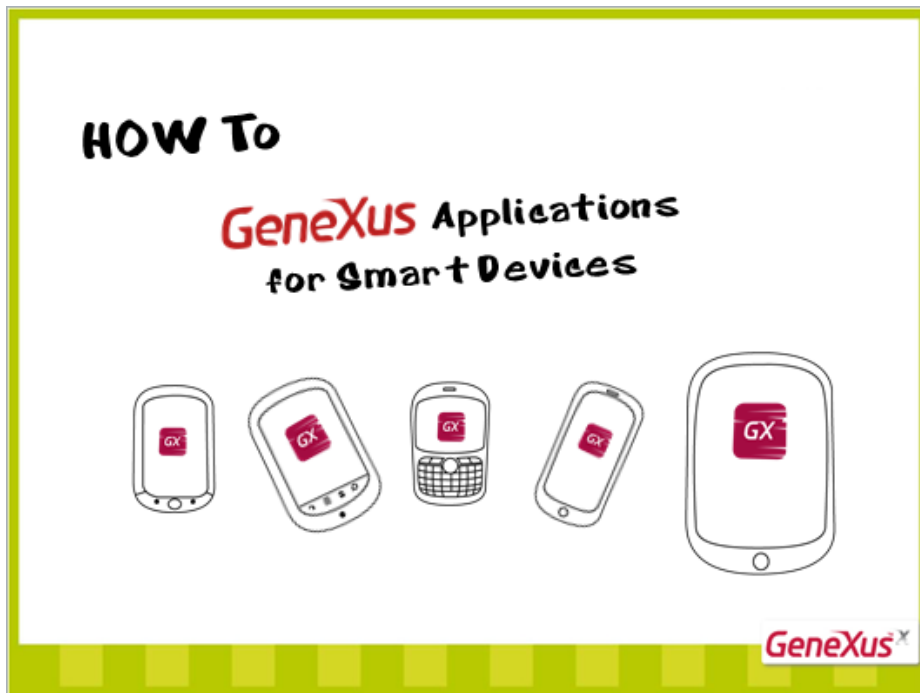
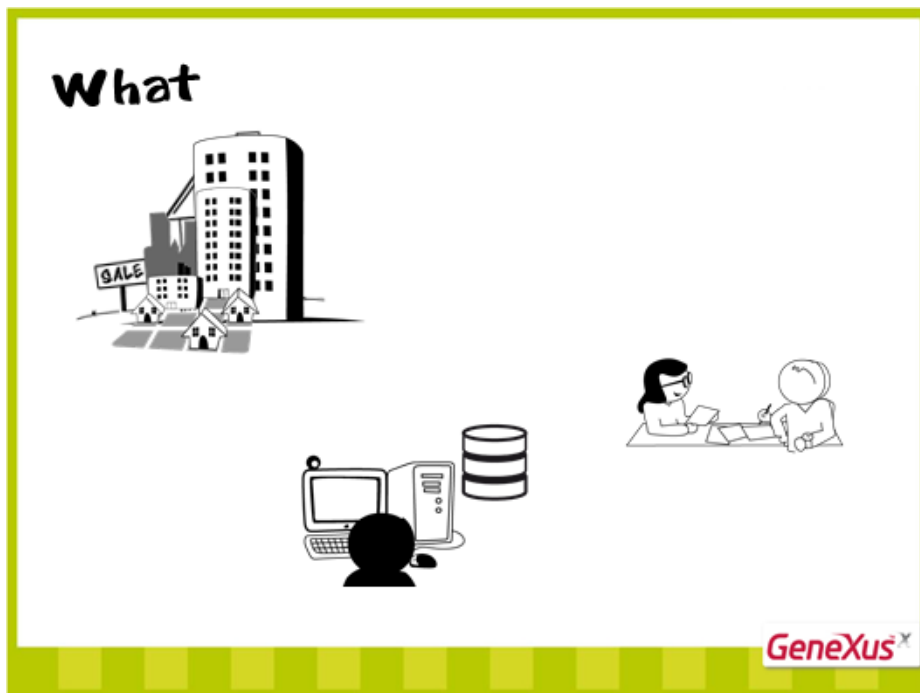


1.



Como implementar aplicativos para Smart Devices...com GeneXus.

2.

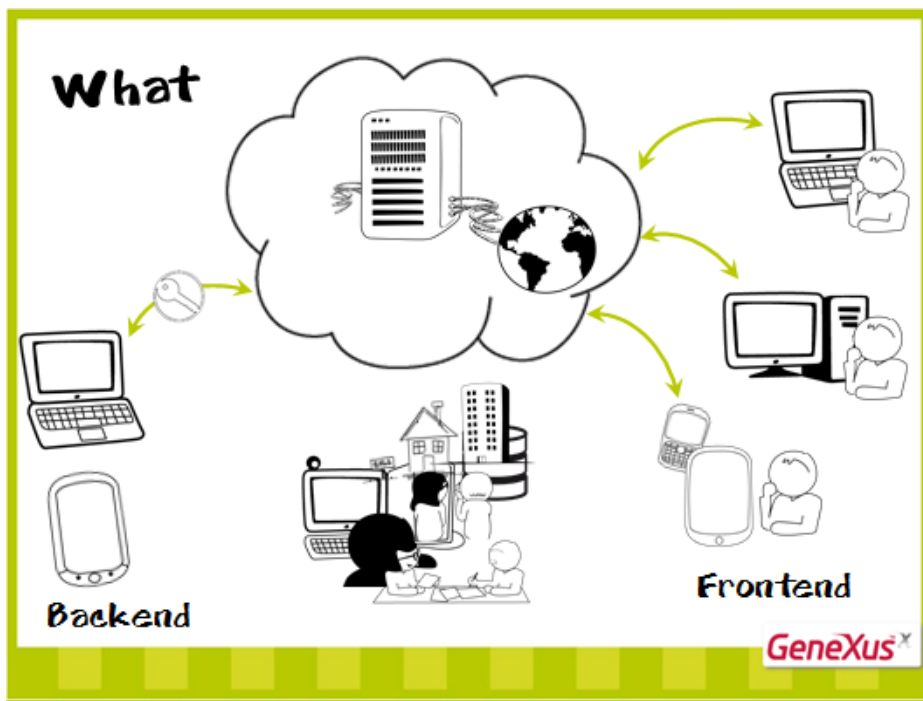


Suponhamos que precisamos desenvolver um aplicativo simplificado para uma imobiliária...

A imobiliária trabalha em certas vizinhanças e zonas da cidade e oferece casas e apartamentos para venda ou aluguel. Ela tem clientes com os quais seus agentes imobiliários realizam contratos de venda ou aluguel...

Para lidar com suas informações, a imobiliária deve contar com um sistema de informática que registre as vizinhanças, as propriedades, os clientes, os agentes imobiliários e contratos: todas as entidades com as quais trabalha.

3.



Para fazê-lo acessível com qualquer navegador, de qualquer lugar com conexão à Internet e sem preocupação com a infraestrutura, podemos desenvolvê-lo em uma nuvem.

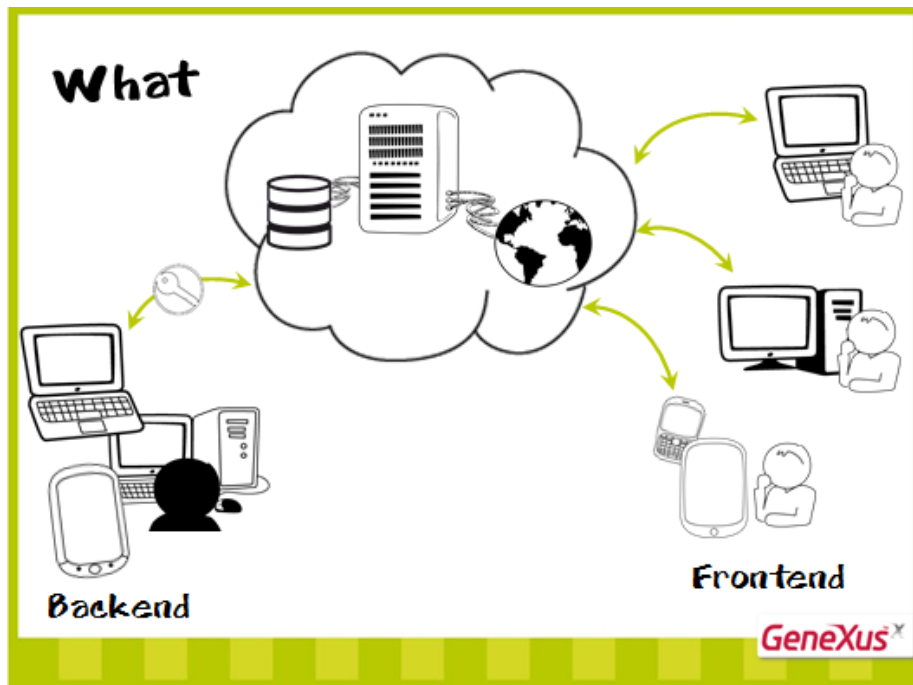
O sistema terá uma parte restrita, interna, utilizada só pelos funcionários da imobiliária (o que se conhece como backoffice ou backend),

Onde se registram as informações das vizinhanças e propriedades, dos agentes, dos clientes, dos contratos e se trabalha com elas.

Também terá uma parte pública, utilizada por quem esteja buscando uma propriedade para alugar ou comprar. Assim, é possível visualizar a oferta de propriedades, filtrar a informação que lhe interessa e, eventualmente, solicitar alguma visita. É o que conhecemos como Frontend.

Visto que praticamente não há diferenças funcionais entre a implementação do backend e do frontend, nos concentraremos em uma: o backend.

4.



5.



Este pode ter uma parte web, implementada por exemplo em ruby , em Java ou em .net...para ser utilizada com um navegador.

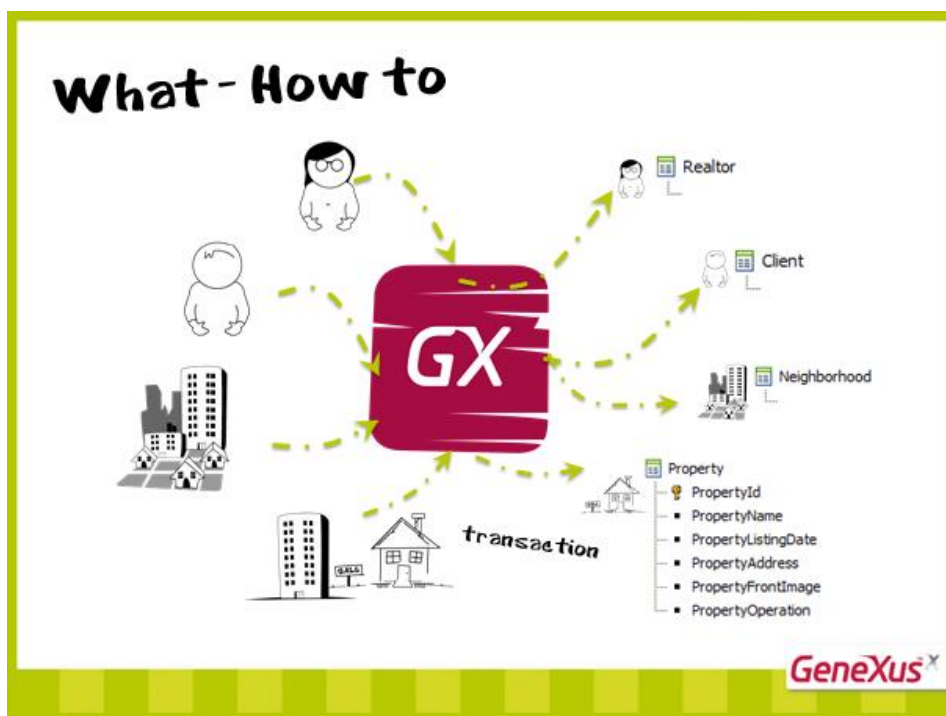
Mas também, para possibilitar o trabalho móvel dos agentes imobiliários, haverá uma parte implementada para dispositivos inteligentes. Dependendo dos dispositivos, escolhe-se a linguagem de programação.

Se queremos desenvolver este aplicativo para vendê-lo a distintas imobiliárias, não podemos fazê-lo para um dispositivo particular. Mas...temos que programá-los diretamente nas linguagens de cada dispositivo?!

Não se usamos GeneXus.

Da mesma forma, quando queremos desenvolver um aplicativo web...Programaremos o aplicativo em GeneXus e ele o construirá na linguagem do ou dos dispositivos escolhidos.

6.



