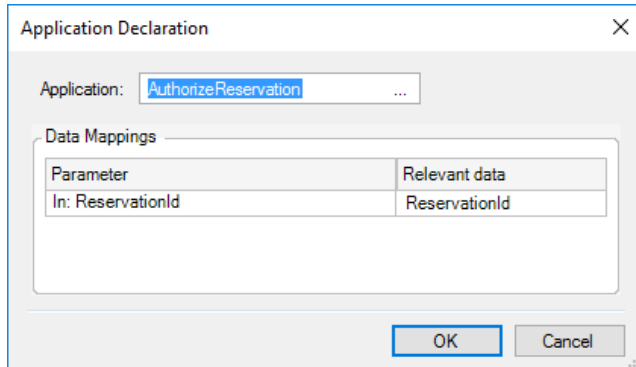


Programando recordatorios y fin del proceso de reserva de pasajes

Volvemos al diagrama para concentrarnos en la tarea de autorización de la reserva, por parte del gerente de atención al cliente.

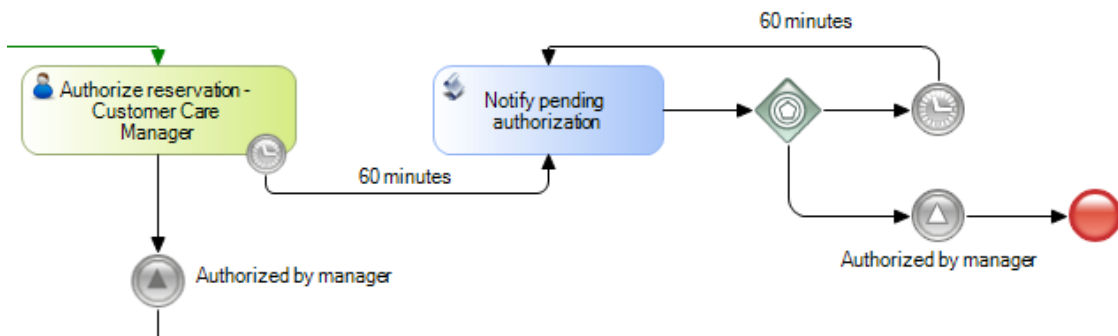
Vamos al diagrama y asociamos la tarea **Authorize reservation – Customer Care Manager** al webpanel **AuthorizeReservation**, mapeando el dato relevante &ReservationId.



El webpanel mostrará los datos de la reserva y tendrá 2 botones para autorizar o rechazar en forma similar al que vimos para evaluar el cliente.

Presionamos OK y vamos a crear un dato relevante **&AuthorizeReservation**, que almacenará la decisión que tomó el gerente de atención al cliente.

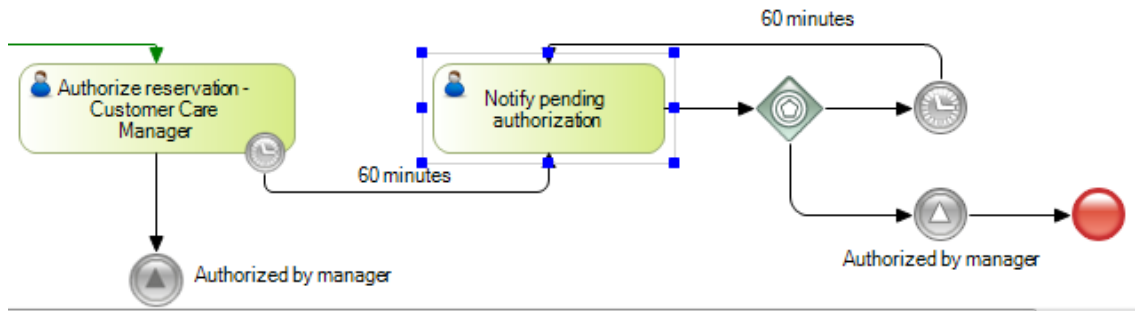
Ahora, recordemos los avisos automáticos que habíamos definido.



La tarea **Notify pending authorization** es una tarea batch, que ejecutará un procedimiento en el que se envía un aviso al gerente de atención al cliente, por ejemplo por mail.

