## Como implementar um menu de entrada para a aplicação móvel (How to create an input menu for the mobile application)

Neste vídeo veremos como implementamos um menu de entrada para a aplicação, deslizável, que oferece duas opções: ir para o painel que mostra a lista de Materiais, que implementamos nos primeiros dois vídeos, ou ir a outro painel, analogamente, que mostra a lista de clientes e permite ver a informação detalhada de qualquer um destes.

A implementação em GeneXus dos painéis para trabalhar e visualizar os clientes, é semelhante aos painéis havíamos desenvolvido para os materiais.

## Aqui vemos o Work With:

GX Sta	rt Page 🗙 🖪 WWSDCustomers 🗙 🕞 ViewSDCustomer 🗙 🖛	
-	Application Bar	l
	&ErrorWarning_m	
	GRID	
	&Customers.item	
	ID &Customers.item	
Style		1
IJ.	<b></b>	
Lay	Any Platform, Default Orientations 👻 🕄 Add Layout 😮 Delete Layout	
😺 Lay	yout 🗟 Rules 😰 Events 🚩 Conditions 🕼 Variables 🙆 Documentation	

De onde chamaremos um procedimento:



que é o que realizará uma chamada RFC à Bapi do SAP que devolve a lista de clientes:



Para a qual teremos que já ter importada essa Bapi no GeneXus.

Se formos no módulo Enterprise de onde estávamos armazenando os objetos importados do ERP, vemos que criamos duas folders, para organizar a informação:



Na do nome Material está o que já tínhamos relativo aos materiais, e no segundo, os tipos de dados estruturados e o objeto externo que importamos para trabalhar com os clientes.

🐼 Start Page 🗴 🕞 WWSDCustomers 🗴 🕞 ViewSDCu	stomer 🗴 🚡 SAPCustomerGETLIST	× 🔀 sapCusto	mer 🗙 👻
Structure	Туре	Is Collection	Description
□ 🗶 sapCustomer			Customer
🖨 🚰 Properties			
	Character(10)		Customer Number
😑 =😡 Methods			
Er=↓ GETDETAIL2	None		Customer Details
	Character(8)		CompanyCode
	BAPICUSTOMER_04, Enterprise		CustomerAddress
	BAPICUSTOMER_KNA1, Enterprise		CustomerGeneralDetail
	BAPICUSTOMER_05, Enterprise		CustomerCompanyDetail
	BAPIRET1, Enterprise		Return
	BAPICUSTOMER_02, Enterprise	$\checkmark$	CustomerBankDetail
	None		List with Addresses
	Character(2)		CPDOnly
MAXROWS	Character(4)		MaxRows
RETURN	BAPIRETURN1, Enterprise		Return
	BAPICUSTOMER_IDRANGE, Enterprise	~	IdRange
ADDRESSDATA	BAPICUSTOMER_ADDRESSDATA, En	~	AddressData
SPECIALDATA	BAPICUSTOMER_SPECIALDATA, Ent	$\checkmark$	SpecialData
🖇 Events			
Structure 🔗 Help 📓 Documentation			

Vemos que são importado dois métodos: o GETLIST para obter a lista de clientes, é o que já havia sido utilizado neste procedimento. E o GETDETAIL2, será utilizado para obter a informação de um cliente em particular:

Este método, como já imaginam será utilizado no procedimento chamado pelo painel ViewSDCustomer:

GX Sta	rt Page 🗙 🕫 WWSDCustomers 🗙 🖪 ViewSDCustomer 🗙 🖾 SAPCustomerG 🔻	< ►
G	&ErrorWarning_m	
	Customer &Customer.Custc	
	Name &Customer.Name	
	City &Customer.City	
	Postal Code &Customer.Posta	
	Region &Customer.Regio	=
	Street &Customer.Street	
	Phone &Customer.Phon	
	Country &Customer.Coun	
	Language &Customer.Langu	
Style	Transport Zone &Customer.Trans	
out		-
	Any Platform, Default Orientations 🗸 🚯 Add Layout 😢 Delete Layout	
😺 La	yout 🗟 Rules 🗑 Events 🝸 Conditions 🚱 Variables 📓 Documentation	

