de GXflow y ejecutamos la tarea ReservationAvailability, haciendo doble clic sobre la misma.



Ahora se nos abre el formulario de la transacción Reservation, para que ingresemos si la reserva está disponible o no. Supongamos que sí, así que marcamos el check box y presionamos Confirmar.

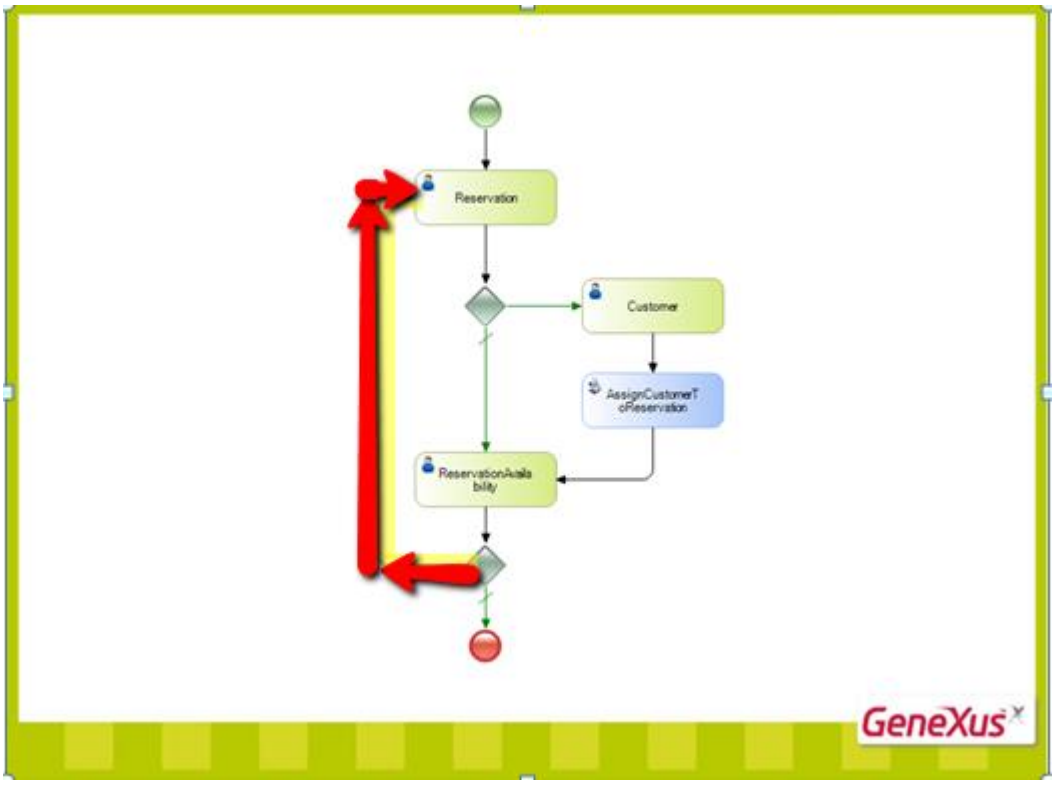


Vemos que los datos se ingresaron correctamente, así que cerramos la ventana de la transacción y presionamos el botón Send para finalizar la tarea.



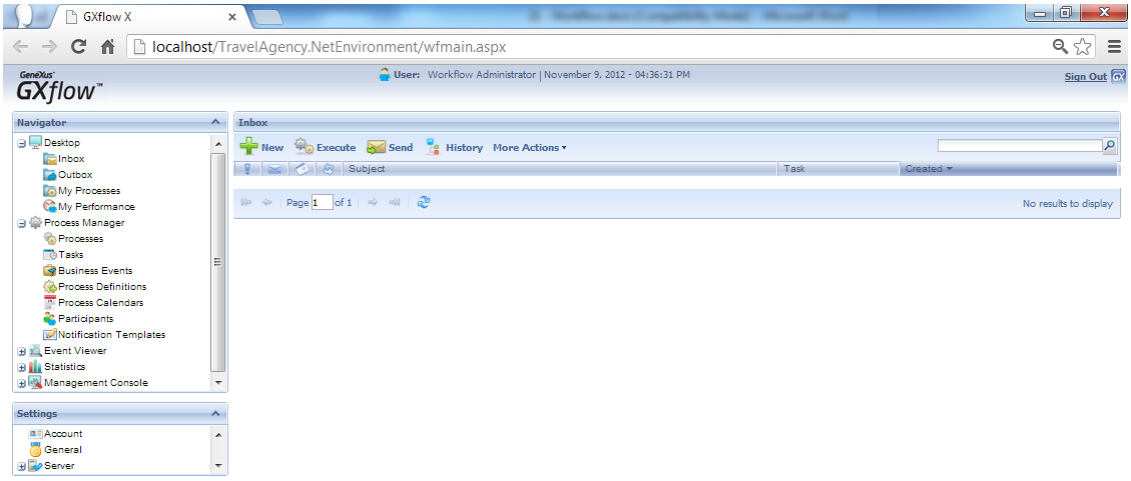
Ahora la bandeja de entrada se nos muestra vacía. Esto significa que no hay más tareas pendientes para ejecutar ya que hemos llegado al fin del proceso.

