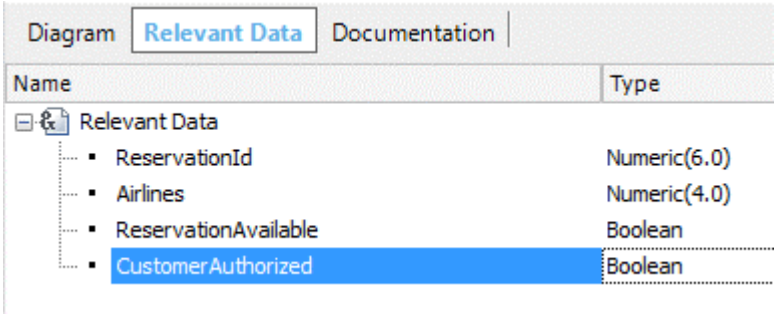
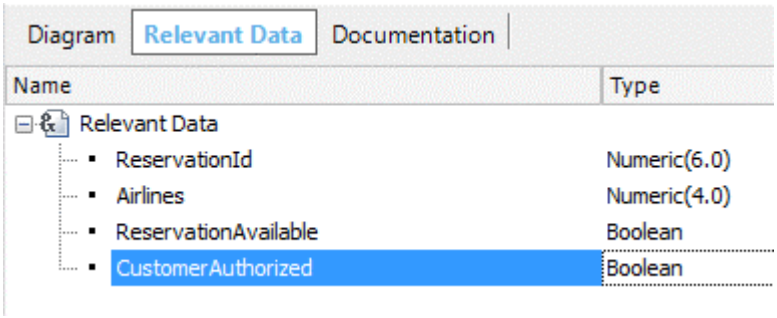
Modificación de dato relevante, evento timer y calendarios

Continuando con el diagrama, la tarea **Evaluate Customer** deberá evaluar la situación financiera del cliente y almacenar la decisión en un dato relevante, que será consultado más adelante en el diagrama, cuando se autorice la reserva.

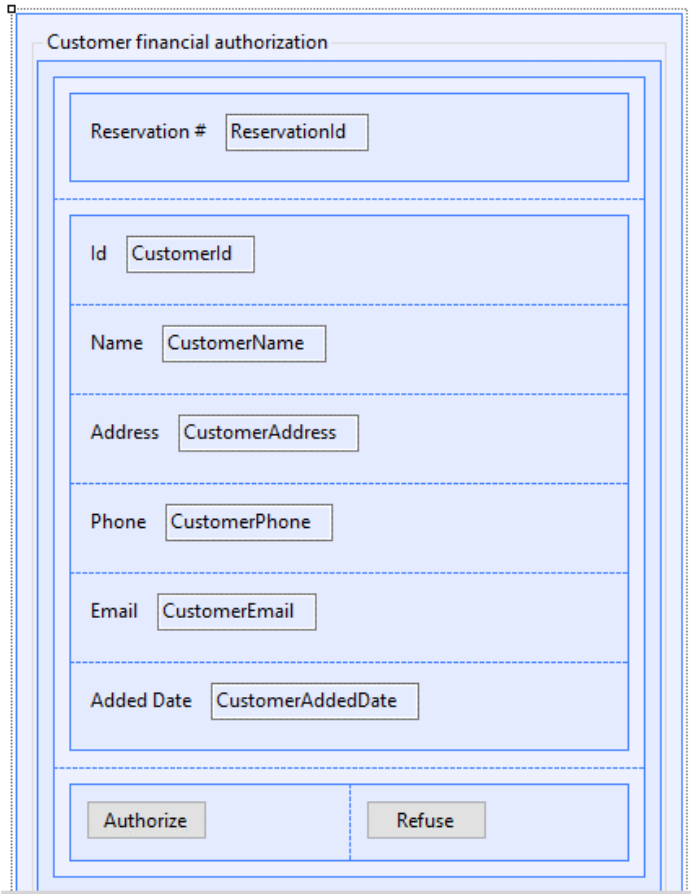
Así que vamos a la solapa Relevant Data y creamos el dato relevante CustomerAuthorized, del tipo boolean.



