

# Exercícios práticos

Curso GeneXus Advanced

## **GeneXus<sup>TM</sup> 18**

Dezembro 2022

*© GeneXus. Todos os direitos reservados. Este documento não pode ser reproduzido em nenhum meio sem a autorização expressa de GeneXus<sup>TM</sup>. As informações contidas neste documento são apenas para uso pessoal.*

*Marca registrada:*

*GeneXus é uma marca comercial ou uma marca registrada de GeneXus S.A. Todas as outras marcas comerciais mencionadas neste documento são propriedade de seus respectivos donos*

## CONTEÚDO

CONTEÚDO .....	2
O PROBLEMA.....	5
NOVO PROJETO, NOVA BASE DE CONHECIMENTO .....	5
MODELO DE DADOS: TRANSAÇÕES E SEU DESENHO INICIAL.....	6
INSIRA DADOS EM EXECUÇÃO .....	8
REGRAS.....	9
Client Side Validation Behaviour [sugerido] .....	11
Transação “Schedule” .....	11
MAIS REGRAS PARA DEFINIR COMPORTAMENTO.....	12
opcional.....	17
EVENTOS DE DISPARO DE REGRAS EM TRANSAÇÕES.....	17
Eventos de disparo de regras em transações (continuação).....	18
Regras e eventos no nível das linhas .....	19
Opcional .....	20
Árvore de avaliação de disparo de regras e fórmulas .....	21
Novo requisito:.....	24
Novo requisito:.....	26
opcional (controles sobre linhas em modo update).....	27
opcional: Exercício mental sobre árvore de avaliação e eventos .....	28
EVENTOS DE DISPARO DE REGRAS EM TRANSAÇÕES (FINAL).....	29
Permitir valores nulos em parte de uma chave estrangeira composta.....	31
Consequências de adicionar um atributo a uma tabela que já tem dados [SUGERIDO] .....	31

Transações Paralelas [SUGERIDO].....	32
<b>SUBTIPOS.....</b>	<b>33</b>
Subtipos: Referência Múltipla e Especialização .....	33
Mais casos de uso de subtipos .....	35
<b>FÓRMULAS .....</b>	<b>35</b>
Fórmulas versus regras .....	35
Fórmulas horizontais (opcional) .....	36
Fórmulas aggregate.....	37
Transações Dinâmicas .....	39
Eventos em Transações .....	40
Mais sobre For eachs aninhados. Casos e navegação.....	40
Sub-rotinas .....	42
Cláusula Unique.....	43
Data Selector.....	45
<b>CLÁUSULAS DA PAGINAÇÃO.....</b>	<b>47</b>
Paginação em For each, Pattern e grid (OPCIONAL).....	47
<b>BUSINESS COMPONENT.....</b>	<b>48</b>
Atualização de Encarregados .....	48
Trabalhando com os Filmes e seus Atores .....	48
<b>ATUALIZAÇÃO DA BASE DE DADOS COM PROCEDIMENTO.....</b>	<b>49</b>
<b>TELAS WEB: WEB PANELS.....</b>	<b>49</b>
Variáveis em grid e ações no pattern Work With (OPCIONAL).....	50
Criando telas interativas.....	50

---

Múltiplos Grids .....	51
Grid sem tabela base .....	53
Carregando Coleções .....	54
Salvar informações de contexto (OPCIONAL).....	54
Tela especial para a venda de tickets .....	54
<b>TELAS MULTI-CHANNEL: PANELS .....</b>	<b>57</b>
Testando a aplicação em smart devices.....	59
<b>APLICAÇÕES WEB PROGRESSIVAS [SUGERIDO] .....</b>	<b>60</b>
<b>WEB SERVICES [SUGERIDO] .....</b>	<b>64</b>

## O PROBLEMA

Uma empresa responsável pela administração de um Cinema o contrata para que desenvolva um sistema com a finalidade de armazenar, manipular e exibir as informações com que trabalha. Imagine que o sistema é composto por estes módulos:

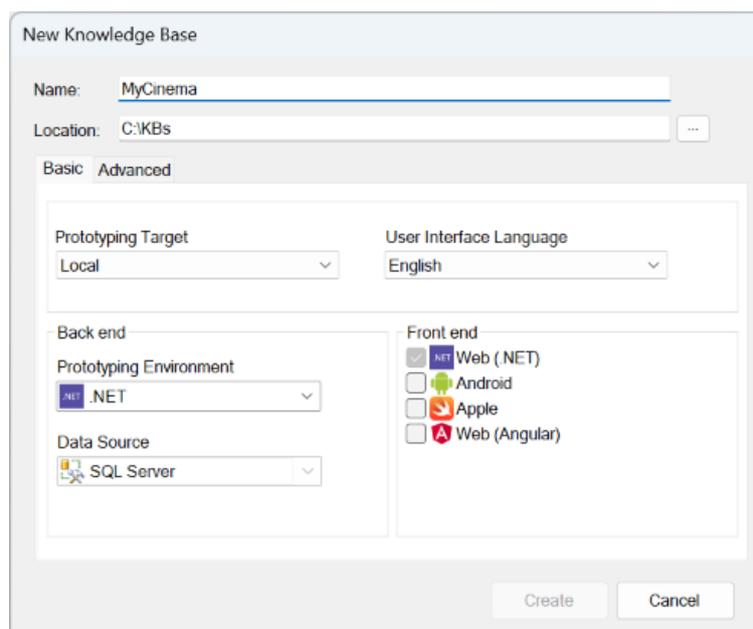
- **Backoffice:** parte da aplicação que deverá ser executada em um servidor web, de tal maneira que os funcionários da empresa possam manipular as informações a partir de qualquer lugar com conexão à internet.
- **Frontoffice:** parte da aplicação que deverá ser executada em um servidor web, de tal maneira que os clientes possam ver as informações com uma estética elegante.

## NOVO PROJETO, NOVA BASE DE CONHECIMENTO

Entrar em GeneXus e criar uma base de conhecimento chamada *MyCinema* para iniciar o desenvolvimento da aplicação.

Sugerimos:

- Escolher como ambiente de desenvolvimento Net Core (.Net). Certifique-se de ter instalado tudo o que é necessário (incluindo SQL Server). Se usa GeneXus Trial, o ambiente de geração com Net Core e SQL Server já está predefinido, prototipando na nuvem da Amazon.
- Não criar a base de conhecimento na pasta “Meus Documentos” ou qualquer outra pasta que esteja em “Documents and Settings”, pois estas pastas têm permissões especiais concedidas pelo Windows.



NOTA: Este prático contém dois tipos de exercícios:

1. Exercícios que deverão ser feitos no IDE de GeneXus
2. Exercícios que podem ser omitidos na aplicação, mas há um espaço no prático para que o aluno escreva as respostas. (Estes exercícios também podem ser replicados no IDE)

## MODELO DE DADOS: TRANSAÇÕES E SEU DESENHO INICIAL

O cinema administra os painéis publicitários de filmes que são exibidos em uma série de salas distribuídas pela cidade. As pessoas poderão comprar ingressos (tickets) para uma exibição determinada de um filme que é exibido em alguma das salas, conforme o painel publicitário.

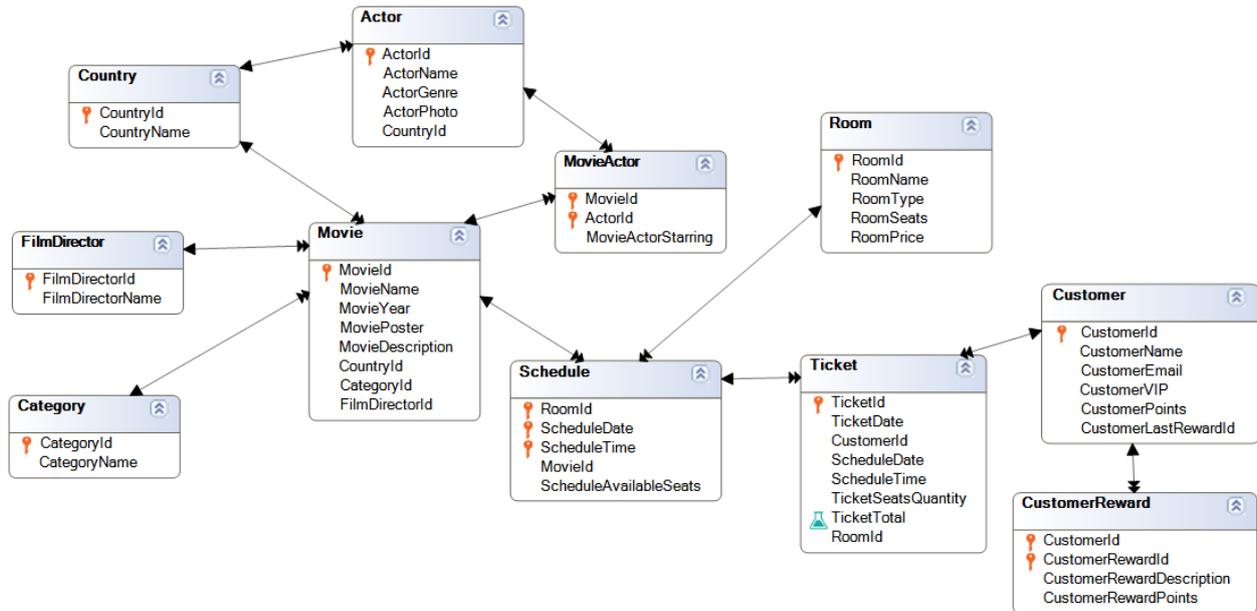
Interessa gerenciar informação dos filmes, seu diretor, os atores principais, a categoria do filme, seu país, etc.

E também dos clientes do cinema, que poderão ter benefícios de acordo com o uso que fizerem dos serviços.

Para inicializar a KB com as transações básicas para começar a modelar esta realidade, você deve importar para a KB o arquivo MyCinema.xpz anexado. No *Folder Trn* você verá, entre outras coisas, as transações:

- Actor
- Category
- Country
- Customer
- FilmDirector
- Movie
- Room
- Schedule
- Ticket

Analise atentamente sua estrutura e infira as relações entre as entidades da realidade que modelam. Para fazer isso, recomendamos criar um diagrama de transações. Você também pode montar um de tabelas (lembre-se que com View > Tables você pode obter o nó Tables no KB Explorer com todas as tabelas juntas, para arrastá-las de uma só vez para o objeto de tipo Diagram). Se você não quer investir tempo fazendo isso, já fornecemos feito o de tabelas (objeto Diagram1).



A transação Movie é aquela que registra a informação relevante de um filme: seu nome, ano, o cartaz ou poster que o representa, uma descrição ou sinopse, o país do filme, sua categoria (comédia, drama, suspense, etc.) e seu diretor, além da lista de atores (identificando com o atributo booleano MovieActorStarring se o ator é protagonista ou coadjuvante).

Observe que os atores poderão, é claro, ser de países diferentes do país do filme.

Por outro lado, temos as salas (Room) em que os filmes serão exibidos. Cada sala, além de um nome e um tipo (se é sala normal, VIP ou 3D), registra a quantidade de assentos que possui e o preço do ingresso de cinema para essa sala. Aqui, obviamente, está sendo considerado que os preços da exibição dependem da sala.

Por outro lado, temos a própria “apresentação” ou “exibição”, que se dá pelo painel publicitário, Schedule, onde é registrado para cada sala, dia e hora de exibição, o filme que será exibido e a quantidade de lugares ainda disponíveis (não vendidos) para essa exibição. Observe o identificador composto por: sala, data e hora da exibição (as exibições ocorrem em 4 horários fixos e por isso ScheduleTime é do tipo enumerado). De acordo com esta modelagem, um mesmo filme poderá ser exibido em duas salas diferentes no mesmo dia e no mesmo horário.

Quando uma pessoa compra bilhetes para uma exibição, é registrado um Ticket, onde são armazenados os dados da exibição e a quantidade de lugares adquiridos, e é calculado o preço total do ticket por esse conceito. Se a pessoa existir como cliente (customer), ela é associada ao ticket.

Além disso, o cliente gera pontos por compra de tickets, que depois poderá trocar por prêmios específicos, que também serão registrados.

Observe que foram criados índices unique na maioria das tabelas para estabelecer chaves candidatas por nome.

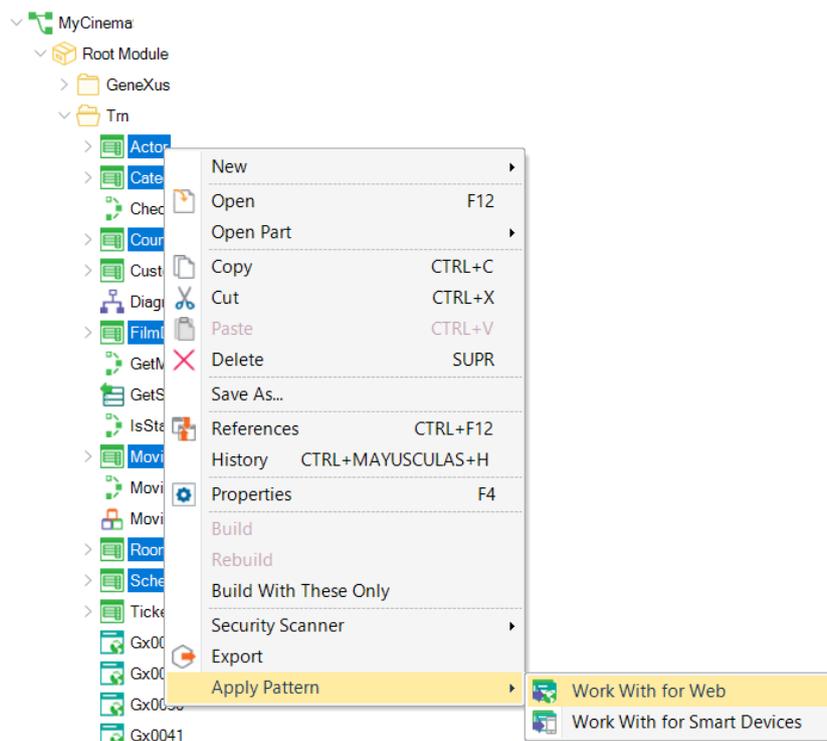
## INSIRA DADOS EM EXECUÇÃO

Clique em Run Developer Menu. Pressione Create na listagem Impact Analysis, e enquanto GeneXus cria as tabelas, especifica e gera, observe o seguinte, levando em conta que o domínio Id é autonumerado:

1. O que significa o aviso sobre as tabelas CustomerReward e Customer na listagem de análise de impacto? Lembre-se que somente podem ser autonumerados atributos que sejam, por si só, PK da tabela. Se são parte de uma PK, ou não são PK de forma alguma, não podem ser autonumerados. Simplesmente é ignorada sua propriedade AutoNumber.
2. Observe que para algumas tabelas (Room, FilmDirector, CustomerReward, Customer, Actor, MovieActor, Movie, Country e Category) é informado que, embora a tabela “is new”, será inicializada com um Data Provider. Se você observar embaixo dessas transações, encontrará esses Data Providers. Serão utilizados para carregar automaticamente dados nas tabelas na próxima execução. Entraremos neste assunto quando estudarmos este tema mais adiante. Por enquanto somente repare em sua existência.

Navegue pelos dados carregados até se familiarizar com as transações. Observe que as transações Schedule e Ticket não possuem dados pré-carregados.

Aplique o pattern Work With for Web a todas as transações, exceto Ticket e Customer. Nota: Lembre-se de que você pode selecionar todas elas no KB Explorer e, a partir dali, clicar com o botão direito:



Execute, e enquanto espera que GeneXus termine, continue lendo.

## REGRAS

Observe a transação Room e suas regras (aquelas não geradas pelo pattern):

```
1 | Msg("Room name is expected")
2 | |   if RoomName.IsEmpty();
3 |
4 | |   Error("Room seats must be greater than 0")
5 | |   if Insert and RoomSeats = 0;
6 |
7 | |   Msg("Room price not specified")
8 | |   if RoomPrice.IsEmpty();
9 |
```

Implementam adequadamente os requisitos solicitados a seguir?:

- É desejado avisar o usuário se o nome da sala estiver vazio, se o preço do ingresso da sala ficou vazio e se o tipo de sala ficou sem especificar (observe o tipo de dados do atributo RoomType).
- Deve ser impedida a inserção de uma sala com zero assentos.

Teste em execução:

- Deixar vazio o nome de uma sala previamente carregada. Verá que permite que você faça isso, mas mostrando a mensagem. É que a regra não está condicionada a um modo em particular. Poderá gravar a sala com nome vazio?
- Tentar deixar vazio “valor (None)” o combo para RoomType. Observe que sem ter programado nenhuma regra, está sendo exibida a mensagem “Field Room Type is out of range” e não permite gravar.

Observe que para o atributo RoomType, de tipo enumerado, modificamos a propriedade do grupo Control Info para que permita o valor vazio. Caso contrário, nem sequer veria no combo a opção “(None)”.

<b>▼ Type Definition</b>	
Supertype	
Based on	<b>RoomType:Domain</b>
Data Type	Numeric
Length	1
Decimals	0
Signed	False
Initial value	
<b>&gt; Validation</b>	
<b>▼ Control Info</b>	
Control Type	Combo Box
Values	Normal:1,VIP:2,3D:3,(None):0
Empty Item	<b>True</b>
Empty Item Text	GX_EmptyItemText
Notify Context Char	False

Mas esse valor não é um valor válido do domínio. O colocamos assim no nível de atributo para que automaticamente em todos os Web Panels onde vá ser utilizada uma variável baseada no atributo, permita escolher esse “(none)”, vazio. Mas não faz sentido no nível do atributo na transação, porque esse valor não é um valor válido para o atributo. Se quisesse que GeneXus permita inserir um valor vazio no atributo, ou seja, que o usuário possa não especificá-lo, modifique o domínio adicionando-o:

Name	Description	Value
Normal	Normal	1
VIP	VIP	2
in3D	3D	3
None	(None)	0

Buttons: Add, Remove, Edit, Move Up, Move Down, OK, Cancel

Observe que é o valor 0 o que fará com que a pergunta para “lsempty()” resulte True. Então, retorna o valor False para a propriedade Empty Item do atributo RoomType e executa agora novamente a transação com esta alteração, deixando em (None) o valor do atributo. Agora não tem nenhum problema deixando o tipo de sala sem especificar. Teste.

Deveria adicionar uma **regra Msg** para avisar o usuário que deixou o valor sem especificar, caso não fosse o que queria.

- Adicione uma nova sala com zero assentos. Não vai deixar você fazer isso. Mas, e se para uma que já existia você mudar o valor para 0? Não deveríamos permitir. Modifique a regra para que **nunca** seja possível deixar em zero esse valor quando se utiliza a transação.

Observação: É possível repetir o nome de uma sala? Observe que criamos um índice único por RoomName e por isso está sendo realizado o controle de duplicados.

#### CLIENT SIDE VALIDATION BEHAVIOUR [SUGERIDO]

Antes de continuar testando, interessa que em todas as transações não seja permitido continuar para o próximo campo quando seja disparada uma regra error.