Notemos que si no hubiéramos marcado el check box, el workflow nos hubiera llevado nuevamente a la transacción Reservation al principio del diagrama.



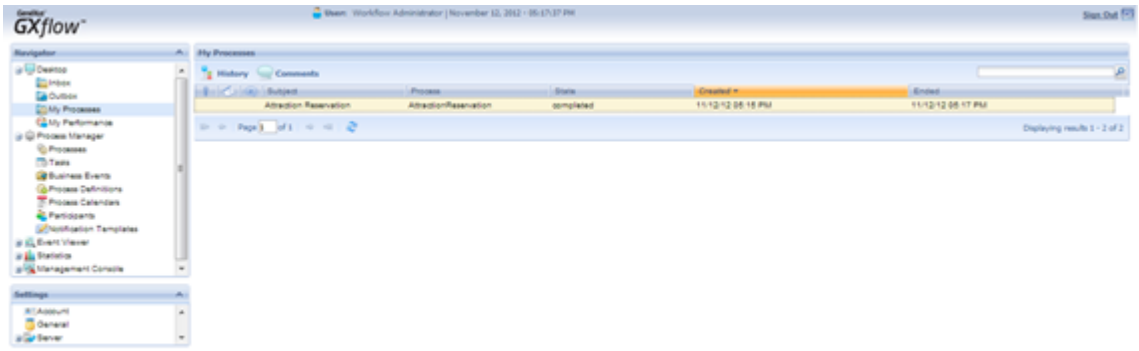
Algo que no mencionamos es que todas estas tareas fueron ejecutadas por el mismo usuario.

Esto es así mientras desarrollamos y probamos el proceso, pero en el uso real, habrá diferentes usuarios que ejecutarán las distintas tareas, de acuerdo a los perfiles que tengan en la organización.

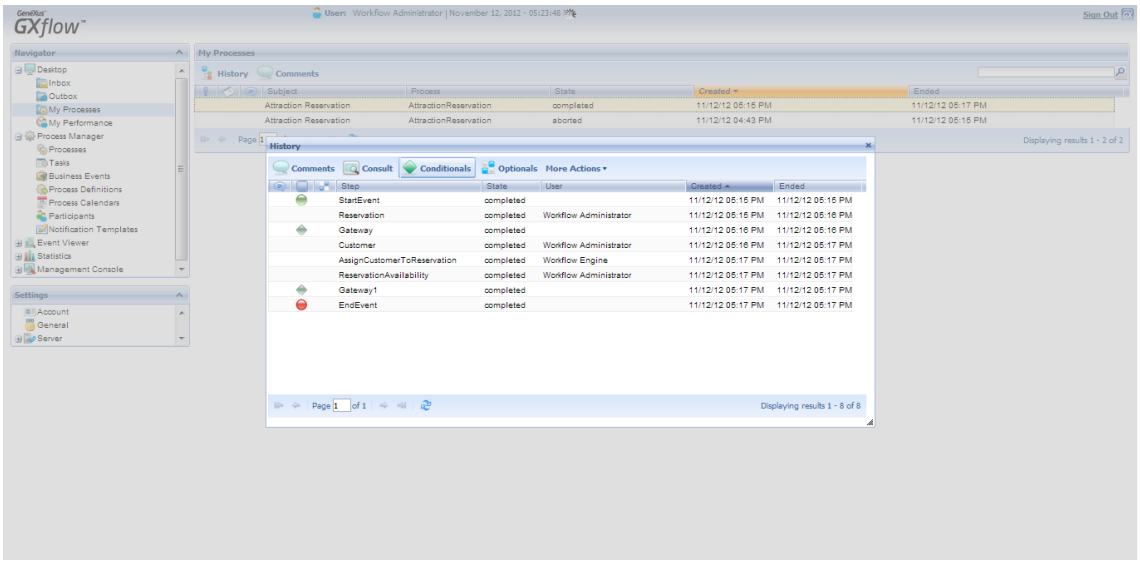


Otra cosa interesante es que podemos ver la historia del proceso, es decir, qué caminos del diagrama se recorrieron cuando se ejecutó el proceso.