A efectos de simplificar el chequeo de este patrón, vamos a cambiar la tarea batch por una tarea interactiva asociada a un webpanel, que simplemente muestre el aviso por pantalla.

Para eso vamos a las propiedades de la tarea, cambiamos su propiedad **Type** a User y asociamos la tarea al webpanel **"NotifyPendingAuthorization"**.

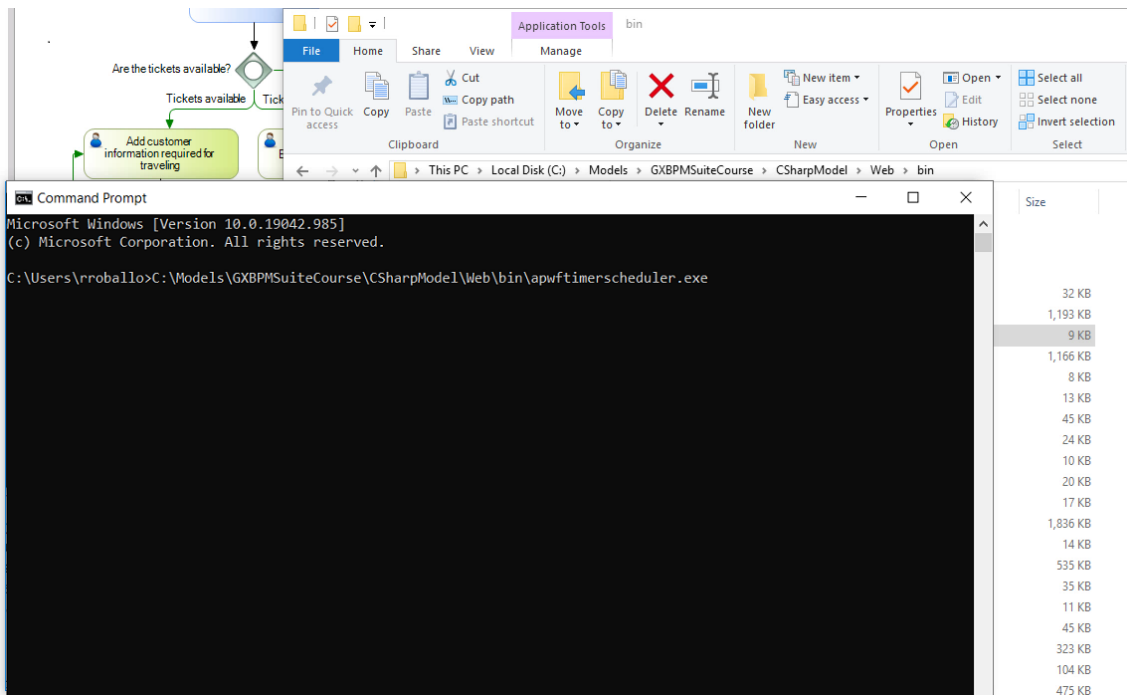


A efectos de poder mostrar el comportamiento en este video, cambiamos los tiempos de espera a 2 minutos, así que editamos la propiedad rule de ambos timers y ponemos el valor 2.

Para que las acciones provocadas por los eventos tipo timer, las podamos ver reflejadas en nuestra inbox, es necesario ejecutar un programa que dispare los eventos del tipo timer, si no lo ejecutamos, estos eventos no se dispararán.

Este utilitario se ejecuta por línea de comando, así que vamos a Tools, Explore Target Environment Directory y hacemos clic en la carpeta bin.

Ahora abrimos una ventana de comando y arrastramos desde la ventana del explorador de Windows al ejecutable: **apwftimerscheduler.exe**. Agregamos el valor 60 y presionamos Enter. En nuestro caso, se dispararán los deadlines cada 60 segundos.



Además del utilitario **Timer Scheduler**, también disponemos del **Timer Control**, que se ejecuta una única vez. El timer scheduler lo usamos típicamente para prototipar, mientras que el timer control lo usamos cuando el workflow está en producción, usando las facilidades del sistema operativo para agendar su ejecución.

Volvamos a GeneXus, ejecutemos el diagrama **FlightTicketReservation** y avancemos hasta la pantalla donde nos habíamos quedado anteriormente.

