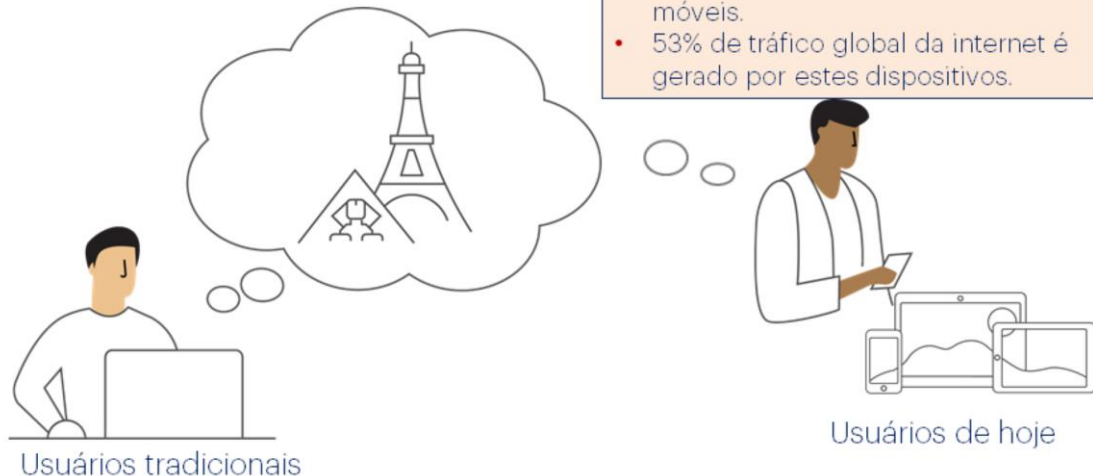


Introdução ao desenvolvimento de aplicações para dispositivos móveis

GeneXus 16

Necessidades dos usuários: Ter acesso de qualquer lugar.

Até poucos anos os usuários usavam nossas aplicações de um PC porém hoje em dia os usuários demandam por poder realizar suas tarefas com o dispositivo que tenham à mão nesse momento, no escritório seguiram utilizando um PC porém enquanto estão à caminho, no transporte fazem o trabalho utilizando dispositivos menores como celulares ou tablets ou inclusive relógios inteligentes, uma vez em casa podem utilizar até mesmo um televisor inteligente.

Todos estes dispositivos nos fornecem uma capacidade de produção ao usuário que antes era impossível, segundo publicações estima-se que os usuários passam mais de 5 horas com seus dispositivos móveis e mais de 53% do tráfego de internet global é gerado atualmente destes dispositivos.

Porém... porquê necessitamos uma aplicação para Smart Devices se já contamos com uma web na qual os usuários podem acessar a partir de seus dispositivos já que são até Responsivas?

Neste vídeo vamos tratar de responder a esta pergunta, adiante vamos ver as características mais importantes das aplicações que podemos desenvolver com GeneXus e ao finalizar vamos ver uma aplicação e como foi desenvolvida.

Aplicações Web Responsivas

Prós



- Disponíveis para todos os usuários.
- Fáceis de atualizar e manter.
- Custos menores de desenvolvimento



Contras



- Sempre necessitam de conexão.
- Forma de navegação não intuitiva.
- Sem acesso às funcionalidades do dispositivo
- Menos compromisso do usuário.

Primeiro vejamos as aplicações Web Responsive.

As vantagens deste tipo de aplicação é que

- estão disponíveis para todos os usuários
- são fáceis de manter, sobretudo com GeneXus
- e os custos de desenvolvimento são menores.

Por outro lado suas desvantagens são que

- somente podem trabalhar Online, sempre é necessário ter uma conexão,
- a forma de navegação não é natural, não é nativa do dispositivo já que usa um browser,
- tampouco teremos acesso a certas características dos dispositivos, como por exemplo a câmera, o GPS
- e por último podemos mencionar geram muito menos compromisso do usuário com a aplicação, isto por que não é uma aplicação instalada no dispositivo.

