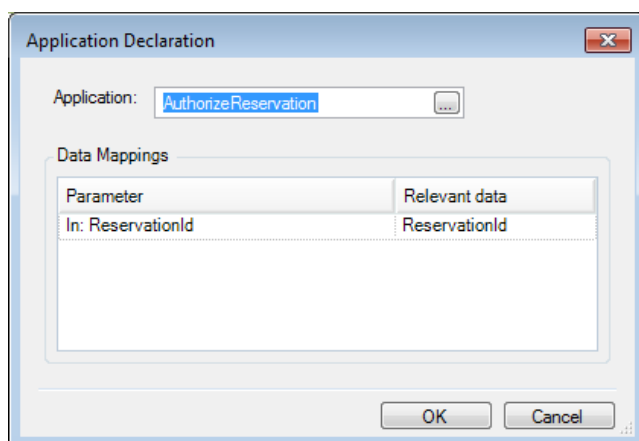


Programando lembretes e fim do processo de reserva de passagens

Voltemos ao diagrama para nos concentrarmos na tarefa de autorização da reserva, que deve ser realizada pelo gerente de atendimento ao cliente.

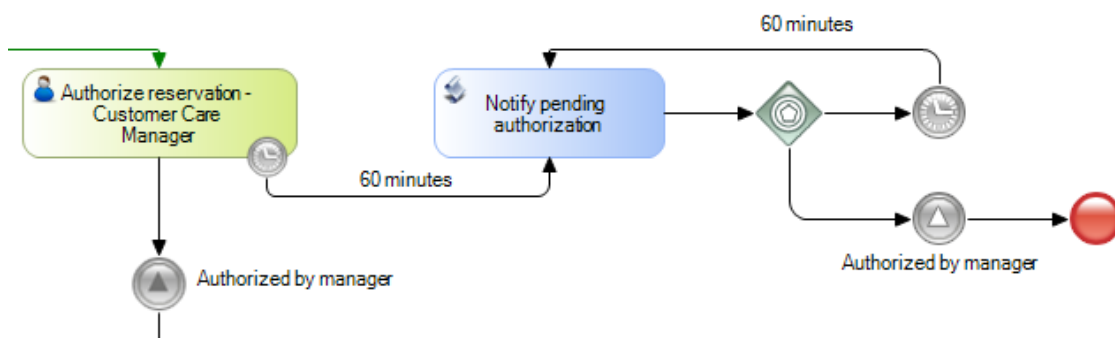
Vamos ao digrama e associamos a tarefa **Authorize reservation – Customer Care Manager** à webpanel **AuthorizeReservation**, mapeando o dado relevante &ReservationId.



A webpanel mostrará os dados da reserva e terá 2 botões para autorizar ou rejeitar de forma similar à que vimos para avaliar o cliente.

Pressionamos OK e vamos criar um dado relevante **&AuthorizeReservation**, que armazenará a decisão que o gerente de atendimento ao cliente tomou.

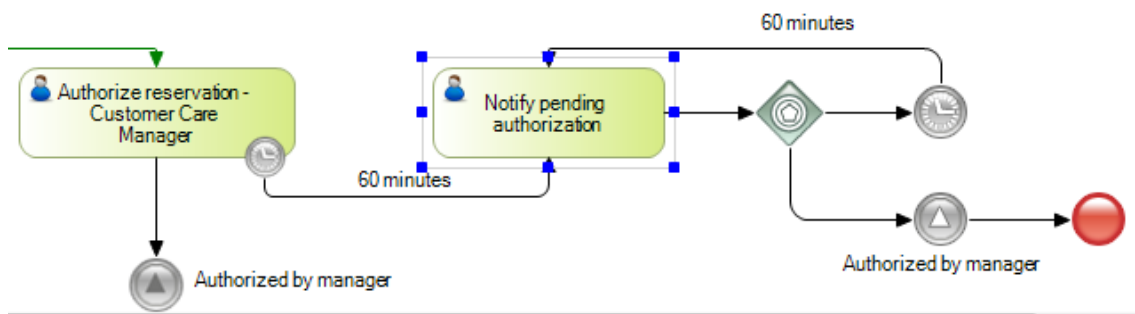
Lembremos agora dos avisos automáticos que tínhamos definido.