Damos Send a la tarea **"Add customer information required for traveling"** y vemos que no avanzamos en el diagrama ya que todavía está pendiente de ejecutar la tarea "Evaluate Customer".

La ejecutamos y vemos que se abre la pantalla del webpanel para evaluar al cliente.

Evaluate Customer Status

Customer financial authorization

Reservation	52
Id	1
Name	John Parker
Address	6500 Main Street, House USA
Phone	555-123456
Email	jparker@example.com
Added Date	02/23/23

AUTHORIZE

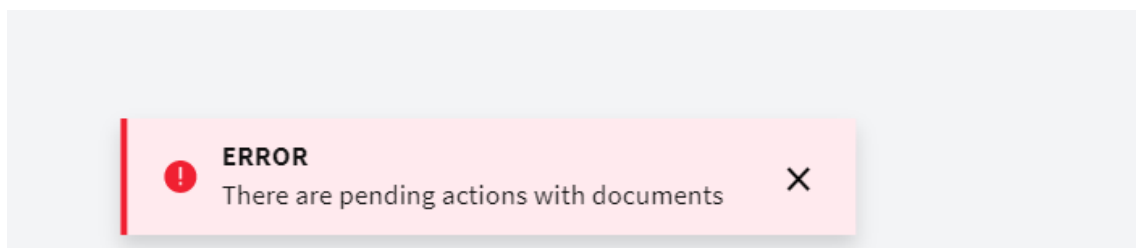
REFUSE

Presionamos el botón **Authorize** y completamos la tarea **"Evaluate Customer"**. Vemos que ahora están pendientes de ejecución las tareas **"Check reservation documents"** y **"Authorize reservation – Customer Care Manager"**.

Vemos que la tarea de revisar documentos, también tiene documentos asociados, porque habíamos definido que la operación de lectura de la visa fuera obligatoria.

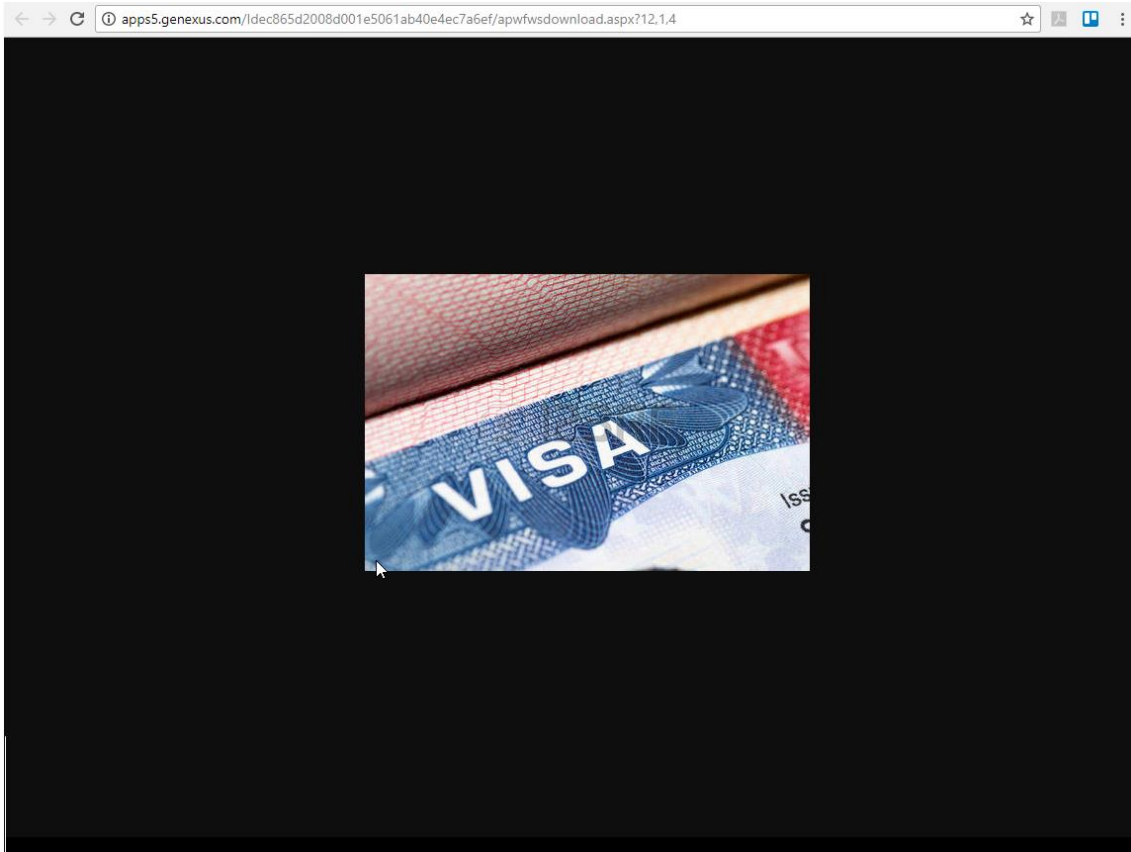
Ejecutamos la tarea y vemos que se abre el trabajar con documentos. En lugar de presionar el botón de "Read", cerremos la ventana y presionemos Send sobre la tarea.