Aplicações nativas (Smart Devices)

Prós



- Nem sempre necessitam de conexão.
- Funcionalidades específicas do dispositivo.
- Melhor experiência de usuário.
- Compromisso.



Contras



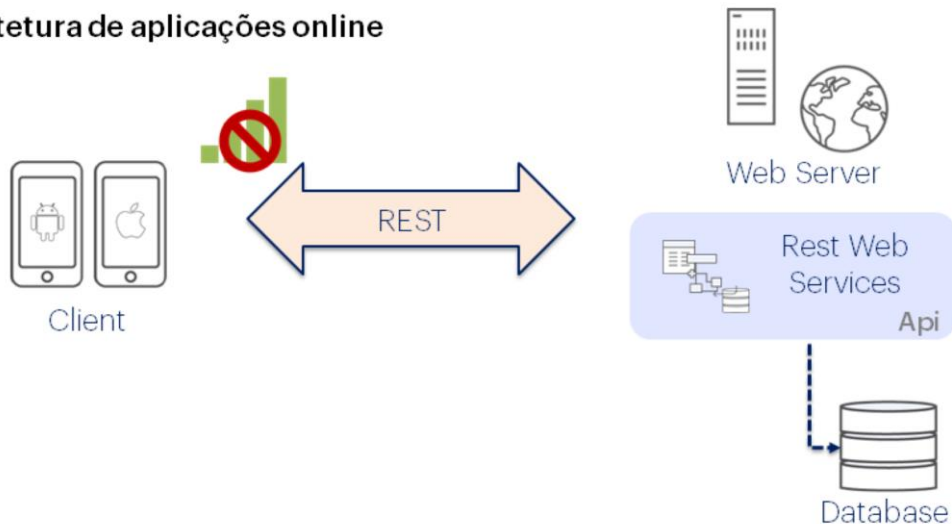
- Desenvolver para várias plataformas.
- Custos de atualização e manutenção.
- Requerem instalação.

A respeito das aplicações nativas para Smart Devices

As vantagens são que nem sempre necessitam uma conexão com Internet, podemos ter aplicações offline (parcialmente conectadas ou totalmente desconectadas), podemos usar as características próprias do hardware como a câmera, o gps, leitor de digitais, tudo de forma natural, e isto nos dará uma melhor experiência do usuário, além de que como estas aplicações devem ser instaladas isso gera uma maior compromisso do usuário.

Do lado negativo temos que é necessário desenvolver para múltiplas plataformas (Android, iOS) algo que resolve GeneXus, temos maiores custos de atualização e manutenção e como já mencionamos, requerem instalação.

Arquitetura de aplicações online



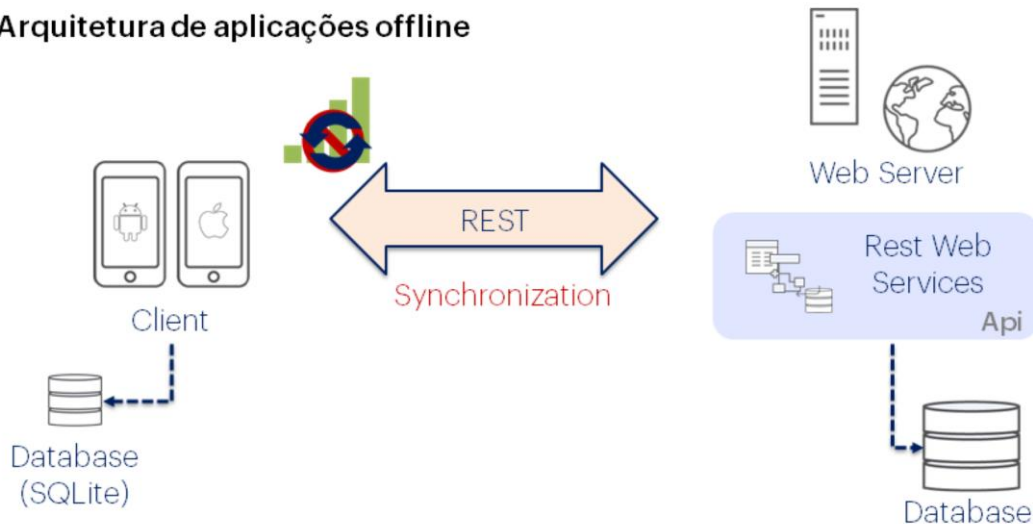
Neste diagrama podemos ver qual é a arquitetura das aplicações Online desenvolvidas com GeneXus.