A tarefa **Notify pendind authorization** é uma tarefa batch que executará um procedimento no qual um aviso é enviado ao gerente de atendimento ao cliente - que pode ser feito por e-mail, por exemplo.

Para simplificar a checagem desse padrão, vamos trocar a tarefa batch por uma tarefa interativa associada a uma webpanel que simplesmente exiba o aviso na tela.

Para isso, vamos às propriedades da tarefa, alteramos sua propriedade **Type** para User e associamos a tarefa à webpanel **"NotifyPendingAuthorization"**.

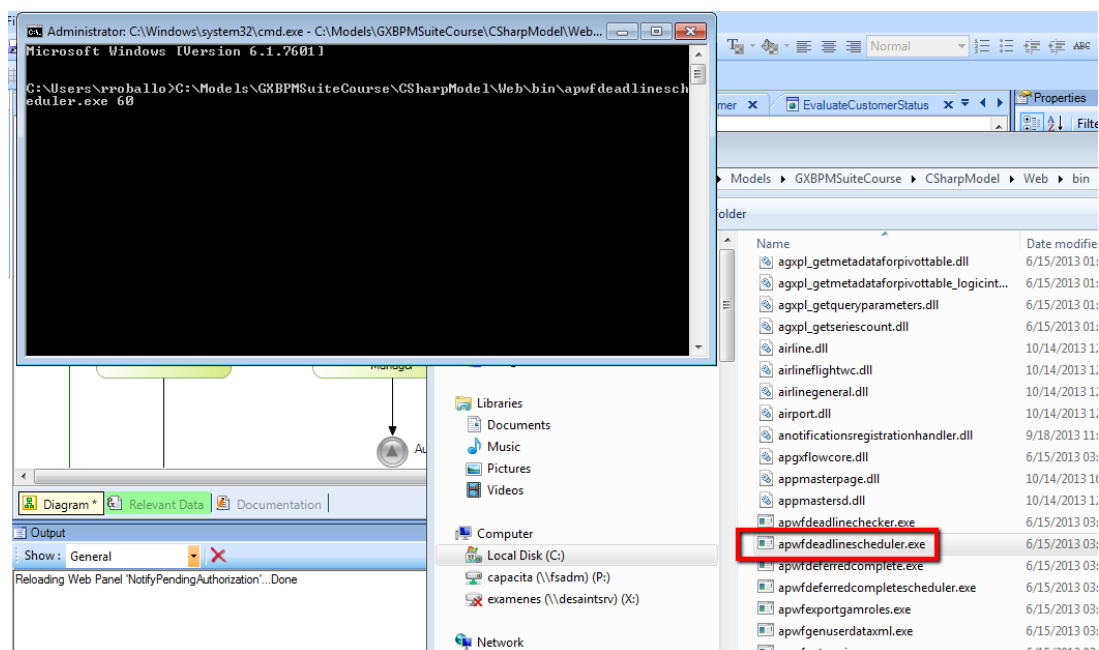


Para poder mostrar o comportamento neste vídeo, mudamos os tempos de espera para 2 minutos. Dessa forma, editamos a propriedade rule de ambos os timers e colocamos o valor 2.

Para que possamos ver as ações provocadas pelos eventos do tipo timer refletidas em nossa caixa de entrada é preciso executar um programa que cheque os deadlines.

Este utilitário é executado por linha de comando. Vamos a Tools, Explore Target Environment Directory e clicamos na pasta bin.

Agora, abrimos uma janela de comando e arrastamos da janela do explorador de Windows ao executável: **apwfdelinescheduler.exe**. Adicionamos o valor 60 e pressionamos Enter. Em nosso caso, os deadlines serão checados a cada 60 segundos.



