

Device Resources



Semantic Domains & Data Types

Smart Devices APIs

Os aplicativos para dispositivos móveis irão precisar integrar-se com recursos do dispositivo, tanto físicos como lógicos.

Por exemplo, eles precisam ser capazes de fazer chamadas telefônicas, enviar mensagens através de programas de chat ou email instalados, acessar e usar o aplicativo de contatos, a câmera, o gps, além de aplicativos de mapas, além de interagir com o Facebook e Twitter .

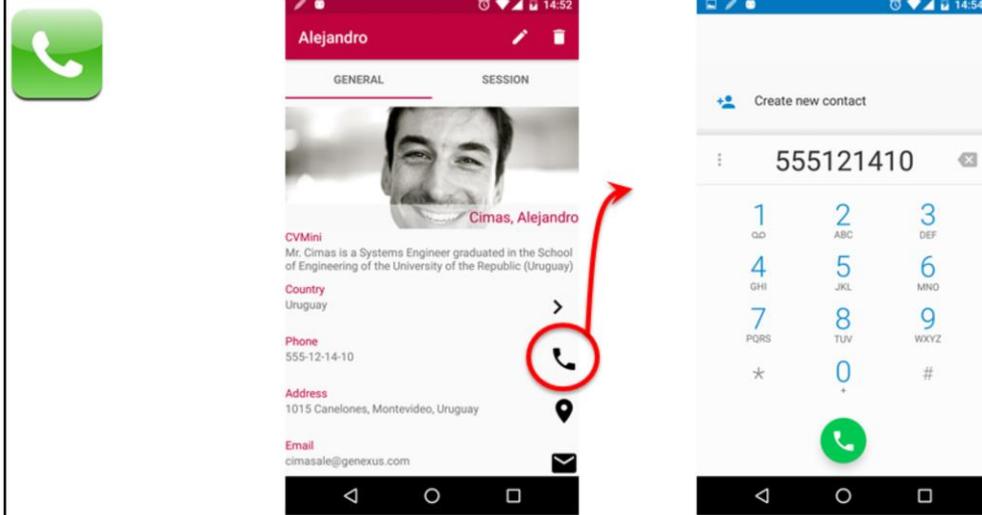
Algumas dessas funcionalidades foram implementadas automaticamente através de domínios semânticos ou tipos de dados com semântica ... Como por exemplo ...

Semantic Domains

Name	Type
Speaker	Speaker
SpeakerId	Id
SpeakerName	Name
SpeakerSurname	Surname
SpeakerFullName	VarChar(60)
SpeakerImage	Image
SpeakerCVMini	VarChar(1K)
CountryId	Id
CountryName	Name
SpeakerPhone	Phone, GeneXus
SpeakerAddress	Address, GeneXus
SpeakerEmail	Email, GeneXus

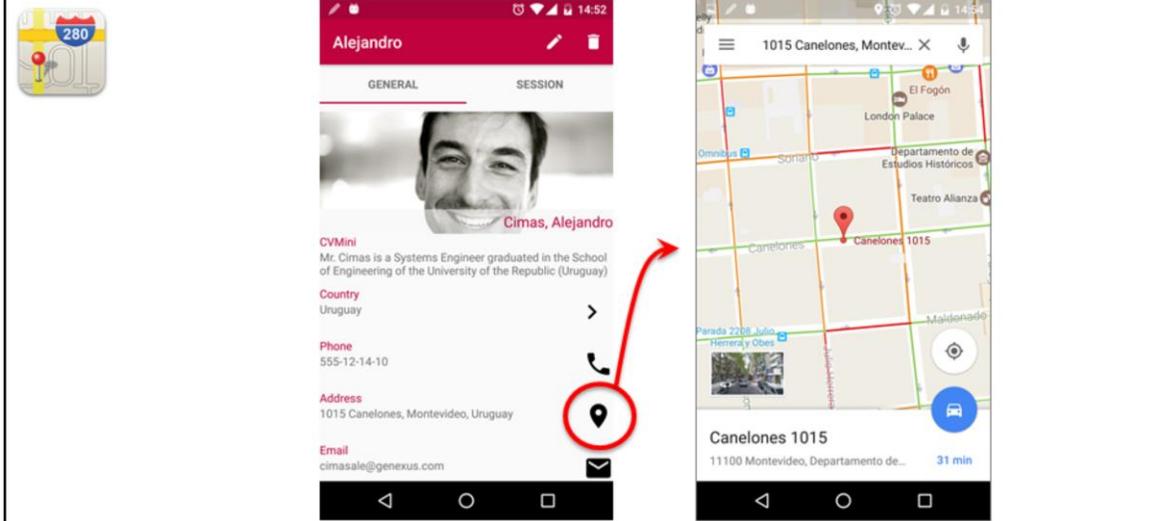
Os domínios pré-definidos Phone, Address, Email.