Vemos que se despliega un mensaje de error, indicándonos que hay documentos pendientes de lectura.

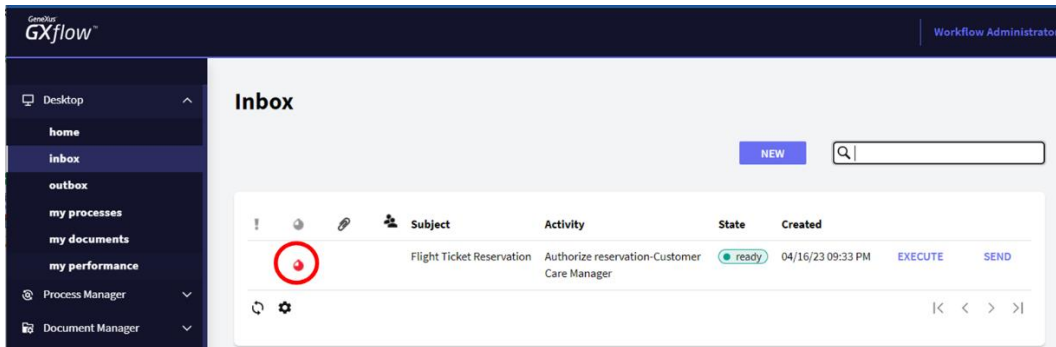


Así que volvemos a ejecutar la tarea, seleccionamos el documento Visa y presionamos Read.

Vemos que abre una ventana del browser para poder ver el contenido del documento, en este caso la imagen de la visa.



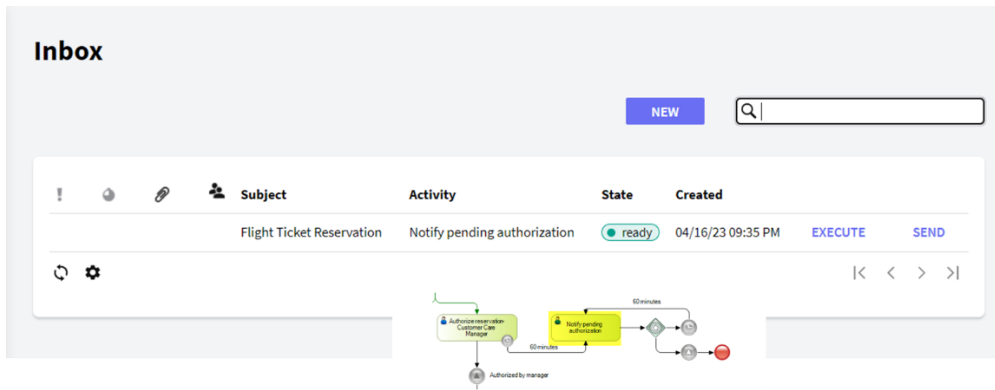
Cerramos la ventana, completamos la tarea y vemos que efectivamente no está más en el inbox.



Ahora en lugar de ejecutar la tarea **"Authorize reservation-Customer Care Manager"** vamos a esperar unos minutos.