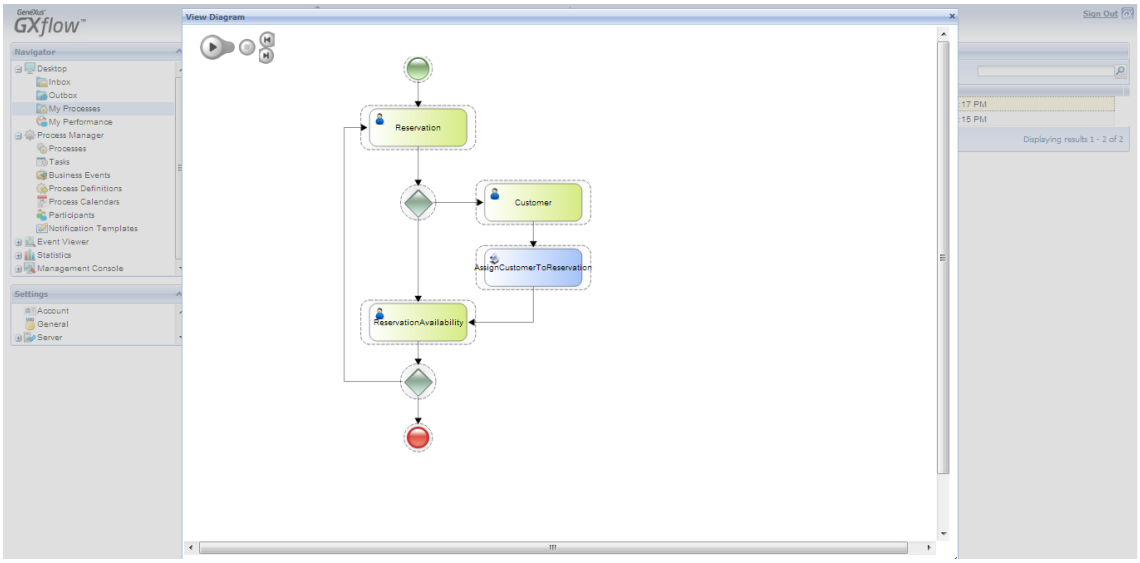
Para ver la historia, seleccionamos My Processes en la ventana Navigator. Vemos que se muestra el proceso que acabamos de ejecutar (Attraction Reservation) que aparece en estado completado.



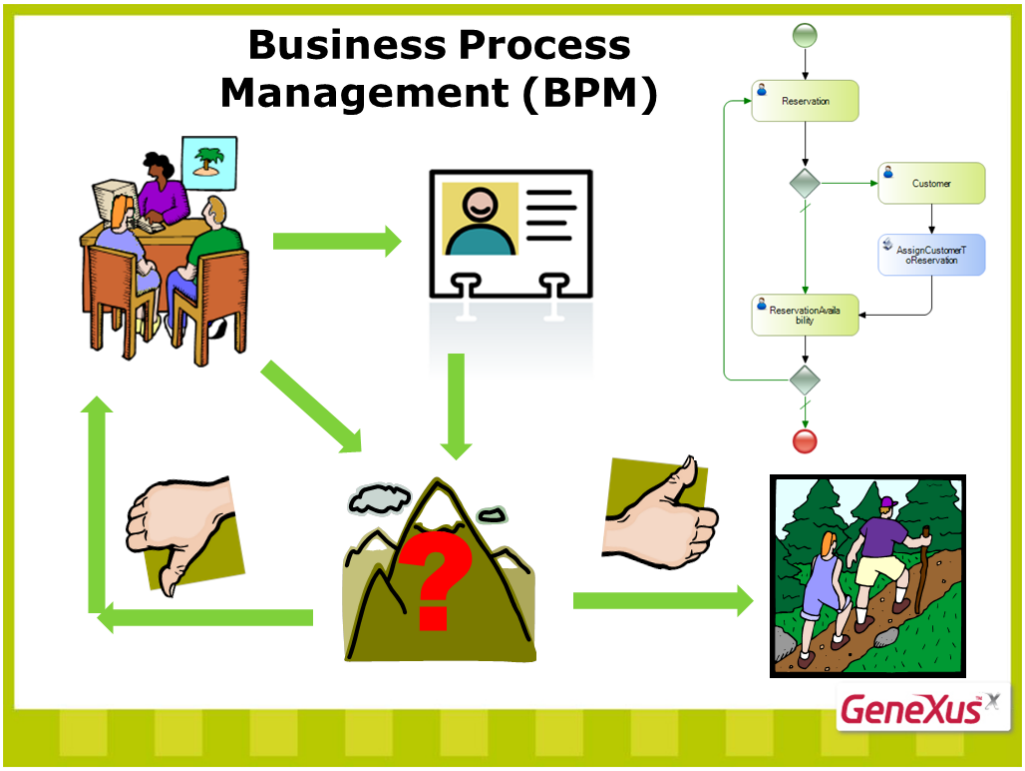
Hacemos doble clic sobre el mismo, y vemos que se abre una ventana que nos muestra la historia del proceso