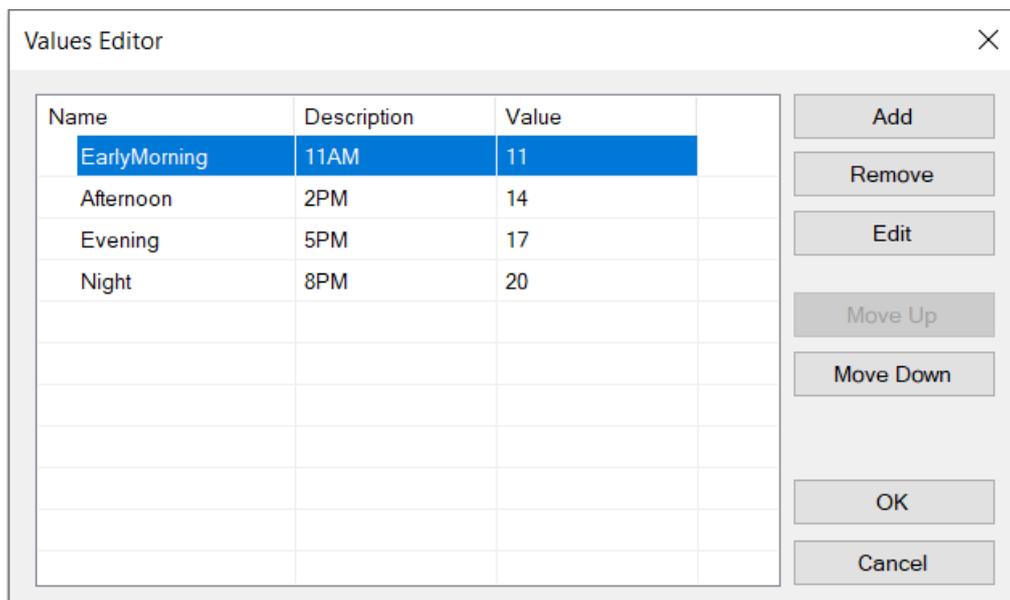
Também interessa que todas as mensagens das regras apareçam na parte inferior do campo.

Faça o que for necessário para obter este comportamento e volte a testar com Room.

Agora concentre-se na seguinte transação.

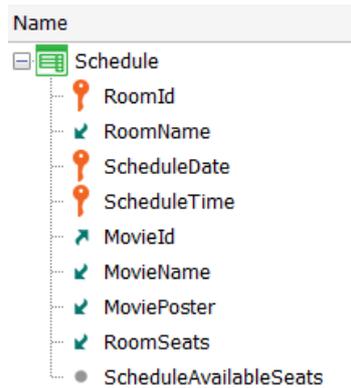
#### TRANSAÇÃO “SCHEDULE”

O painel publicitário é composto pela sala (Room), a data e o horário em que serão apresentados os filmes (apenas nestes horários: 11:00 AM, 2:00 PM, 5:00 PM e 8:00 PM). Para registrar esses quatro horários foi criado um domínio enumerado, MovieTime.



Aqui não adicionamos valor (None), mas adicionamos a possibilidade do elemento vazio no combo.

Observe que optamos por utilizar um identificador composto e não um autonumerado. Com este desenho, um mesmo filme poderá ser exibido em duas salas diferentes no mesmo dia e horário?



A resposta é sim e isto está correto porque nos informam que faz parte do requisito.

O atributo ScheduleAvailableSeats armazenará a quantidade de assentos disponíveis (ainda não vendidos), portanto é inicializado com a capacidade da sala (RoomSeats).

Observe as regras declaradas na transação (fora daquelas do pattern) e teste em execução inserir uma nova exibição, onde:

- RoomId seja inexistente. Qual mensagem é exibida? Coloque um existente.
- A data da exibição seja anterior a hoje. Coloque uma posterior.
- Deixar sem especificar o horário. Acontece o mesmo que vimos para o caso de RoomType. Aqui precisamos sem falta que o usuário insira um valor válido, já que o horário faz parte da chave. Insira um dos quatro.
- Deixar sem especificar o MovieId ou atribuir um inexistente. Coloque um válido.
- Por que RoomSeats e AvailableSeats têm o mesmo valor e são readonly? Em que momento foi visto o valor de RoomSeats e o de AvailableSeats? Caso não tenha percebido, confirme e recomece inserindo outro Schedule. Preste atenção a estes campos quando sai de RoomId.

#### MAIS REGRAS PARA DEFINIR COMPORTAMENTO

Observe a transação Ticket, que corresponde aos ingressos que são vendidos a uma pessoa para uma exibição. Se a pessoa está inserida no sistema como cliente, então ela será atribuída ao ticket, caso contrário, o campo será deixado vazio. A data do ticket será a data de emissão.

Para o ticket, são registrados os dados da exibição (conforme Schedule) e a quantidade de lugares que estão sendo vendidos para a pessoa, apresentando o preço total (calculado de acordo com a quantidade de lugares e o custo da exibição nessa sala).

Name	Type	Description	Formula	Nullable
Ticket	Ticket	Ticket		
TicketId	Id	Ticket Id		No
TicketDate	Date	Ticket Date		No
CustomerId	Id	Customer Id		Yes
CustomerName	Name	Customer Name		
RoomId	Id	Room Id		No
RoomName	Name	Room Name		
RoomType	RoomType	Room Type		
ScheduleDate	Date	Schedule Date		No
ScheduleTime	MovieTime	Schedule Time		No
MovieId	Id	Movie Id		
MovieName	Name	Movie Name		
RoomSeats	Numeric(4.0)	Room Seats		
RoomPrice	Price	Room Price		
TicketSeatsQuantity	Numeric(4.0)	Ticket Seats Quantity		No
TicketTotal	Price	Ticket Total	$RoomPrice * TicketSeatsQuantity$	

Observe as regras declaradas e reflita se elas implementam adequadamente os requisitos listados a seguir. Se não o fizerem, modifique-as para que o façam:

#### Ticket

- Por padrão, deve ser sugerida a data do dia de hoje. Não permitir modificá-la quando estiver sendo inserida. Mas sim em outro modo.
- A data do filme não deve ser menor que a data de compra. Isto apenas em caso de inserção.

Execute a aplicação e adicione um par de tickets de venda, testando as regras. Por exemplo:

- Insira um ticket e observe em que momento são disparadas as regras que envolvem TicketDate.
- Edite o ticket que acabou de inserir e observe se agora você pode alterar a data do ticket. Coloque uma data posterior à do filme. Permite continuar e até gravar? Por que não está sendo disparada a regra de erro?
- Preste atenção ao seguinte: Pode modificar o cliente de um ticket previamente inserido? e a exibição (Schedule)? Verdade que sim? Não são campos readonly. A partir da adição das seguintes regras, você verá que não poderá mais modificá-los.

É requerido adicionar o seguinte comportamento:

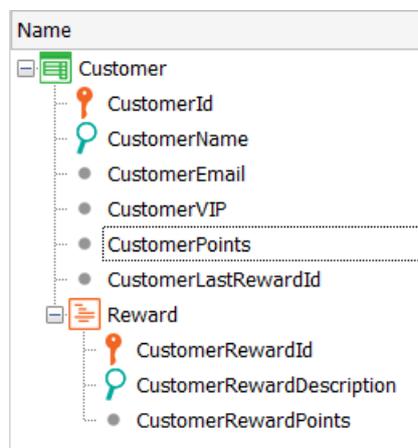
1. Verificar se a quantidade de lugares que estão querendo comprar para a exibição (Schedule) está disponível. Caso contrário, que não seja permitida a venda.
2. Seja diminuída a quantidade de lugares disponíveis para a exibição (ScheduleAvailableSeats) de acordo com a quantidade de lugares que estão sendo comprados.

3. Caso a venda esteja associada a um cliente, deverão ser somados à conta do cliente os pontos gerados por esta compra. Cada compra de ingressos gera 10% do total do ticket como pontos.

Adicione as regras à transação Ticket que você acha que são necessárias para alcançar este comportamento.

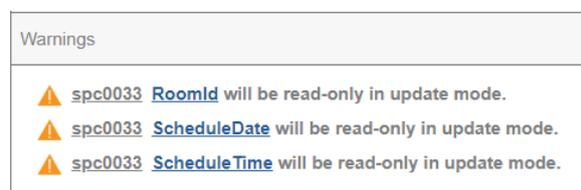
Nota: lembre-se que se precisar utilizar um atributo nas regras da transação você deverá tê-lo declarado na estrutura, mesmo que seja inferido.

Observe que na transação Customer tem o atributo CustomerPoints para armazenar os pontos que o cliente gerou até o momento.



Com as alterações que realizou, teste em execução:

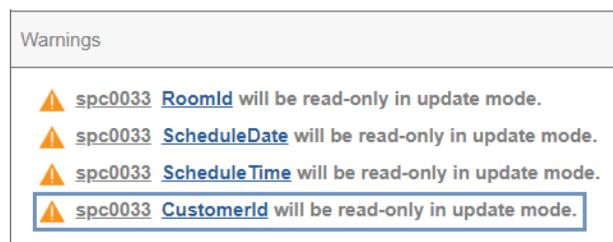
1. Inserir um novo Ticket sem associar cliente e usando uma quantidade de lugares que não exceda os disponíveis. Observe que fique atualizada a quantidade de lugares disponíveis da sala para a exibição (Schedule).
2. Edite o ticket que acabou de inserir. Pode modificar a sala, a data ou o horário da exibição? Não mais! Volte ao GeneXus e observe estes warnings que apareceram na listagem de navegação (por enquanto vamos deixar de lado o de CustomerId):



Apareceram quando adicionou a regra Subtract. Tente comentá-la e obter novamente a listagem de navegação (sem gerar o objeto: posicionando-se sobre sua aba e com botão direito do mouse escolhendo View Navigation). Você verá que não estão mais lá os

warnings. Por que aparecerá esta restrição? Deixe novamente a regra subtract descomentada.

3. Insira um novo Ticket associando agora um cliente, e usando uma quantidade de lugares que não exceda os disponíveis. Observe que foram atualizadas a quantidade de assentos disponíveis da sala e a quantidade de pontos do cliente.
4. Edite o ticket anterior. Pode mudar o cliente do ticket? Na listagem de navegação também encontrará o aviso, que está relacionado coma regra Add. Por que você acha que a partir da introdução dessa regra não permite mais modificar o cliente?



5. Para o ticket que está atualizando, modifique a quantidade de assentos que foram inseridos. Confirme que são atualizadas tanto a quantidade de assentos disponíveis da sala quanto os pontos do cliente.
6. Edite ou insira um ticket e tente usar mais assentos do que os disponíveis. Não deverá permitir gravar.
7. Exclua um ticket com cliente e observe que os pontos desse cliente são restituídos.

Ficou assim a estrutura da transação?

Name	Type	Description	Formula	Nullable
Ticket	Ticket	Ticket		
TicketId	Id	Ticket Id		No
TicketDate	Date	Ticket Date		No
CustomerId	Id	Customer Id		Yes
CustomerName	Name	Customer Name		
CustomerPoints	Numeric(4,0)	Customer Points		
RoomId	Id	Room Id		No
RoomName	Name	Room Name		
RoomType	RoomType	Room Type		
ScheduleDate	Date	Schedule Date		No
ScheduleTime	MovieTime	Schedule Time		No
MovieId	Id	Movie Id		
MovieName	Name	Movie Name		
RoomSeats	Numeric(4,0)	Room Seats		
RoomPrice	Price	Room Price		
ScheduleAvailableSeats	Numeric(4,0)	Schedule Available Seats		
TicketSeatsQuantity	Numeric(4,0)	Ticket Seats Quantity		No
TicketTotal	Price	Ticket Total	RoomPrice*TicketSeatsQuantity	
TicketPoints	Numeric(4,0)	Ticket Points	TicketTotal*.10	

O tipo de sala não pode ser alterado a partir da transação Ticket porque é inferido. Pode conseguir que o usuário possa modificá-lo a partir desta mesma tela, sem necessidade de ter que fazê-lo indo para a transação Room? Tente.

Como {RoomId, ScheduleDate, ScheduleTime} formam juntos uma chave estrangeira para a tabela Schedule, até que o usuário insira o último dos valores, não é feita a verificação de integridade sobre essa tabela e a mensagem pode ser confusa para o usuário.

Se em vez desta mensagem:

Room Id	6
Room Name	Room 6
Room Type	VIP
Schedule Date	05/28/21
Schedule Time	8PM
Movie Id	0

No matching 'Schedule'.

queremos exibir esta outra:

Room Id	6
Room Name	Room 6
Room Type	VIP
Schedule Date	05/28/21
Schedule Time	8PM
Movie Id	0

No movie for Room: 6, Day:05/28/21, Time: 8PM

Como você faz isso? Se precisa ver a sintaxe exata da regra quando a chave estrangeira é composta, como neste caso, procure-a em [wiki.genexus.com](http://wiki.genexus.com).

Você pode montar a string para a saída de maneira simples assim:

```
"No movie for Room:" + RoomId.ToString() + ", Day:" + ScheduleDate.ToFormattedString() + ", Time: " + ScheduleTime.EnumerationDescription()
```

Execute e teste as regras que você adicionou.

## OPCIONAL

Por outro lado, para cada cliente são registradas as recompensas que ele pode trocar por seus pontos:

Name	Type	Description	Formula	Nullable
Customer	Customer	Customer		
CustomerId	Id	Customer Id		No
CustomerName	Name	Customer Name		No
CustomerEmail	Email, GeneXus	Customer Email		No
CustomerVIP	Boolean	Customer VIP		No
CustomerPoints	Numeric(4,0)	Customer Points		No
CustomerLastRewardId	Id	Customer Last Reward Id		No
Reward	Reward	Reward		
CustomerRewardId	Id	Customer Reward Id		No
CustomerRewardDescription	Numeric(4,0)	Customer Reward Description		No
CustomerRewardPoints	Numeric(4,0)	Customer Reward Points		No

Estas recompensas são exclusivas do cliente e é por isso que foram modeladas como entidade fraca. Para cada recompensa é registrada a quantidade de pontos que vale. Portanto, quando um cliente deseja trocar alguns de seus pontos gerados por uma recompensa, ao inseri-la como linha da transação, deverão ser atualizados os pontos do cliente (CustomerPoints).

Obviamente, não deve ser permitida uma recompensa se o cliente não tiver os pontos que vale.

Implemente este comportamento e teste-o em execução.

## EVENTOS DE DISPARO DE REGRAS EM TRANSAÇÕES

Antes de iniciar esta série de exercícios, recomendamos que você esboce em um pedaço de papel a ordem dos eventos de disparo de regras em transações que conhece até o momento, o que às vezes também chamamos de “momentos de disparo”. Partir desse esboço, claro, será de grande utilidade para aproveitar ao máximo os exercícios. Propomos exercícios ligados a cada um dos três vídeos que trabalham este tema. Você pode ver cada vídeo em ordem e ir fazendo os exercícios relacionados, ou pode ver os três e em seguida fazer todo o bloco do prático.

Recomendamos que você faça todas as provas sobre a transação Movie com o filme “Gone with the Wind”, do diretor Victor Fleming, dos Estados Unidos, tipo Drama e atores “Clark Gable”, “Olivia de Havilland” e “Vivien Leigh”. Também recomendamos tornar invisível o atributo MovieDescription do arquivo de instância do pattern Work With Web.

## EVENTOS DE DISPARO DE REGRAS EM TRANSAÇÕES (CONTINUAÇÃO)

Suponhamos que temos um Web Service para o qual podemos passar o nome de um filme e ele retorna seu ano. Para tornar isso mais simples (porque ainda não vimos nada de Web Services), em seu lugar utilizaremos o procedimento GetMovieYear que vem com o xpz.

Qual será o comportamento da transação Movie se adicionarmos a seguinte regra:

```
MovieYear = GetMovieYear(MovieName) if Insert;
```

Antes de testá-lo em execução, pense em que momento deverá ser disparado. Claramente será unicamente quando estiver sendo inserido um filme novo, e não em outro modo.

1. Quando é aberta a transação em modo Insert e antes de passar pelo campo MovieName?
2. Quando sai do campo MovieName e entra no campo MovieYear?
3. Depois de sair do campo MovieYear?
4. Somente quando se confirma?

Antes de testá-lo, mova MoviePoster para que entre MovieName e MovieYear haja outro atributo.

Agora sim, teste inserindo o filme “Gone with the Wind” (escrito exatamente assim). Procure na pasta Movies que fornecemos com o material, a imagem para o pôster. Depois de inserido mude o nome e observe como a data não é modificada.

Agora mude a regra anterior, condicionando-a desta outra forma:

```
MovieYear = GetMovieYear(MovieName) if MovieYear.IsEmpty() on BeforeInsert;
```

Quando será disparada?

Tente inserir novamente o filme “Gone with the Wind”, deixando o campo MovieYear vazio e observe que não será disparada a regra enquanto não confirme. Por quê?

Exclua o filme e volte a inseri-lo, mas agora deixando no campo MovieYear um valor qualquer diferente de 1939. É disparada?

Que diferença haverá entre condicionar a regra como você a tinha e condicioná-la de acordo com a opção 2 seguinte?

1. `MovieYear = GetMovieYear(MovieName) if MovieYear.IsEmpty() on BeforeInsert;`
2. `MovieYear = GetMovieYear(MovieName) if MovieYear.IsEmpty() and Insert on AfterValidate;`

Resposta: o evento AfterValidate é imediatamente anterior ao BeforeInsert / BeforeUpdate / BeforeDelete, mas entre eles nenhuma ação ocorre, então são quase equivalentes.

Teste agora alterar o evento de disparo da regra para AfterInsert, o momento imediatamente posterior ao BeforeInsert, e tente inserir novamente o filme “Gone with the Wind”, deixando vazio o valor MovieYear. Atribui o valor 1939 após confirmar ou deixa vazio?

Volte a deixar a regra condicionada corretamente: “`if MovieYear.IsEmpty() on BeforeInsert;`”

E exclua o filme.

---

## REGRAS E EVENTOS NO NÍVEL DAS LINHAS

Observe que com o xpz também veio um procedimento IsStarring, que utiliza o Data Provider GetStarrings e o SDT MovieStarringActors. A ideia é que ao procedimento seja passado um nome de filme e nome de ator e retorne true se esse ator é protagonista (starring) do filme. Outra vez, estaríamos emulando a utilização de um web service que nos forneça essa informação.

Portanto, poderia adicionar a seguinte regra à transação Movie:

```
MovieActorStarring = IsStarring(MovieName, ActorName) if Insert;
```

Volte a inserir o filme “Gone with the Wind”, com os atores “Clark Gable”, “Olivia de Havilland” e “Vivien Leigh”.

A regra será disparada no cliente (além de ser disparada depois também no servidor). Quando é disparada no cliente? Pense no momento preciso para a execução mencionada em que insere os três atores. Quantas vezes terá sido disparada antes que o usuário pressione o Confirm? E depois do Confirm, no servidor, quando serão disparadas?

\*\*\*

E se, em vez de atribuir diretamente valor ao atributo MovieActorStarring a partir do procedimento, quisermos deixar que o usuário insira o valor, mas verificar contra o procedimento e perguntar se tem certeza de que deseja deixar esse valor, caso não coincida? Ou seja, algo assim:

```
msg("Are you sure about starring?") if MovieActorStarring <> IsStarring(MovieName,  
ActorName);
```

Outra possibilidade seria que, no caso em que o usuário tenha deixado em false o campo, então ali seja sobrescrito com o valor retornado pelo procedimento:

```
MovieActorStarring = IsStarring(MovieName, ActorName) if Insert and MovieActorStarring  
= False;
```

Observe inserindo “Gone with the Wind” que é somente quando sai do campo MovieActorStarring que a regra é disparada (teste com Clark Gable deixando em False o campo e com Olivia de Havilland (que não é protagonista) também em False).

Observe que neste caso marcará o check box para Clark Gable, e se depois for explicitamente e o desmarcar, o marcará novamente, ou seja, volta a disparar a regra.