Si observamos la tarea, tiene un símbolo de reloj

Presionamos el botón de Refresh del navegador



Vemos que a los 2 minutos apareció en la bandeja de entrada la tarea interactiva **Notify Pending Authorization**, que es el aviso disparado por el timer. Vamos a ejecutarla y vemos el aviso de que hay una reserva pendiente de autorización.

Notify Pending Authorization

You have a reservation pending of authorization, please check your inbox.

Confirm

Presionamos Confirm y se cierra la ventana. Damos Send para completar la tarea.

Mientras el gerente no ejecute la tarea de autorización, seguirán apareciendo notificaciones cada 2 minutos. Ejecutemos nuevamente la tarea del aviso....

Ahora ejecutemos la tarea de autorización y presionemos Authorize.

Authorize Reservation

Reservation Information

Id 53

Date 04/14/23

Qty 2

Customer Name John Parker

Departure Airport Carrasco , Montevideo , Uruguay

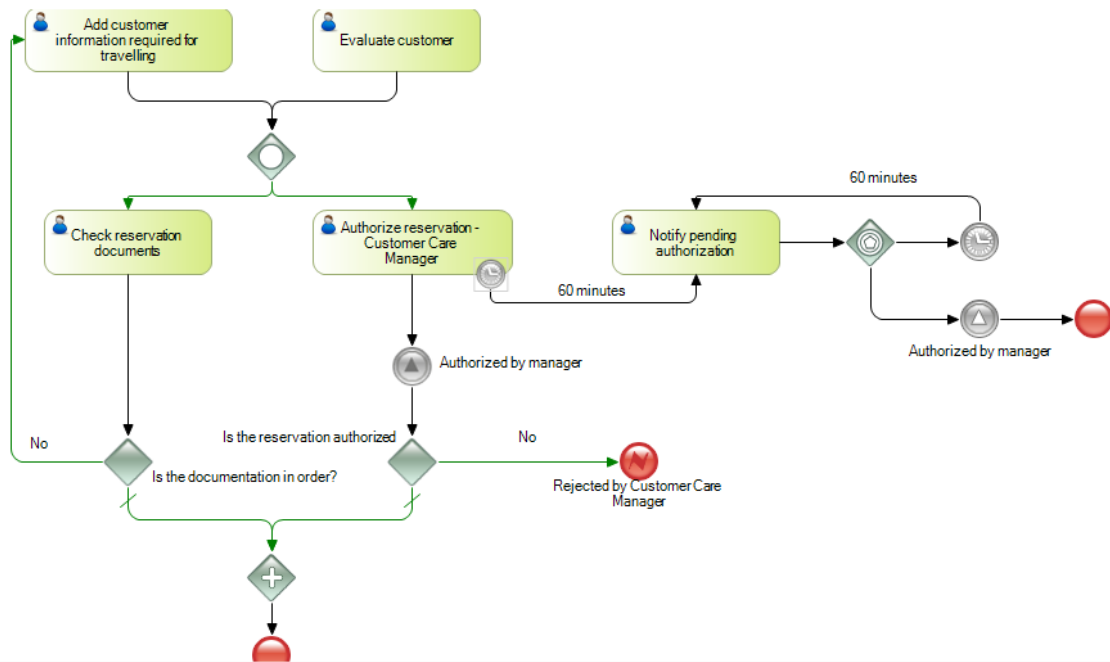
Arrival Airport Guarulhos , Sao Paulo , Brazil

AUTHORIZE

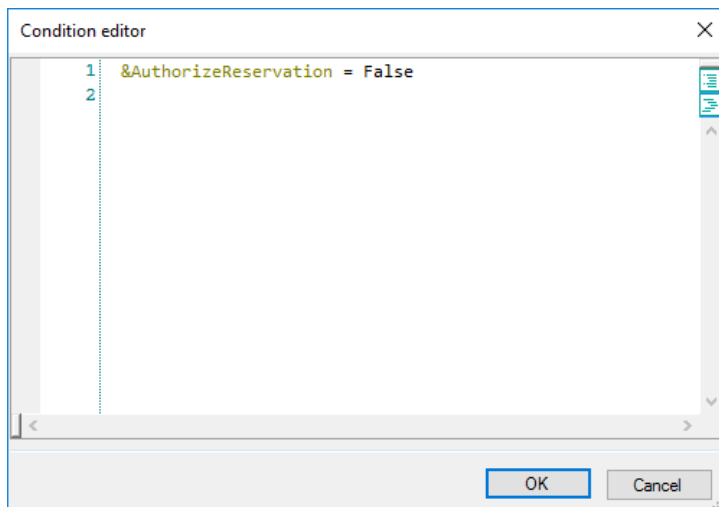
REFUSE

Completamos la tarea y esperamos unos minutos...