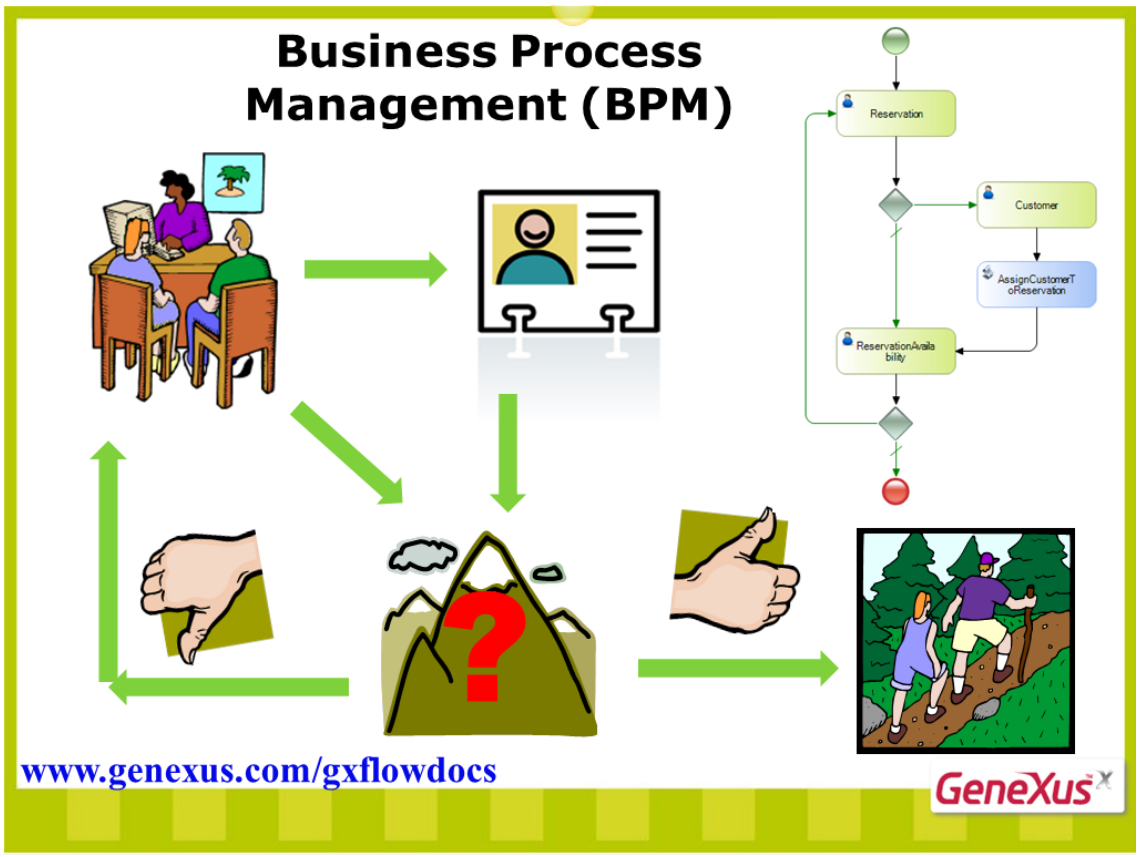
En esta ventana vemos todas las tareas que se fueron ejecutando. Si vamos a More Actions, View Diagram, podemos ver la historia en forma de animación. Presionamos Play…



Hemos visto así, que GeneXus nos permite modelar procesos de negocios en forma intuitiva, resolviendo automáticamente qué objeto debe ejecutarse en cada caso, de acuerdo a la definición que hicimos a través del diagrama.



Para saber más sobre este tema, visite el sitio en la dirección que se muestra en pantalla


Business Process Management (BPM)
Reservation
Customer
AssignCustomerToReservation
ReservationAvailability
www.genexus.com/gxflowdocs
GeneXus