Start Page 🗙	ⓑ WWSDCustomers 🗴 ⓑ ViewSDCustomer × ಔ SAPCustomerGETLIST × ∑sapCustomer ×	-
	Events	-
1 - Eve	ent Refresh	1
2	SAPCustomerGETDETAIL ( & CustomerId, & Customer, & Messages )	-
3 🛓	If &Messages.Count = 0	=
4	<pre>&amp;ErrorWarning_msg.Visible = False</pre>	Â
5	else	
6	&ErrorWarning_msg.Visible = True	
7 🛱	for &Messages_Item in &Messages	
8	<pre>&amp;ErrorWarning_msg = &amp;ErrorWarning_msg + ' ' + &amp;Messages_Item.Description</pre>	
9 -	endfor	
10 -	endif	
11 - End	levent	
12		
13		-
	III	•
😺 Layout 🗟 R	ules 😰 Events 🝸 Conditions 🛍 Variables 📓 Documentation	

Através de uma variável, por requerer ser instanciado com um identificador de cliente:

🕫 ViewSDCustomer 🗴 🚋 SAPCustomerGETLIST 🗙 🔀 sapCustomer 🗙 🚋 SAPCustomerGETDETAIL* 🗙 📼 4
Subroutines
1 SAPSessionManagerConnect( &SAPSessionManager, &Messages )
2
3 □ If & SAPSessionManager.ErrorCode.IsEmpty()
4 //success
5 SABLUSTOMER.CUSIOMERNO = SCUSTOMEDADDRESS SCUSTOMEDADDRESS SCUSTOMEDADDRESS
asabcustomer.GetDetATL2 (acomparitobe, acostomerabbress, acostomerabbress, acostomerabbress, acostomerabbress)
9 if SETIEN, NUMBER, ISEMPLY()
10 //success
11 & Customer.CustomerId = & CustomerId
12 & Customer.Name = & CUSTOMERADDRESS.NAME
13 &Customer.City = &CUSTOMERADDRESS.CITY
14 &Customer.PostalCode = &CUSTOMERADDRESS.POSTL_CODE
15 &Customer.Region = &CUSTOMERADDRESS.REGION
16 δCustomer.Street = δCUSTOMERADDRESS.STREET
17 &Customer.Phone = &CUSTOMERADDRESS.TELEPHONE
18 δCustomer.Country = δCUSTOMERADDRESS.COUNTRY
19 &Customer.Language = &CUSTOMERADDRESS.LANGU_ISO
20 &Customer.TransportZone = &CUSTOMERADDRESS.TRANSPZONE
21
22 else
23 & Messages_Item.Id = & RETURN.NUMBER.ToString()
24 SMessages_Item.Description = fRetOwn.Message.Irim()
25 Messages_item.ivpe = Messageiypes.trror
20 anessages.Add(anessages_ttem)
2/ = EIULI
🐱 Source * 😺 Layout 🗟 Rules 🝸 Conditions 🚯 Variables 🥏 Help 🙆 Documentation

Agora vejamos como implementamos o menu deslizável que vemos em execução.

Se trata do painel que chamamos do SAPAppMenu. No layout temos incluído um controle text block, cuja caption é Welcome:

GX	Start Page 🗙 🕫 WWSDMaterials 🗙 🕫 WWSDCustomers 🗙 🐻 SAPAppMenu 🗴 📼	Properties	<b>д X</b>
	- Application Bar	Participation of the second secon	
		textblock: Textblock	d
		Control Name	Textblock1
		Caption	Welcome
		Auto Grow	False
	GRID	<ul> <li>Appearance</li> </ul>	
	&OptionName	Class	TextBlock
		Visible	True
		Invisible Mode	Keep Space
		Enabled	True
		Cell information	
		Row Span	1
		Col Span	1
		Horizontal Alignment	Default
		Vertical Alignment	Default
	j		
		- Catagorian	, ,
		Categories	
l 😤			
N N			
Ħ			
နို	Any Platform Default Orientations - Add Layout O Delete Layout		
-	Any Flation, Default Orentations • 😈 Add Layout 😈 Defete Layout		
V	Layout 🗟 Rules 😰 Events 🍸 Conditions 🚯 Variables 👫 Documentation		

E um grid, que contém dois controles correspondentes as variáveis de nomes OptionIcon e OptionName. A primeira do tipo Imagem, e a segunda do tipo Character.

🐼 Start Page 🗙 🕞 WWSDMate	rials 🗙 🛅 WWSDCustomers 🔉	K 🗊 SAPAppMenu 🗙	-
Name	Туре	Is Collection	Description
🖃 🚷 Variables			
🕀 🚱 Standard Variables			
····   OptionIcon	Image		Option Icon
• OptionName	Character(20)		Option Name

Este grid mostrará as opções do menu, com suas imagens correspondentes.