Se o que é desejado é que o usuário insira em sua tela todos os atores marcando o check box como quiser e somente quando confirmar para aqueles atores que estavam em False seja atribuído o valor de acordo com o procedimento, como você deve modificar a regra? Propomos 4 opções:

1. `MovieActorStarring = IsStarring(MovieName, ActorName) if MovieActorStarring = False on BeforeInsert;`
2. `MovieActorStarring = IsStarring(MovieName, ActorName) if MovieActorStarring = False on AfterInsert;`
3. `MovieActorStarring = IsStarring(MovieName, ActorName) if MovieActorStarring = False on AfterLevel Level ActorId;`
4. `MovieActorStarring = IsStarring(MovieName, ActorName) if MovieActorStarring = False on Aftercomplete;`

Se agora observar estas duas regras:

1. `MovieYear = GetMovieYear(MovieName) if MovieYear.IsEmpty() on BeforeInsert;`
2. `MovieActorStarring = IsStarring(MovieName, ActorName) if MovieActorStarring = False on BeforeInsert;`

Como GeneXus sabe que a primeira será disparada imediatamente antes de inserir o cabeçalho na tabela Movie, e em vez disso a segunda será disparada imediatamente antes de inserir a linha na tabela MovieActor (para cada linha)?

---

#### OPCIONAL

Se tiver tempo e quiser, pode tentar adicionalmente o seguinte: faça um “Save as” da transação Movie, MovieCopy1, removendo todas as regras que o pattern adicionou e todos os eventos (da aba Events), bem como colocando em False a propriedade Data Provider. Desta forma, poderá trabalhar diretamente sobre ela (mais adiante verá o tema Transações Paralelas onde verá que trabalhará sobre a mesma tabela que Movie) invocando-a a partir do Developer Menu.

E adicione as seguintes regras a ela, definindo a variável &ActorLine de tipo Numeric:

```
&ActorLine = &ActorLine + 1
    on AfterInsert
    Level ActorId;

Msg(&ActorLine.ToString() + " actors. Directed by: " + FilmDirectorName)
    If Insert
    on AfterLevel
    Level ActorId;
```

A ideia é somar 1 depois de inserir cada linha na tabela MovieActor e disparar a mensagem depois de terem sido inseridas todas as linhas. Teste em execução.

Em seguida, altere a cláusula Level da primeira regra, para que agora seja o level de FilmDirectorId e observe o que acontece.

### ÁRVORE DE AVALIAÇÃO DE DISPARO DE REGRAS E FÓRMULAS

Na transação Movie já havíamos nos perguntado quando seria disparada esta regra:

```
MovieYear = GetMovieYear(MovieName) if Insert;
```

Se no layout web os controles estiverem nesta ordem:

Se pensarmos que cada regra será disparada “o mais breve possível”, e que “o possível” tem a ver com qual informação precisa ter para isso, ficará claro que a atribuição ao MovieYear dependerá do modo da transação (Insert), e de ter o valor que o usuário tenha dado ao atributo MovieName.

- Se a transação é executada recebendo por parâmetro o modo (como quando é invocada pelo Work With) então assim que for aberta já saberá se o modo é Insert ou não. Mas como ainda precisa saber o valor de MovieName, não poderá disparar a regra até que o usuário saia desse campo. Se, ao contrário, a regra tivesse sido escrita:

```
MovieYear = GetMovieYear() if Insert;
```

Ou seja, sem precisar de nenhuma outra coisa além do modo para ser disparada, então assim que for aberta a transação em Insert, já será disparada a regra, e veremos o cursor posicionado sobre o campo MovieName vazio, mas com um valor já carregado em MovieYear.

- Se, em vez disso, a transação é executada sem regra parm (como seria o caso da transação paralela MovieCopy1 com a qual finalizamos o exercício anterior), para saber o modo,

deverá esperar que o usuário saia do campo MovieId. Mas ainda não poderá ser disparada, pois ainda não passamos pelo campo MovieName, que é necessário já que a regra depende desse valor. Se a regra tivesse sido escrita como:

```
MovieYear = GetMovieYear() if Insert;
```

Então, assim que o usuário sai de MovieId, a transação já disparará a regra.

E se nessa transação tivermos escritas nesta ordem as seguintes regras, quando será disparada cada uma?

```
MovieYear = GetMovieYear(MovieName) if Insert;  
  
error("The name is empty")if MovieName.IsEmpty();
```

Para qualquer uma das duas, é necessário que o usuário tenha saído do campo MovieName. Então serão avaliadas e disparadas na ordem em que estão definidos. Obviamente, a segunda será disparada tanto em modo Insert como nos demais.

Se agora adicionarmos a regra Msg seguinte, antes das outras duas, quando será disparada?

```
msg("The movie belongs to the classic film era")  
    if MovieYear < 1950;  
  
MovieYear = GetMovieYear(MovieName)  
    if Insert;  
  
error("The name is empty")  
    if MovieName.IsEmpty();
```

Observe que, por depender do atributo MovieYear, será após, não apenas do disparo da segunda regra, aquela que atribui valor a MovieYear, mas também será após sair do campo MovieYear, pois o usuário pode querer modificar o valor que lhe foi atribuído na segunda regra. Aqui é onde fica clara a importância da ordem das regras de acordo com a árvore de dependências, mas também o lugar dos controles na tela e o fluxo de interações do usuário.

Pense no seguinte fluxo:

1. Entra em modo Insert
2. Escreve algo no campo MovieName ou deixa-o vazio e sai dele. Será avaliada a regra error e também será disparada a regra de atribuição, então você verá em tela para MovieYear o valor retornado pela proc, mesmo que ainda não tenha passado por esse campo.
3. Insere uma foto em MoviePoster.
4. Passa para o campo MovieYear. Pode deixar o valor que a regra atribuiu ou modificá-lo. De qualquer forma, somente quando sair do campo será entendido que deve ser avaliada a condição da regra msg.

5. Decide retornar ao campo MovieName para alterar o nome do filme. O que será disparado novamente quando sair do campo?

Observe a listagem de navegação da transação. Informa a ordem de execução das regras? Por padrão, não faz isso. Se quiser ver esse relatório detalhado, terá que ir a Tools > Options e na opção Build, alterar para True o valor da propriedade Detailed Navigation.

Agora obtenha novamente a listagem de navegação. Aqui mostramos o da trn. MovieCopy1 (apenas com as regras anteriores; todas as demais comentadas):

```

Level Movie
READ Movie
WHERE
  Movie.MovieId = MovieId
INTO MovieYear MovieName MoviePoster MoviePoster.Uri MovieDescription CountryId CategoryId FilmDirectorId

READ FilmDirector
WHERE
  FilmDirector.FilmDirectorId = Movie.FilmDirectorId
INTO FilmDirectorName

READ Category
WHERE
  Category.CategoryId = Movie.CategoryId
INTO CategoryName

READ Country
WHERE
  Country.CountryId = Movie.CountryId
INTO CountryName

READ Movie UNIQUE
WHERE
  Movie.MovieName = MovieName

MovieYear = GetMovieYear ( MovieName ) IF insert;
Error( "The name is empty" ) IF MovieName.isempty()
Msg( "The movie belongs to the classic film era" ) IF MovieYear < 1950

INSERT INTO Movie (MovieYear, MovieName, MoviePoster, MoviePoster.Uri, MovieDescription, CountryId, CategoryId, FilmDirectorId)
UPDATE Movie (MovieYear, MovieName, MoviePoster, MoviePoster.Uri, MovieDescription, CountryId, CategoryId, FilmDirectorId)
DELETE FROM Movie

Referential integrity controls on delete:
  • Schedule ( MovieId )
  
```

Por último, o que acontecerá se a alguma das regras anteriores adicionar um evento de disparo? Por exemplo:

```

msg("The movie belongs to the classic film era")
  if MovieYear < 1950;

MovieYear = GetMovieYear(MovieName)
  on BeforeInsert;

error("The name is empty")
  if MovieName.IsEmpty();
  
```

A partir daqui, a regra de atribuição torna-se independente da árvore de dependências que vai sendo disparada à medida que se vá passando pelos campos (tanto interativamente no browser, quanto no Server, antes de ser validado o cabeçalho e inseri-lo). A regra será disparada apenas no Server, e será muito depois que no server se vá passando por cada campo e disparando suas regras. Será, de fato, quando tiver terminado de passar por todos os campos do cabeçalho, e tiver sido validado tudo relacionado a esse cabeçalho (a integridade referencial, que os valores estejam dentro dos intervalos permitidos, etc.) e um instante antes de inserir esse cabeçalho na tabela Movie.

Observe agora a listagem de navegação da transação e compare com a anterior.

```

Level Movie
READ Movie
  WHERE
    Movie.MovieId = MovieId
  INTO MovieYear MovieName MoviePoster MoviePoster.Uri MovieDescription CountryId CategoryId FilmDirectorId
READ FilmDirector
  WHERE
    FilmDirector.FilmDirectorId = Movie.FilmDirectorId
  INTO FilmDirectorName
READ Category
  WHERE
    Category.CategoryId = Movie.CategoryId
  INTO CategoryName
READ Country
  WHERE
    Country.CountryId = Movie.CountryId
  INTO CountryName
READ Movie UNIQUE
  WHERE
    Movie.MovieName = MovieName
Error( "The name is empty" ) IF MovieName.isempty()
Msg( "The movie belongs to the classic film era" ) IF MovieYear < 1950
Before Insert Rules
  MovieYear = GetMovieYear ( MovieName);
INSERT INTO Movie (MovieYear, MovieName, MoviePoster, MoviePoster.Uri, MovieDescription, CountryId, CategoryId, FilmDirectorId)
UPDATE Movie (MovieYear, MovieName, MoviePoster, MoviePoster.Uri, MovieDescription, CountryId, CategoryId, FilmDirectorId)
DELETE FROM Movie
Referential integrity controls on delete:
  • Schedule ( MovieId )
Level MovieActor

```

#### NOVO REQUISITO:

Nos pedem para controlar que ao adicionar um filme pelo menos sejam registrados dois atores, caso contrário, a aplicação não deverá nos permitir salvar o filme.

## Movie

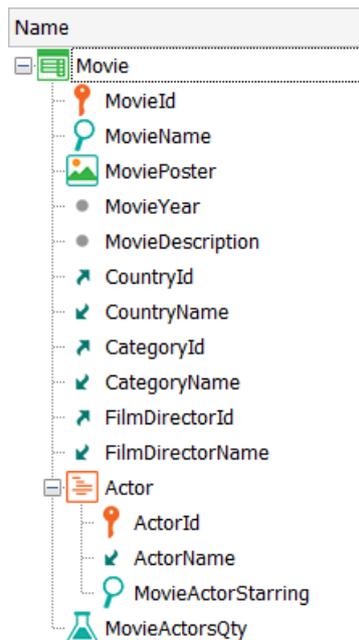
**!** Select at least two actors

Id	1
Name	<input type="text" value="Avatar 2: The way of water"/>
Year	<input type="text" value="2022"/>
Poster	
Description	Jake Sully and Ney'tiri have formed a family and are doing everything to stay together. However, they must leave their home and explore the regions of Pandora. When an ancient threat resurfaces, Jake must fight a difficult war against the humans.

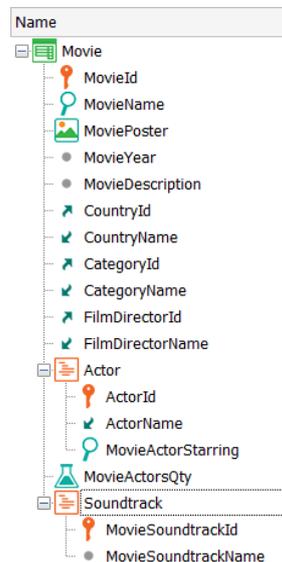
Como condiciona a regra do erro?

```
Error("Select at least two actors")
  If <condition>
    on <Event>;
```

A estrutura da transação ficou assim para você agora?



Existe alguma diferença neste caso entre escolher o evento “AfterLevel Level MovieActorStarring” e o “BeforeComplete”? E se houvesse outro nível paralelo a Actor e posterior a ele, por exemplo, Soundtracks, haveria diferença?



#### NOVO REQUISITO:

Às vezes são realizados festivais de cinema, para os quais as diferentes produtoras enviam emprestadas cópias de seus filmes para serem exibidos, e não é raro que uma produtora tome a decisão de enviar o filme de última hora.

Por este motivo, queremos dar a oportunidade ao usuário de inserir os dados de um filme: cabeçalho e linhas e por último, antes do Commit, somente ali verificar se foi recebida a permissão com a cópia para exibição ou não. Se não foi recebida, então não deve ser inserido o filme no sistema.

Temos na KB o procedimento MovieInStock que o que faz é verificar se chegou ao Cinema essa permissão. Não importa seu código (o que fornecemos na KB produz um verdadeiro ou falso aleatório, apenas para ser possível testar).

E o invocamos nas regras da transação Movie:

```
&IsInStock = MovieInStock(MovieId) If Insert;
```

Fazemos isto supondo que no procedimento será acessada a tabela Movie para ser possível recuperar o MovieName, por exemplo. Ou seja, precisamos que o registro do cabeçalho, pelo menos, esteja inserido.

Mas, na verdade, tenha em mente que é preferível que já tenha sido realizado tudo o que for possível, para dar mais tempo. Ou seja, queremos atrasar a verificação até o último momento.

Qual evento de disparo deveríamos adicionar a ele? E o que mais devemos fazer para cumprir o requisito de que o filme não seja gravado se não chegaram a permissão e a cópia?

Para saber o que retornou para você a proc (já que foi programada como aleatório) você pode modificá-la para que retorne False em um caso e True em outro e assim você pode controlar o comportamento.

#### OPCIONAL (CONTROLES SOBRE LINHAS EM MODO UPDATE)

Agora mudamos um pouco as coisas, e adicionamos um atributo MovieStatus de um domínio enumerado de mesmo nome (veio criado no xpz), que permite dois valores Pending e Completed. Desta forma permitiremos ir inserindo um filme em mais de uma instância, e só consideramos bom quando o usuário passa para Completed o valor desse campo.

Deixaremos a confirmação de que estão corretos os dados de “starring” dos atores para quando o usuário considerar concluída a inserção dos dados, marcando para MovieStatus o valor Completed.

A estrutura da transação, então, ficará:

Name	Type
Movie	Movie
MovieId	Id
MovieName	Name
MoviePoster	Image
MovieYear	Numeric(4.0)
MovieDescription	LongVarChar(1M)
CountryId	Id
CountryName	Name
CategoryId	Id
CategoryName	Name
FilmDirectorId	Id
FilmDirectorName	Name
MovieStatus	MovieStatus
Actor	Actor
ActorId	Id
ActorName	Name
MovieActorStarring	Boolean
MovieActorsQty	Numeric(4.0)

Remove, é claro, a regra que atribuía valor ao atributo MovieActorStarring e as que você não deseja que sejam disparadas para testar este novo requisito -por exemplo, as mensagens-. Você deverá adicionar as novas regras (definindo para &ok o tipo Boolean):

```
Default(MovieStatus, MovieStatus.Pending);

&ok = CheckMovieActors(MovieId)
      if MovieStatus = MovieStatus.Completed
```

```

    on <event>;

    Error("Problem with starrings")
    if not &ok and MovieStatus = MovieStatus.Completed
    on <event>;

```

Observe que o procedimento CheckMovieActors vem dentro do xpz que importou no início. Qual evento escolheu?

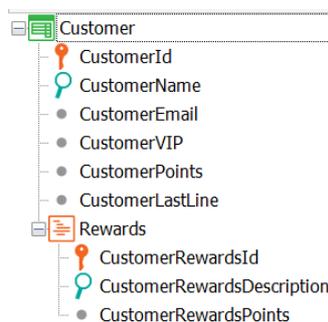
Antes de testar, certifique-se que foi adicionado ao Web Layout o atributo MovieStatus. Se não o fez, vá para o menu principal e escolha Edit > Apply Default (Web Layout).

Teste (você terá que reorganizar) inserindo novamente o filme “Gone with the Wind”. Lembre-se que é o Data Provider GetStarrings que determina quais são e quais não são atores protagonistas. Jogue com esses dados, deixando primeiro o filme em status Pending, mas com algum dado de starring errado. Você verá que permite gravar. Em seguida, entre em modo Update, altere o status para Completed e você verá exibida a mensagem de erro sobre o campo MovieStatus. Modifique os erros nos starrings dos atores até permitir que você grave com sucesso.

---

#### OPCIONAL: EXERCÍCIO MENTAL SOBRE ÁRVORE DE AVALIAÇÃO E EVENTOS

Levando em conta a seguinte estrutura para a transação Customer:



A seguir é exibida uma lista das regras programadas na referida transação.

Coloque entre os parênteses a ordem correta de execução (não precisa testar):

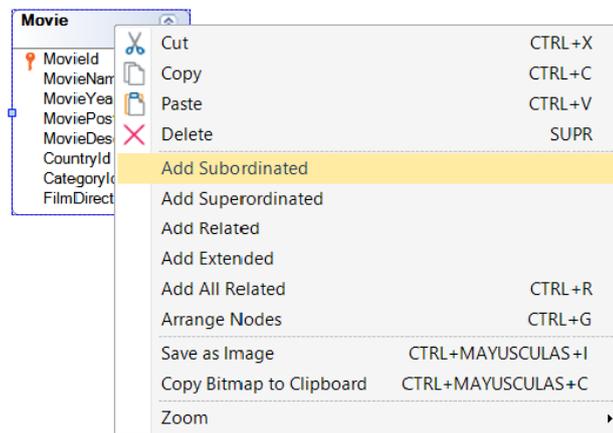
```

( )error('The name is empty') if CustomerName.IsEmpty();
( )msg('The email is empty') if CustomerEmail.IsEmpty();
( )PrintCustomerList() on afterComplete;
( )Subtract(CustomerRewardsPoints, CustomerPoints);
( )error("Not enough points") if CustomerPoints < 0;
( )PrintRewards(CustomerId, CustomerRewardsId) on AfterInsert;
( )serial(CustomerRewardsId, CustomerLastLine, 1);
( )noaccept(CustomerRewardsId);
( )Checkpoints(CustomerId) if update on AfterInsert;

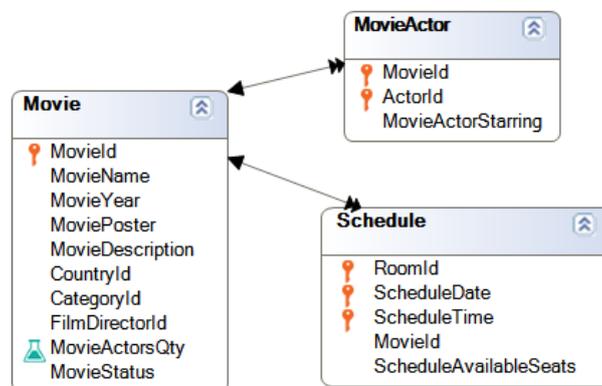
```

## EVENTOS DE DISPARO DE REGRAS EM TRANSAÇÕES (FINAL)

Crie um novo objeto diagrama e arraste ali dentro a tabela Movie. Em seguida, posicionado sobre a tabela clique com o Botão direito do mouse > Add Subordinated, para obter todas as tabelas que têm chaves estrangeiras referenciando Movie.



Serão apenas 2:



Uma é a óbvia, aquela que corresponde ao segundo nível da transação. A única outra que temos é a da transação Schedule.

Portanto, o que acontecerá se você tentar, a partir da transação Movie, excluir um filme para o qual existem registros em Schedule relacionados?

Tente. Controlará a integridade referencial e não permitirá excluir o filme. Mas... em que momento exato da execução da transação Movie será realizado este controle?