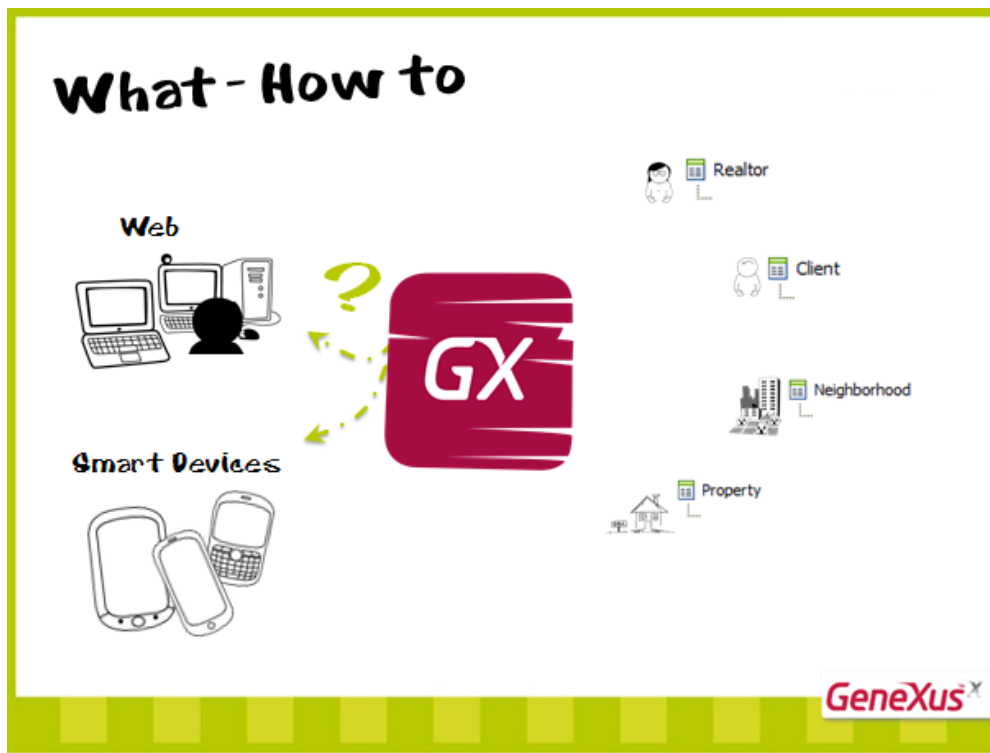
Como?

A primeira coisa que devemos fazer é representar em GeneXus as entidades da realidade com as que trabalhará o aplicativo: agentes, clientes, vizinhanças e propriedades.

Como fazemos? Criando um objeto GeneXus transação para cada entidade...e, em cada objeto, definindo a informação que registraremos de cada um (seus atributos)...

Para que parte do Backend?

7.



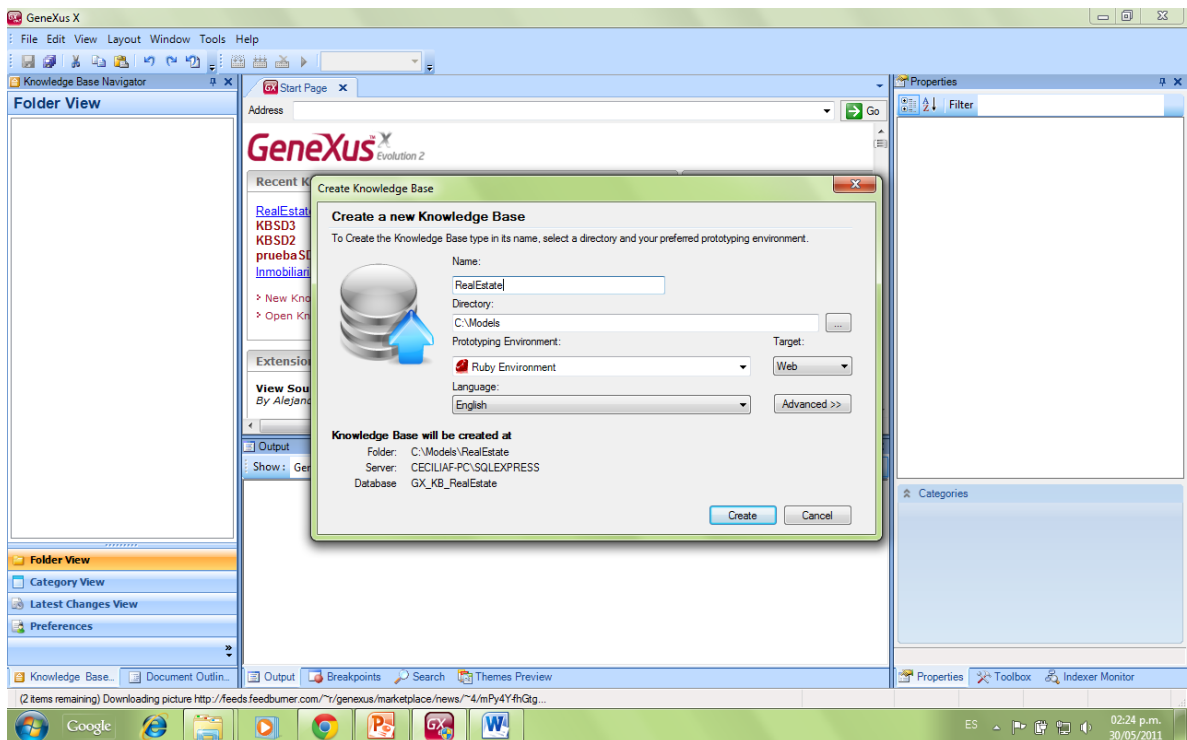
Web ou Smart Devices? Para ambos!!

Vamos a GeneXus...

Mostraremos a criação da KB (com protótipo Ruby) e a transação Property. Quem já conhece estes temas, pode passar diretamente ao vídeo seguinte, que mostra como prototipar na nuvem.

(Em GeneXus:)

Criamos a base de conhecimento que chamaremos RealEstateAgency (imobiliária). Esta conterá todo o conhecimento necessário para implementar nosso backend.

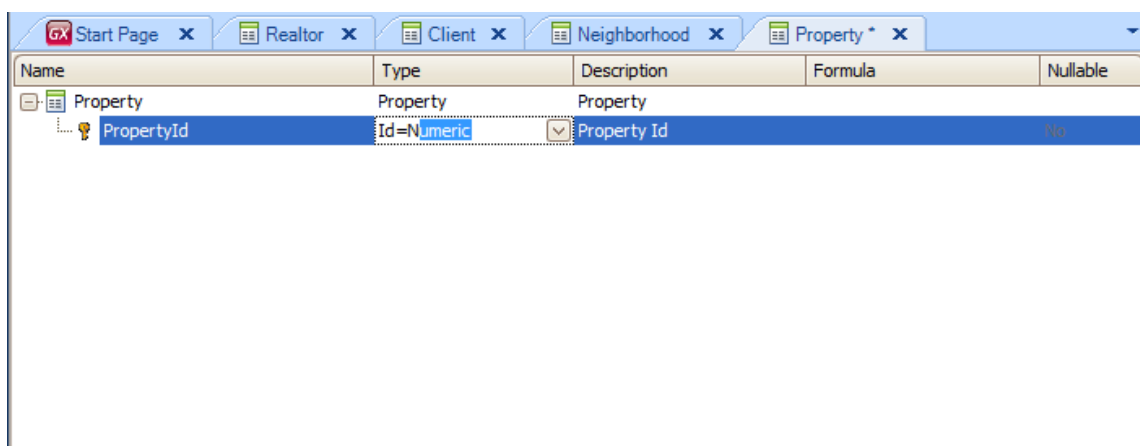


Devemos escolher a plataforma de geração automaticamente. Escolhamos a pré-determinada, Ruby.

A partir deste ponto podemos criar transações que identificamos antes.

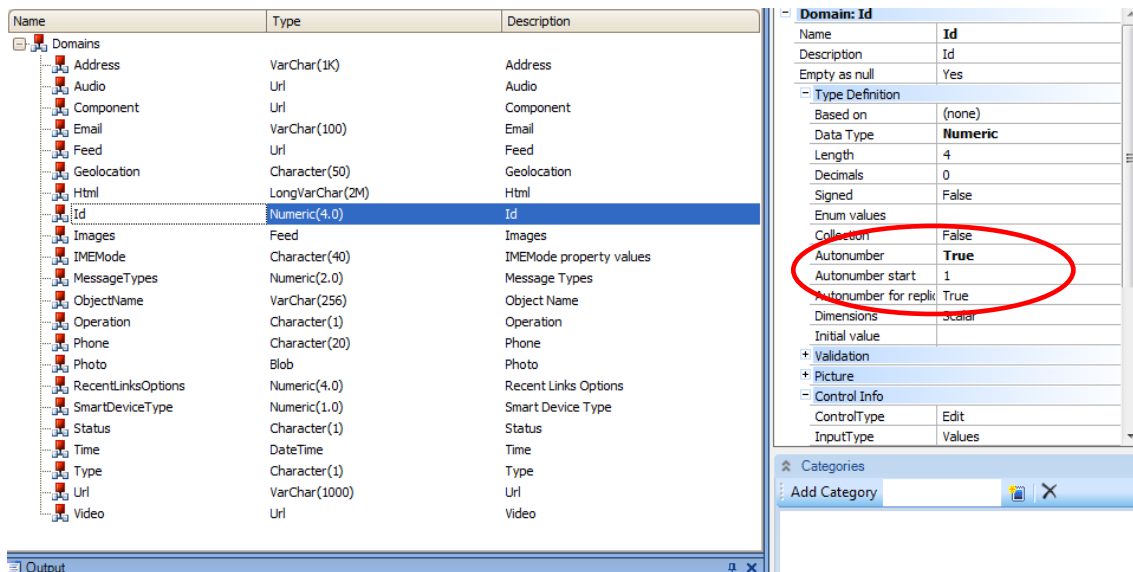
Trabalharemos com a transação de Propriedades, que é onde nos deteremos. As demais serão análogas.

Criamos então o novo objeto GeneXus...transação...Property

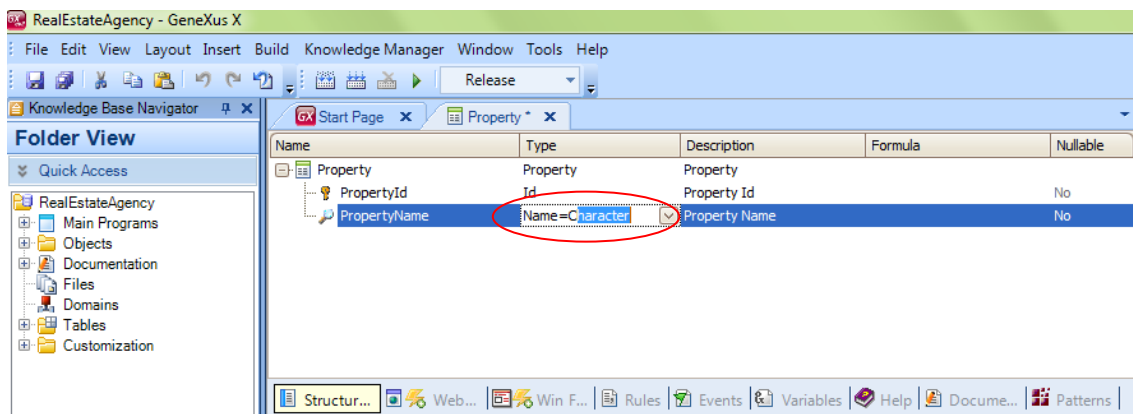


Criemos seu atributo identificador e definamos um domínio que chamaremos "Id", que caracterize todos os identificadores.

Vejamos ele entre os domínios pré-definidos.

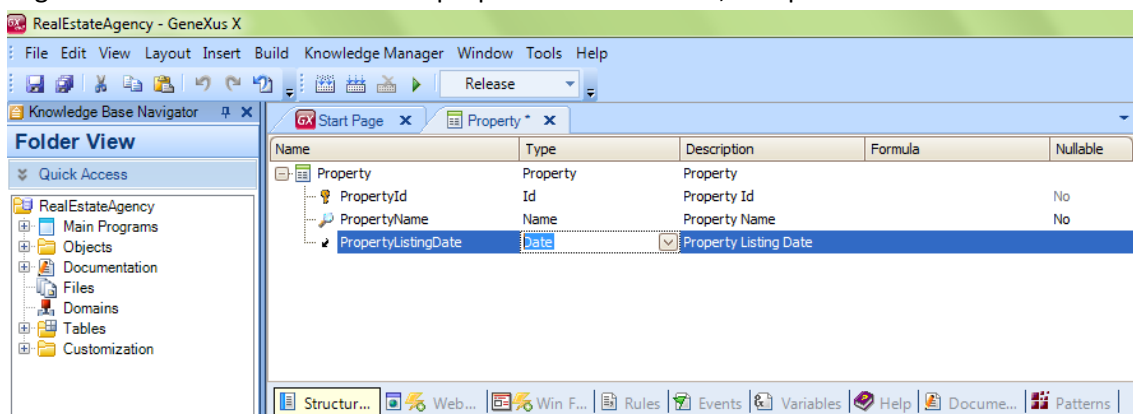


Podemos indicar-lhe que seja um domínio autonumerado para que todos os atributos que estejam baseados neste domínio se numerem automaticamente.