Vemos que no aparecen más la notificación periódica, ya que el signal event que está a continuación de la tarea envió un mensaje al otro signal event y canceló este bucle de notificación.



Volviendo al diagrama, vemos que el evento de señal "Authorized by manager" va a un Exclusive Gateway el cual evalúa si la reserva fue autorizada o no. Para establecer esta decisión, hacemos doble clic en el conector que une el Gateway con el Error End Event "Rejected by Customer Care Manager" y en el editor de condiciones escribimos `&AuthorizeReservation = False`.

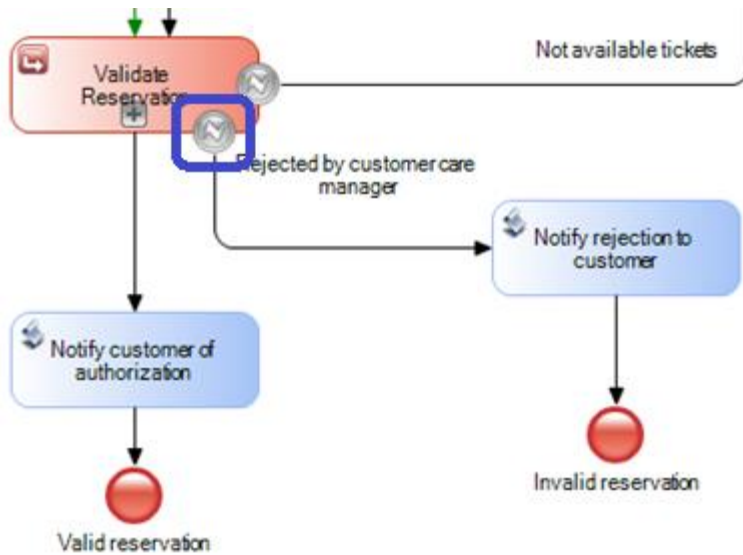


Este evento de error le comunicará al proceso FlightTicketReservation que el proceso de validación de la reserva se cancela porque el gerente de atención al cliente no autorizó la reserva. Recordemos viendo sus propiedades, el error code que tiene asignado.

▼ End Event: Rejected by Customer Care Manager

Name	Rejected by Customer Care Manager
Trigger	Error
Error code	REJECTED_BY_CUSTOMER_CARE_MANAGER

En el proceso de FlightTicketReservation, tenemos presente un Signal Error Event definido como catch, con el mismo nombre.



Y sobre todo con el mismo Error Code, lo que asegura que capturará el evento disparado en el subproceso ValidateReservation lo que hará que se emita un aviso de notificación al cliente.

▼ **Intermediate Event: Rejected by Customer Care Mana**

Name	Rejected by Customer Care Mana
Trigger	Error
Error code	REJECTED_BY_CUSTOMER_CARE_MANAGER