Se quisermos mudar este comportamento de forma que se houver “exibições” relacionadas (registros de Schedule) elas sejam excluídas imediatamente antes de que o controle de integridade

referencial seja realizado na transação Movie e, portanto, não falhe e, conseqüentemente, nos permita remover o filme, como conseguimos isso?

Observe que tem o procedimento DeleteSchedules já implementado, que exclui todos os registros de Schedule do MovieId recebido por parâmetro.

Como dica, deverá invocá-lo a partir das regras da transação Movie, condicionado ao modo, mas... em qual evento de disparo?

```
DeleteSchedules(MovieId)
  If Delete
  on .....;
```

Lembre-se: deve ser imediatamente antes da realização do controle de integridade referencial relativo à exclusão do registro do cabeçalho da tabela Movie. Qual é esse momento? Escolha-o entre os seguintes:

1. BeforeValidate
2. AfterValidate
3. BeforeDelete
4. AfterDelete
5. BeforeComplete
6. AfterComplete

E teste em execução excluir um filme que tenha schedules associados (sem tickets).

Se houver tickets para o Schedule que está querendo excluir dentro do procedimento, terá uma exceção de base de dados. O motivo você entenderá mais adiante. E se tentar a exclusão através do Business component Schedule (terá que ativar a propriedade Business Component da transação para poder utilizar variável do tipo Schedule) no procedimento DeleteSchedules? Não permitirá excluir, mas pelo menos não travará o programa.

```
parm(in:MovieId);

For each Schedule
  &schedule.Load(RoomId, ScheduleDate, ScheduleTime)
  &schedule.Delete()

Endfor
```

Exclua os tickets do filme e teste novamente.

## PERMITIR VALORES NULOS EM PARTE DE UMA CHAVE ESTRANGEIRA COMPOSTA

Às vezes podem ser vendidos ingressos de cinema “abertos” para que o cliente assista à exibição no horário que mais lhe convém, ou seja, é possível vender um ingresso para uma determinada sala no dia de hoje mas o cliente pode chegar ao Cinema na hora que quiser.

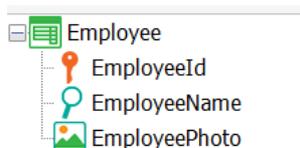
Que alteração deve ser feita na transação Ticket para que não seja levado em consideração o controle da integridade referencial na tabela Schedule entre RoomId, ScheduleDate e ScheduleTime?

Realize as alterações necessárias e tente inserir um ticket sem atribuir a ele uma sala, uma data ou um horário da exibição. Deveria permitir gravar sem problemas.

Agora reverta as alterações feitas para deixar como estava anteriormente.

## CONSEQUÊNCIAS DE ADICIONAR UM ATRIBUTO A UMA TABELA QUE JÁ TEM DADOS [SUGERIDO]

Outro requisito é adicionar um catálogo de funcionários. Adicione a seguinte transação à KB.



Execute a aplicação e adicione alguns registros para o funcionário.

Solicitam-nos a seguinte alteração: é requerido que para cada sala haja um funcionário responsável e cada funcionário pode ser responsável por uma ou mais salas. **Como modelaria isto em GeneXus levando em consideração as transações Employee e Room?**

Replique em GeneXus o que acabou de analisar.

A análise de impacto nos indica o seguinte. **O que acontecerá na tabela de Empregado se pressionarmos Reorganize?**

---

---

---

**Database needs to be reorganized.**

This report describes Database changes and how they will be handled by reorganization programs. Please select Reorganize to proceed or Cancel.

Reorganize Cancel

Pattern:

Room

**Table Room specification**

Table name: [Room](#)

Room needs conversion

Warnings

- rgz0007 Attribute [EmployeeId](#) does not allow nulls and does not have an Initial Value. An empty default value will be used.
- rgo0002 Table ['Employee'](#) is referenced by table ['Room'](#) and at least one attribute in its foreign key ([EmployeeId](#)) does not allow nulls. A default record will be added if table ['Room'](#) has records and a record with default values does not exist in table ['Employee'](#).

Table Structure

Attribute	Definition	Previous values	Takes value from
<a href="#">RoomId</a>	Numeric (4), Not null, Autonumber		<a href="#">Room. RoomId</a>
<a href="#">RoomType</a>	Character (20), Not null, NLS		<a href="#">Room. RoomType</a>
<a href="#">RoomSeats</a>	Numeric (4), Not null		<a href="#">Room. RoomSeats</a>
New <a href="#">EmployeeId</a>	Numeric (4), Not null		0

Indexes

0 Errors 1 Warnings 0 Success

Se não queremos que aconteça isto que nos informam os warnings, o que podemos fazer?

---



---



---

## TRANSAÇÕES PARALELAS [SUGERIDO]

As companhias cinematográficas compartilham certas informações das pré-estreias com todos os Cinemas. É compartilhado: O nome do filme, o ano, um pôster, diretor, país e sua categoria; os dados da descrição e os atores são confidenciais até certo momento depois, portanto, no sistema, não são colocados até que o filme seja exibido no Painel publicitário.

Para registrar os dados iniciais do filme e não visualizar os demais campos, é requerida uma transação paralela onde apenas sejam registradas as informações disponíveis.

Realize esta implementação.

Lembre-se que já havíamos definido uma transação paralela antes: era MovieCopy1, paralela de Movie. Fizemos isso para ter uma sem o pattern (sem parâmetros ou tudo que adicionava o pattern a Movie).

**SUBTIPOS****SUBTIPOS: REFERÊNCIA MÚLTIPLA E ESPECIALIZAÇÃO**

São requeridas as seguintes melhorias na aplicação:

Lembremos que um filme tem um país, e que pode ter 1 ou mais atores associados a ele. É requerido adicionar, também para seus atores, a informação de seu país.

**O que devemos mudar em nossa estrutura de transações para salvar o País do Filme e também fazer referência ao País do Ator?** Pense em 3 desenhos diferentes que resolvam este requisito. Escreva para cada um suas vantagens e desvantagens (se as tiverem) e escolha aquele que considere mais adequado para implementá-lo em GeneXus.

Após a análise, realize a implementação na aplicação que estamos desenvolvendo.

Leve em consideração que, com a implementação mais conveniente, também deverá realizar alterações no DP da transação Movie. Bem como nas transações MovieCopy1 e MoviePreview.

**DESENHO 1:**

---

---

VANTAGENS	DESVANTAGENS

**DESENHO 2:**

---

---

VANTAGENS	DESVANTAGENS

DESENHO 3:

---

---

VANTAGENS	DESVANTAGENS

Os Funcionários do Cinema podem ser categorizados em:

- **Administrativos:** são responsáveis pela venda de ingressos e atendimento ao público em geral. Deste tipo de funcionários, interessa saber se falam algum idioma adicional.
- **Operadores:** são responsáveis por programar os filmes nas diferentes salas nos dias e horários em que são projetados. Estes funcionários devem ter mais de 22 anos, portanto interessa saber a sua idade.
- **Técnicos:** são responsáveis pela manutenção do Cinema no que diz respeito a consertos de equipamento, instalações, etc. Deste tipo de funcionários, interessa conhecer sua formação técnica pertinente ao cargo.

Como podemos fazer esta distinção em GeneXus? Implemente-a.

## MAIS CASOS DE USO DE SUBTIPOS

É solicitada uma transação onde possamos ver a informação dos Funcionários do cinema e seu chefe responsável. Como deve ser modelada se for levado em conta que o chefe também é um funcionário?

Na transação de Empregado, deverá ser exibido um erro se tentarmos atribuir como chefe um funcionário que não possui essa posição, e também não deve permitir salvar a informação se o chefe e o funcionário forem a mesma pessoa.

Como devem ser programadas essas regras?

Realize a implementação em GeneXus e escreva também as regras para testar o comportamento.

```
error("The employee and the Manager can't be the same")
  if _____ = _____;

error("The employee is not a Manager")
  if _____ and _____;
```

The screenshot shows a web form titled "Employee" with the following fields and controls:

- Navigation: << < > >> SELECT
- Id: Input field with value "1"
- Name: Input field with value "Jonh Smith"
- Is Manager: Check box (checked)
- Employee Id: Input field with a dropdown arrow and a tooltip that says "The employee is not a Manager"
- Employee Name: Input field with value "Billie Robin"
- Is Manager: Check box (unchecked)
- Buttons: CONFIRM, CANCEL, DELETE

## FÓRMULAS

### FÓRMULAS VERSUS REGRAS

Volte sobre o controle dos pontos que o cliente dispõe para trocar por prêmios.

Name	Type	Description	Formula	Nullable
Customer	Customer	Customer		
CustomerId	Id	Customer Id		No
CustomerName	Name	Customer Name		No
CustomerEmail	Email, GeneXus	Customer Email		No
CustomerVIP	Boolean	Customer VIP		No
CustomerPoints	Numeric(4.0)	Customer Points		No
CustomerLastRewardId	Id	Customer Last Reward Id		No
Reward	Reward	Reward		
CustomerRewardId	Id	Customer Reward Id		No
CustomerRewardDescription	Numeric(4.0)	Customer Reward Description		No
CustomerRewardPoints	Numeric(4.0)	Customer Reward Points		No

É possível resolvê-lo utilizando fórmulas globais em vez das regras add e subtract?

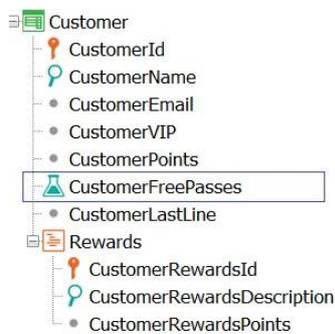
Obtemos o mesmo comportamento?

FÓRMULAS HORIZONTAIS (OPCIONAL)

Antes de continuar testando, é solicitado um novo requisito:

Os clientes podem obter passes livres (CustomerFreePasses) no mês corrente, dependendo da quantidade de tickets registrados:

- Se registra no mês mais de 2 e menos de 6 tickets, gera 1 passe livre.
- Se registra no mês mais de 5 e menos de 9 tickets, gera 2 passes livres.
- Se registra no mês mais de 8 tickets, gera 3 passes livres.



Que definição você faria para o atributo fórmula CustomerFreePasses?

Uma opção pode ser a seguinte:

```

Formula Editor
1 IF count(TicketDate, TicketDate.month()=month(today()))>2 and count(TicketDate, TicketDate.month()=month(today())) < 6;
2 IF count(TicketDate, TicketDate.month()=month(today()))>5 and count(TicketDate, TicketDate.month()=month(today())) < 9;
3 IF count(TicketDate, TicketDate.month()=month(today())) > 8;
    
```

Outra possibilidade é definir um procedimento FreePasses que retorne o valor correspondente. Implemente esta opção.

Como será sua regra Parm?

Como será a sua invocação para que o atributo CustomerFreePasses tome o valor retornado pelo procedimento?

**FÓRMULAS AGGREGATE**

Anteriormente, o preço dos ingressos era definido no momento da compra, porém, agora o preço será definido de acordo com a sala que está sendo selecionada. Para isto, é necessário guardar um histórico de preços para a sala e a data em que foi guardado o referido preço (NOTA: a data não deve poder ser repetida) e o valor que será utilizado no cálculo do ticket será o da data mais recente.

Com base no exposto, qual seria a opção correta para modelar o requisito?



Qual fórmula de agregação você usaria para retornar o valor do preço de acordo com a data mais atual?

RoomUpdatedPrice = \_\_\_\_\_

Uma vez atualizada a transação Room, **que alterações você deve fazer em Ticket? Ticket Price continua válido?**

---



---



---

Realize as alterações necessárias (na trn Room e na trn Ticket) e teste em execução.

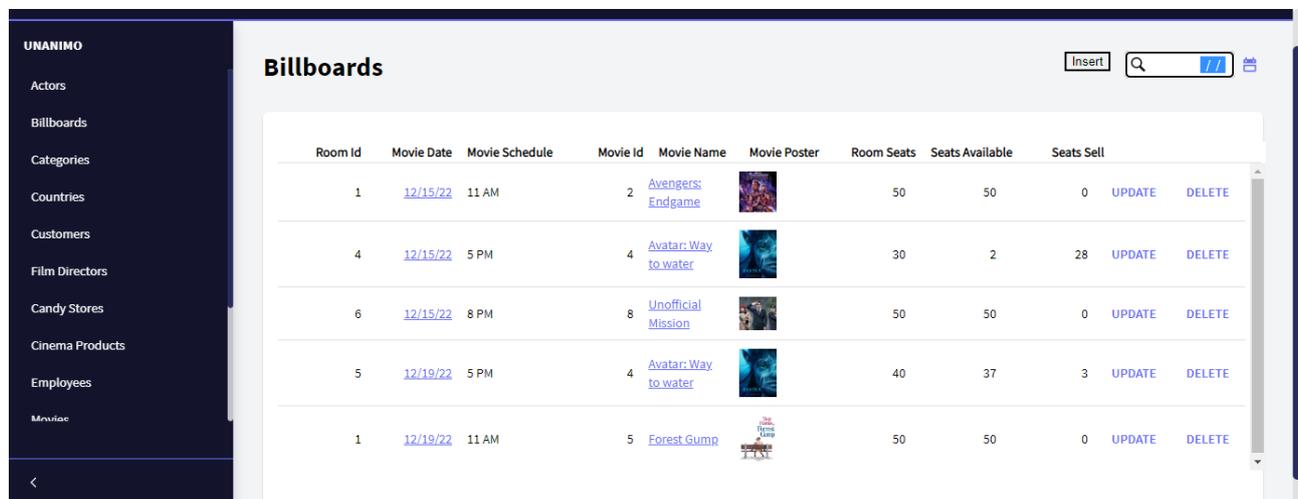
---

Dentro da transação Movie, é desejado adicionar um campo que permita saber se o filme é Premium ou não.

Caso o filme seja Premium, deverá ser adicionado no momento da compra um acréscimo de 7% de seu valor e se o filme não tem este campo especificado, então deverá assumir o preço normal.

Room Type	VIP
Movie Id	1
Movie Name	Avengers End Game
Seats Quantity	<input type="text" value="1"/>
Room Updated Price	135.00
Total Price	144.45
Points	14

Deseja-se saber para cada registro de Painel publicitário os ingressos vendidos e os assentos disponíveis.



Room Id	Movie Date	Movie Schedule	Movie Id	Movie Name	Movie Poster	Room Seats	Seats Available	Seats Sell		
1	12/15/22	11 AM	2	Avengers: Endgame		50	50	0	UPDATE	DELETE
4	12/15/22	5 PM	4	Avatar: Way to water		30	2	28	UPDATE	DELETE
6	12/15/22	8 PM	8	Unofficial Mission		50	50	0	UPDATE	DELETE
5	12/19/22	5 PM	4	Avatar: Way to water		40	37	3	UPDATE	DELETE
1	12/19/22	11 AM	5	Forest Gump		50	50	0	UPDATE	DELETE

Caso já tenham sido vendidos todos os ingressos para aquela sala, o sistema não deve permitir salvar a compra.

## TRANSAÇÕES DINÂMICAS

Outro dos requisitos do sistema de Cinemas é unir a informação e ter um catálogo dos filmes e dos produtos que são vendidos na doceria. Será salvo: o identificador do produto, seu nome, uma imagem e seu preço.

Com finalidade de conhecer todos os produtos oferecidos pelo Cinema (filmes e doces), é requerido um relatório onde são listados todos os filmes e os produtos da doceria. É necessário ver: seu identificador, seu nome, uma foto e o tipo (se é filme ou item da doceria).

**Sugestão**

Utilize uma transação dinâmica para recuperar as informações de ambos os catálogos e poder listá-las no relatório anteriormente solicitado.



### All Products

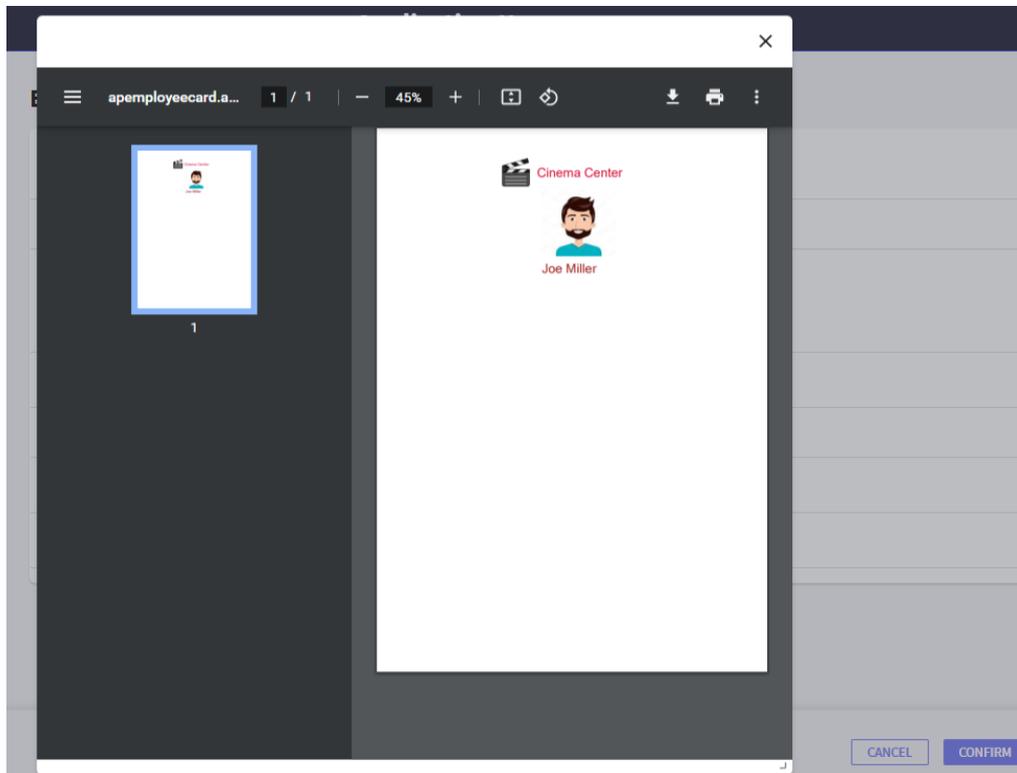
Id	Type	Product Name	Photo
1	Candy	Popconm	
1	Movie	Avengers End Game	
2	Candy	Soda	
2	Movie	Catch me if you can	
3	Candy	Gummies	
3	Movie	Forrest Gump	
4	Candy	Hot Dog	

Antes de mais nada, deverá ser criada uma transação (CandyStoreProduct) onde possa ser inserido, modificado e eliminado os doces que serão vendidos. Destes nos interessa registrar seu nome, uma foto e seu preço.

Que modificações deveria fazer na implementação proposta se o usuário final desejasse poder inserir, modificar ou excluir filmes e doces através do form de uma nova transação que será dinâmica?