Ahora asociaremos la tarea **Evaluate Customer** al webpanel **EvaluateCustomerStatus** y mapeamos el dato relevante ReservationId



Si abrimos el objeto webpanel, vemos que tiene el identificador de la reserva, los datos del cliente y dos botones: Autorizar y Rechazar.



Si vamos a la solapa Eventos, vemos que el mismo carga el valor del dato relevante que acabamos de ingresar con el valor True o False, dependiendo de si presionamos los botones Authorize y Refuse respectivamente.

Vemos que en este webpanel estamos utilizando métodos de la API de workflow para recuperar y modificar el dato relevante "CustomerAuthorized".

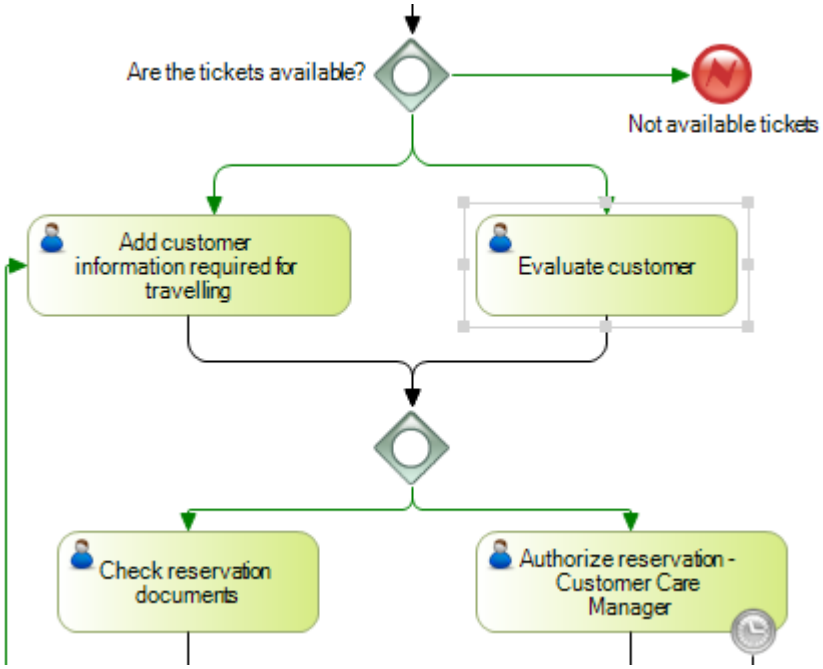
```
Event 'Authorize'
    &CustomerAuthorizedWorkflowApplicationData = &WorkflowContext.ProcessInstance.GetApplicationDataByName("CustomerAuthorized")
    &CustomerAuthorizedWorkflowApplicationData.BooleanValue = True
    return
Endevent

Event 'Refuse'
    &CustomerAuthorizedWorkflowApplicationData = &WorkflowContext.ProcessInstance.GetApplicationDataByName("CustomerAuthorized")
    &CustomerAuthorizedWorkflowApplicationData.BooleanValue = False
    return
Endevent
```

Recordemos que en los webpanels, a diferencia de los procedimientos, el mapeo de valores entre los datos relevantes y las variables presentes en la regla Parm, es válido solamente para variables de entrada, mientras que en los procedimientos el mapeo es válido tanto para variables de entrada como de salida.

Por esa razón, en un procedimiento alcanza con definir una variable que se llame exactamente igual a un dato relevante y ponerla como variable de salida en la regla Parm y su valor se trasladará al dato relevante del diagrama, sin necesidad de usar métodos de la API para acceder al mismo.

Volvemos al diagrama y vemos que hay un inclusive Gateway que está después de las tareas **Evaluate Customer** y **Add customer information required for traveling**, que cierra los caminos abiertos por otro inclusive Gateway.