Por um lado teremos uma aplicação na linguagem nativa que será instalada nos dispositivos, seja este Android ou iOS e serão os clientes dentro da arquitetura.

No lado do servidor web vamos ter uma camada de serviços web que usam o protocolo Rest, estes serviços ou API que os clientes utilizarão estão compostos por Data Providers, Processos e Business Components aos quais resolvem todo o acesso aos dados que a aplicação necessita.

Sendo uma aplicação Online, quando não contamos com conexão não poderemos utilizar a aplicação já que não poderá acessar o servidor para executar os serviços Rest que fornecem os dados para a aplicação.

Arquitetura de aplicações offline



Nas aplicações Offline do lado cliente além da aplicação propriamente dita vamos contar com uma base de dados SQLite com todas as tabelas que a aplicação necessite.

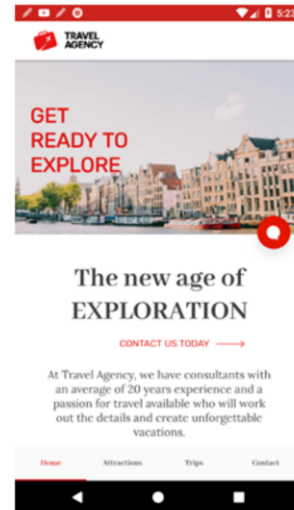
Toda a lógica de acesso aos dados estará na própria aplicação e todas as operações que se realizem sobre os mesmos serão feitas no cliente sobre esta base de dados local.

No servidor web contaremos com o acesso à base de dados centralizada e na camada de serviços Rest contaremos com os serviços de sincronização necessários para prover ao cliente os dados atualizados cada vez que este os requeira. Além disso teremos Business Components utilizados para que o cliente nos informe das mudanças ocorridas nos registros localmente.

Nesse esquema quando não temos conexão, a aplicação pode seguir funcionando normalmente, somente que não contará com a última versão dos dados, os quais são sincronizados automaticamente quando a aplicação detecte que a conexão foi restabelecida.

Aplicações móveis com GeneXus

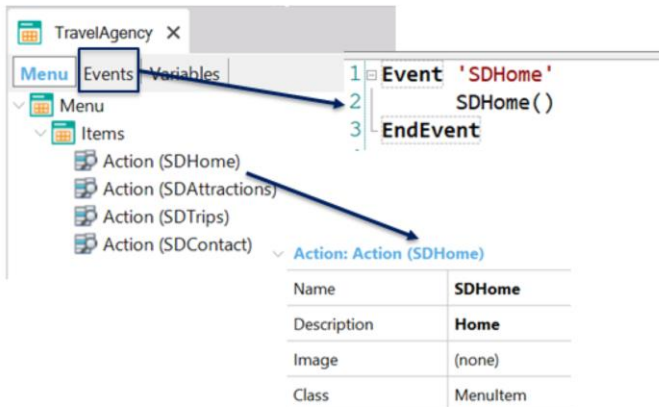
- Objetos específicos
 - Menu for Smart Devices
 - Panel for Smart Devices
 - Theme for Smart Devices
 - Work With for Smart Devices
- Outras funcionalidades
 - Domínios semânticos
 - Integração por meio de APIs
 - Online & Offline
 - GAM
 - Stencils
 - Controles específicos