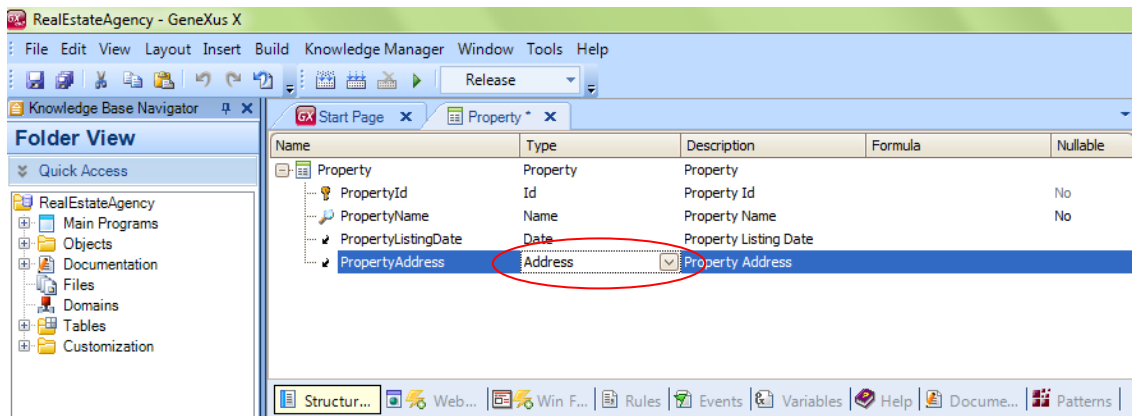


Depois é a vez do atributo que representará o nome da propriedade, do tipo carácter, para que possamos criar um domínio "Name", de maneira a reutilizá-lo em outros atributos que signifiquem também nome.

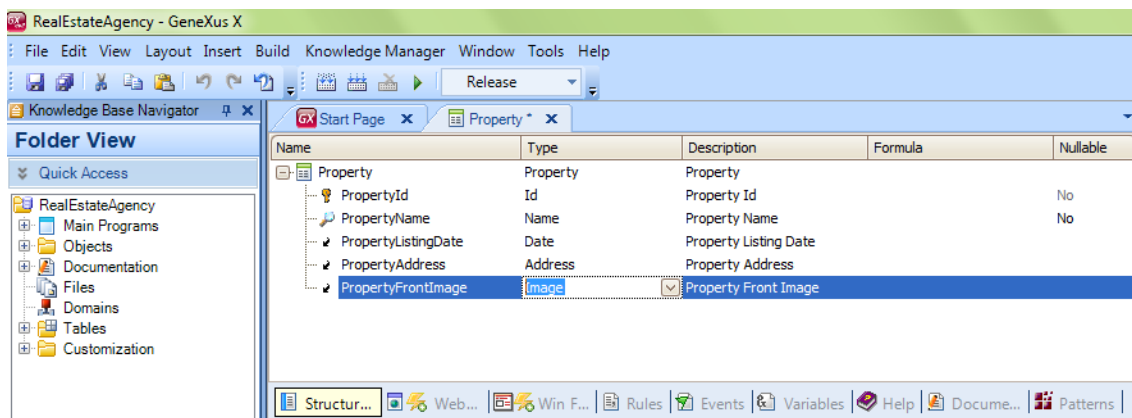
Registraremos a data de entrada da propriedade no sistema, do tipo de dado Date:



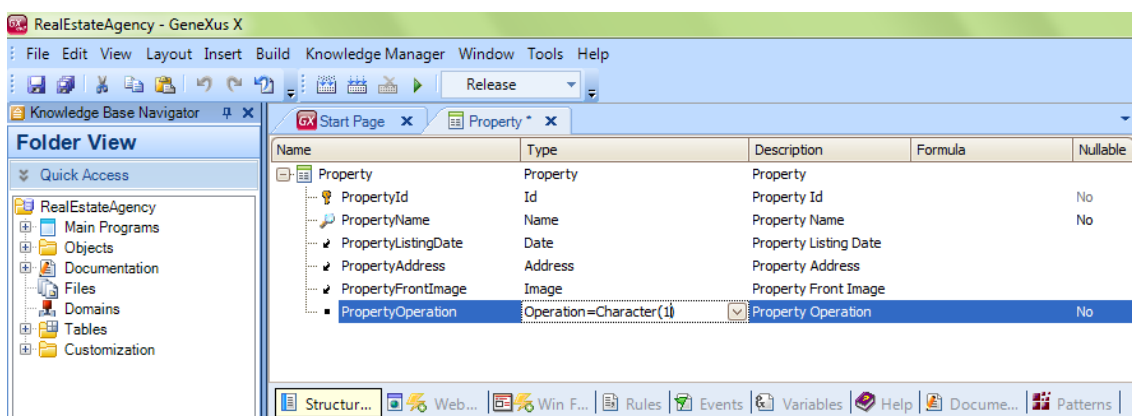
Em seguida, agregamos o atributo PropertyAddress que, como vemos, assume automaticamente o domínio pré-definido Address.

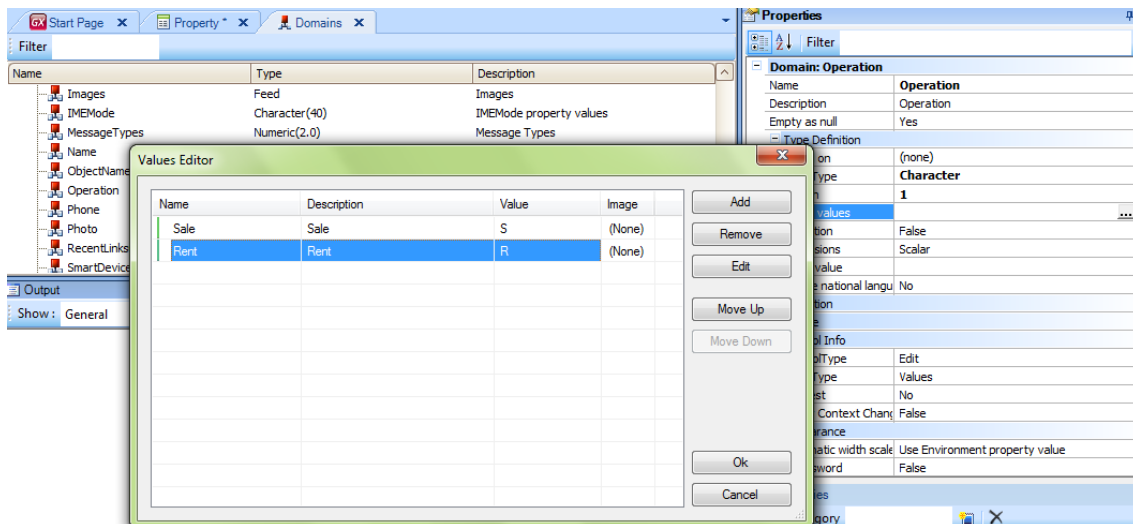


Depois precisamos de um atributo para registrar a foto da fachada da propriedade. Será do tipo Imagem.



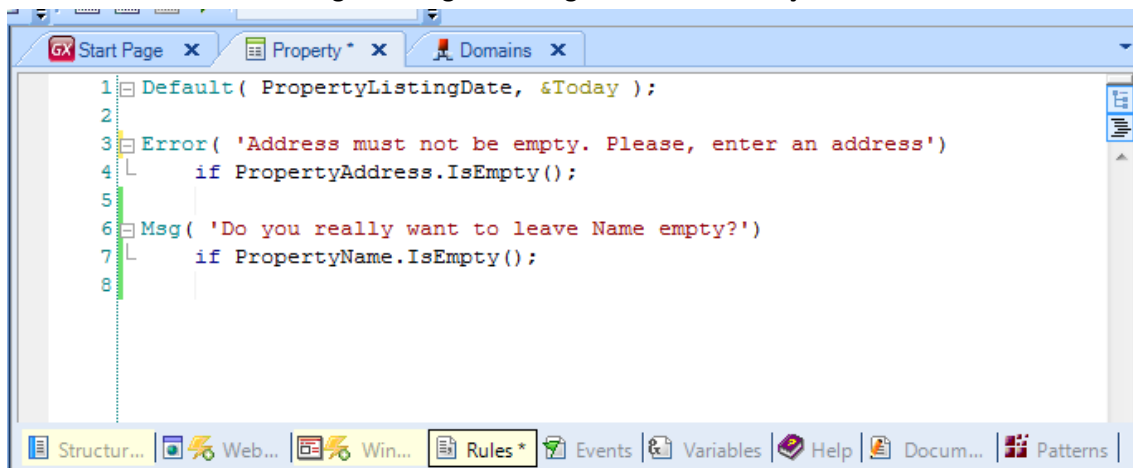
E, por último, um atributo para registrar o tipo de operação: se será uma venda ou um aluguel. Será um domínio enumerado, que assumirá dois valores: Sale (venda) e Rent (aluguel).





E assim, seguiríamos adicionando os demais atributos que a realidade nos demande...Para esta demonstração, estes nos bastam.

Podemos adicionar ainda algumas regras de negócio a esta transação:



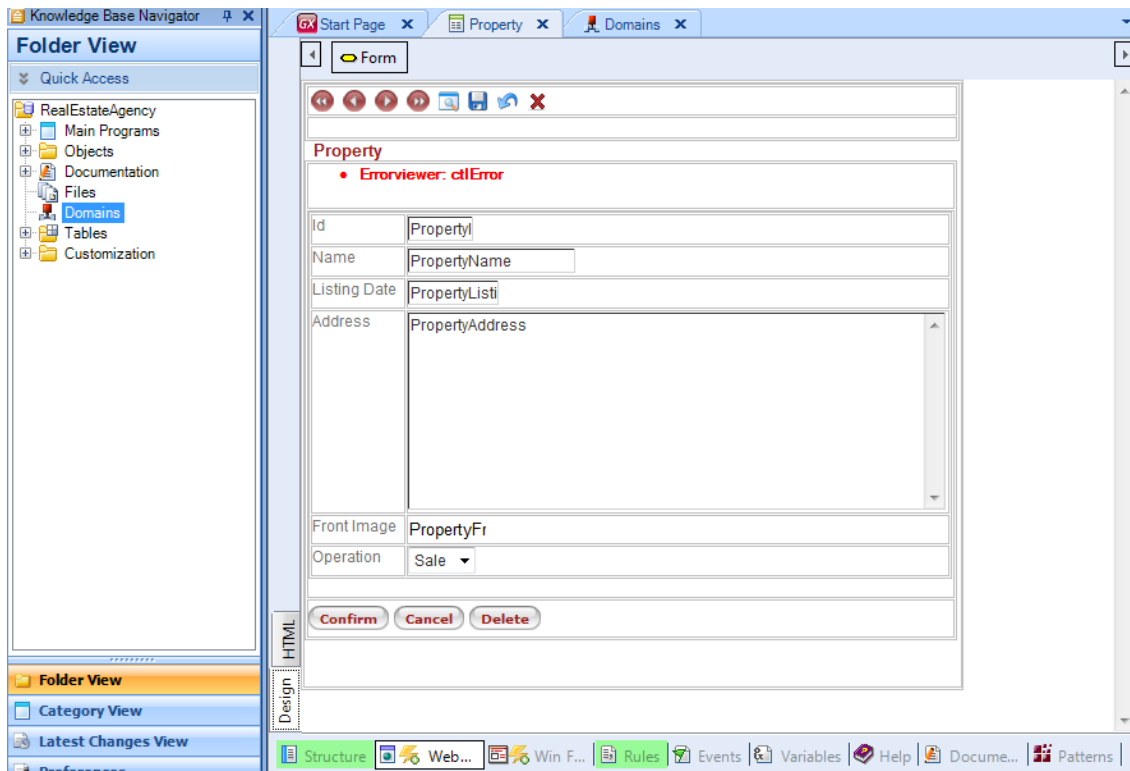
...atribuir a data de hoje como valor automático para a data de entrada da propriedade no sistema,

Um erro... se não se preenche o endereço

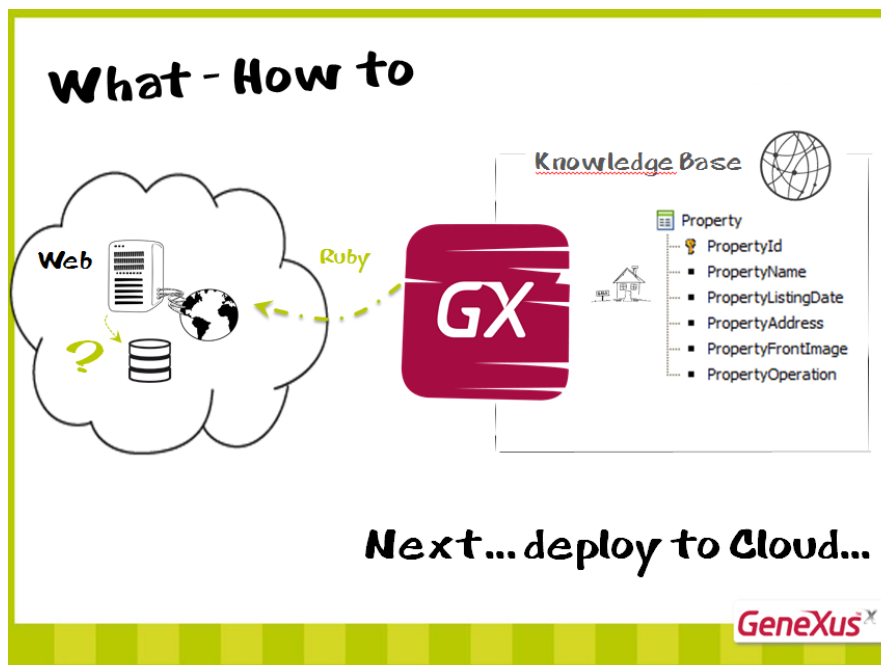
E uma mensagem...advertindo quando se deixou o nome da propriedade vazio...