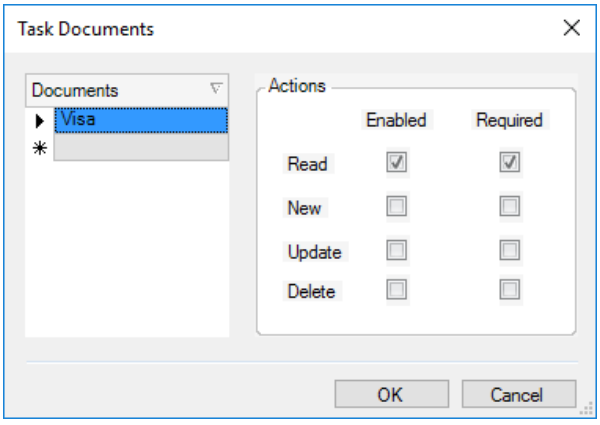
En este caso, el segundo Inclusive Gateway sincronizará los caminos que llegan al mismo, y pero a diferencia del Paralell Gateway que espera a **todos los caminos del diagrama**, el inclusive Gateway sincroniza solamente los caminos **que en la ejecución efectivamente llegan al Gateway** y no todos los que están presenten en el diagrama.

Como en este caso ejecutamos ambas tareas, el flujo continuará hacia las tareas "**Check reservation documents**" y "**Authorize reservation – Customer Care Manager**".

Sigamos con la tarea de la izquierda.

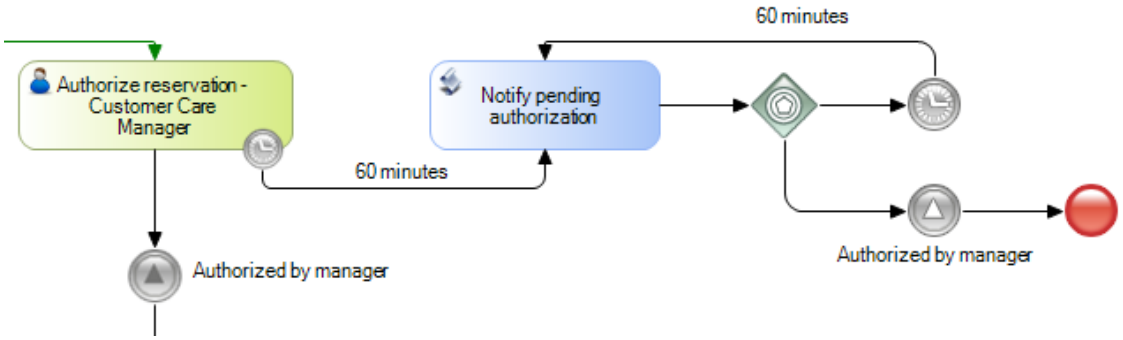
Hasta ahora hemos asociado documentos a la tarea "**Add customer information required for traveling**". Sin embargo, otras tareas pueden acceder a esos documentos, por ejemplo la tarea "**Check reservation documents**".

Para hacer esto vamos a la propiedad Work With Documents y ponemos su valor en True. Luego seleccionamos el documento, por ejemplo, "Visa".



En este caso definimos que la acción es la lectura y dicha acción es obligatoria.

Presionamos OK y volvemos al diagrama para concentrarnos en la tarea de autorización de la reserva, por parte del gerente de atención al cliente.



Si recordamos los requerimientos de la agencia, en esta tarea el gerente de atención al cliente deberá recibir un aviso del sistema cada hora, para recordarle que tiene una autorización pendiente de realizar. Una vez que el gerente autoriza la tarea, el sistema deberá cancelar dichos avisos y se deberá continuar con el proceso.

Para modelar este tipo de temporización y aviso, usamos en primer lugar un evento intermedio del tipo **timer**, asociado a la tarea. Si vamos a las propiedades de este evento, vemos que se configuró el mismo como un deadline de 60 minutos.