Além do utilitário **deadline scheduler**, também contamos com o **deadline Checker**, que ao ser executado faz a checagem uma única vez. Utilizamos o deadline schedule principalmente para prototipar, enquanto o deadline Checker é utilizado quando o workflow está em produção, usando as facilidades do sistema operacional para agendar sua execução.

Voltemos a GeneXus, executamos o diagrama **FlightTicketReservation** e avançamos até a tela onde tínhamos parado.

Clicamos em Send para tarefa **"Add customer information required for traveling"** e vemos que não avançamos no diagrama, já que a execução da tarefa **"Evaluate Customer"** ainda está pendente.

Executamos essa tarefa e vemos abre a tela da webpanel para avaliar o cliente.

Evaluate Customer Status

Customer financial authorization

Id 1
 Name John Parker
 Is ACompany ☐
 Address
 Phone 555-123456
 Email
 Added Date 09/13/13

Authorize **Refuse**

Pressionamos o botão **Authorize** e completamos a tarefa **"Evaluate Customer"**. Vemos que agora as tarefas pendentes são **"Check reservation documents"** e **"Authorize reservation – Customer Care Manager"**.

Vemos que também há documentos associados à tarefa de revisar documentos porque tínhamos definido que a operação de leitura fosse obrigatória.

Executamos a tarefa e vemos que abre o trabalhar com documentos. Em vez de pressionar o botão "Read", fechamos a janela e pressionamos Send sobre a tarefa.

Vemos que abre uma mensagem de erro, indicando que há documentos cuja leitura está pendente.

User: Workflow Administrator | October 14, 2013 - 04:33:57 PM

Error
 There are pending actions with documents

INBOX

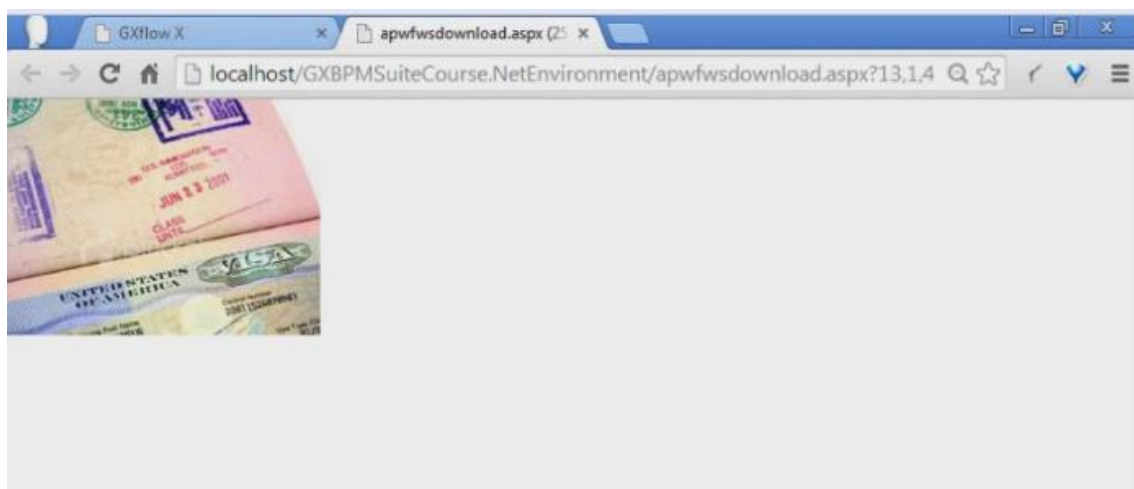
| Subject | Task | Created |
|---------------------------|---|-------------------|
| Flight Ticket Reservation | Check reservation documents | 10/14/13 05:15 PM |
| Flight Ticket Reservation | Authorize reservation - Customer Care Manager | 10/14/13 05:15 PM |