Gravamos...

Aqui podemos ver a interface web default que GeneXus cria automaticamente para a transação:



Voltar às ppts



Por enquanto, só dissemos a GeneXus que é chegado o momento, vamos pedir-lhe que gere o aplicativo web implementando-o em ruby.

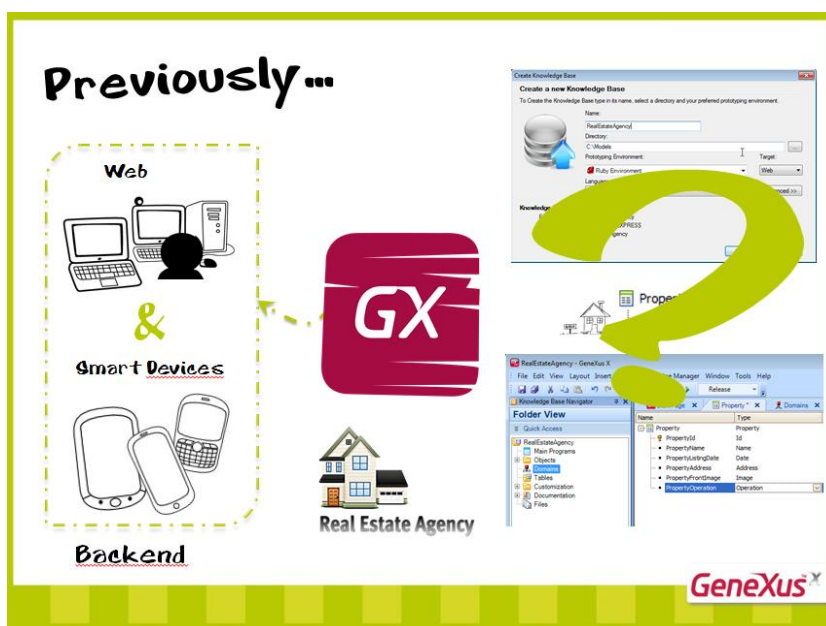
Dessa forma, criadas as transações, GeneXus terá que implementá-las nesta plataforma.

Porém, para isso, ainda nos resta terminar de configurar esse ambiente: por exemplo, dizer-lhe onde estará a base de dados, qual será seu nome, entre outras coisas.

Poderíamos localizar os programas e a base de dados localmente...ou na nuvem...faremos o último simplesmente mudando o valor de uma propriedade.

Será o tema do próximo vídeo...

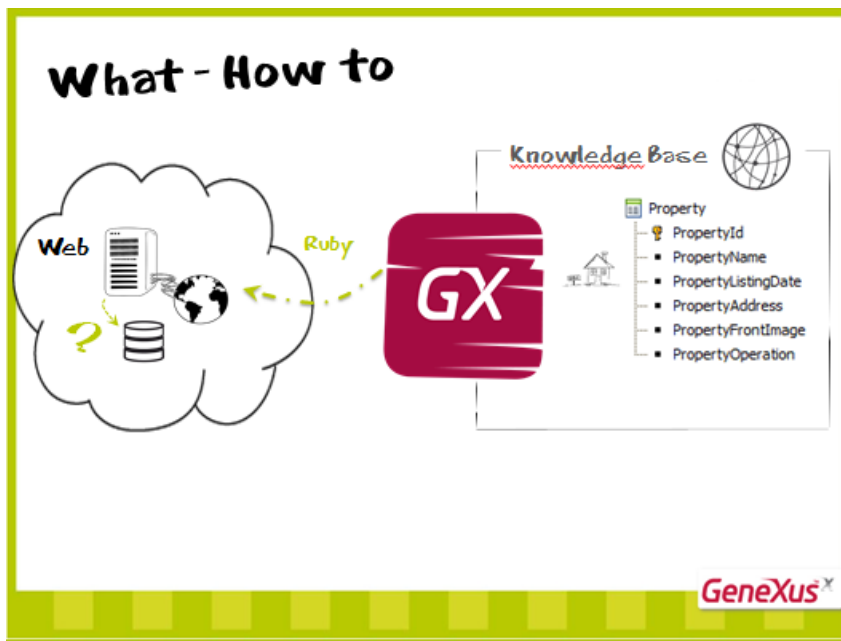
Segundo vídeo



Anteriormente...

Havíamos definido nosso problema: construir um backend web e um para smart devices para uma imobiliária.

Começamos a criar em GeneXus uma KB e uma transação: Property.



Por enquanto, só dissemos a GeneXus que é chegado o momento, vamos a pedir-lhe que gere o aplicativo web implementando-a em ruby.

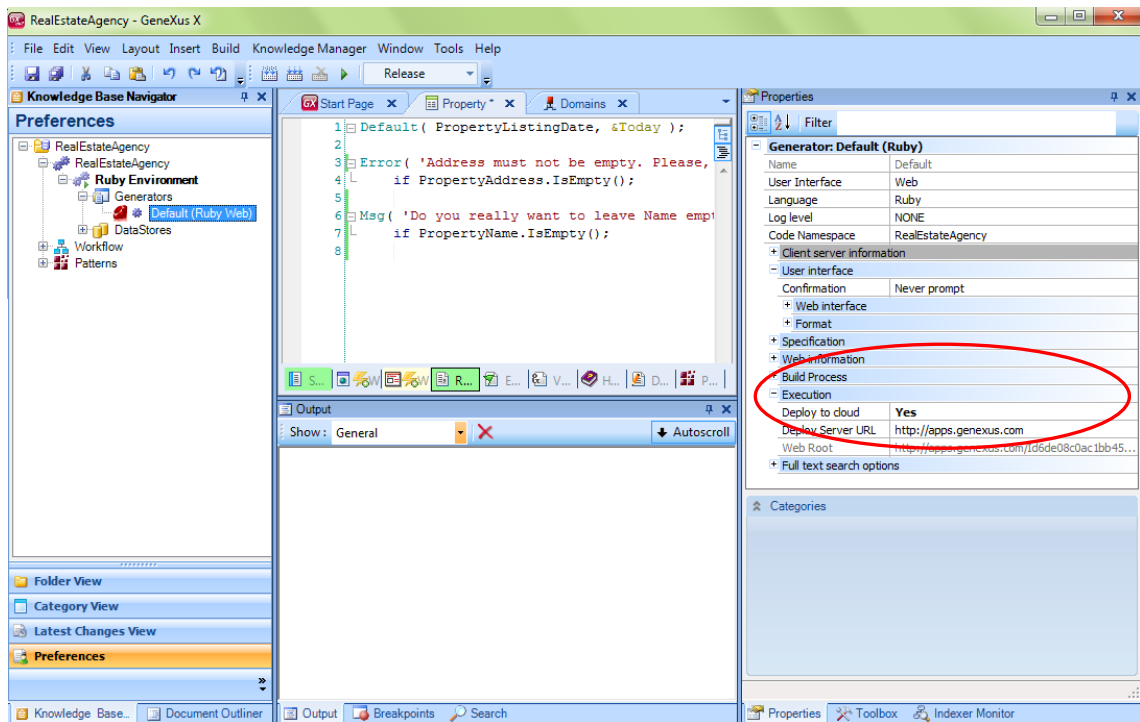
Por isso, criadas as transações, GeneXus terá que implementá-las nesta plataforma.

Para isso, ainda nos resta terminar de configurar esse ambiente: por exemplo, dizer-lhe onde estará a base de dados, qual será seu nome, entre outras coisas.

Poderíamos localizar os programas e a base de dados localmente...ou na nuvem...faremos este último, simplesmente mudando o valor de uma propriedade.

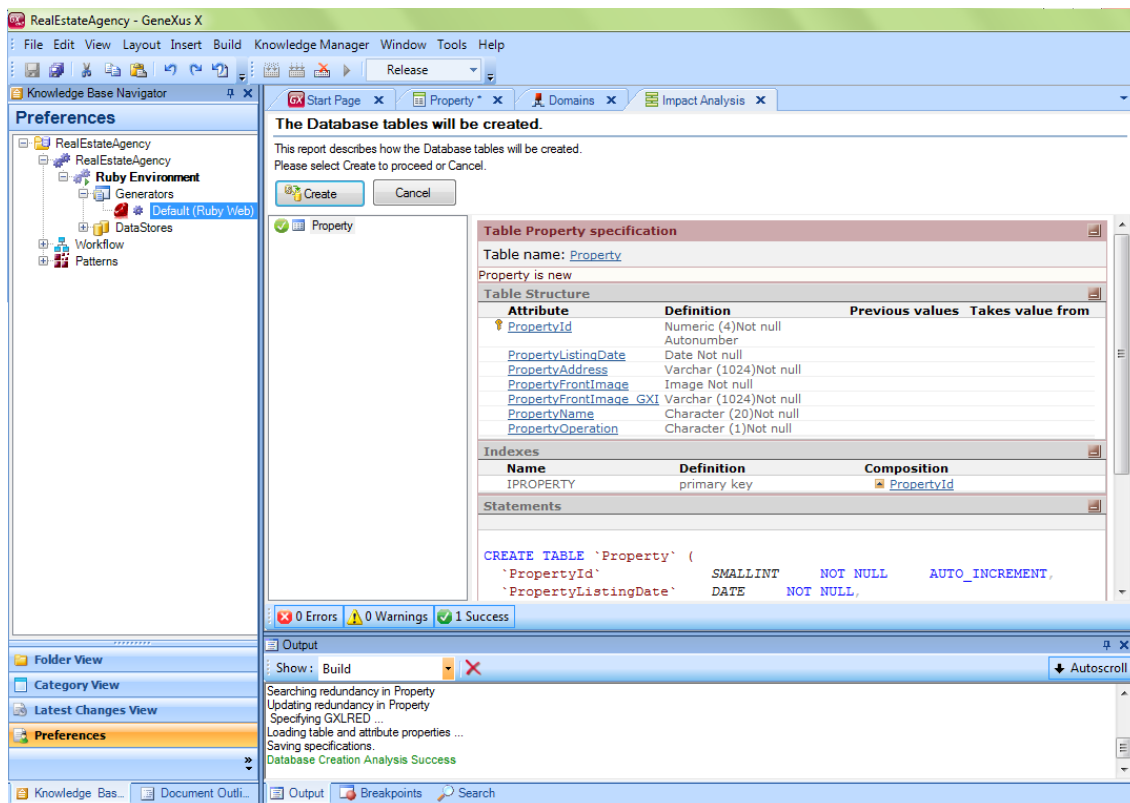
Para isso, vamos ao setor de Preferences na base de conhecimento.

Entre as propriedades do gerador Default (no nosso caso Ruby), buscamos a propriedade Deploy to Cloud, que automaticamente está em Não e mudamos para Sim.



Fazendo simplesmente isso, toda a informação necessária ficará automaticamente configurada apontando para a nuvem.

Temos tudo pronto para testar nosso aplicativo web Ruby. Vamos pedir a GeneXus que gere. F5...



Vemos que GeneXus nos informa sobre as tabelas que serão criadas na base de dados na nuvem. Em nosso caso, será a tabela sobre a qual trabalhará a transação Property. Pedimos a ele que a crie.

Depois ele o fará e, logo após gerar os programas ruby para nossas transações (neste caso uma Property), terminando por subí-los à nuvem e abrir-nos no navegador o menu de desenvolvimento que cria automaticamente para que possamos executar nossos objetos e testá-los. Ele é conhecido como Developer Menu.



Se você já desenvolveu em GeneXus para Web, pode pular esta parte e prosseguir para o vídeo seguinte.

Registremos como exemplo uma propriedade.

Podemos ver a interface web que criou automaticamente GeneXus.