Intermediate Event: IntermediateEvent

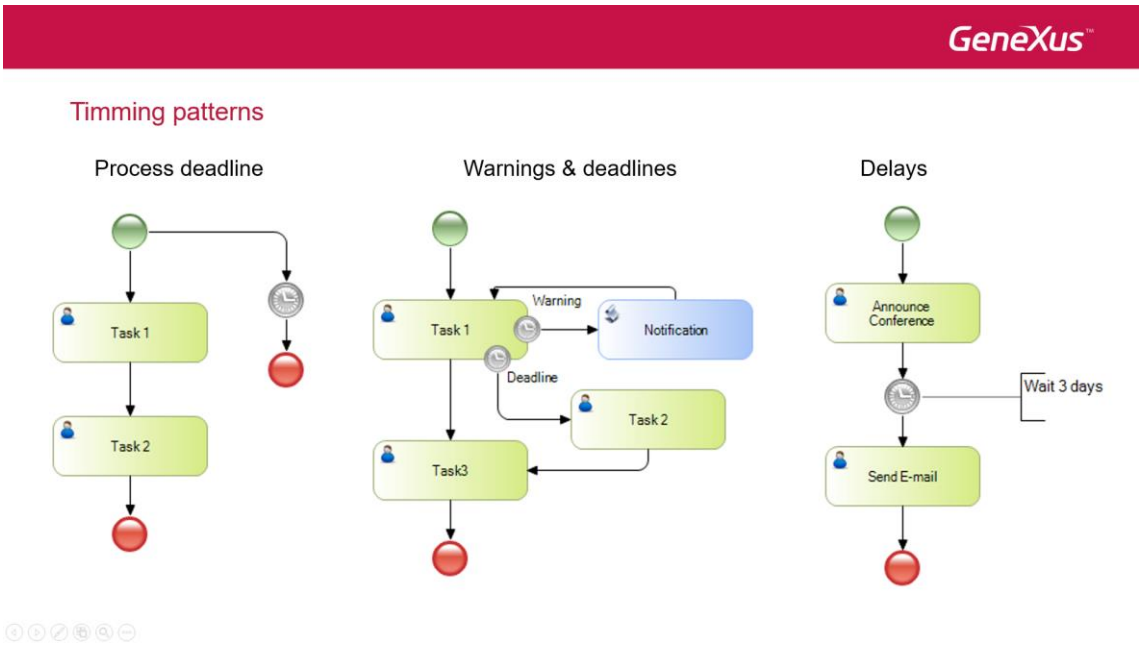
Name	**IntermediateEvent**
Trigger	**Timer**
Interrupts activity	**False**
Timer usage	Deadline
Submit to calendar	False
Time unit	Minutes
Lapse expression type	Rule
Lapse expression rule	60

La propiedad **Interrupts activity** en False, asegura que el vencimiento de este timer no interrumpirá la tarea al vencerse el plazo de los 60 minutos.

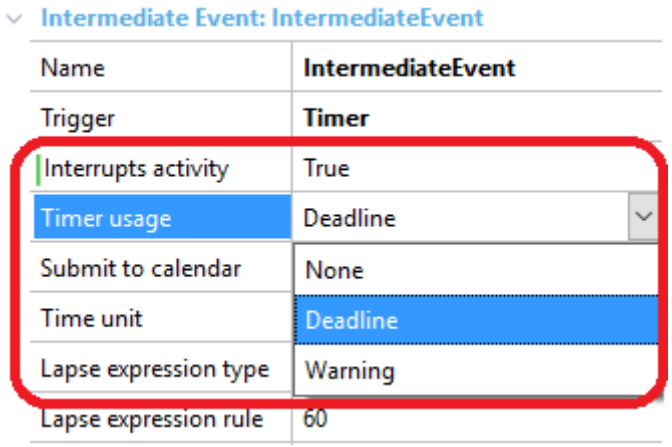
Una vez transcurrido ese tiempo, se ejecutará la tarea batch "**Notify pending authorization**" que dará aviso al gerente. Una vez ejecutado el aviso, un Gateway del tipo evento, asociado a otro timer, asegurarán que el mensaje se repita cada 60 minutos.