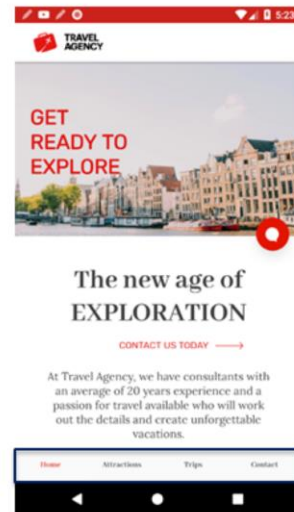
GeneXus nos oferece a possibilidade de poder desenvolver aplicações para Smart Devices, quase da mesma forma que desenvolvemos aplicações Web:

Para isto contaremos com objetos de uso específico para Smart Devices os quais veremos com detalhes na sequência, além de outras características como Domínios Semânticos, Integração por meio de APIs com o hardware do dispositivo, a possibilidade de gerar aplicações conectadas ou desconectadas, segurança através do GAM, uso de Stencils e uma grande variedade de Controles específicos para Smart Devices.

Menu for Smart Devices



Name	SDHome
Description	Home
Image	(none)
Class	MenuItem



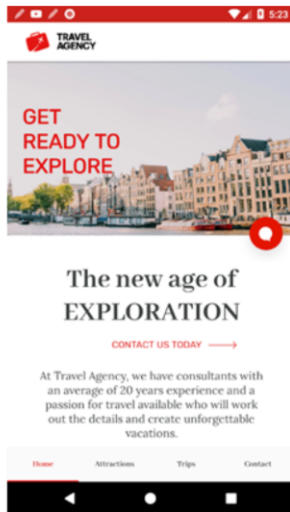
Vamos ver agora os objetos específicos para o desenvolvimento de aplicações para Smart Devices.

O objeto Menu for Smart Devices nos permite definir menus para as aplicações e pode ser mostrado de diferentes maneiras : pode ser como Tabs, como uma Tabela ou como uma Lista. À direita podemos ver a estrutura que o objeto possui, onde podemos definir os Itens que esse menu terá, cada item possui um nome , uma descrição e podemos associar um ícone que é uma imagem da base de conhecimento, um objeto de tipo image, e nos eventos podemos definir o que acontece quando acessamos cada item do Menu.

Normalmente estes objetos serão os pontos de entrada da aplicação e por padrão serão objetos Main.

Em tempo de execução veremos o menu e que neste caso foi utilizado o do tipo de controle Tabs como pode ser visto na parte inferior.

Panel for Smart Devices



- Layout flexível
- Mostrar ou solicitar dados
- Telas complexas (Assistentes com vários passos)
- Menu (Ponto de entrada ao app – Main Object)
- Componentes reutilizáveis
- Mostrar dados devolvidos por Data Providers ou REST Services
- Múltiplos Grids
- E muito mais

Os objetos Panel for Smart Devices são análogos aos Web Panels que usamos no desenvolvimento Web, contam com um Layout abstrato no qual inserimos os controles que necessitemos como tabelas, text blocks, grids, imagens, etc.

Este painel que vemos foi construído com este layout em um objeto Panel for Smart Devices. Sua principal característica é que:

- Seu Layout é totalmente flexível, quando criamos um objeto deste tipo toda a implementação ficará a cargo do desenvolvedor.
Deste modo podemos utilizá-lo para:
- mostrar informação ao usuário ou para que o usuário entre com informação ou mesmo uma combinação de ambas as opções ao mesmo tempo.
- Podemos desenvolver telas complexas, por exemplo para criar um Assistente com vários passos
- Poderíamos usá-lo para criar um Menu já que também pode ser Main e portanto um Entry Point de nossa aplicação.
- Podemos utilizá-lo como componentes reutilizáveis similares a como usamos os Web Components.
- Podemos utilizá-lo para mostrar informação devolvida por Data Providers ou Servicios REST
- Podemos implementar múltiplos Grids
- E muitos outros usos que vocês encontrarão além destes mencionados.