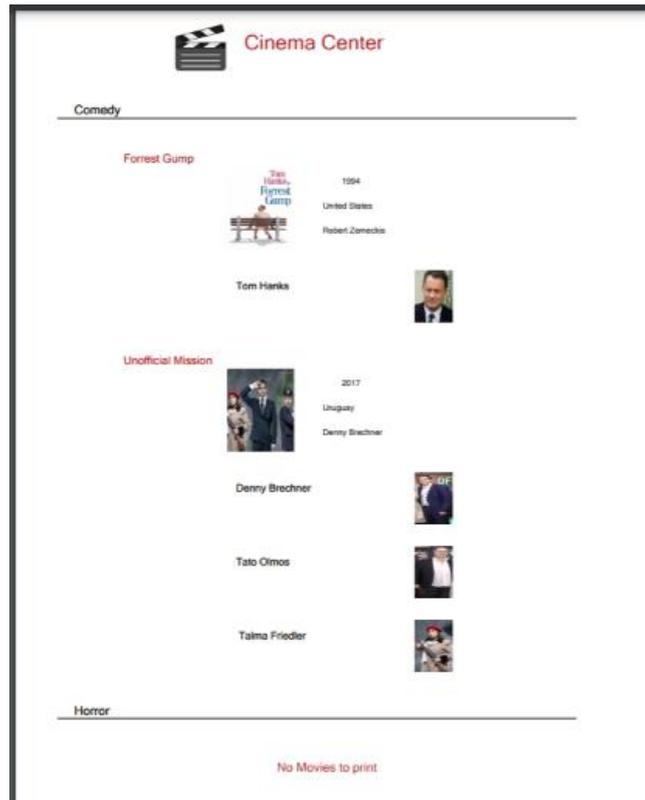
## EVENTOS EM TRANSAÇÕES

O sistema de cinemas requer que, no momento de salvar um funcionário novo, seja exibido em tela um Pop-up com suas informações no estilo de uma credencial:



## MAIS SOBRE FOR EACHS ANINHADOS. CASOS E NAVEGAÇÃO

Nos solicitam um relatório que tenha o seguinte formato. Colocar as categorias e seus filmes relacionados, bem como os atores que participam desses filmes. Caso não haja filmes associados à categoria, deverá exibir uma mensagem informando que não há informação:

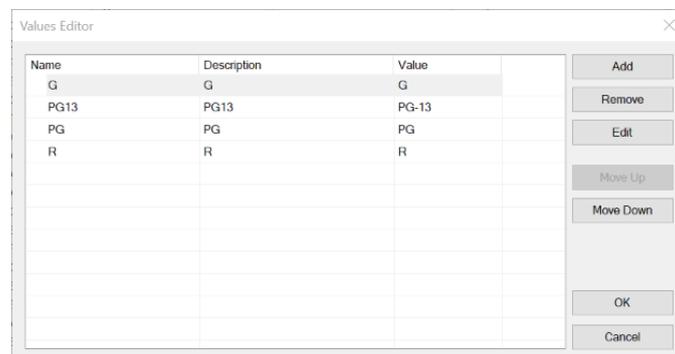


Se nos é solicitada uma listagem com o formato da anterior com a única restrição de que apenas sejam listadas as categorias que tenham filmes associados. Como você conseguiria isso? Escreva o código a seguir com as alterações que você deve fazer.

Faça um Save As do primeiro procedimento e faça as modificações adequadas para poder receber como parâmetro uma Categoria a partir de um web panel. O procedimento deve mostrar a informação da categoria selecionada e caso não seja selecionada nenhuma categoria, deve ser mostrada toda a informação das Categorias com seus Filmes e Atores correspondentes.

## SUB-ROTINAS

Na transação Movie, adicione um novo atributo chamado MovieRating, onde salvaremos a classificação etária para cada filme (G, PG, PG-13, R). Para isso, faça um domínio enumerado chamado Rating que conterá os seguintes dados:

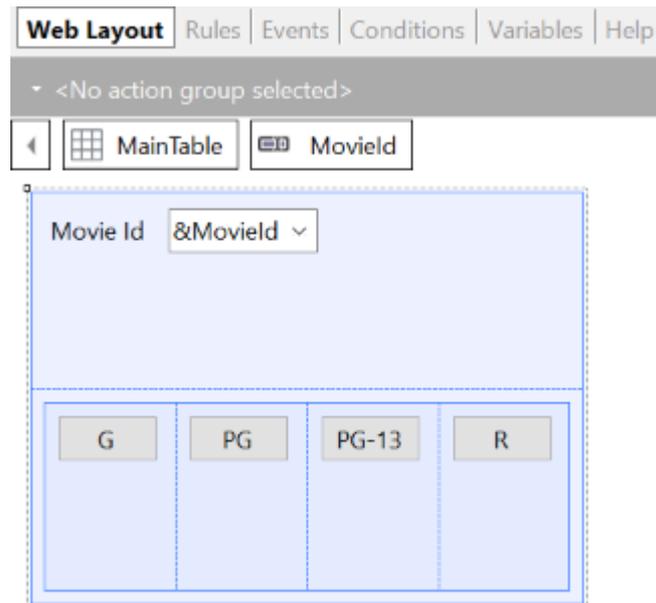


Name	Description	Value
G	G	G
PG13	PG13	PG-13
PG	PG	PG
R	R	R

### Sobre os domínios enumerados:

Nas colunas "Name" e "Description" não são permitidos nomes que contenham hifens ou outros caracteres especiais. Para colocar o nome da classificação PG-13, faça-o sem o hífen.

A seguir, crie o seguinte web panel onde selecionaremos um filme e clicando em um botão atualizaremos sua classificação de acordo com o valor selecionado.



Observe que determinado código deverá ser repetido para cada botão, portanto podemos fazer uso das sub-rotinas para usar o mesmo código, mas atualizar conforme o selecionado.

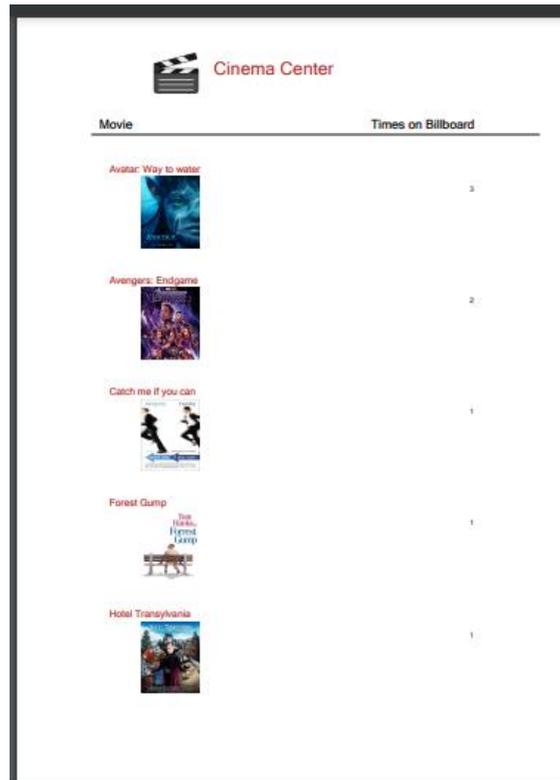
```
1 Event 'G'  
2     &Rating = Rating.G  
3     do "UpdateRating"  
4 Endevent
```

### Sugestão

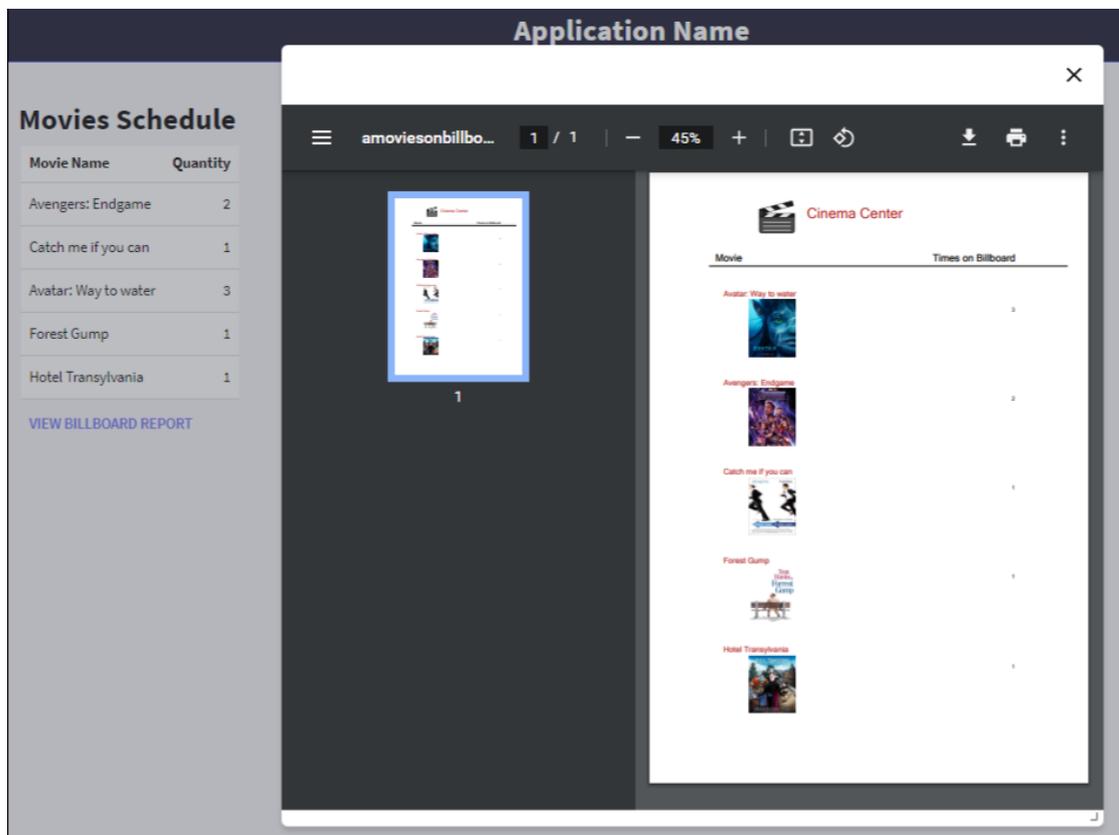
Utilize uma tabela responsiva para colocar os botões e altere a classe dos botões para uma que combine com a estética do programa, exemplo: BTN Cancel.

## CLÁUSULA UNIQUE

Outro requisito do Cinema é poder visualizar um relatório dos filmes que aparecem no painel publicitário, Schedule, sem repetições e com a quantidade de aparições que tiveram no painel publicitário.



Um novo requisito é poder visualizar as informações anteriores em um web panel. Coloque um botão onde possa consultar a lista anterior como Pop-Up.



### Sugestão

Altere as classes do Grid para WorkWith e a do Botão para Btn Cancel para dar mais estética à aplicação.

### DATA SELECTOR

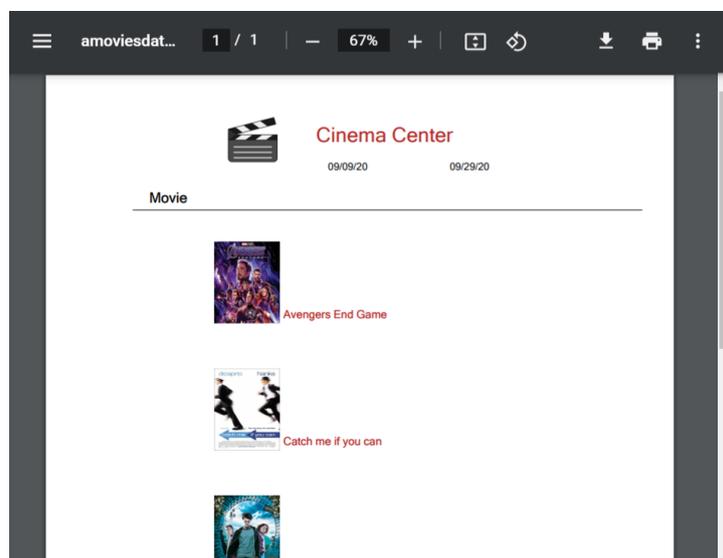
É requerida uma tela web que permita selecionar um intervalo de datas e que com um botão seja exibida como pop-up a lista de todos os Filmes que aparecem no painel publicitário, mas apenas os filmes que estão no intervalo de datas selecionado.

Qual é a navegação resultante? Explique ...

---

---

---



NOTA: Por se tratar de uma consulta que de forma recorrente será utilizada dentro do sistema de Cinemas, gere um novo objeto data selector para poder utilizá-lo posteriormente.

Na mesma tela web deseja-se gerar outra lista pop-up onde se consulte as informações dos filmes que serão exibidos, seu horário, a sala e a data, em um determinado intervalo de datas.

Qual é a diferença entre esta navegação resultante e a anterior? Explique...

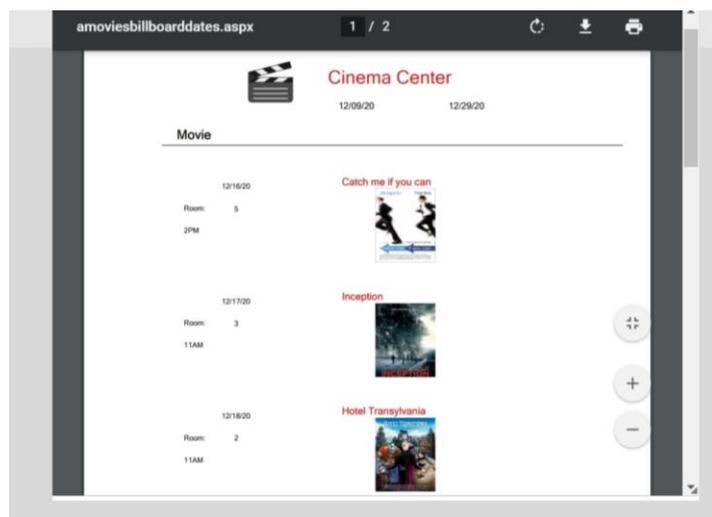
---



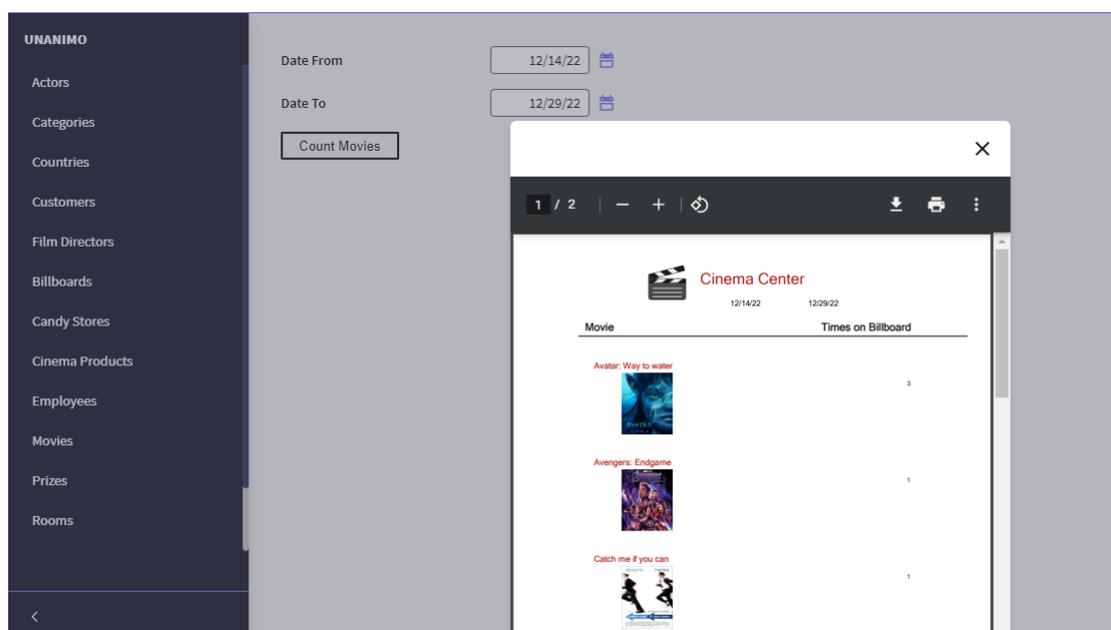
---



---



Além disso, deve permitir visualizar uma lista Pop-up com os filmes que aparecem no painel publicitário, sem repetições e contando somente a quantidade de aparições que tiveram no painel publicitário neste intervalo de datas.



Qual é a diferença entre esta navegação resultante e as anteriores? Explique...

---

---

---

## CLÁUSULAS DA PAGINAÇÃO

Na tela trabalhar com associada ao Painel publicitário me interessa apenas ver 5 registros por vez, além de que o formato de saída seja um scroll infinito. Aplique a alteração solicitada.

## PAGINAÇÃO EM FOR EACH, PATTERN E GRID (OPCIONAL)

É requerida uma tela onde o usuário possa chamar uma lista com as informações dos atores. Em particular, na tela deve selecionar quantos registros deseja pular e quantos possa visualizar depois. Exemplo: “quero pular os 3 primeiros registros e poder trabalhar com os próximos 5 para trabalhar pontualmente com esses dados.”



### List of Actors

4	Daisy Ridley	United Kingdom
5	Emma Watson	United Kingdom
6	Robert Downey Jr	United States
7	Chris Hemsworth	Australia
8	Adam Sadler	United States

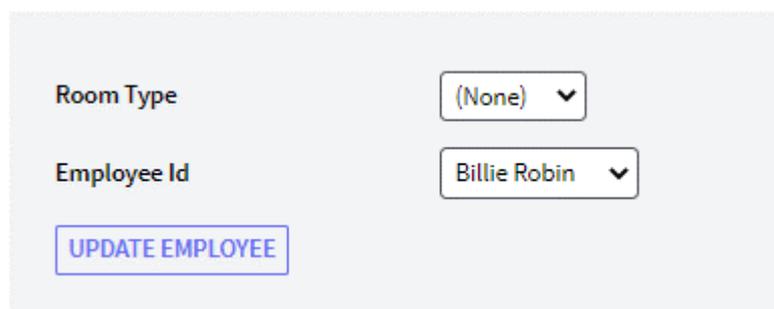


**BUSINESS COMPONENT**

Realizaremos uma série de operações na Base de Dados, por meio de Business Component.

**ATUALIZAÇÃO DE ENCARREGADOS**

Se requer de uma tela especializada que permita selecionar um tipo de sala (VIP, 3D, etc.) e selecionar um funcionário. O funcionário ficará encarregado de todas as salas que sejam desse tipo.



Room Type (None) ▾

Employee Id Billie Robin ▾

UPDATE EMPLOYEE

**TRABALHANDO COM OS FILMES E SEUS ATORES**

É solicitada uma tela com formato especial que nos permita selecionar um filme para poder trabalhar com seus atores. Serão oferecidos dois botões:

- Um que nos permita modificar o papel do ator para estrear ou remover o papel estrear.
- Outro que permita eliminar o ator do filme selecionado.

**Sugestão**

Utilize a variável &Messages de GeneXus para programar as mensagens do sistema em caso de que a atualização falhe ou tenha sucesso.

O que poderia fazer para que na variável associada ao ator somente sejam vistos os atores que participam do filme selecionado?

---

---

---

**Application Name**

▲ Data has been successfully updated.

Movie Id

Actor Id  Is Starring

## ATUALIZAÇÃO DA BASE DE DADOS COM PROCEDIMENTO

Na mesma tela onde foi feita a atualização dos funcionários encarregados das salas de acordo com seu tipo, implemente um botão que efetue a mesma atualização citada anteriormente, mas agora usando um procedimento.

Como poderia apagar os filmes e seus atores associados através de um procedimento?

Insira um filme novo (o que deseje) com seus respectivos atores.

Programe um botão na tela que criou anteriormente para trabalhar com os filmes e todos os seus atores, esse botão deve excluir um filme selecionado e remover os atores associados. Exclua o Filme que acabou de criar.

O que acontece quando tenta excluir Filmes que têm relação com outras tabelas?