Una vez que el gerente autoriza la tarea, el flujo del proceso continuará desde la tarea hacia abajo, por lo que alcanzará un evento intermedio del tipo señal, configurado como "**throw**" (vean el color oscuro de la flecha) que enviará una señal la que será capturada por otro evento intermedio del tipo señal, configurado como "**catch**" (con su flecha en color claro), el que terminará este patrón de avisos con un evento de fin.

Los timers nos permiten modelar diferentes patrones relacionados con el tiempo, como establecer un plazo de vencimiento para el proceso o para las tareas (deadlines), emitir un alerta de un vencimiento próximo (warnings), o establecer un retraso entre actividades (delay).

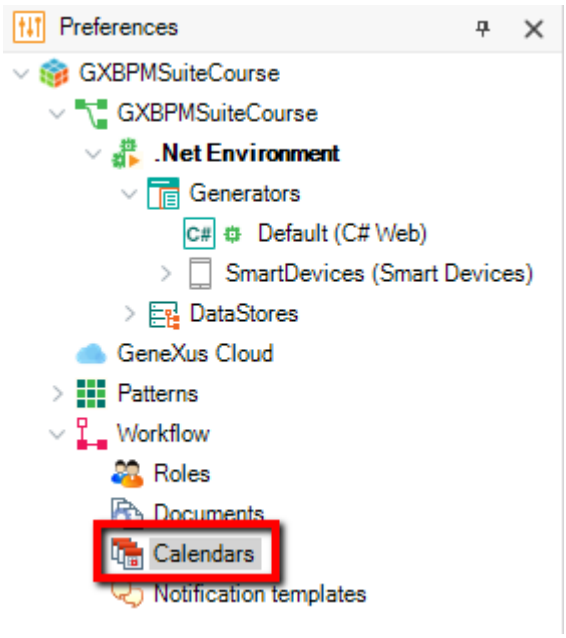


La configuración se realiza mediante las propiedades del timer, donde a través de la propiedad **Timer usage** puede establecerse si se desea implementar un deadline o un warning y mediante la propiedad **Interrupts activity** si se desea que la tarea sea interrumpida al finalizar el tiempo establecido, o no.

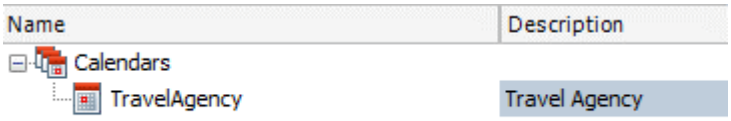


Observemos que tenemos una propiedad **Submit to calendar**. Esta propiedad nos permite definir que el deadline o warning sea asociado a un calendario particular, es decir, teniendo en cuenta los días y horas hábiles de trabajo, excluyendo los feriados.

Para definir un calendario, vamos a la ventana de Preferences y bajo Workflow hacemos doble clic en Calendars.