The screenshot shows a web browser window with the address bar displaying 'apps.genexus.com/Id6de08c0ac1bb45d88559f672d0:'. The page has a red header with the text 'Application Header'. Below the header, there is a 'Recents: Property' section. The main content area is titled 'Property' and contains a form with the following fields: 'Id' (text input with value '0'), 'Name' (text input), 'Listing Date' (date picker with value '06/13/11'), and 'Address' (text area). Below these fields is a 'Front Image' section with a 'Choose File' button and the text 'No file chosen'. The 'Operation' section has a dropdown menu with 'Sale' selected. At the bottom of the form are three buttons: 'Confirm', 'Cancel', and 'Delete'. The footer of the page is red and contains the text 'Footer Info'.

Não atribuímos id já que se autonumerará.

Se tentamos sair deste campo...recebemos a advertencia que o deixamos vazio. Registremos um nome.

Aqui podemos ver como a data de entrada tem o valor pré-determinado de hoje, devido à regra default que havíamos especificado. Poderíamos mudá-la ou deixar a que estava.

Se queremos sair do campo de endereço sem especificar um valor, não é permitido devido à regra error. Registremos um endereço.

Agora selecionemos a foto da fachada da casa.

E por último escolhamos a operação.

Confirmamos.

Depois, podemos verificar que a propriedade ficou efetivamente registrada no sistema.

Application Header

Recents: Property


Property

Id:

Name:

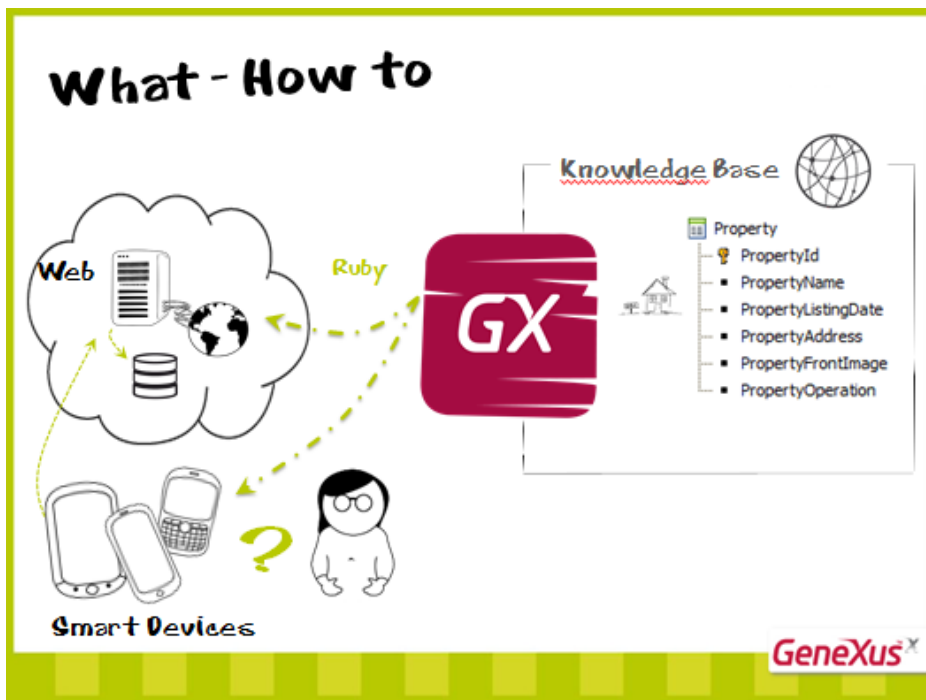
Listing Date:

Address:

Front Image: 

Choose File: Pty.0387.bmp

Operation:



Por este caminho, continuaríamos desenvolvendo todo o backend web. Mas...e o backend para os dispositivos móveis?

Que tipo de aplicativo estamos imaginando para o Smart Device?

Será o tema do próximo vídeo

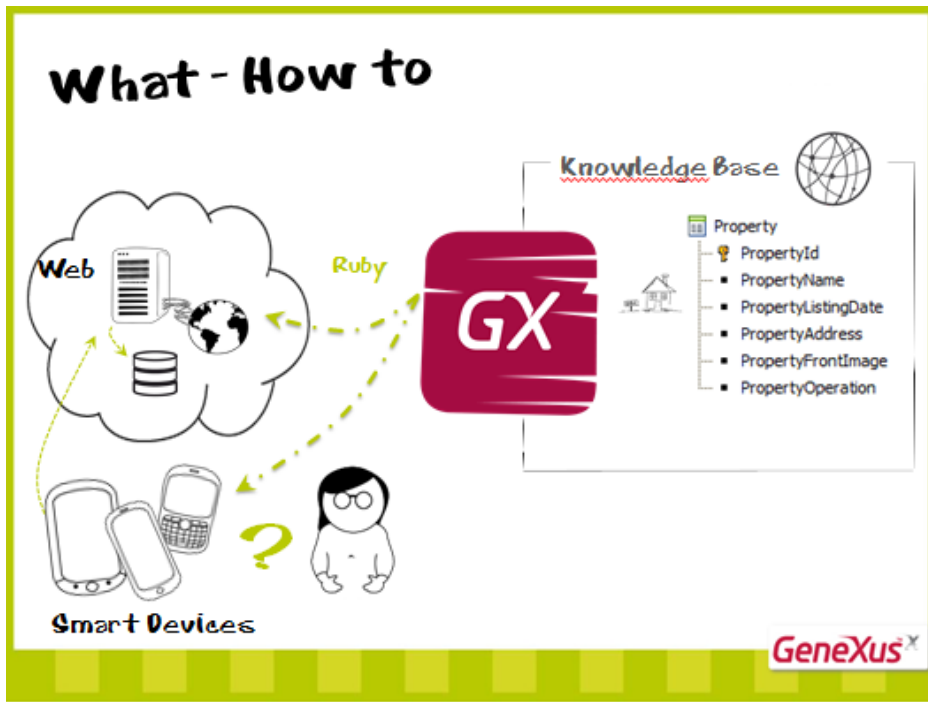
Terceiro vídeo



Em vídeos anteriores...

Havíamos definido nosso problema: construir um backend para uma imobiliária: tanto web como para Smart Devices.

Para isso, havíamos começado criando uma KB, a transação Property, e havíamos definido que o backend web será gerado em ruby, na nuvem, conforme havíamos pedido a GeneXus que construísse o aplicativo, e havíamos registrado uma propriedade no sistema através do navegador.

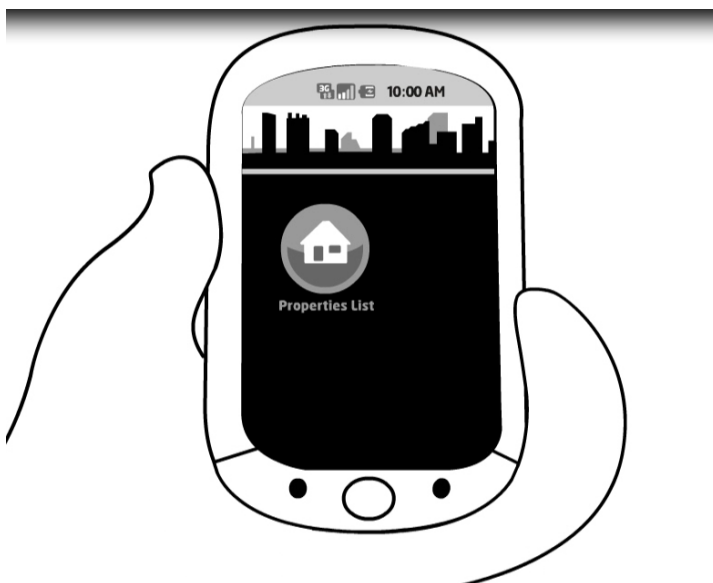


Por esse caminho, continuaríamos desenvolvendo todo o backend web. Mas...e o backend para os dispositivos móveis?

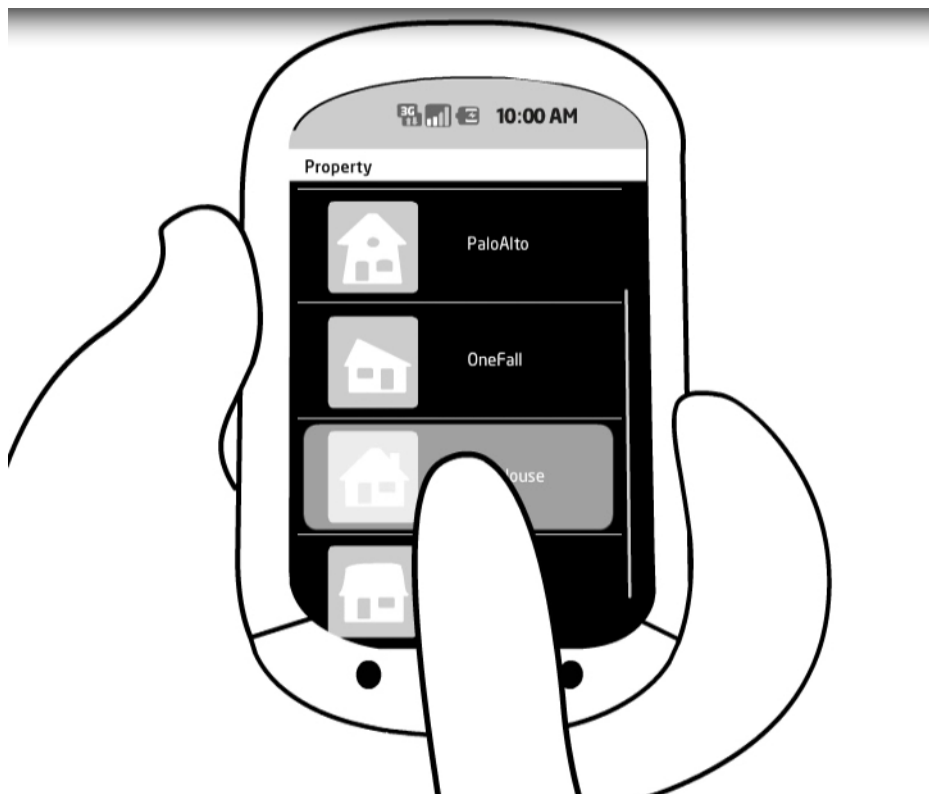
Que tipo de aplicativo estamos imaginando para o Smart Device?

Em uma que será usada pelos agentes imobiliários.

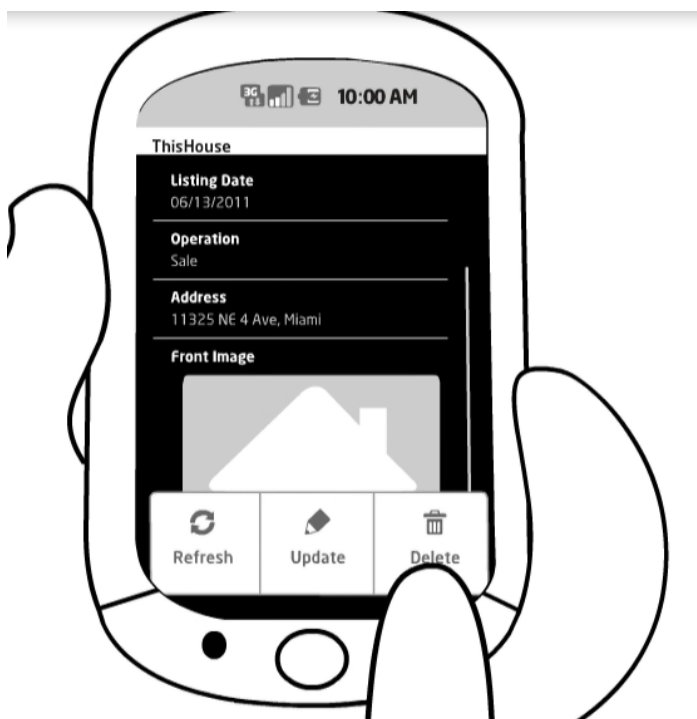
Começando pelo mais simples...



Necessita ver a lista de propriedades...



Escolha ver a informação detalhada de uma propriedade da lista...