Page 1 of 1

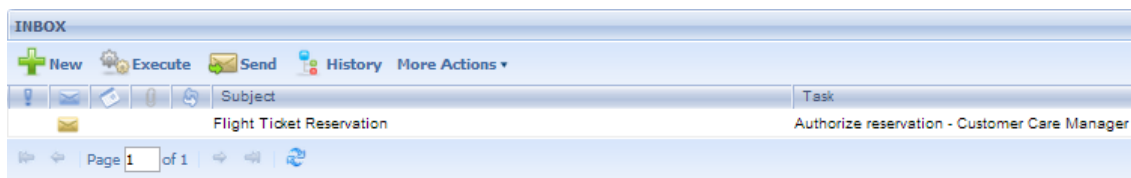
Displaying results 1 - 2 of 2

Devemos voltar a executar a tarefa. Então selecionamos o documento "Visa" e pressionamos Read.

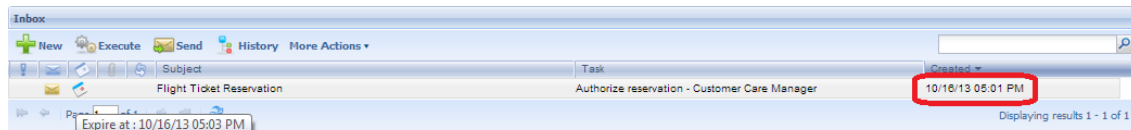
Então abre uma janela do browser para poder ver o conteúdo do documento, neste caso a imagem do visto.



Fechamos a janela, completamos a tarefa e vemos que já não está mais na caixa de entrada.

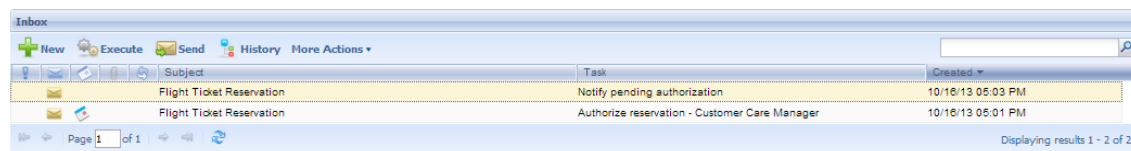


Agora, em vez de executar a tarefa “**Authorize reservation-Customer Care Manager**”, vamos esperar alguns minutos.

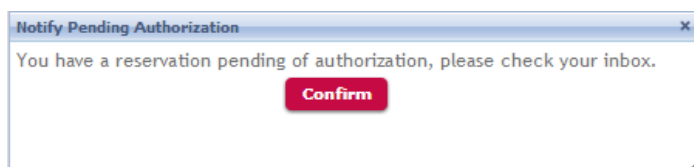


Se observarmos a tarefa, vemos que há um símbolo de calendário e, se colocarmos o mouse sobre ele, somos informados de que a tarefa expira dois minutos depois de ter sido criada.

Pressionamos o botão Refresh do navegador.

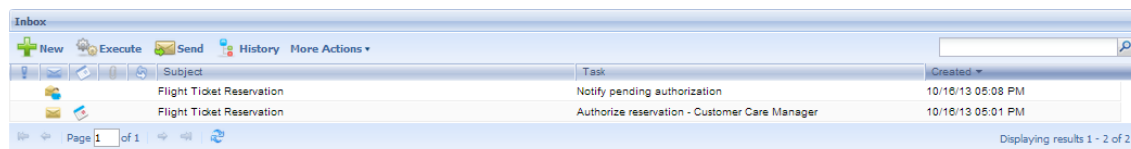


Vemos que dois minutos depois apareceu na caixa de entrada a tarefa interativa **Notify Pending Authorization**, que é o aviso disparado pelo timer. Vamos executá-la e vemos o aviso de que há uma reserva cuja autorização ainda está pendente.



Pressionamos Confirm e a janela se fecha. Pressionamos Send para completar a tarefa.

Enquanto o gerente não executar a tarefa de autorização, continuarão aparecendo notificações a cada 2 minutos.



Executemos novamente a tarefa do aviso...

Agora, executamos a tarefa de autorização e pressionamos Authorize.