---

---

---

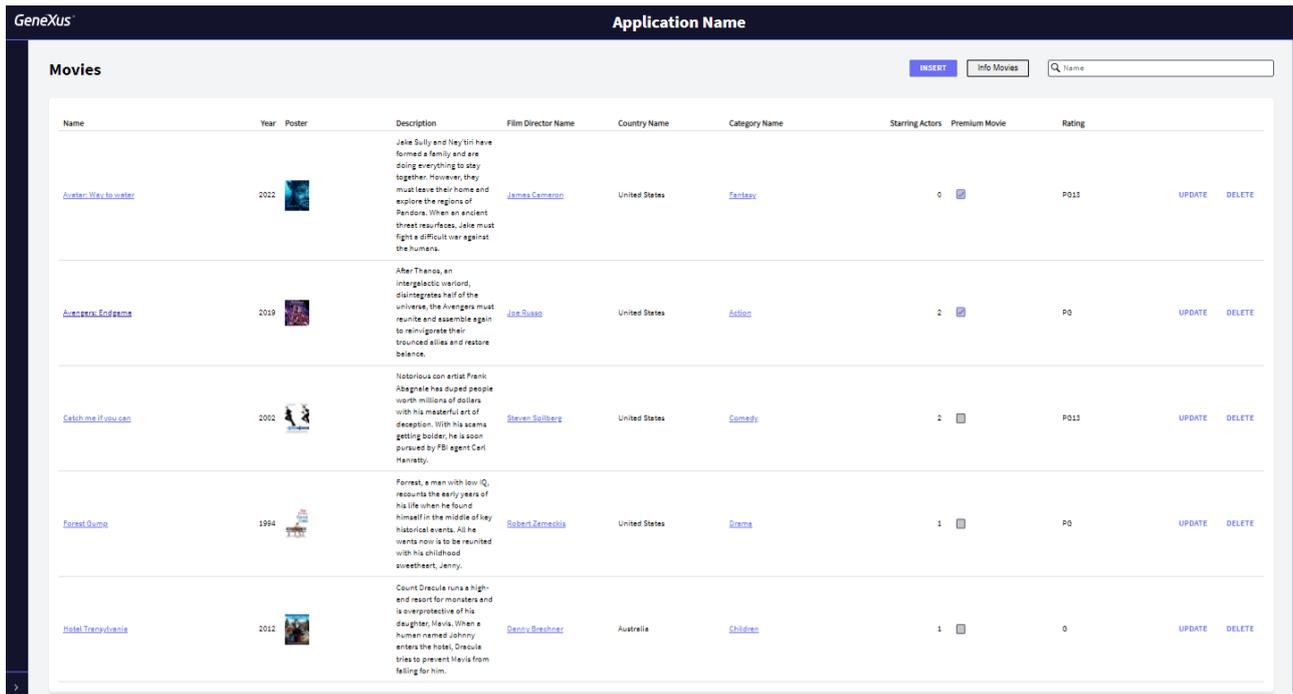
## TELAS WEB: WEB PANELS

A seguir, começaremos a gerar telas interativas personalizadas que nos foram solicitadas para o sistema de Cinemas.

## VARIÁVEIS EM GRID E AÇÕES NO PATTERN WORK WITH (OPCIONAL)

Na tela que nos permite trabalhar com os filmes, nos solicitam as seguintes alterações:

- Não devem ser visualizados os Id
- Um campo calculado que nos mostre quantos atores protagonistas tem o filme.
- Um botão que nos mostre a lista de informações dos filmes anteriormente criada onde se visualiza a Categoria, o filme e os atores que o compõem.



Name	Year	Poster	Description	Film Director Name	Country Name	Category Name	Starring Actors	Premium Movie	Rating	
<a href="#">Avatar: Way to water</a>	2012		Jelie Sully and Neytiri have formed a family and are doing everything to stay together. However they must leave their home and explore the regions of Pandora. When an ancient threat resurfaces, she must fight a difficult war against the humans.	<a href="#">James Cameron</a>	United States	<a href="#">Fantasy</a>	0	<input checked="" type="checkbox"/>	PG13	<a href="#">UPDATE</a> <a href="#">DELETE</a>
<a href="#">Avengers: Endgame</a>	2019		After Thanos, an intergalactic overlord, disintegrates half of the universe, the Avengers must reunite and assemble again to reinvigorate their trounced allies and restore balance.	<a href="#">Joe Russo</a>	United States	<a href="#">Action</a>	2	<input checked="" type="checkbox"/>	PG	<a href="#">UPDATE</a> <a href="#">DELETE</a>
<a href="#">Catch me if you can</a>	2002		Notorious con artist Frank Abagnale has duped people worth millions of dollars with his masterful art of deception. With his seams getting tighter, he is soon pursued by FBI agent Carl Hanratty.	<a href="#">Steven Spielberg</a>	United States	<a href="#">Comedy</a>	2	<input type="checkbox"/>	PG13	<a href="#">UPDATE</a> <a href="#">DELETE</a>
<a href="#">Forest Gump</a>	1994		Forrest, a man with low IQ, recounts the early years of his life when he found himself in the middle of key historical events, all he wants now is to be reunited with his childhood sweetheart, Jenny.	<a href="#">Robert Zemeckis</a>	United States	<a href="#">Drama</a>	1	<input type="checkbox"/>	PG	<a href="#">UPDATE</a> <a href="#">DELETE</a>
<a href="#">Hotel Transylvania</a>	2012		Count Dracula runs a high-end resort for monsters and is overprotective of his daughter, Mavis. When a human named Johnny enters the hotel, Dracula tries to prevent Mavis from falling for him.	<a href="#">Genndy Tartakovsky</a>	Australia	<a href="#">Children</a>	1	<input type="checkbox"/>	G	<a href="#">UPDATE</a> <a href="#">DELETE</a>

## CRIANDO TELAS INTERATIVAS

- É solicitada uma tela onde se visualize as informações do Painel publicitário ordenado pela data de exibição. Deve ser permitido filtrar por:
  - A data, que seja maior ou igual ao dia de hoje. (O sistema deve sugerir automaticamente tomar a data de hoje.
  - Por horário.
  - Por tipo de Sala.
  - Por nome do filme.

Se os filtros estiverem vazios, deverá ser vista todas as informações do painel publicitário.

Para cada registro é requerido um botão que, ao ser clicado, nos mostre o detalhe do filme selecionado. Na tela do detalhe, implemente um botão que nos permita retornar ao web panel anterior.

GeneXus™ **Application Name**

Room Movie Date  

Room Movie Schedule  ▼

Room Type  ▼

Movie Name

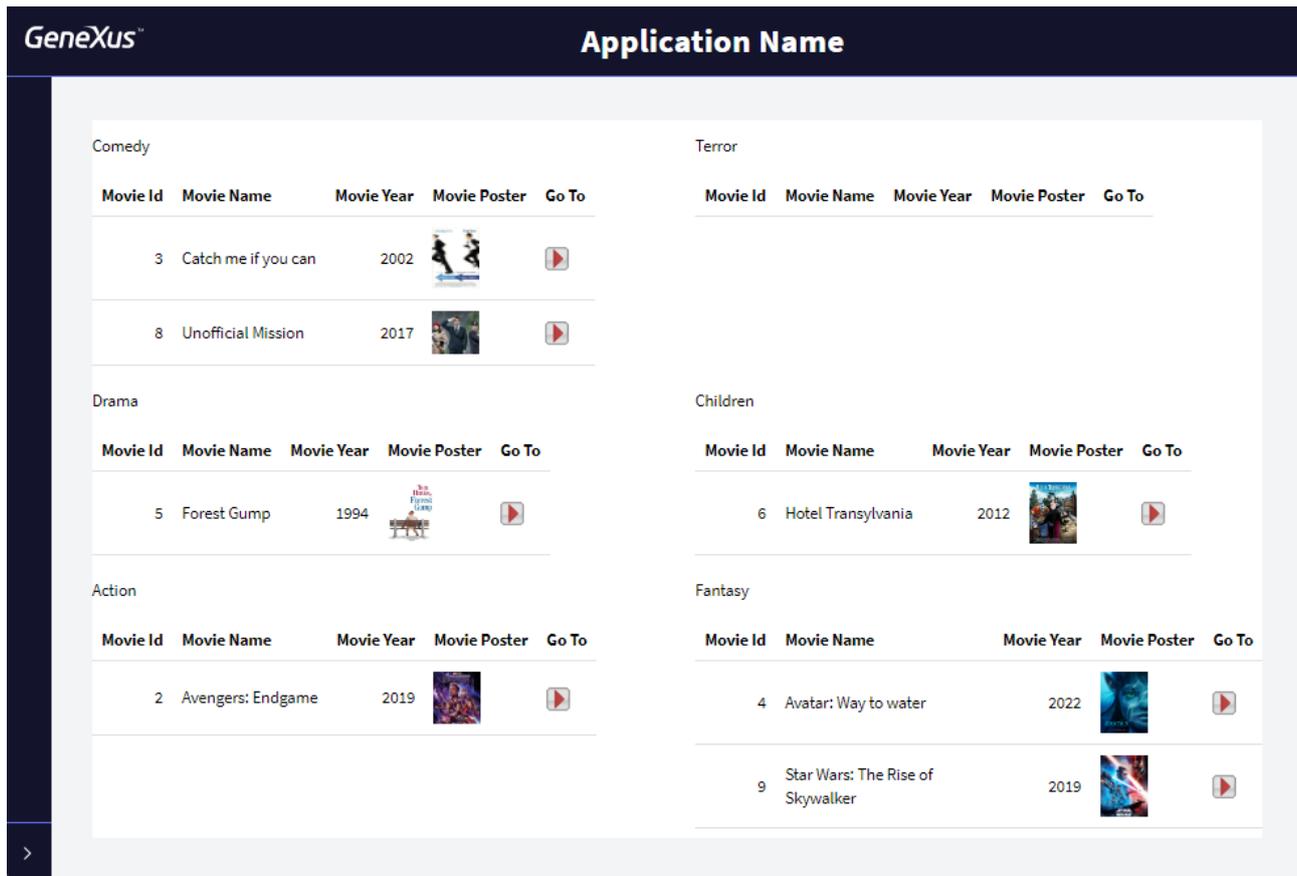
Room Type	Room Movie Date	Room Movie Schedule	Movie Name	Movie Poster	Goto
VIP	12/15/22	5 PM	Avatar: Way to water		
VIP	12/19/22	5 PM	Avatar: Way to water		
Normal	12/21/22	5 PM	Hotel Transylvania		

## MÚLTIPLOS GRIDS

É solicitado um Web Panel que permita ver todas as Categorias e para cada uma delas seus filmes associados.

Para cada filme, é requerido um botão que, ao ser clicado, nos mostre o detalhe do filme selecionado. Na tela do detalhe, implemente um botão que nos permita retornar ao web panel anterior.

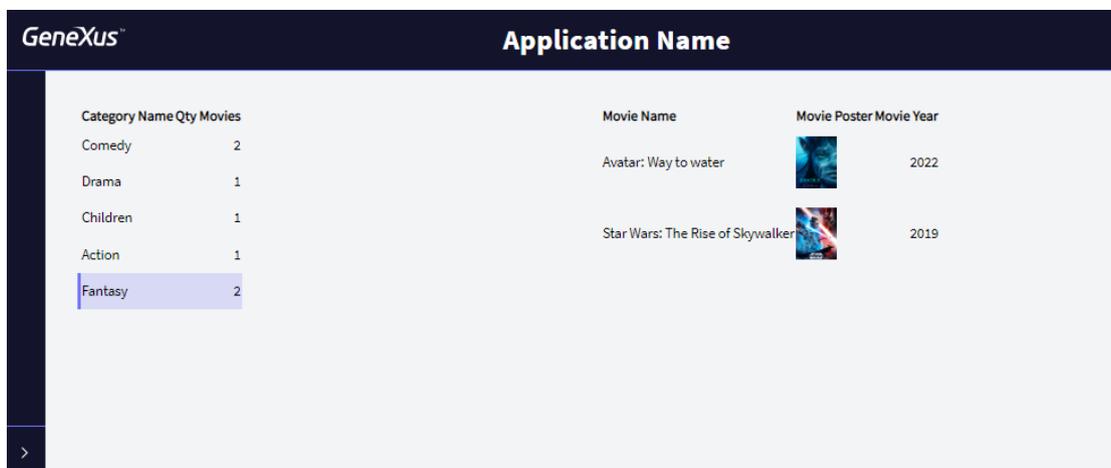
Nota: Pode ser reutilizado o web panel do detalhe do filme.



Como poderia garantir que sejam visualizadas apenas as categorias que possuem filmes associados?

Implemente essa mudança.

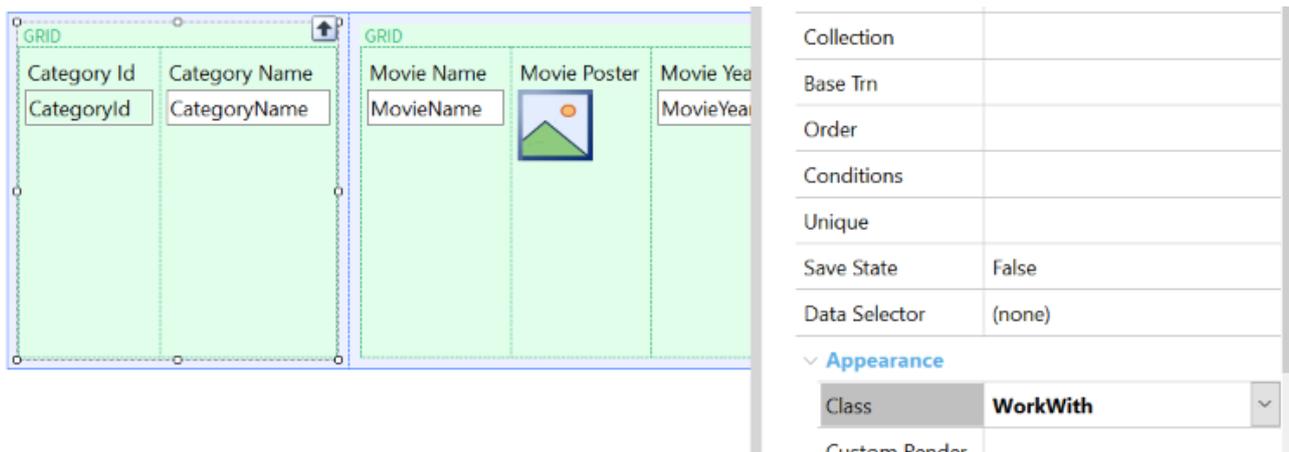
É requerida uma tela onde se mostre a lista das categorias que possuem filmes associados (sem repetições) e sua quantidade de filmes, por sua vez, será possível selecionar uma categoria e a seguir serão exibidos os filmes associados na mesma tela.



Qual deve ser a tabela base do Grid que mostra as informações das Categorias?

---

Nota: Recomenda-se que seja aplicada a classe WorkWith ao Grid para melhorar a estética



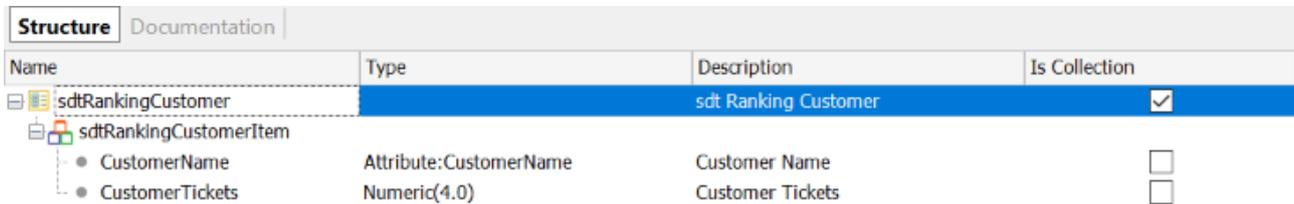
### GRID SEM TABELA BASE

É requerida uma tela especial que nos mostre para cada data os filmes a serem apresentados. O formato é o seguinte:



## CARREGANDO COLEÇÕES

É requerido um ranking de Clientes de acordo com os assentos de cada ticket que tenham comprado. Para resolver o anterior, utilize uma estrutura.



The screenshot shows the 'Structure' window in GeneXus. It displays a tree view on the left and a table on the right. The tree view shows a folder 'sdtRankingCustomer' containing a folder 'sdtRankingCustomerItem' which has two sub-items: 'CustomerName' and 'CustomerTickets'. The table on the right has the following columns: Name, Type, Description, and Is Collection. The 'sdtRankingCustomer' row is highlighted in blue and has a checked checkbox in the 'Is Collection' column. The 'CustomerName' and 'CustomerTickets' rows have unchecked checkboxes.

Name	Type	Description	Is Collection
sdtRankingCustomer		sdt Ranking Customer	<input checked="" type="checkbox"/>
sdtRankingCustomerItem			
CustomerName	Attribute:CustomerName	Customer Name	<input type="checkbox"/>
CustomerTickets	Numeric(4.0)	Customer Tickets	<input type="checkbox"/>

Além do anterior, a tela deve permitir que ao selecionar um cliente mostre as informações em formato PDF do cliente com sua respectiva quantidade de tickets.

### Sugestão

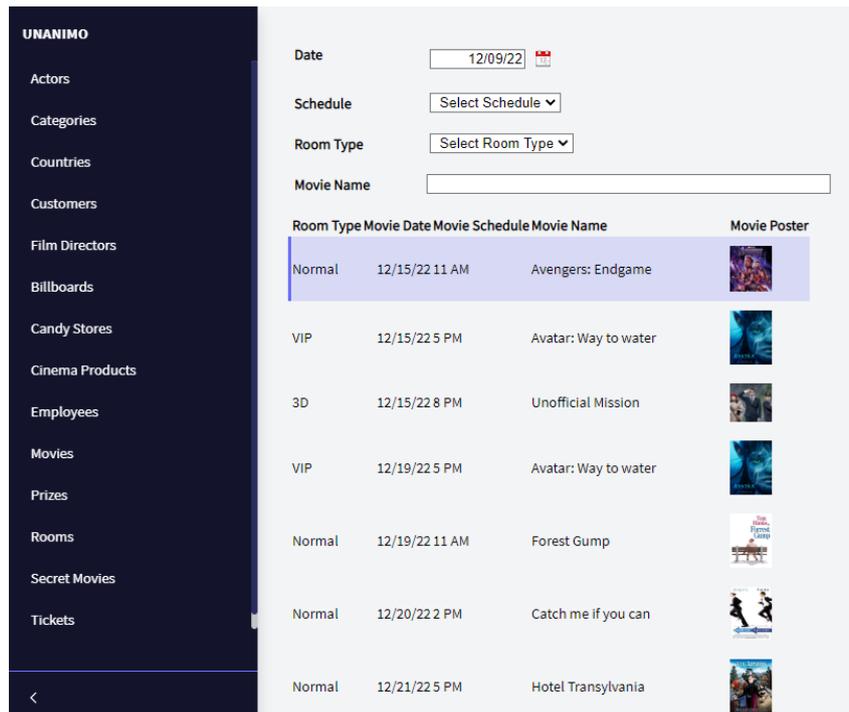
Utilize a propriedade AllowSellection do Grid e o evento OnLineActivate.

## SALVAR INFORMAÇÕES DE CONTEXTO (OPCIONAL)

Abra o web panel que contém as informações completas do Painel publicitário. É requerido que no momento de retornar à referida tela após a visualização do detalhe do filme, sejam salvos os valores dos filtros.