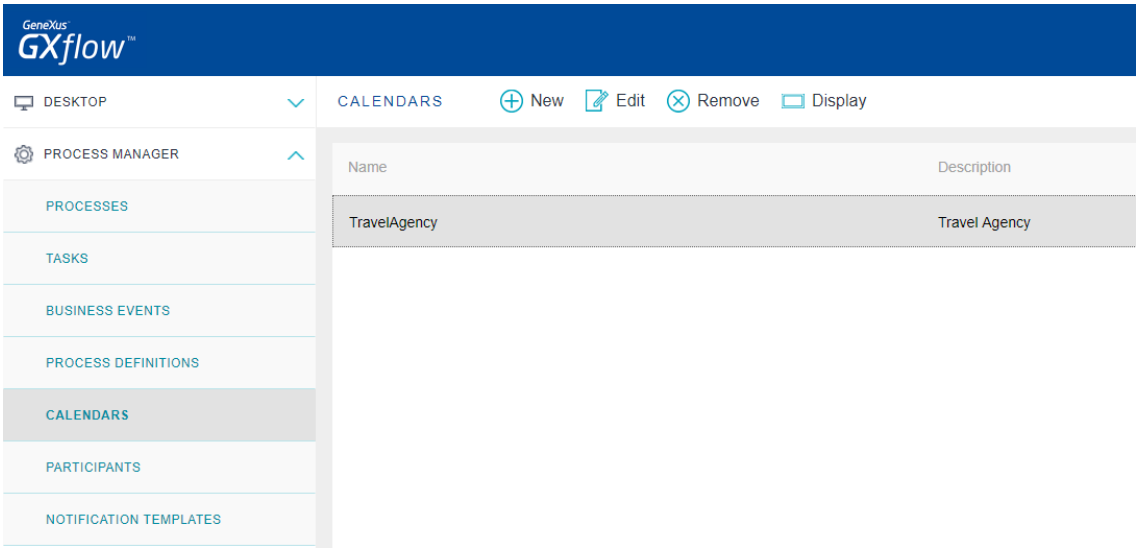


Ingresamos el nombre del calendario, por ejemplo Travel Agency. Ahora seleccionamos el diagrama **Validate Reservation**, en la propiedad Calendar elegimos **Travel Agency** y salvamos.

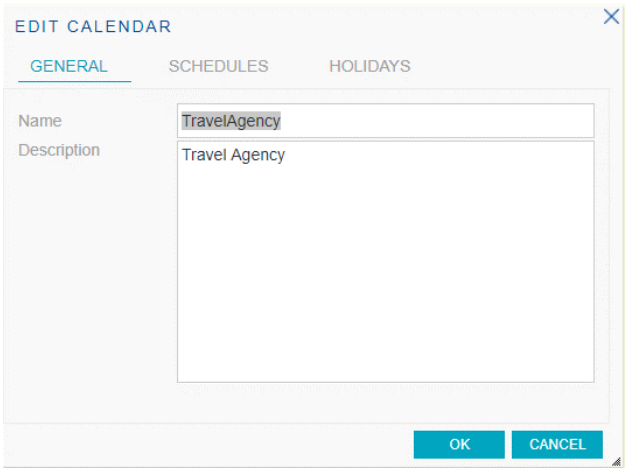


La configuración de los días y horas hábiles, así como los feriados no laborables, se hacen desde el cliente de workflow. Vamos a ejecutar el proceso **FlightTicketReservation** para hacerlo.

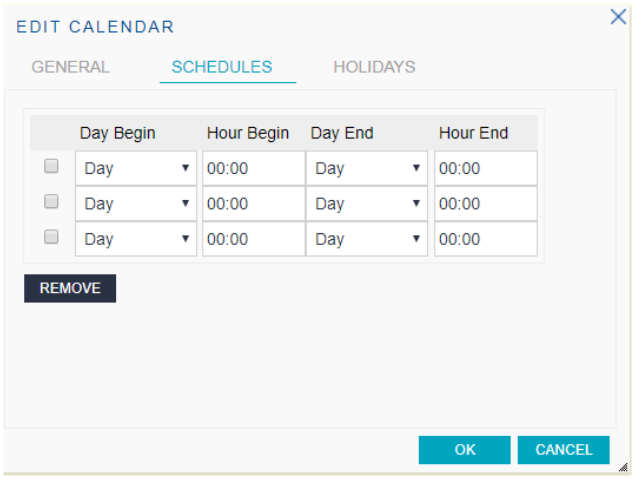
En la ventana del navegador, bajo Process Manager, seleccionamos Calendars.



Vemos que aparece el calendario Travel Agency que habíamos definido. Seleccionamos el mismo y presionamos Edit.



Vamos a Schedules y elegimos los días y horas hábiles de la agencia de viajes.



También podemos ir a la ventana de "Holidays" y definir los días feriados para la empresa, por ejemplo el 1° de mayo.