Authorize Reservation

Reservation Information

Id

22

Date

10/15/13

Qty

1

Customer Name

John Parker

Departure Airport

Carrasco, Montevideo, Uruguay

Arrival Airport

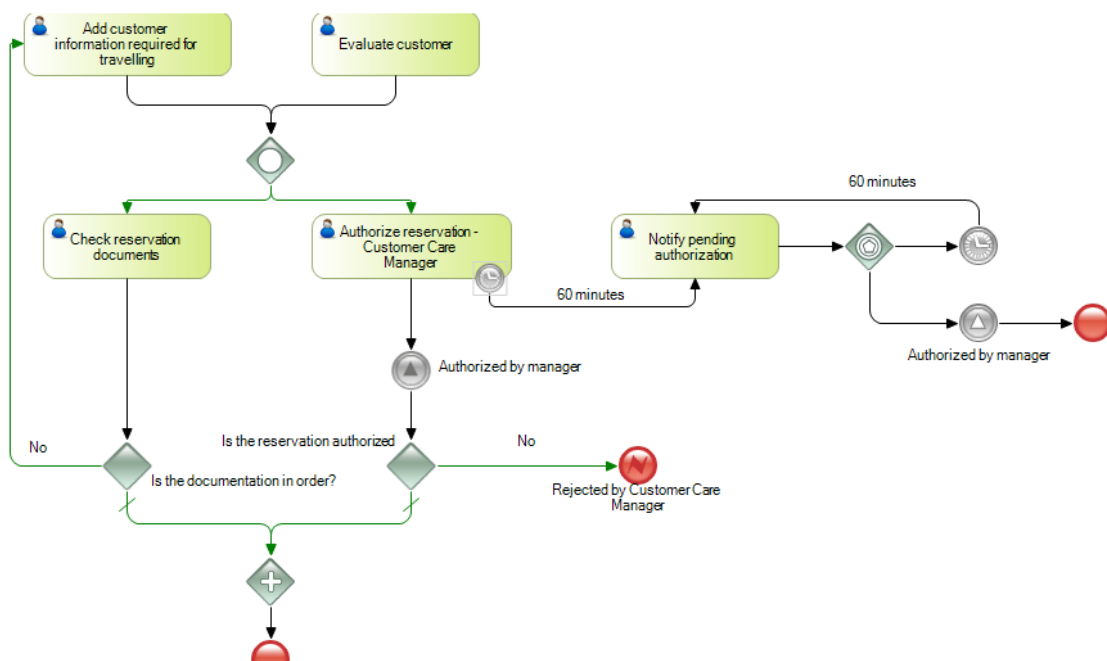
Guarulhos, Sao Paulo, Brazil

Authorize

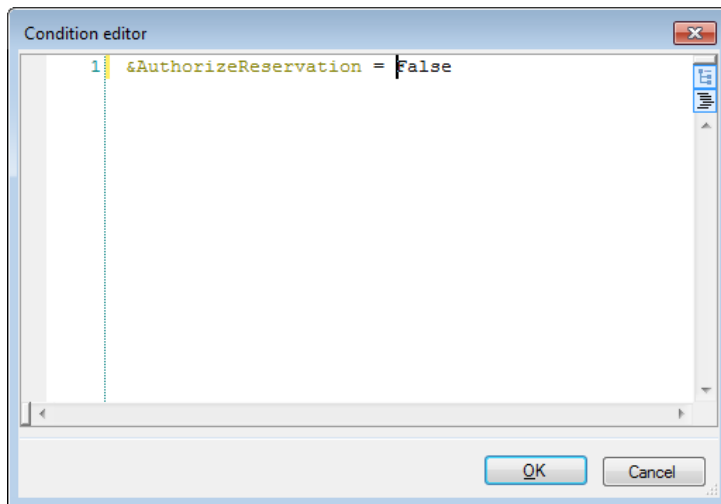
Refuse

Completamos a tarefa e esperamos alguns minutos...

Vemos que a notificação periódica não aparece mais, já que o signal event que está depois da tarefa enviou uma mensagem ao outro signal event e cancelou este loop de notificação.