## TELA ESPECIAL PARA A VENDA DE TICKETS

Faça uma cópia da tela anterior e modifique as classes do web panel para que tenha uma estética semelhante à seguinte:



Room Type	Movie Date	Movie Schedule	Movie Name	Movie Poster
Normal	12/15/22 11 AM		Avengers: Endgame	
VIP	12/15/22 5 PM		Avatar: Way to water	
3D	12/15/22 8 PM		Unofficial Mission	
VIP	12/19/22 5 PM		Avatar: Way to water	
Normal	12/19/22 11 AM		Forest Gump	
Normal	12/20/22 2 PM		Catch me if you can	
Normal	12/21/22 5 PM		Hotel Transylvania	

## Sugestão

Utilize a classe *WorkWith* do Grid e as classes *WWColumn* e *WWOptionalColumn*.

Ao selecionar uma linha do Painel publicitário, deve ser exibido um pop-up que nos permita ver os dados do filme selecionado. O Funcionário poderá selecionar um cliente ou não, e o número de assentos a comprar; ao clicar no botão Buy, a compra deverá ser realizada. Nota: há ocasiões em que não poderá ser realizada a compra devido a algum controle que programamos, portanto, deverão ser visualizadas as mensagens do sistema na tela:

GeneXus Application Name

Date: 12/15/22  
Schedule: 5 PM  
Room Type: Select Room Type  
Movie Name:

Room Type	Movie	Date	Movie Schedule	Movie Name	Movie Poster
VIP	Avatar: Way to water	12/15/22 5 PM			
VIP	Avatar: Way to water	12/19/22 5 PM			
Normal	Hotel Transylvania	12/21/22 5 PM			
VIP	Avatar: Way to water	12/31/22 5 PM			

### Buy Tickets

⚠ Not enough seats

Room	Date	Schedule	Movie Name	Seats Available
4	12/15/22	5 PM	Avatar: Way to water	2

Customer Id: Anne Parker

Ticket Seats Quantity: 1, 2, 3, 4, 5, 6

BUY

GeneXus Application Name

Date: 12/15/22  
Schedule: 5 PM  
Room Type: Select Room Type  
Movie Name:

Room Type	Movie	Date	Movie Schedule	Movie Name	Movie Poster
VIP	Avatar: Way to water	12/15/22 5 PM			
VIP	Avatar: Way to water	12/19/22 5 PM			
Normal	Hotel Transylvania	12/21/22 5 PM			
VIP	Avatar: Way to water	12/31/22 5 PM			

### Buy Tickets

⚠ Data has been successfully added.

Room	Date	Schedule	Movie Name	Seats Available
5	12/19/22	5 PM	Avatar: Way to water	40

Customer Id: Anne Parker

Ticket Seats Quantity: 1, 2, 3, 4, 5, 6

BUY

## TELAS MULTI-CHANNEL: PANELS

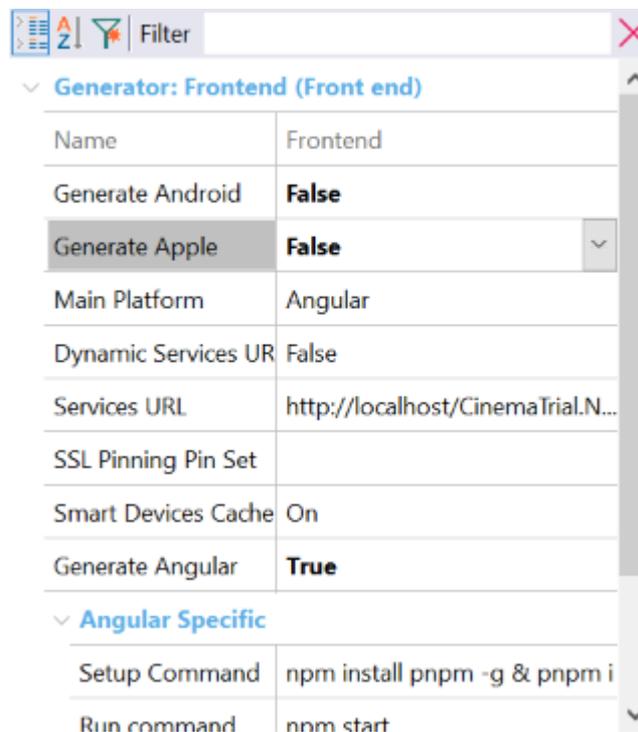
Para realizar o seguinte exercício, você precisará verificar se possui os [requisitos necessários para gerar em Angular](#).

É requerido Panel gerado com Angular que contenha as informações do painel publicitário

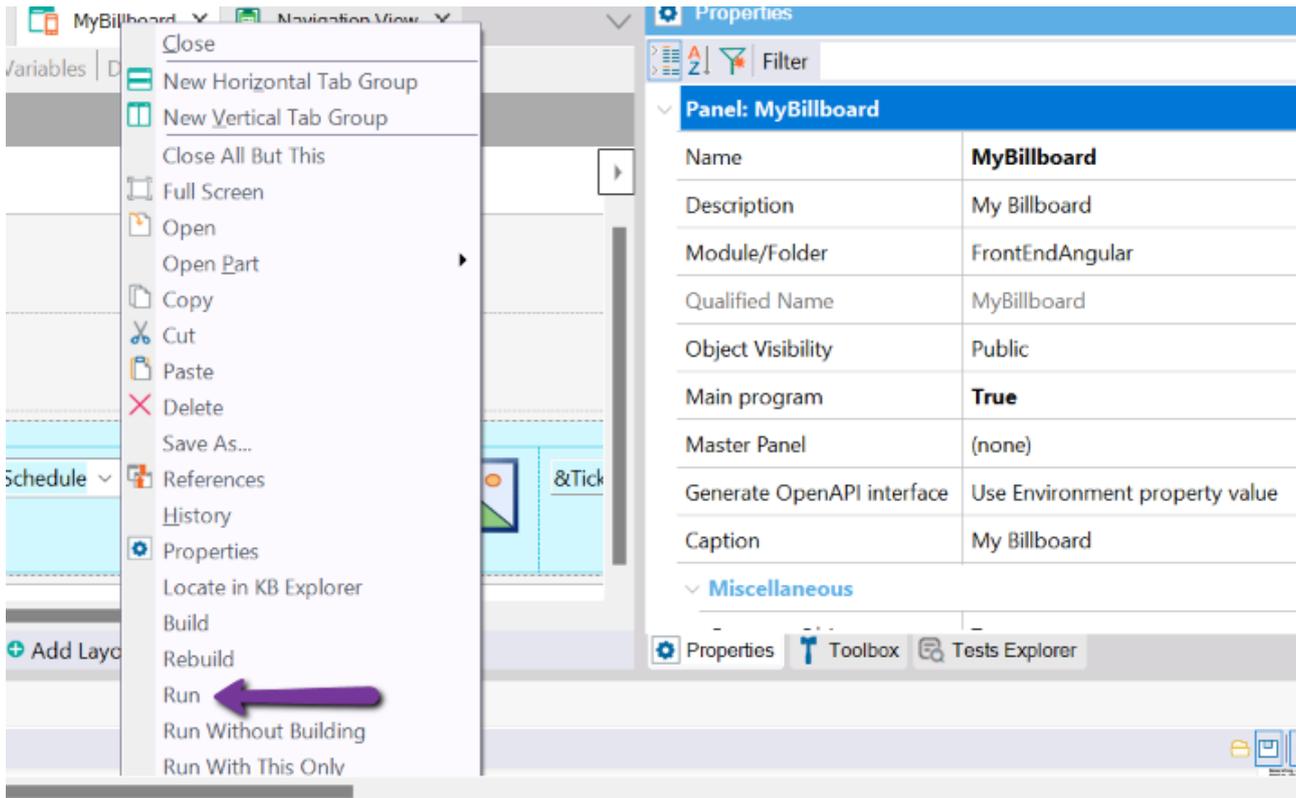
Para solucionar o anterior, dê Ctrl+n e selecione o objeto Panel. No panel, arraste um Grid e selecione a data do filme, o horário, a sala, o nome do filme e o pôster.

Coloque a propriedade Autogrow do Grid como True.

Uma vez salvo o objeto panel, para começar a executar em Angular, vá para *FrontEnd* no *KBExplorer*. Altere a propriedade *Generate Angular* para *True* e, como no momento não vamos desenvolver para aplicações móveis, altere as propriedades *Generate Android* e *Generate Apple* para *False*.



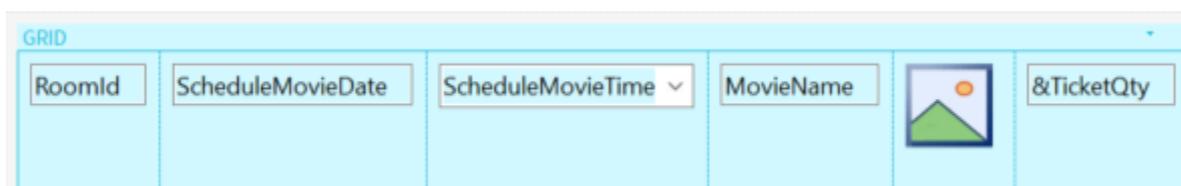
Nas propriedades do panel, altere a propriedade Main Program para True, clique com o botão direito do mouse no nome do objeto e clique em Run para executar a aplicação.



Verifique se a aplicação é vista desta maneira.

3	7/5/2021	5 PM	Hotel Transylvania	
4	6/5/2021	11 AM	Avengers End Game	
6	30/4/2021	8 PM	Inception	

Agora adicione uma variável que calcule o número de vendas emitidas por Painel publicitário. *Nota: temos um atributo que calcula o número de assentos vendidos, não queremos mostrar essa informação, mas sim o número de tickets ou vendas.*



Programo o necessário na seção de eventos e teste o funcionamento da aplicação.

Nos interessa ver que as informações apareçam ordenadas por data e programar filtros pelo nome do filme e que mostre as exibições a partir de uma data

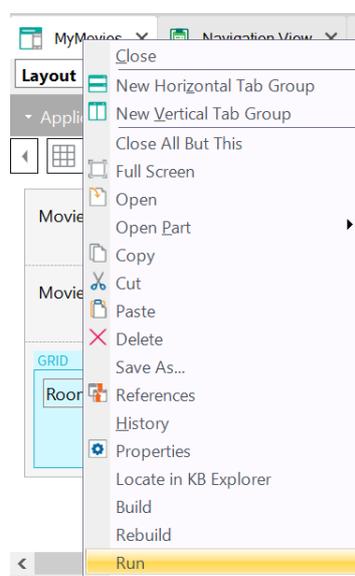
Movie Date	04/05/2021			
Movie Name	av			
<b>6/5/2021</b>	<b>11 AM</b>	<b>4</b>	<b>Avengers End Game</b>	 <b>3</b>

## TESTANDO A APLICAÇÃO EM SMART DEVICES

Altere a propriedade Generate Android para True para começar a testar a aplicação em Smart Devices e verifique se a propriedade Main Platform foi alterada para Android.

Generators	
Generate Android	<b>True</b>
Generate Huawei	False
Generate Apple	<b>False</b>
Generate Angular	<b>False</b>

Agora clique com o botão direito no nome do panel e execute a aplicação. Deverá ser exibida uma tela em Android que será a mesma gerada anteriormente para web.



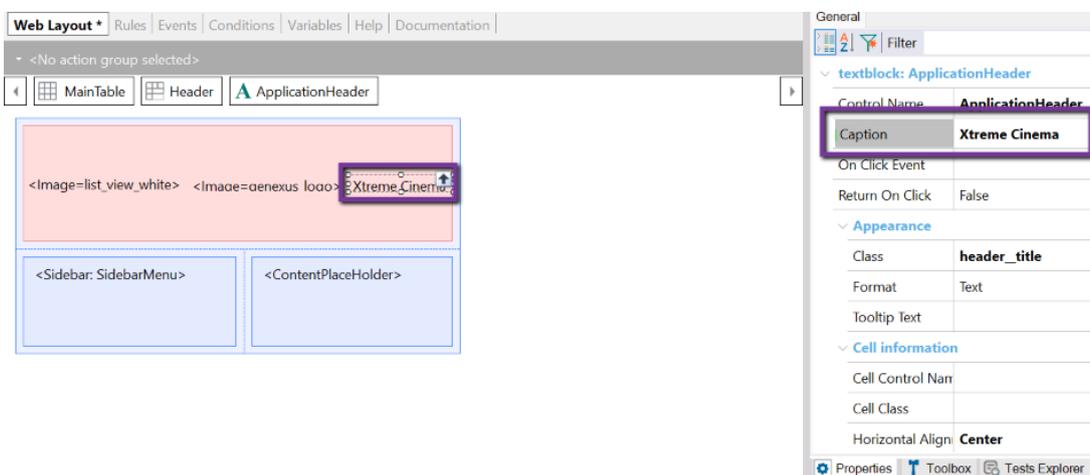
## APLICAÇÕES WEB PROGRESSIVAS [SUGERIDO]

Vamos mudar um pouco a estética da aplicação para que nossa PWA tenha uma aparência melhor. Vá para o KB Explorer e procure no filtro “RWD Master Page”. Esta é a Master Page associada a todos os objetos GeneXus de nossa KB.

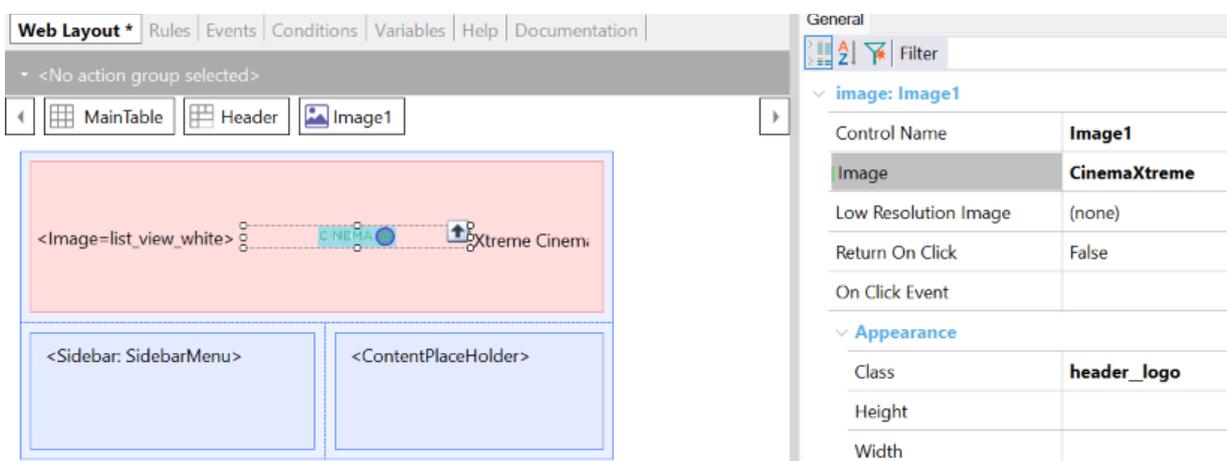
Observe que conta com três seções:

- Um Header onde podemos localizar o nome da aplicação e o logotipo.
- Um objeto web component que é o encarregado de nos oferecer o menu dos links recentes.
- Um content place holder onde são exibidos os objetos de nossa aplicação.

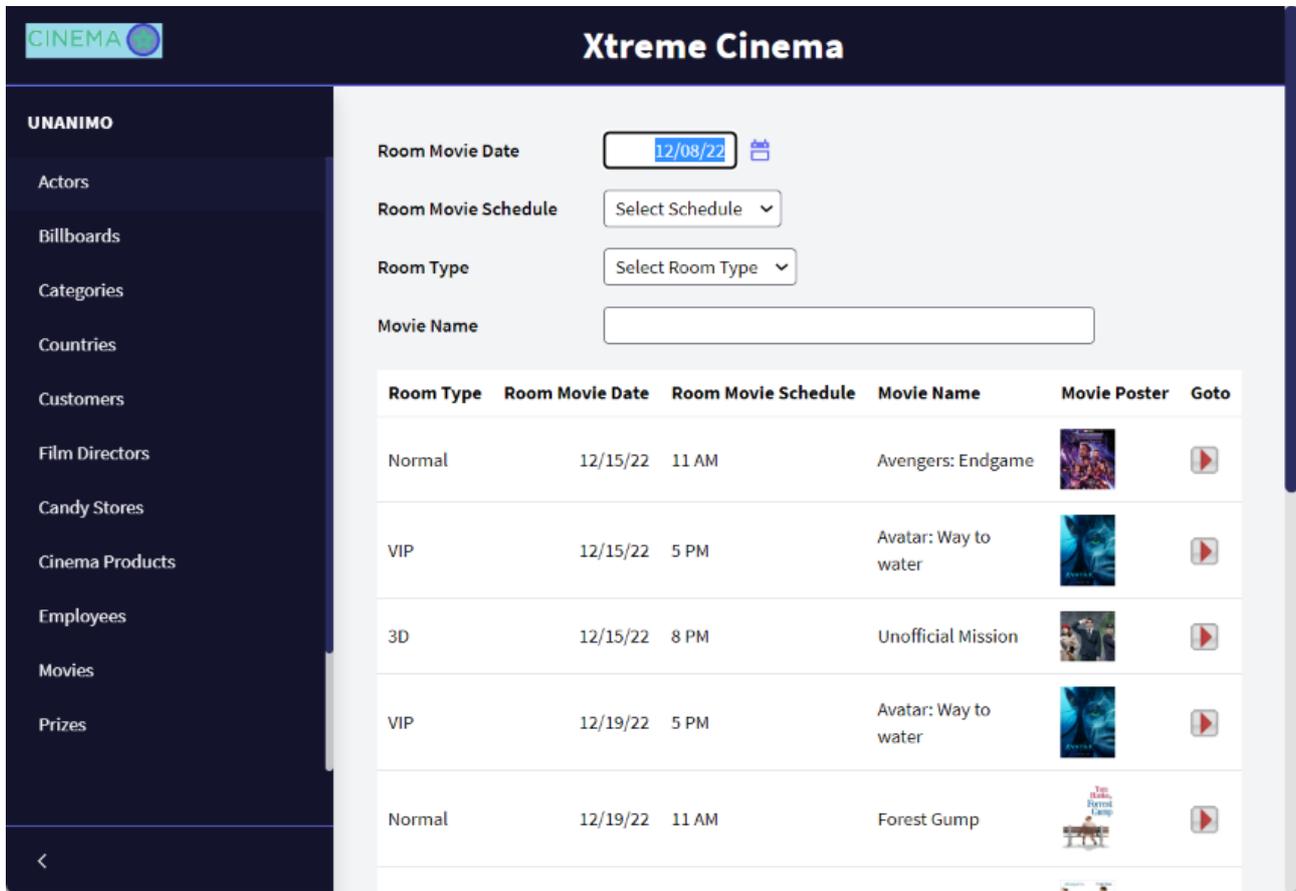
No *Header*, selecione o controle *ApplicationHeader* e altere a propriedade *caption* para o nome da aplicação: My Cinema.



Selecione o controle *Image1* e na propriedade *Image* selecione uma imagem como logotipo da empresa, pode usar as imagens destinadas ao prático. Verifique que tenha a classe *PoweredBy* aplicada.



Faça uma cópia do web panel onde são exibidas todas as informações do painel publicitário:



The screenshot shows the 'Xtreme Cinema' web panel. On the left is a dark sidebar menu with the 'UNANIMO' logo at the top and a list of categories: Actors, Billboards, Categories, Countries, Customers, Film Directors, Candy Stores, Cinema Products, Employees, Movies, and Prizes. The main content area has a header with the 'Xtreme Cinema' title and a search bar. Below the header are four filter fields: 'Room Movie Date' (with a date picker set to 12/08/22), 'Room Movie Schedule' (a dropdown menu), 'Room Type' (a dropdown menu), and 'Movie Name' (a text input field). Below these filters is a table with the following data:

Room Type	Room Movie Date	Room Movie Schedule	Movie Name	Movie Poster	Goto
Normal	12/15/22	11 AM	Avengers: Endgame		
VIP	12/15/22	5 PM	Avatar: Way to water		
3D	12/15/22	8 PM	Unofficial Mission		
VIP	12/19/22	5 PM	Avatar: Way to water		
Normal	12/19/22	11 AM	Forest Gump		

Dentro do novo web panel, selecione o Grid e em sua propriedade class altere para WorkWith.