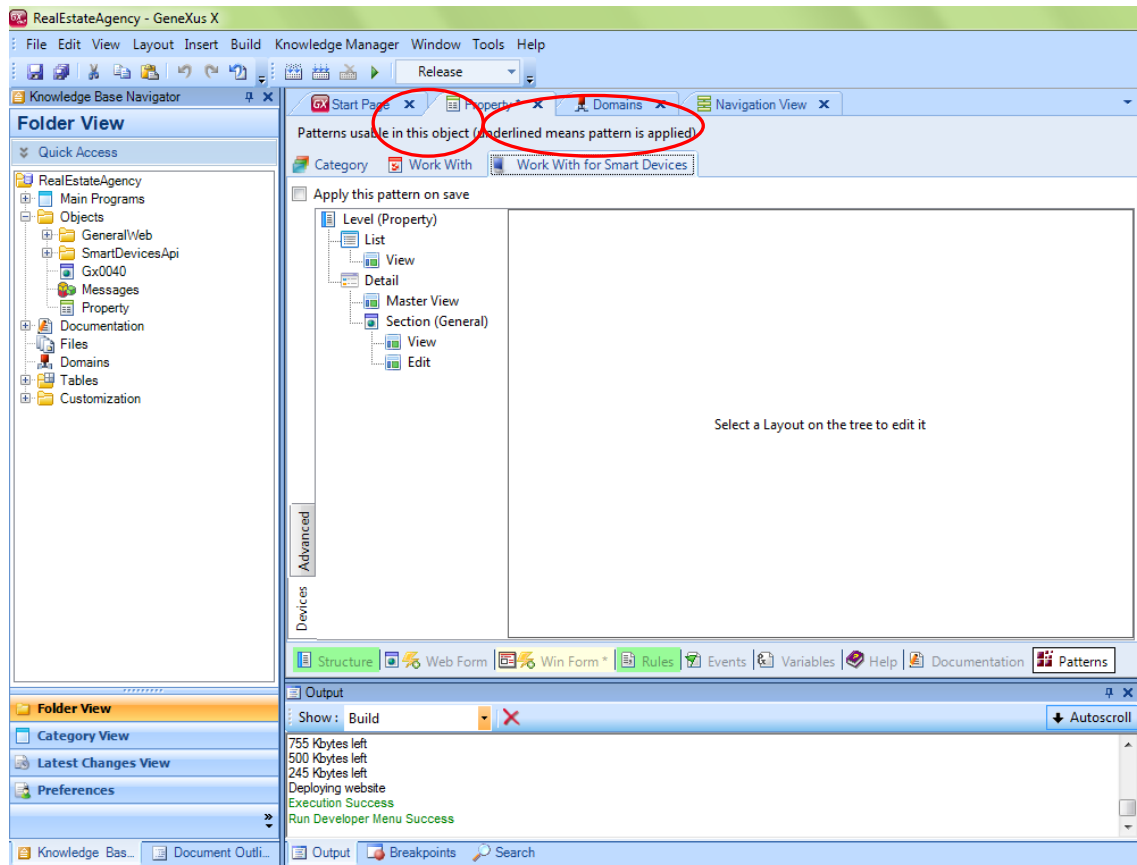
Da qual pode precisar, eventualmente, modificar alguma informação...



Como implementamos o painel para trabalhar com propriedades?

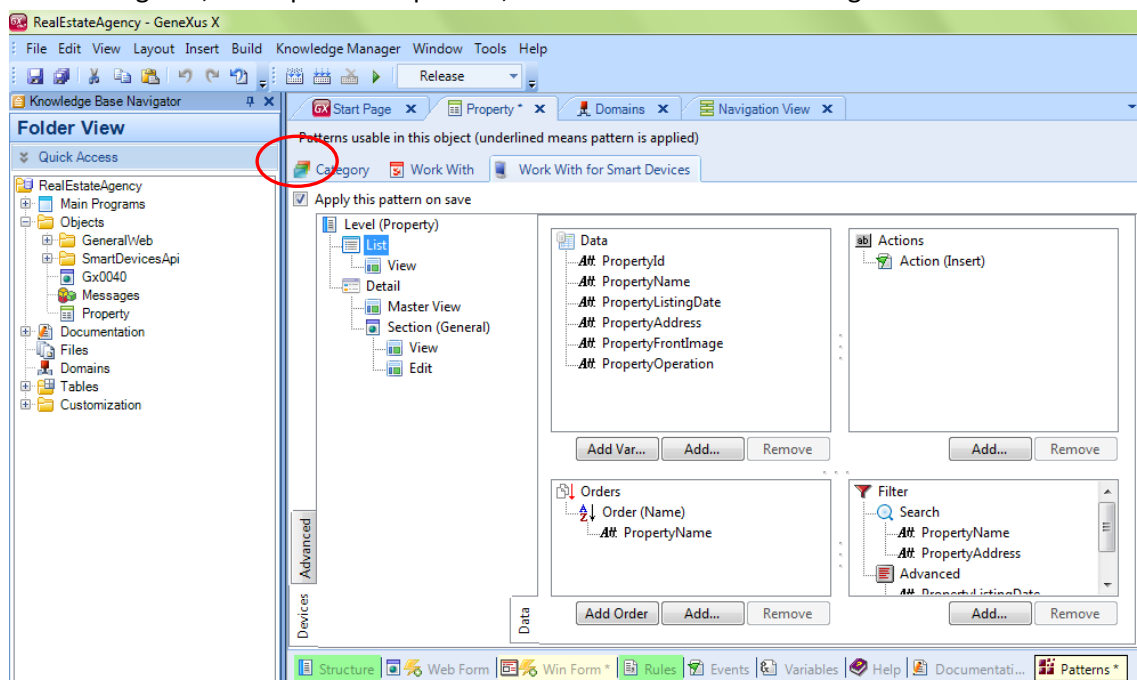
Na transação, vamos à seção Patterns para escolher os padrões que podem ser aplicados à transação, o “Trabalhar com”...esta entidade, ou seja, trabalhar com propriedades...mas não

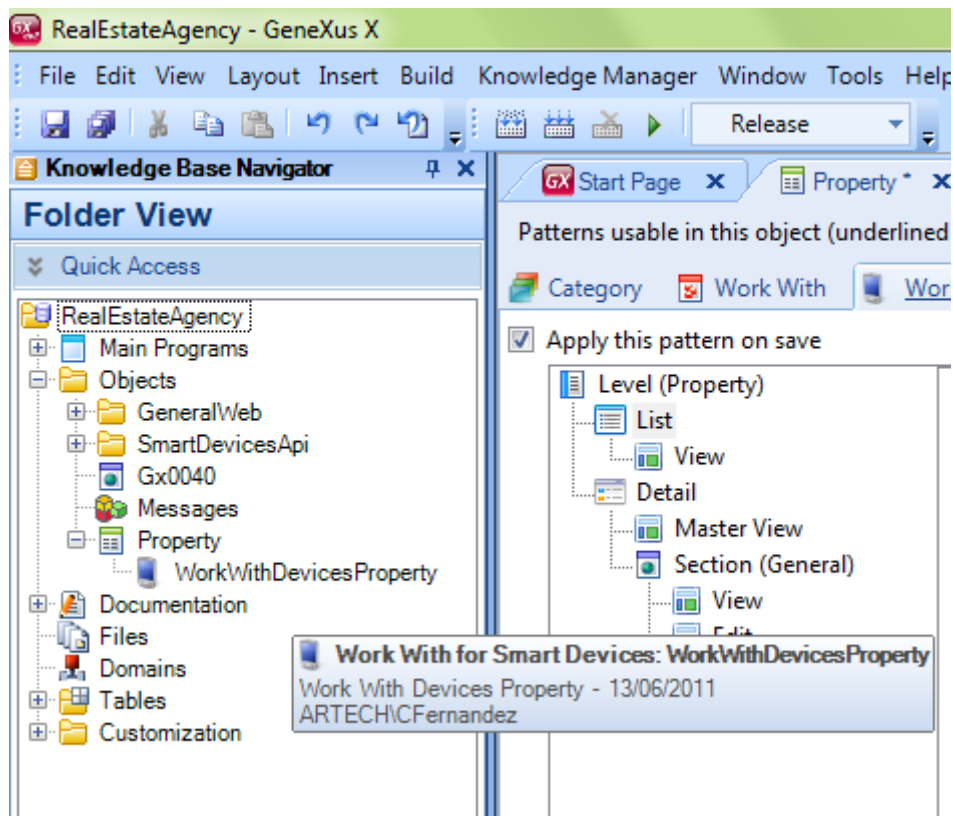
para web, sim para Smart Devices.



Este padrão implementará a lista de “Propriedades Imobiliárias” e para cada propriedade, seu detalhe.

Para consegui-lo, como para todo pattern, basta marcar o check box e gravar.





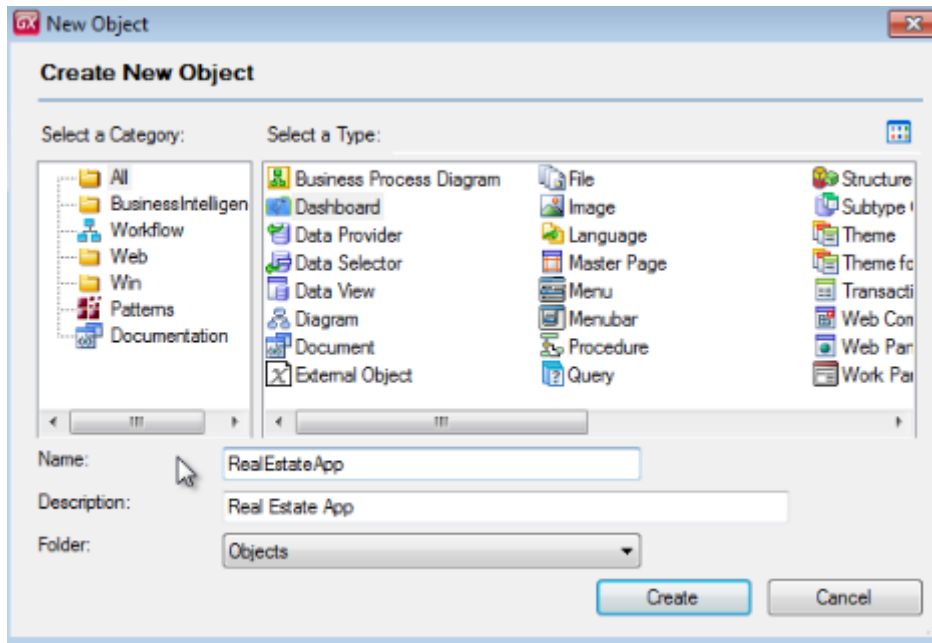
Aqui podemos ver que está criado

O que nos falta para gerar e experimentar nosso aplicativo para Smart Devices?

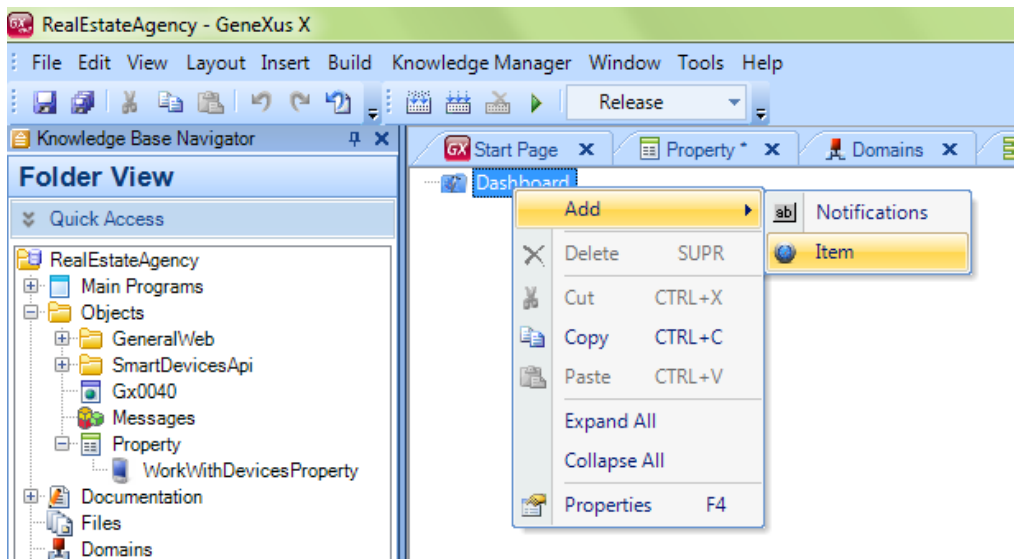
Fazer um menu ou ponto de entrada (para web, GeneXus criava-o automaticamente: era o Developer Menu):

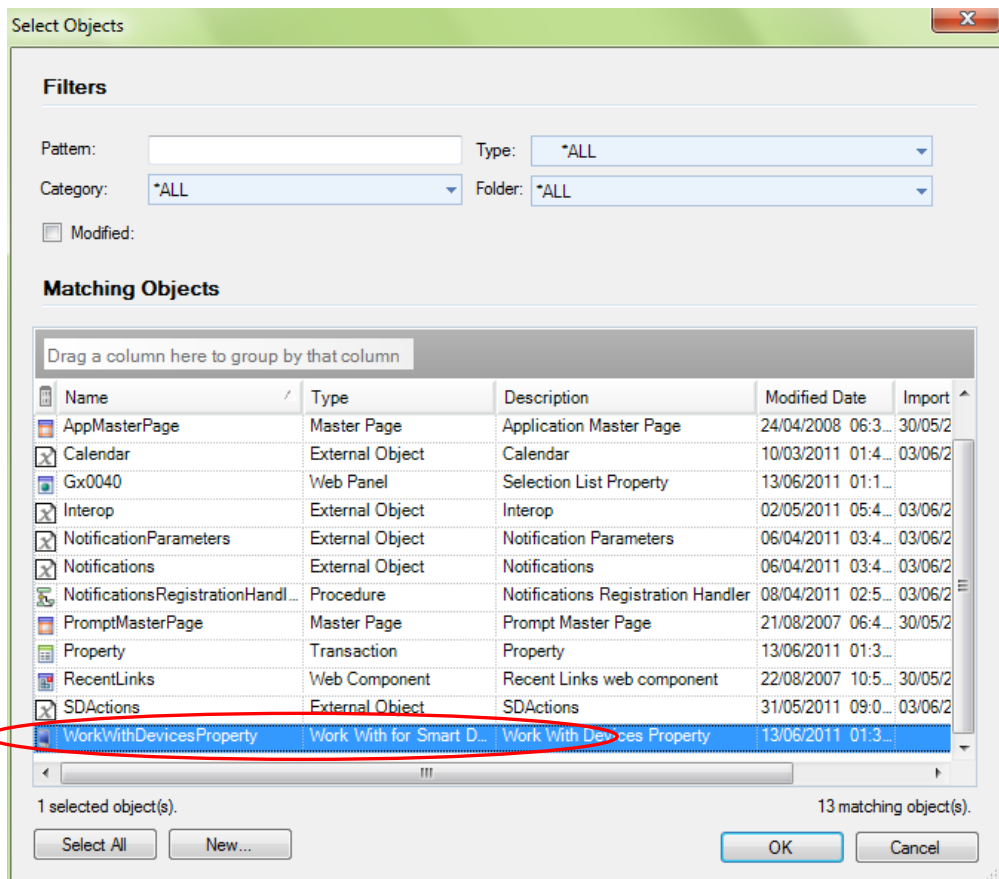


Para fazer este menu de entrada para Smart Devices, criamos um objeto de tipo Dashboard.