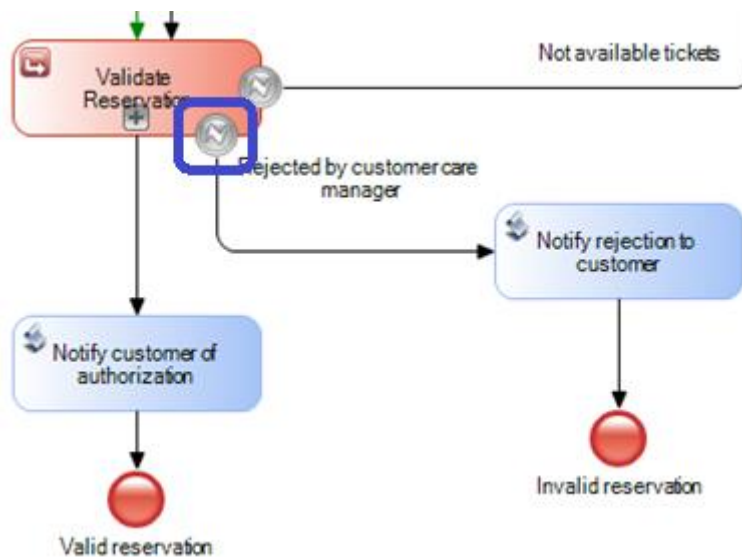
Voltando ao diagrama, vemos que o evento de sinal "Authorized by manager" vai a uma Exclusive Gateway que avalia se a reserva foi autorizada ou não. Para estabelecer essa decisão, damos um duplo clique no conector que une o Gateway com o Error End Event "Rejected by Customer Care Manager" e no editor de condições escrevemos `&AuthorizeReservation = False`.



Este evento de erro comunicará ao processo FlightTicketReservation que o processo de validação da reserva foi cancelado porque o gerente de atendimento ao cliente não autorizou a reserva. Recordemos isso vendo em suas propriedades o error code atribuído.

| | |
|---|-----------------------------------|
| - End Event: Rejected by Customer Care Manager | |
| Name | Rejected by Customer Care Manager |
| Trigger | Error |
| Error code | REJECTED_BY_CUSTOMER_CARE_MANAGER |

No processo de FlightTicketReservation, temos um Signal Error Event definido como catch, com o mesmo nome.



E sobretudo com o mesmo Error Code, o que garante que capturará o evento disparado no subprocesso ValidateReservation, o que fará com que se emita um aviso de notificação ao cliente.

| | |
|--|-----------------------------------|
| - Intermediate Event: Rejected by Customer Care Manager | |
| Name | Rejected by Customer Care Manager |
| Trigger | Error |
| Error code | REJECTED_BY_CUSTOMER_CARE_MANAGER |

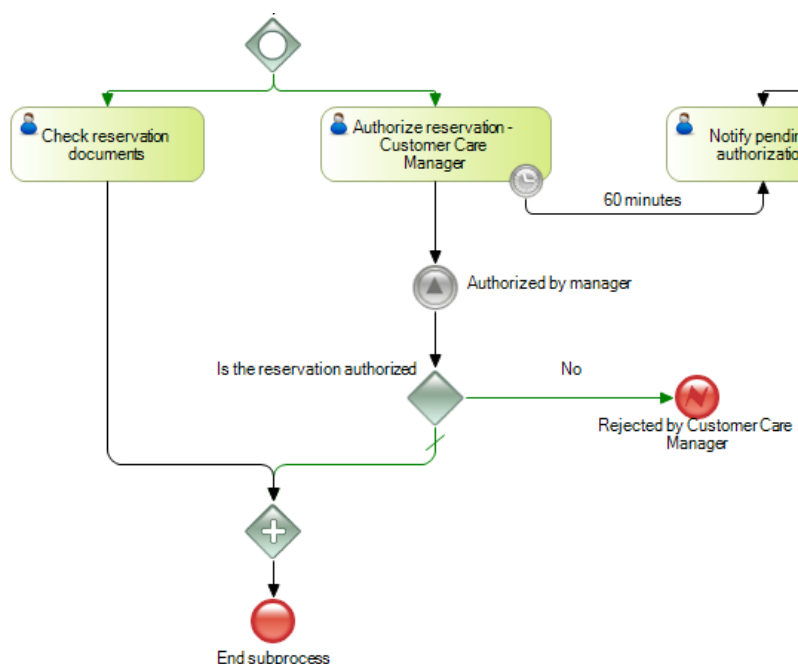
Voltando ao subprocesso de validação da reserva, a tarefa de revisão de documentos também termina em um Exclusive Gateway. Neste caso, deve-se avaliar se a documentação está em ordem. Se não estiver, a tarefa **"Add customer information required for traveling"** precisa ser executada novamente.