Oculte também o Id do Filme, remova a variável que nos permite ver os detalhes do filme e seu evento associado.

Selecione os atributos RoomId, ScheduleMovieDate, ScheduleMovieTime e MovieName; na propriedade class coloque: *WWColumn*. E para cada coluna na propriedade Title, mude seus nomes para nomes mais curtos.

Selecione os atributos RoomType, MovieId e MoviePoster; na propriedade class coloque: *WWColumn* *WWOptionalColumn*. Igualmente coloque nomes mais curtos para cada coluna.

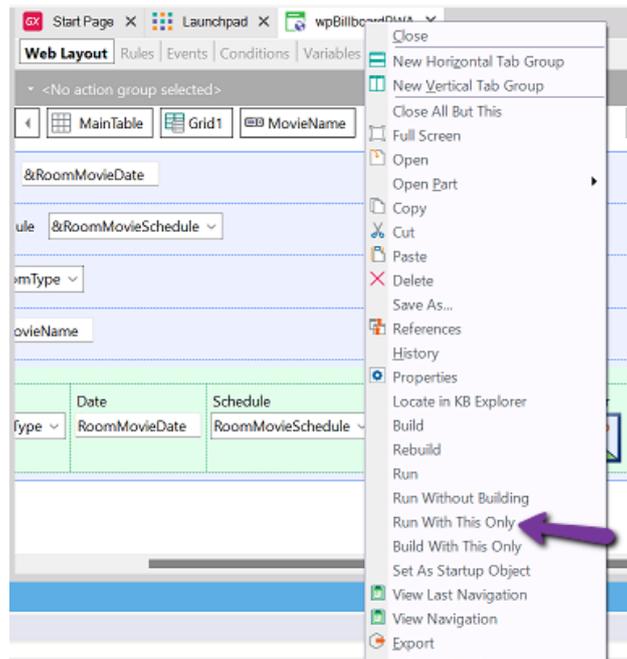
Dentro das propriedades do web panel onde se tem programado o Grid aninhado, altere *Main Program* para *True*. Observe que foi ativado um novo grupo de propriedades denominado *Main object properties*. Dentro desse bloco, altere a propriedade *Web Application* para *Progressive*.

Filter	
Type	Web Page
Master Page	MasterUnanimoSidebar
Show Master Page when Pop-u	False
Main program	<b>True</b>
On session timeout	Ignore
Focus control	Use Environment property value
Cache expiration lapse	
Automatic refresh	Yes
Auto compress http traffic	Use Environment property value
Qualified Name	wpBillboardPWA
Object Visibility	Public
<b>▼ Main object properties</b>	
Web Application	<b>Progressive</b>
Location	
Generator	Default (.NET)

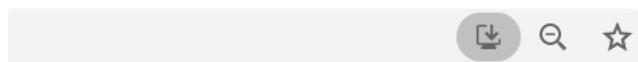
Ao mudar a propriedade default para Progressive, foi ativado um novo grupo de propriedades chamado Web Application que iremos configurar desta forma:

Filter	
<b>▼ Web Application</b>	
Web Application Name	<b>Xtreme Cinema</b>
Web Application Short Name	<b>XC</b>
Web Application Description	<b>Cinema Progressive Web App description</b>
Primary Text Direction	Auto
Display	<b>Fullscreen</b>
Background Color	<input type="color" value="#D4D8D0"/> #D4D8D0
Theme Color	<input type="color" value="#0000CD"/> #0000CD
Prefer Related Applications	False
Web Application Icon	<b>movies</b> <input type="button" value="..."/>
Android Alternative App	
Android Alternative App Ider	
iOS Alternative App	
Offline Object	<b>WWBillboard</b>
<b>▼ Compatibility</b>	

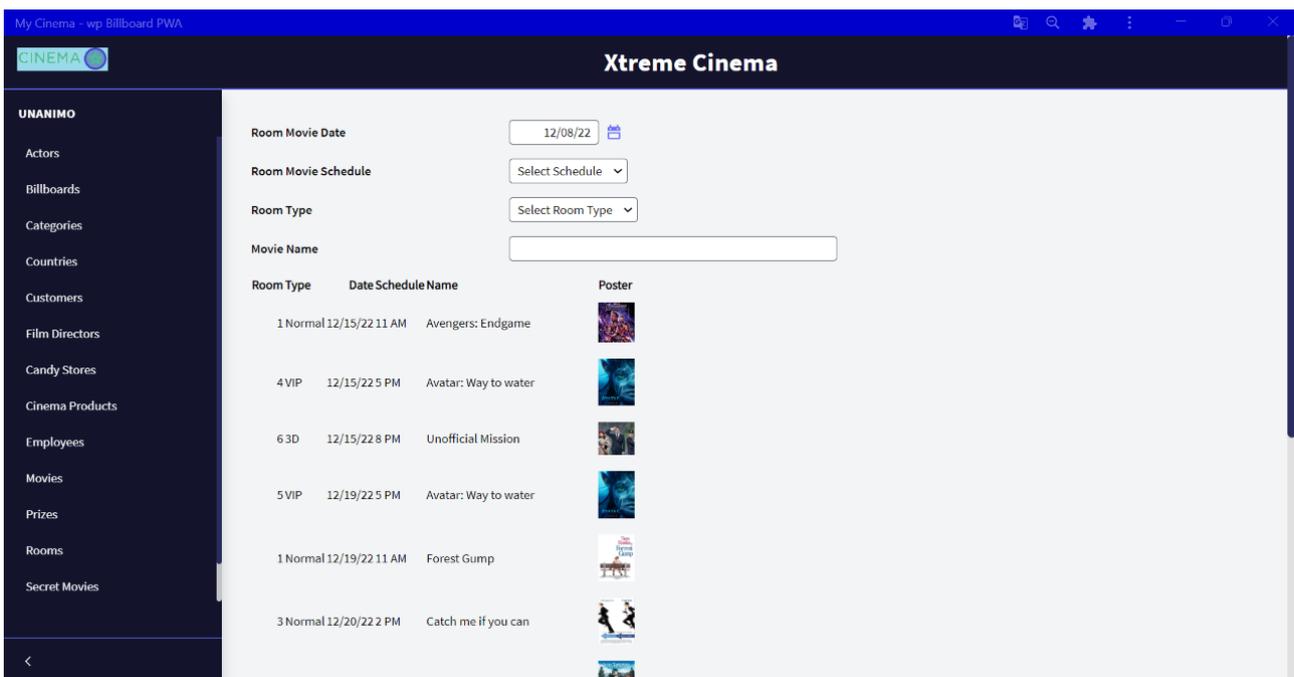
Agora, clique com o botão direito no web panel e selecione a opção Run with this only.



No navegador, observe a opção de instalar a aplicação, clique nela e veja o atalho que foi criado na área de trabalho.



Abra a aplicação clicando duas vezes. Observe que as cores, o ícone e que ao mudar o tamanho da tela a aplicação responde à configuração que fizemos.



Room	Date	Schedule Name
1	12/15/22	11 AM Avengers: Endgame
4	12/15/22	5 PM Avatar: Way to water
6	12/15/22	8 PM Unofficial Mission
5	12/19/22	5 PM Avatar: Way to water
1	12/19/22	11 AM Forest Gump
3	12/20/22	2 PM Catch me if you can
2	12/21/22	5 PM Hotel Transylvania
3	12/22/22	2 PM Star Wars: The Rise of Skywalker
6	12/22/22	8 PM Avatar: Way to water

## WEB SERVICES [SUGERIDO]

Nos é solicitado poder publicar um serviço web que permita consultar as informações de um determinado filme consultando por seu nome.

Para implementar o anterior, vamos criar uma coleção SDT baseada na transação Movie. Você pode arrastar a transação para a estrutura ou criar o SDT manualmente.

Em seguida, arraste a estrutura para um data provider e configure o seguinte:

The screenshot shows the GeneXus IDE interface. On the left, a rule named 'sdtMovies' is defined with the following code:

```

1 sdtMovies
2 where MovieName like &MovieName when not &MovieName.IsEmpty()
3 {
4     sdtMoviesItem
5     {
6         MovieId
7         MovieName
8         MovieYear
9         MoviePoster
10        MovieDescription
11        MovieCountryId
12        MovieCountryName
13        CategoryId
14        CategoryName
15        FilmDirectorId
16        FilmDirectorName
17        IsPremiumMovie
18    }
19 }
20

```

On the right, the 'Data Provider: dpViewMovies' configuration is shown. The 'Expose as Web Service' property is set to 'True', and the 'Web Service Protocol' is set to 'REST Protocol'. Other properties include 'Generate OpenAPI interface' (Use Environment property value), 'Use Native Soap' (Use Environment property value), 'Exposed namespace' (CinemaTrial), 'Main program' (False), 'Call protocol' (Internal), 'Module/Folder' (MyServices), 'Qualified Name' (dpViewMovies), 'Object Visibility' (Public), 'Infer Structure' (No), and 'Output' (sdtMovies).

A condição previamente programada nos permitirá recuperar a lista de todos os filmes mesmo que o parâmetro esteja vazio. Vamos expor o serviço como Rest.

Coloque uma regra que receba o atributo pelo qual filtraremos: `parm(in:&MovieName);`

Agora clique com o botão direito no nome do dp e selecione a opção “Build with this only” para expor o serviço. Para verificar se está tudo correto, coloque o seguinte endereço no navegador:

`http://server/baseURL/rest/nombreDelDataProvider`

Deverá aparecer o seguinte:

```

[{"MovieId":1,"MovieName":"Avengers End Game","MovieYear":2019,"MoviePoster":"PublicTempStorage/multimedia/AvengersEnd_4f30d21695f74e7fb87112f132ab0770.jpg","MovieDescription":"Avengers: Endgame is an American superhero film based on the Marvel Comics superhero team the Avengers.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":5,"CategoryName":"Action","FilmDirectorId":1,"FilmDirectorName":"Joe Russo","IsPremiumMovie":true}, {"MovieId":2,"MovieName":"Catch me if you can","MovieYear":2002,"MoviePoster":"PublicTempStorage/multimedia/CatchMe_c115ac88cf742529610e64b27a9f24.jpg","MovieDescription":"The film is based on the life of Frank Abagnale, who, before his 19th birthday, successfully performed cons worth millions of dollars by posing as a Pan American World Airways pilot, a Georgia doctor and a Louisiana parish prosecutor.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":5,"CategoryName":"Action","FilmDirectorId":3,"FilmDirectorName":"Steven Spielberg","IsPremiumMovie":false}, {"MovieId":3,"MovieName":"Forrest Gump","MovieYear":1994,"MoviePoster":"PublicTempStorage/multimedia/ForrestGump_1293eaa799bd4485abd10a7ff600d11c.jpg","MovieDescription":"The story depicts several decades in the life of Forrest Gump (Hanks), a slow-witted but kind-hearted man from Alabama who witnesses and unwittingly influences several defining historical events in the 20th century United States.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":1,"CategoryName":"Comedy","FilmDirectorId":7,"FilmDirectorName":"Robert Zemeckis","IsPremiumMovie":false}, {"MovieId":4,"MovieName":"Hotel Transylvania","MovieYear":2012,"MoviePoster":"PublicTempStorage/multimedia/HotelTransylvania_36a5b8a603c549b289c3b7d925a5.jpg","MovieDescription":"The film tells the story of Count Dracula, the owner of a hotel called Hotel Transylvania where the world's monsters can take a rest from human civilization.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":4,"CategoryName":"Cartoon","FilmDirectorId":8,"FilmDirectorName":"Genndy Tartakovsky","IsPremiumMovie":false}, {"MovieId":5,"MovieName":"The family Man","MovieYear":2000,"MoviePoster":"PublicTempStorage/multimedia/FamilyMan_ce19e3c9ddf64a45b7bb9758fb6ce33b.jpg","MovieDescription":"The film centers on a man who experiences what his life might have been if he had made a different decision earlier in his life.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":3,"CategoryName":"Drama","FilmDirectorId":4,"FilmDirectorName":"Brett Ratner","IsPremiumMovie":false}, {"MovieId":6,"MovieName":"Harry Potter and the Prisoner of Azkaban","MovieYear":2004,"MoviePoster":"PublicTempStorage/multimedia/HPPrisoner_7c5f2c468a19cc64b0898481471a385.jpg","MovieDescription":"The film stars Daniel Radcliffe as Harry Potter, with Rupert Grint as Ron Weasley, and Emma Watson as Hermione Granger. Its story follows Harry Potter's third year at Hogwarts as he is informed that a prisoner named Sirius Black has escaped from Azkaban and intends to kill him.", "MovieCountryId":2,"MovieCountryName":"United Kingdom","CategoryId":6,"CategoryName":"Fantasy","FilmDirectorId":2,"FilmDirectorName":"Alfonso Cuaron","IsPremiumMovie":true}, {"MovieId":7,"MovieName":"Unofficial Mission","MovieYear":2017,"MoviePoster":"PublicTempStorage/multimedia/misionNoOficial_372f49892b604d39b1252d62e431613f.jpg","MovieDescription":"\"u000aUnofficial Mission is a documentary comedy, directed by Alfonso Guerrero, Denny Brechner and Marcos Hecht.", "MovieCountryId":1,"MovieCountryName":"Uruguay","CategoryId":1,"CategoryName":"Comedy","FilmDirectorId":5,"FilmDirectorName":"Denny Brechner","IsPremiumMovie":false}, {"MovieId":8,"MovieName":"Inception","MovieYear":2010,"MoviePoster":"PublicTempStorage/multimedia/Inception_4199fd287e3419f80146fdb045c3e02.jpg","MovieDescription":"The film stars Leonardo DiCaprio as a professional thief who steals information by infiltrating the subconscious of his targets.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":5,"CategoryName":"Action","FilmDirectorId":6,"FilmDirectorName":"Christopher Nolan","IsPremiumMovie":false}]

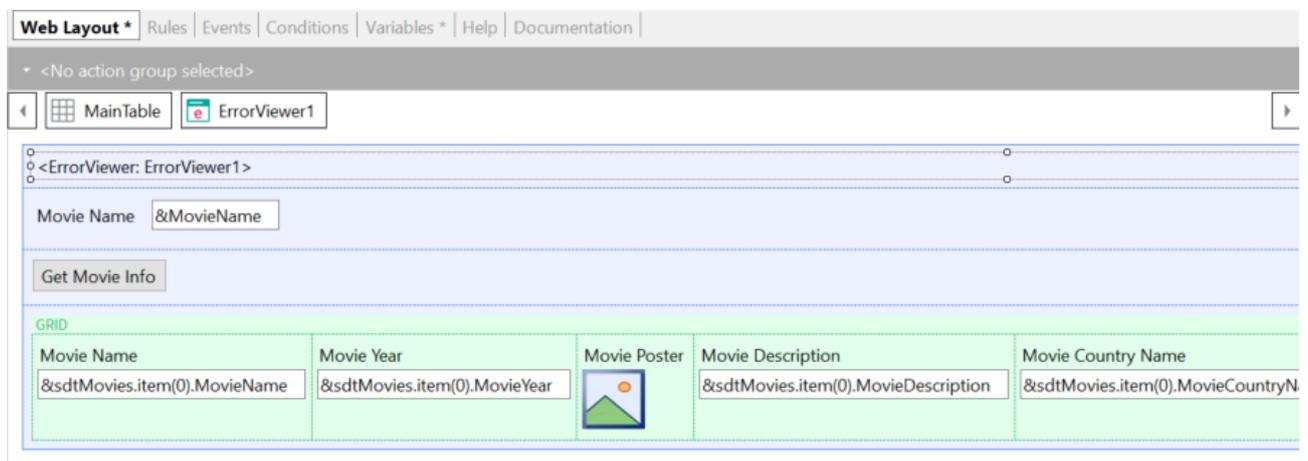
```

Agora construiremos um web panel que nos permitirá consumir o serviço Rest com o tipo de dados HttpClient. Pode estar dentro da mesma KB ou em uma diferente. O web panel deverá ter a seguinte estrutura:

Variáveis:

Variável	Tipo de dado
HttpClient	HttpClient
MovieName	Character(50)
SdtMovies	Baseado no SDT previamente criado
StringParameter	Character(200)

## Web Layout



## Eventos

```

1 Event 'Get Movie Info'
2     &httpClient.Host= "localhost"
3     &httpClient.Port = 8082
4     &httpClient.BaseUrl = "/MyCinemaNETSQLServer/rest/"
5     &httpClient.secure = 0
6     &StringParameter = '<DP_NAME>?' + 'MovieName=' + &MovieName
7     &httpClient.Execute('GET', &StringParameter)
8
9     if &httpClient.StatusCode = 200
10        msg("Everything Ok!")
11        &sdtMovies.FromJson(&httpClient.ToString())
12
13    else
14        msg("Error: " + &httpClient.StatusCode.ToString())
15    endif
16 Endevent

```

Execute a aplicação e verifique se é possível consumir o serviço. Escreva o nome de um filme ou deixe o campo vazio, em ambos os casos deveria poder consumir o serviço.

NOTA: Habilite a propriedade *read only* como true para cada campo do SDT que está sendo impresso na tela.

The screenshot shows the 'Xtreme Cinema' application interface. On the left is a dark sidebar with a menu including 'Actors', 'Billboards', 'Categories', 'Countries', 'Customers', 'Film Directors', 'Candy Stores', 'Cinema Products', 'Employees', 'Movies', and 'Prizes'. The main content area has a header 'Xtreme Cinema' and a status indicator 'Everything Ok!'. Below this is a search bar with 'avatar' entered and a 'GET MOVIE INFO' button. A table displays the search results for 'Avatar: Way to water'.

Movie Name	Movie Year	Movie Poster	Movie Description	Movie Country Name	Category Name	Film Director Name
Avatar: Way to water	2022		Jake Sully and Ney'tiri have formed a family and are doing everything to stay together. However, they must leave their home and explore the regions of Pandora. When an ancient threat resurfaces, Jake must fight a difficult war against the humans.	United States	Fantasy	James Cameron

This screenshot shows the same application interface but with a different search result. The search bar is empty, and the table lists four movies.

Movie Name	Movie Year	Movie Poster	Movie Description	Movie Country Name	Category Name	Film Director Name
Avengers: Endgame	2019		After Thanos, an intergalactic warlord, disintegrates half of the universe, the Avengers must reunite and assemble again to reinvigorate their trounced allies and restore balance.	United States	Action	Joe Russo
Catch me if you can	2002		Notorious con artist Frank Abagnale has duped people worth millions of dollars with his masterful art of deception. With his scams getting bolder, he is soon pursued by FBI agent Carl Hanratty.	United States	Comedy	Steven Spielberg
Avatar: Way to water	2022		Jake Sully and Ney'tiri have formed a family and are doing everything to stay together. However, they must leave their home and explore the regions of Pandora. When an ancient threat resurfaces, Jake must fight a difficult war against the humans.	United States	Fantasy	James Cameron
Forest Gump	1994		Forrest, a man with low IQ, recounts the early years of his life when he found himself in the middle of key historical events. All he wants now is to be reunited with his childhood sweetheart, Jenny.	United States	Drama	Robert Zemeckis