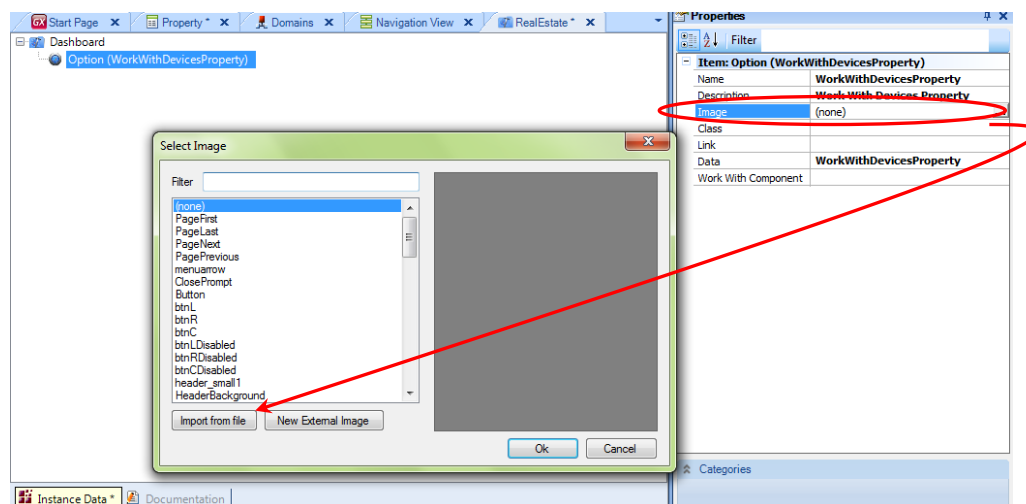


Agregamos um ítem para invocar o “Trabalhar com propriedades” que criamos antes...

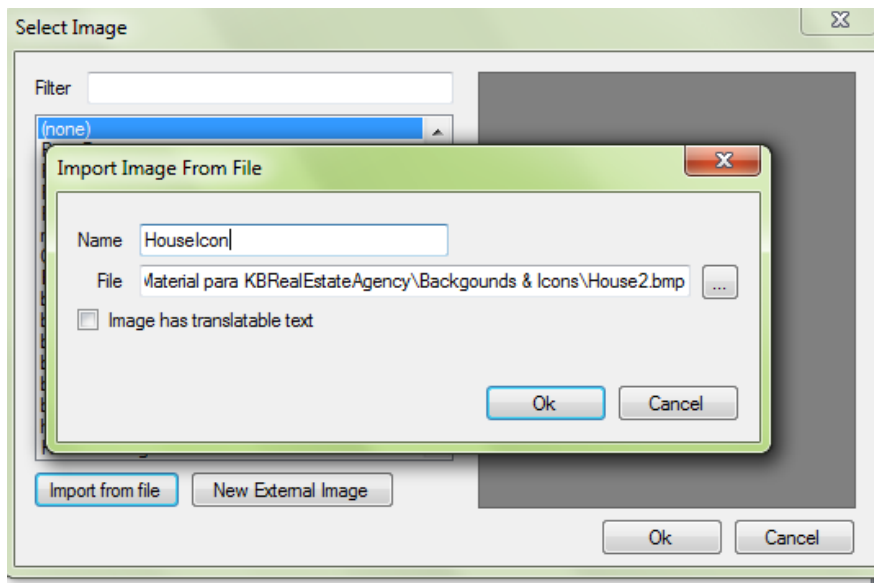




E configuramos suas propriedades, por exemplo, se queremos que apareça o ícone com uma imagem personalizada ao **fazer tap** (sobre ele no dispositivo).

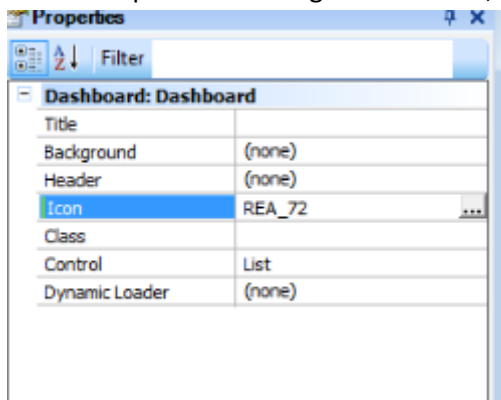


Importamos a imagen de um arquivo...



Podemos mudar a descrição

E também podemos configurar o header, o ícone do dashboard...



Ao chegar neste ponto, podemos dizer a GeneXus que crie os programas. Simplesmente F5...

Para qual dispositivo? Como não modificamos nada, será para a opção automática: Android

Podemos ver que GeneXus nos abre simulador de Android, para poder testar o aplicativo se não temos um dispositivo para fazê-lo.

Aqui vemos a propriedade, que já havíamos registrado...

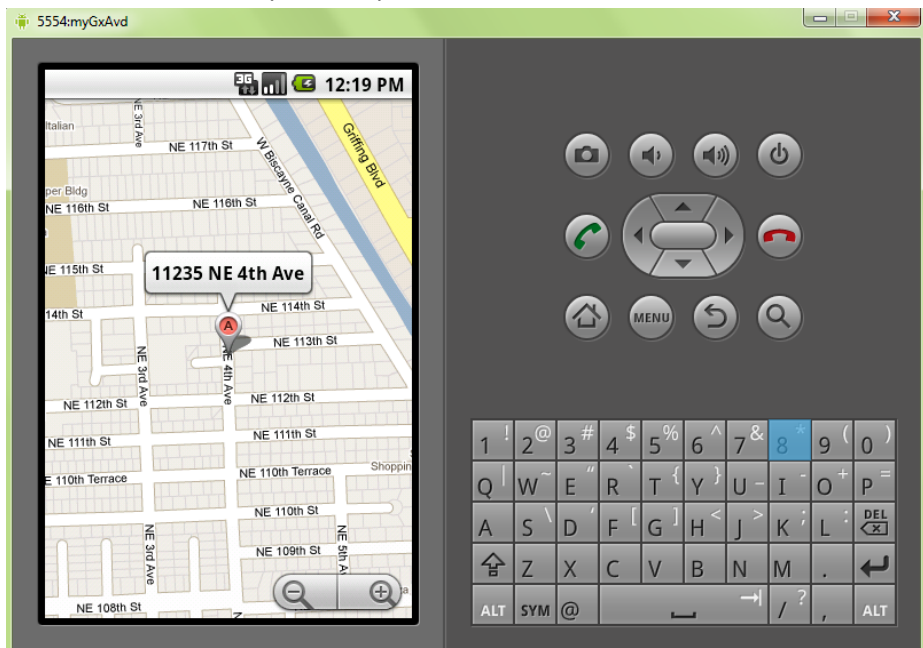
Mas por sua vez, o developer menu abre no navegador e terá também os links para executar o backend web, um QRCode e um link de Download relativos ao aplicativo para Smart Devices. Para quê? Para poder carregar e instalar no nosso dispositivo o aplicativo compilado e experimentá-lo lá, diretamente, em vez de utilizar o simulador.



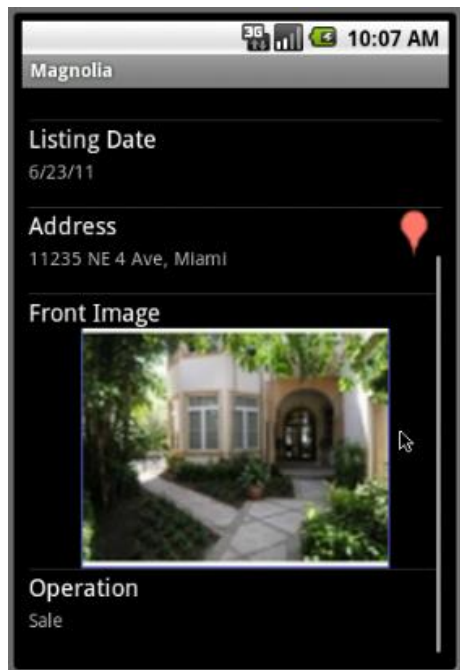
Enviando esta URL a quem desejamos que teste o aplicativo, com o programa para leitura de QR Codes que tenha instalado, será possível baixar e instalar o aplicativo.

Observemos que aparece listada uma só propriedade, aquela que havíamos inserido antes através do Developer Menu web. Se damos **tap** sobre a mesma...nos mostra seu detalhe.

Como o endereço corresponde a um atributo de domínio Address, o aplicativo nos oferece visualizá-lo em um mapa do dispositivo.

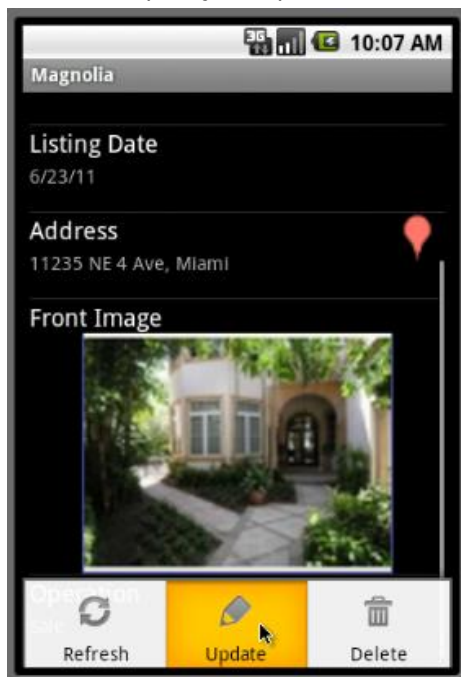


A imagem está aparecendo porque o aplicativo interpreta este tipo de dado e o mostra corretamente no dispositivo.



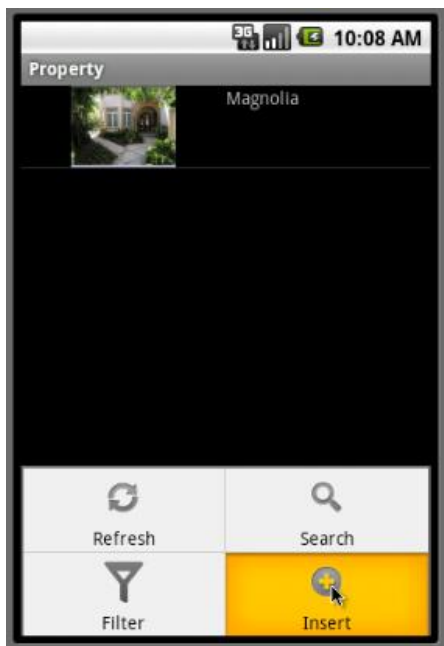
O agente imobiliário poderia querer modificar algum destes dados, por exemplo,

Modificar Operação, e para isso editaria esta informação



...Muda a operação por Rent e Grava.

Ou poderia até estar visitando uma propriedade para registrá-la no conjunto de propriedades que maneja....



Observemos que, como o identificador é autonumerado, não está esperando que seja registrado um valor.


Property Information

Id
0

Name

Listing Date
6/13/11

Address

Front Image


Operation

Registramos o nome.