		<sup>36</sup> 3:49
E 🚺 SA	APApp Menu	
Welcome		
Я	Customers	
\$	Materials	

E deverá permitir que ao selecionar uma linha (ou seja, uma das opções), seja chamado o objeto correspondente.

Page

O primeiro, então, é carregar esse grid com as imagens e nomes das opções. Para isso utilizamos o evento Load, que é do grid e é executado justamente quando este é carregado:



São atribuídos os valores para cada linha do grid, e é carregada. Observe que cada imagem já está previamente importada na base de conhecimento.

Todas as imagens podem ser acessadas através do nó Images abaixo de Customization, na folder View:



Aqui podemos criar novas, ou filtrar por nome, e até mesmo ter uma visualização/preview da imagem:

PageD

GX Start F	Page 🗙 🛅 WWS	DMaterials 🗙 🛅 WWS	DCustomers 🗙 🛅 SAPAppl	Menu 🗙	🔏 Images	x
Filter		🛛 🖬 🕶 🔛 New Image	Advanced 🗸			
Name		Description	Last Update	<b>_</b>	Preview	
ta 👘	ActionDelete	Action Delete	7/20/2015 11:50 AM			
1	ActionDeleteDisa	Action Delete Disabled	7/20/2015 11:50 AM			
	ActionDisabled	Action Disabled	7/20/2015 11:50 AM	=		
1	ActionDisplay	Action Display	7/20/2015 11:50 AM			
1	Action Display Dis	Action Display Disabled	7/20/2015 11:50 AM			
	ActionExport	Action Export	7/20/2015 11:50 AM			
1	ActionInsert	Action Insert	7/20/2015 11:50 AM			
	Action Update	Action Update	7/20/2015 11:50 AM			
1	Action Update Dis	Action Update Disabled	7/20/2015 11:50 AM			×
	asc	asc	7/20/2015 11:50 AM			
	AudioDownload	Audio Download	7/20/2015 11:50 AM			
1	BigButton	Big Button	7/20/2015 11:50 AM			
	blank	blank	7/20/2015 11:50 AM			
1	btn_cancel	btn_cancel	7/20/2015 11:50 AM			
1	btnC	btn C	7/20/2015 11:50 AM			
1	btnCDisabled	btn CDisabled	7/20/2015 11:50 AM			
	btnL	btn L	7/20/2015 11:50 AM			
	btnLDisabled	btn LDisabled	7/20/2015 11:50 AM			
	btnR	btn R	7/20/2015 11:50 AM			
	btnRDisabled	btn RDisabled	7/20/2015 11:50 AM			
	Button	Button	7/20/2015 11:50 AM			
	cabezal_cuerpo	cabezal_cuerpo	7/20/2015 11:50 AM			
	calendar	calendar	7/20/2015 11:50 AM			
1	Cancel	Cancel	7/20/2015 11:50 AM	-	Strech	

Nós já importamos nesta base de conhecimento as duas imagens que chamamos Customers\_icon\_sd, Materials\_icon\_sd e outra que usaremos no seguinte vídeo, Sales\_order\_icon\_sd.

Se filtramos pela primeira, não vemos a imagem pois seu desenho é na cor branca e aqui o fundo é branco:

GX Start Page	🗙 🕞 WWSDMaterials 🗙 🕞 WM	/SDCustomers 🗙 🛅 SAPAppMe	nu 🗙 🖾 Images 🗙 👻
Filter cus	🔢 👻 New Image	e 🛛 Advanced 🛩	
Name	Description	Last Update	Preview
Cu	stomers_icon Customers_icon_sd	8/13/2015 3:15 PM	
N 1			
1			
			Strech

Observa-se que se escolhermos criar uma nova imagem, é aberto este Wizard que nos permite importar uma imagem de nosso computador, da rede ou inclusive da web:

New Image Wizard	
Welcome to the New Image Wizard This wizard helps you to create new image objects.	$\underline{}$
Create Image(s) from File(s) Select this option if source images are stored in files i or network.	in your computer
Create External Image Select this option if you need to reference someone usually on the Web (i.e. an image on GeneXus Web)	else's image, site)
Cancel < Back Next >	Finish

Voltemos a nosso menu.