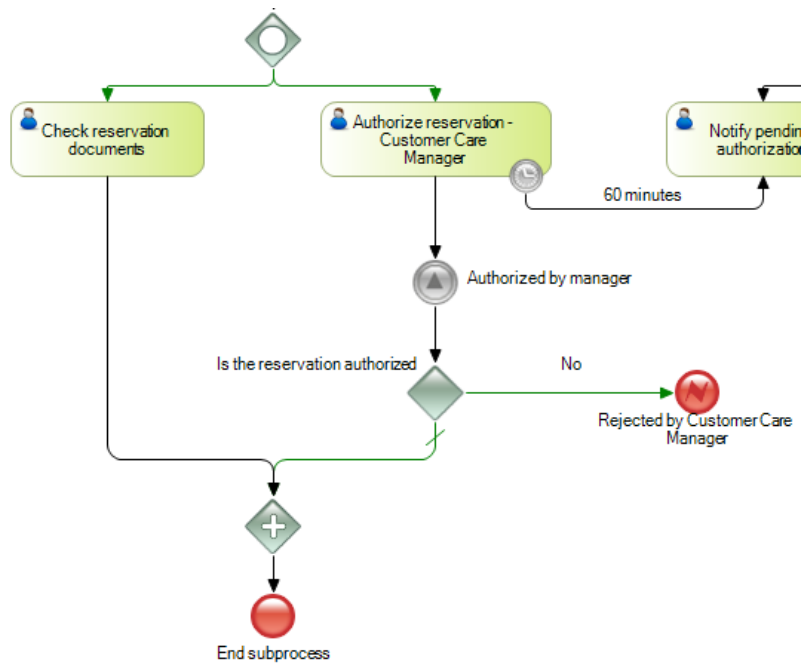
Volviendo al subproceso de validación de la reserva, la tarea de revisión de documentos también termina en un Exclusive Gateway. En este caso se evalúa si la documentación está en orden y si no lo está se deberá volver a ejecutar la tarea **“Add customer information required for traveling”**.

Cuando hicimos el modelo, supusimos que la tarea interactiva **“Check reservation documents”** estaría asociada a un objeto GeneXus que sería el responsable de cargar un dato relevante con un valor, que después se evaluaría en el exclusive Gateway.

Sin embargo, ya vimos que el mismo motor de workflow se encarga de verificar que no podemos dejar de leer el documento, ya que en la tarea habíamos definido una operación de Read y como Required.

Por lo tanto, no es necesario tener un Gateway que controle si la tarea se realizó o no, ya que el control lo realiza el mismo motor cuando ejecutamos la tarea y no nos deja continuar si no hacemos Read sobre el documento.

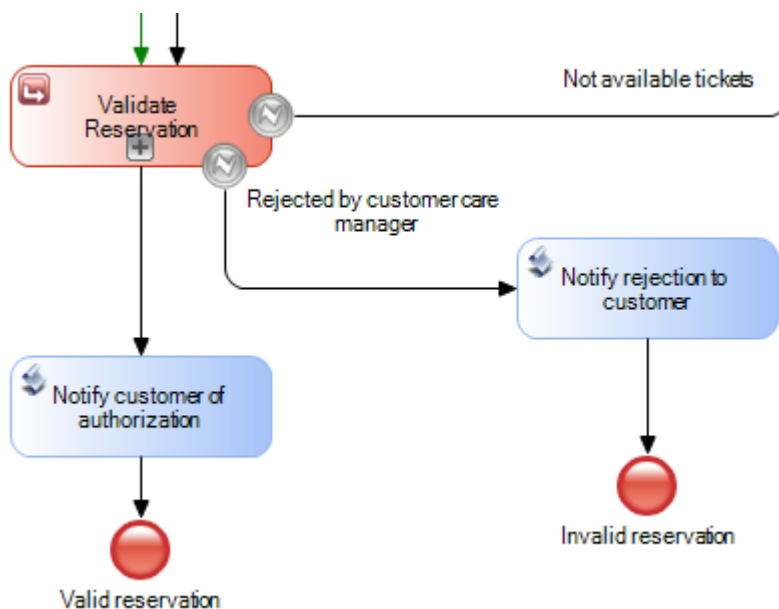
Así que vamos a modificar el diagrama, eliminando el exclusive Gateway **“Is the documentation in order?”**, por lo que ahora la tarea **“Check reservation documents”** se conecta directamente con el parallel Gateway.



El parallel Gateway sincroniza todos los caminos que están conectados a él en el diagrama y espera hasta que todos los caminos lleguen al mismo, para dejar continuar el proceso.

Si ejecutamos la tarea de control de documentos, revisando los documentos y la reserva fue autorizada por el gerente de atención al cliente, el parallel Gateway dejará continuar el flujo, finalizándose el subprocesso.

En este caso, la finalización de la validación fue exitosa, por lo que el control volverá al proceso principal FlightTicketReservation.



Al volverse a este proceso con una validación exitosa, se notificará al cliente de que la reserva fue autorizada y finalizará el proceso de reserva de pasajes.

En videos siguientes contemplaremos cambios al diagrama, debidos a nuevos requerimientos de la agencia de viajes.