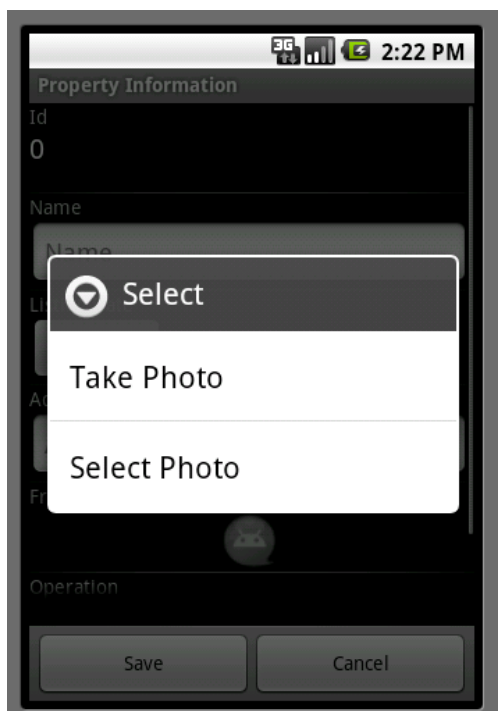
Listing Date assumirá o valor automaticamente da data de hoje (por causa da regra Default que declaramos na transação). Por isso, não temos motivos para registrar valor.

Mas, se desejamos, como é do tipo Date, o aplicativo tem a capacidade de entender seu significado e aparece o “Date Picker”, uma aplicação do dispositivo para escolher facilmente uma data...

Agora, registremos o endereço:

O aplicativo está entendendo que a informação seguinte é uma imagem, por isso nos oferece a possibilidade de tirar uma foto com o dispositivo neste momento, ou seleccionar uma das fotos armazenadas.

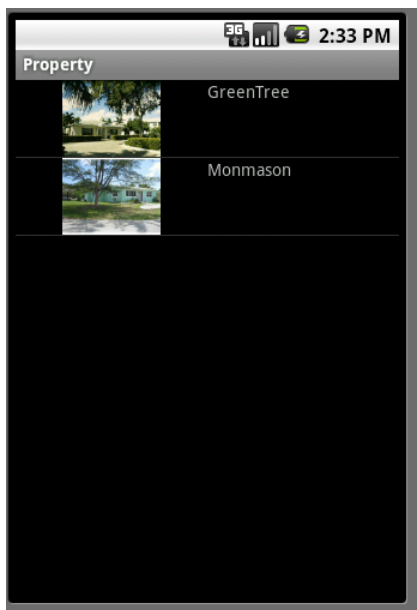
Tiremos a foto da propriedade que temos na frente...



Depois deixamos o campo seguinte, Operation, com o valor Sale.

Gravemos.

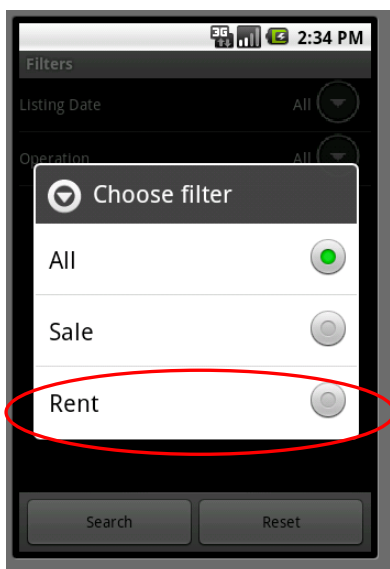
Agora vemos a nova propriedade na lista de propriedades.



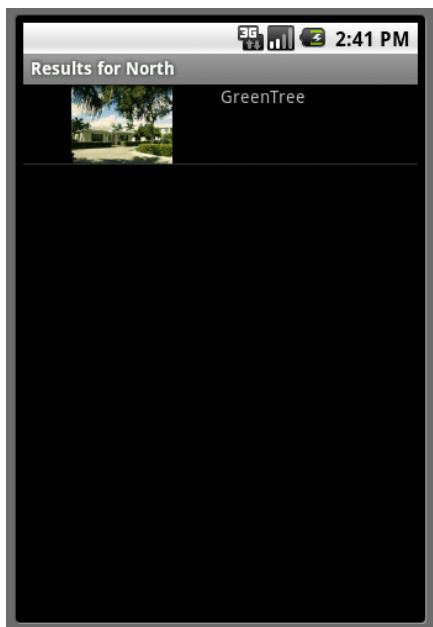
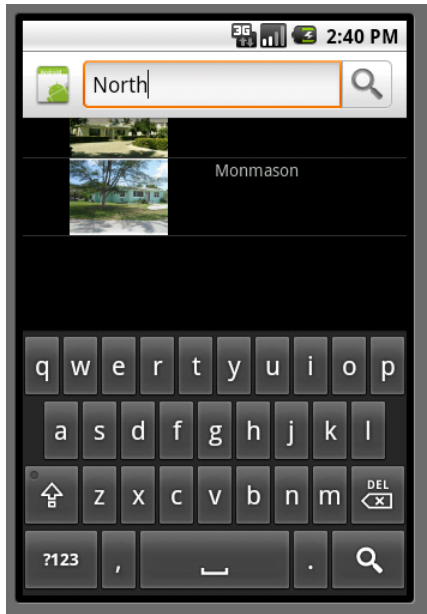
Da mesma forma que no backend web, as regras de negócio que especificamos aparecem. Por exemplo, registrar o endereço era obrigatório. Observemos que se queremos eliminá-la...

...não nos é permitido.

Podemos querer filtrar a lista pela operação de aluguel...

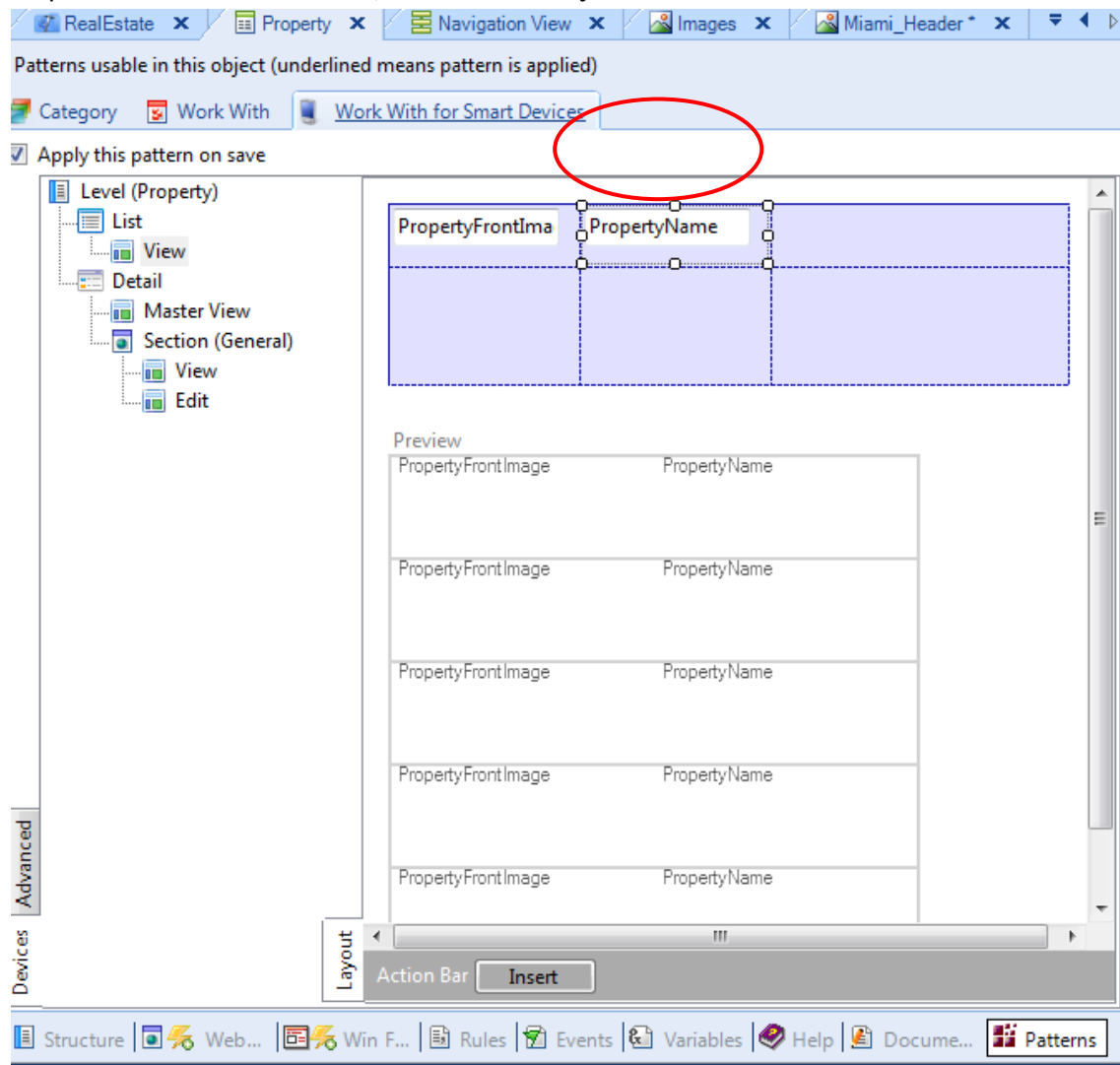


Ou voltar e fazer uma pesquisa mais ampla, por exemplo, por endereço....

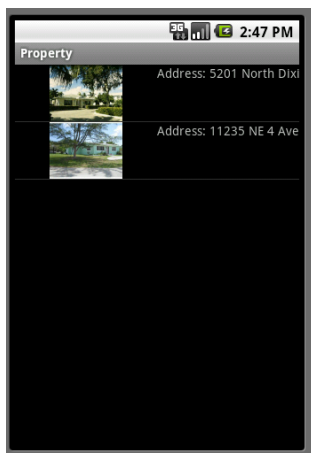


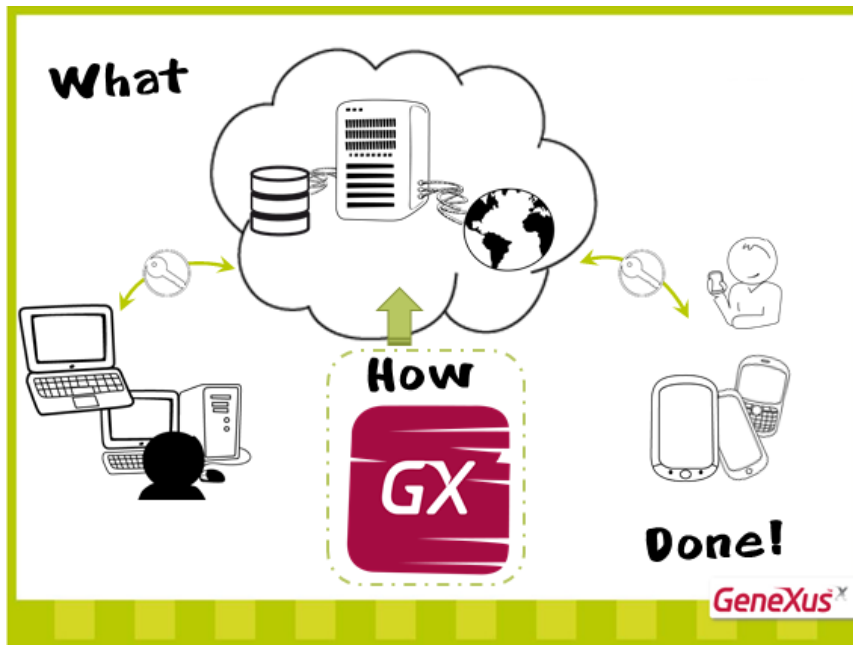
Que informação está aparecendo na lista? A imagem e o nome da propriedade. Poderíamos desejar que saísse o endereço no lugar ou além do nome. Esta e outras personalizações são realizadas facilmente em GeneXus.

Em “trabalhar com para Smart Devices”, vou à seção que implementa o Layout da Lista de Propriedades...e além do nome, coloco o endereço...



Gravo e com o F5, peço a GeneXus que gere novamente o aplicativo.





Com isso, vemos uma pequena amostra de como é fácil personalizar o aplicativo.

Obtivemos assim o que nos havíamos proposto, uma parte do backend Web e outra funcionando em Smart Device...



Com passos simples, podemos pedir a GeneXus que gere o mesmo aplicativo, mas para dispositivos com sistema operativo iOS ou para Blackberry, da mesma maneira que podemos pedir-lhe que o aplicativo Web seja em Java ou em .Net no lugar de Ruby.

Some real life examples...

GeneXus[®]

Vejamos como alguns exemplos de aplicativos da vida real...

What else?



To be continued...

GeneXus[®]

E muito mais:

Quer agregar segurança?

...mais entidades interrelacionadas?

Desejaria personalizar o que se mostra, as interfaces, ordem das informações, buscas?

Necessita adicionar ações, como agendar visitas no calendário?...ou um cliente na agenda de endereços do dispositivo? Ou necessita agregar ações, como agendar visitas no calendário? ...ou um cliente na agenda de endereços do dispositivo? Ou talvez enviar mensagens?

Continuará...