Já carregamos com o Load as duas linhas correspondentes a nossa duas opções. Agora teremos que programar qual painel será chamado ao escolher cada uma das linhas. Para isso,

Page

Properties **μ**Χ 🐼 Start Page 🗴 🔚 WWSDMaterials 🗴 📠 WWSDCustomers 🗴 📠 SAPAppMenu 🗙 🔏 Images 🗴 -Ad ↓ | Filter Grid: Grid1 Control Name Grid1 Welcome Collection Default Action 'Select' GRID ult &OptionName Selection Type Platform Default Selection Flag Selection Flag Field Spec Default Selected Item La (none) - Control Info Control Type Auto Grow False - Appearance Grid Class Visible True Invisible Mode Keep Space Enabled True <default> Style Rows Empty Grid Backgrour (none) Empty Grid Backgrour Image Layou Empty Grid Text 📗 Any Platform, Default Orientations 🝷 🛟 Add Layout 😢 Delete Layout Empty Grid Text Class TextBlock 😺 Layout 🗟 Rules 🕱 Events 🝸 Conditions 🛍 Variables 🙆 Documentation + Data Selector

vejamos que a Default Action do grid, que corresponde à ação produzida ao usuário dar tap sobre uma linha, a temos associada um evento que chamamos de 'Select':

E se olharmos seu código:

GX Start Page 🗙	🕼 WWSDMaterials 🗙 🕼 WWSDCustomers 🗙 🕼 SAPAppMenu 🗙 🖓 Images 🗙	*
	'Select'	-
1 🗆 Eve	ent Load	15
2	<pre>&amp;OptionIcon.FromImage(Customers_icon_sd)</pre>	
3	&OptionName = 'Customers'	-
4	Load	
5		
6	<pre>&amp;OptionIcon.FromImage(Materials_icon_sd)</pre>	
7	&OptionName = 'Materials'	
8	Load	
9 - End	levent	
10		
11 🖂 Eve	ent 'Select'	
12	Composite	=
13	if &OptionName = 'Customers'	
14	WWSDCustomers()	
15	else	
16	<pre>if &amp;OptionName = 'Materials'</pre>	
17	WWSDMaterials()	
18 -	endif	
19 -	endif	
20 -	Endcomposite	
21 - End	levent	-
14		P.
🖬 Layout 🗟 R	ules 😰 Events 🍸 Conditions 🕲 Variables 🙆 Documentation	

É verificado se OptionName da linha que foi escolhida é 'Customers' então é chamado o Work With de clientes que mostramos no princípio, ou se for Materials é chamado o Work With de Materials.

O comando Composite é utilizado nos eventos do cliente (o Load é do sistema). Não entraremos aqui nestes detalhes. Para isso vá ao curso de desenvolvimento de aplicações móveis para Smart Devices com GeneXus.

Queremos que este painel seja o Main da aplicação, e seu ponto de entrada. Para isso, primeiro de tudo o configuramos como Main:

Page

Properties	д	×
A ↓   Filter		
Panel for Smart Dev	rices: SAPAppMenu	~
Name	SAPAppMenu	
Description	SAPApp Menu	
Module/Folder	Root Module	
Qualified Name	SAPAppMenu	
Object Visibility	Public	
Main program	True	
Caption	Your App for SAP	=
Share		
<ul> <li>Miscellaneous</li> </ul>		
Generate Object	True	
<ul> <li>Network</li> </ul>		
Connectivity Support	t Online	
<ul> <li>Main object propertie</li> </ul>	es	
Application Title		
Default Layout Orien	Default	
Enable Ads	False	
Facebook App Id		
Facebook App URL S	ι	
Twitter Consumer Ke	r	
Twitter Consumer Se	9	
Bing Maps Key	AvHhO1IwVBJa-uYU-SxTau	
Enable Notifications	False	
Enable Analytics	Eplag	

E mudaremos a propriedade Main ao Work With de Materiais.

Por outro lado, podemos já especificar que este painel será nosso objeto de inicialização da aplicação. Para isso, cliquemos no botão direito sobre o painel, Set As Startup Object.



Deste modo, para executar a aplicação, já não teremos que posicionar-nos no painel e clicar com o botão direito/run, mas sim simplesmente ao pressionar F5, já compilará a aplicação, e compilada acionará o emulador com este objeto.

Testemos: F5.

Aqui não está aparecendo como menu deslizável. Por que? Porque não é o valor padrão para uma aplicação Android para telefone.

O que devemos fazer é ir no Folder View, abaixo do nó SD Platforms, escolher a que nos interesse e modificar; neste caso Android Phone e mudar a propriedade Navigation Style para "Slide".



Testemos agora :F5.

Agora sim.

No próximo vídeo, adicionaremos uma nova opção para trabalhar com as Sales Orders:



A diferença entre trabalhar com Materials e com Customers, é que aqui veremos como criar a partir da aplicação móvel uma nova sale order no ERP.

Page L

Com isso terminaremos nossa demo.



 ${\sf Page}{11}$