Semantic Domain



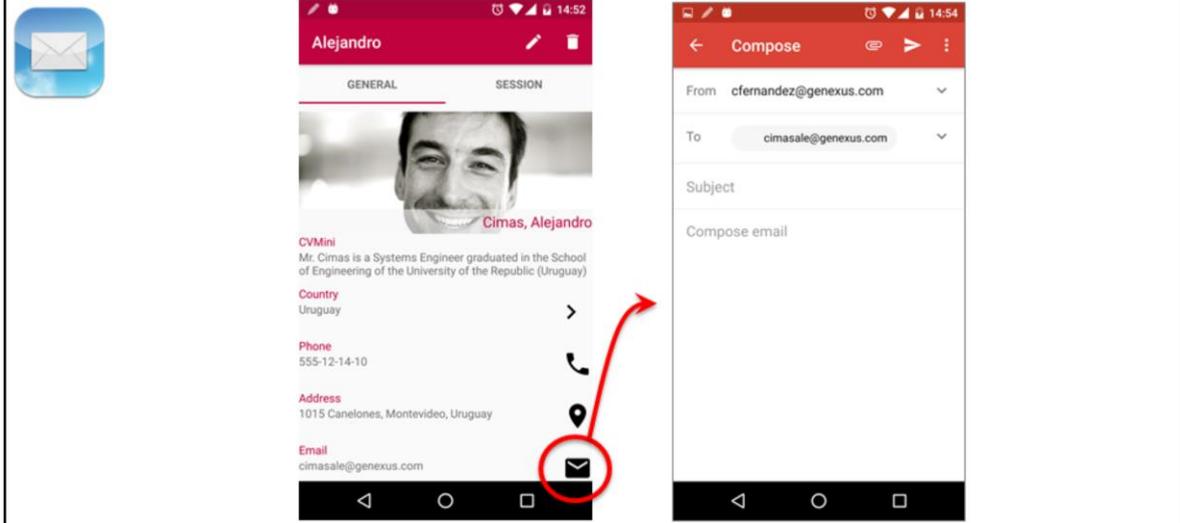
Associando o domínio semântico Phone ao atributo SpeakerPhone, se, no Detail de um orador damos tap no campo phone, vai abrir o aplicativo instalado no dispositivo para fazer a ligação telefônica.

Semantic Domain



Por outro lado, se formos no campo Address e tocamos no endereço, os mapas instalados no dispositivo são abertos, permitindo ver, então, o endereço no mapa.

Semantic Domain



E, finalmente, se tocamos no campo Email, do domínio semântico Email, abrimos o aplicativo instalado no dispositivo para escrever e enviar um e-mail para o orador correspondente.

Device Resources

GeneXus™

Semantic Data Types

Name	Type
Speaker	Speaker
SpeakerId	Id
SpeakerName	Name
SpeakerSurname	Surname
SpeakerFullName	VarChar(60)
SpeakerImage	Image
SpeakerCVMini	VarChar(1K)
CountryId	Id
CountryName	Name
SpeakerPhone	Phone, GeneXus
SpeakerAddress	Address, GeneXus
SpeakerEmail	Email, GeneXus

Alejandro

Id
4

Full Name
Cimas, Alejandro

Name
Alejandro

Surname
Cimas

Image

CVMini
Mr. Cimas is a Systems Engineer graduated i

Country Id
5

Country Name
Uruguay

Select

Take a Photo

Select an Image...

Por outro lado, apenas tendo o tipo de dados image associado a um atributo, isso fará com que o Detail, quando editarmos as informações, nos ofereça o controle correspondente para tirar uma foto usando a câmera do dispositivo. Ou ainda, isso nos permitirá selecionar uma imagem da galeria de imagens no dispositivo.

Device Resources
GeneXus™

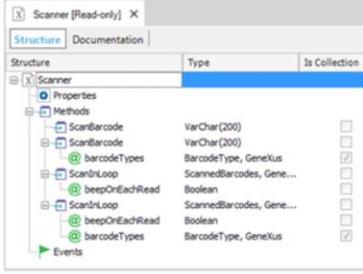
Control type

Smart Devices APIs




Att/var: Varchar(200)
Control Type: SD Scanner

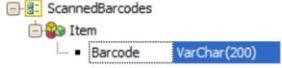




```

Event &code.Tap
...
&code = Sacanner.ScanBarcode()
ShowProduct( &code)
...
endevent

```



Nós também podemos usar a câmera para verificar Qrcodes ou códigos de barras.

Por exemplo, neste aplicativo desenvolvido para uma cadeia de supermercados, podemos escanear o código de barras do produto que temos em casa e que já acabou, buscá-lo no aplicativo e ver o preço atual ou adicioná-lo ao carrinho de compras.

Para implementar isso, temos algumas opções: ter um atributo ou variável de tipo caractere ou varchar, alterando seu tipo de controle para SD Scanner, quando o controle é de entrada, ele irá adicionar automaticamente um botão Scan, que ao ser pressionado, abrirá a aplicação do leitor de código de barras instalado no dispositivo.

A outra alternativa será programar um evento (que poderia ser o Tap na variável), onde usaremos o método ScanBarcode de uma API fornecida pelo GeneXus própria pra isso. É um objeto externo que fornece propriedades, métodos e eventos para abstrair sua implementação e fornecer funcionalidades.

Aqui estaríamos chamando um método que executará o programa de leitor de código de barras instalado no dispositivo, que por sua vez, usará a câmera. O valor lido será devolvido pelo método e então podemos chamar um Panel for Smart devices que mostre todas as informações desse produto.

Em alguns casos, as APIs precisarão trabalhar com tipos de dados estruturados, que virão definidos automaticamente no GeneXus, juntamente com elas. Por exemplo, para poder ler uma série de códigos de barra um após o outro, para processá-los mais tarde, existe uma coleção SDT e um método ScanInLoop.