Quando fizemos o modelo, supomos que a tarefa interativa **"Check reservation documents"** estaria associada a um objeto GeneXus que seria o responsável por carregar um dado relevante com um valor que depois seria avaliado no exclusive Gateway.

No entanto, já vimos que o mesmo motor de workflow se responsabiliza por verificar que não podemos deixar de ler o documento, já que na tarefa tínhamos definido uma operação de Read como Required.

Portanto, não é preciso ter um Gateway que controle se a tarefa foi ou não realizada, já que o controle é feito pelo motor quando executamos a tarefa e ele não nos deixa continuar se não lermos o documento.

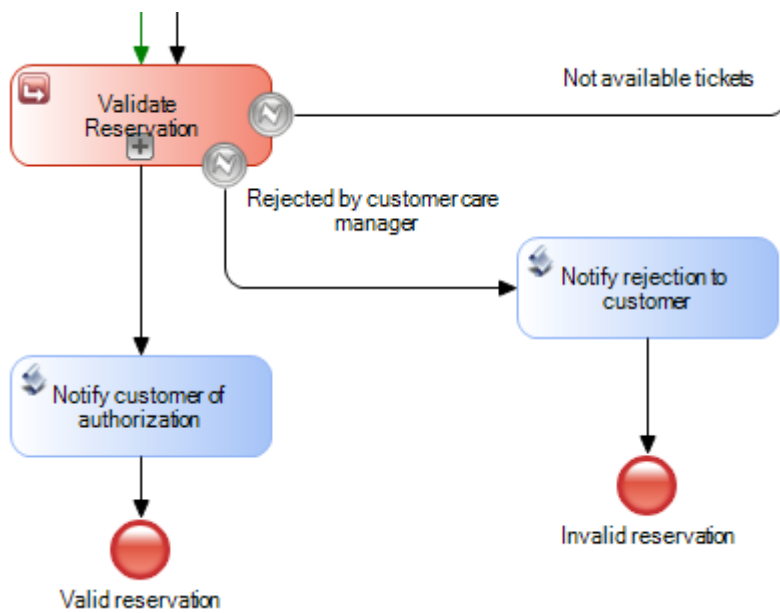
Assim, modificamos o diagrama eliminando o exclusive Gateway **"Is the documentation in order"**, por isso agora a tarefa **"Check reservation documents"** se conecta diretamente com o parallel Gateway.



O parallel Gateway sincroniza todos os caminhos que estão conectados a ele no diagrama e espera até que todos os caminhos cheguem ali para deixar o processo continuar.

Se executamos a tarefa de controle de documentos revisando os documentos e se a reserva foi autorizada pelo gerente de atendimento ao cliente, então o parallel Gateway deixará o fluxo continuar finalizando o subprocesso.

Neste caso, a finalização da validação foi bem-sucedida, de modo que o controle voltará ao processo principal FlightTicketReservation.



Voltando a esse processo com uma validação bem-sucedida, o cliente será notificado de que sua reserva foi autorizada e o processo de reserva de passagens é finalizado.

Nos próximos vídeos, contemplaremos mudanças no diagrama devido a novos requerimentos da agência de viagens.