Introdução ao desenvolvimento de aplicações para dispositivos móveis
GeneXus

Theme for Smart Devices

Android

Platform: Any Android

Name	Any Android
OS	Android
Version	
Device Kind	All
Size	All
Theme	CarmineAndroid
Additional Themes	(none)
Navigation Style	Default
Default Layout Orientation	Any
Bounds Name	
Minimum Shortest Bound	0
Maximum Shortest Bound	0

iOS

Platform: Any iOS

Name	Any iOS
OS	iOS
Version	
Device Kind	All
Size	All
Theme	CarminelIOS
Additional Themes	(none)
Navigation Style	Default
Default Layout Orientation	Any
Bounds Name	
Minimum Shortest Bound	0
Maximum Shortest Bound	0
Minimum Longest Bound	0
Maximum Longest Bound	0

Customization

Localization

Themes

☒ Carmine

☒ CarmineSD

☒ CarminelIOS

☒ CarmineAndroid

SD Platforms

Any Platform

☒ Any Android

☒ Android Phone

☒ Android Tablet 7"

☒ Android Tablet 10"

☒ Any iOS

☒ iPad

☒ iPhone

☒ iPhone 3.5"

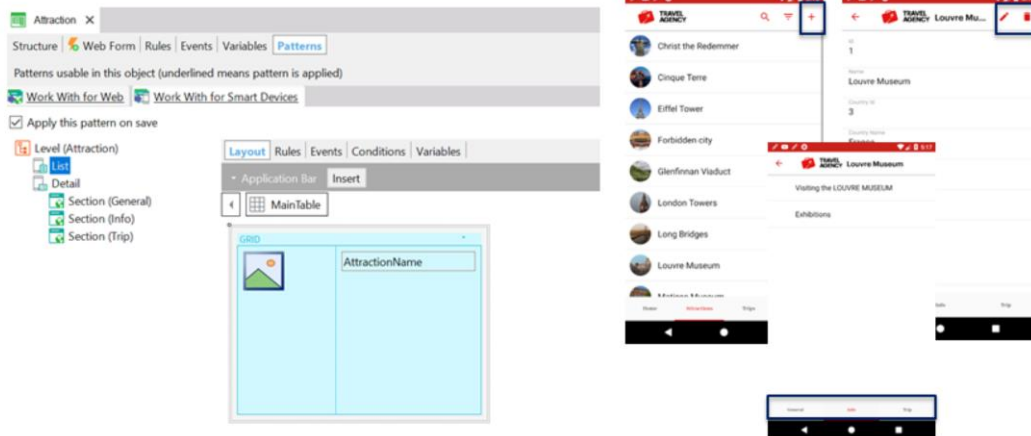
☒ iPhone 4"

☒ iPhone 4.7"

Nossas apps para Smart Devices serão multiplataforma. Ou seja, criamos basicamente a mesma app e faremos o mínimo de trabalho indispensável para personalizá-la de modo que seja executada em dispositivos Android e nos dispositivos iOS, seguindo suas regras de desenho e comportamento específicos.

Para personalizar o Look & Feel contaremos em cada Base de Conhecimento com dois temas já criados, CarminelIOS e CarmineAndroid, e por default se associam estes Themes, um à plataforma Android e outro para a plataforma iOS.

Work With for Smart Devices

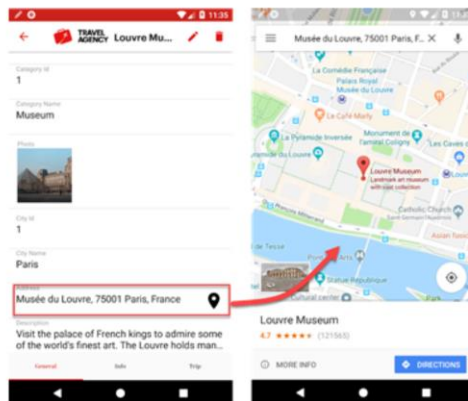


O padrão Work With for Smart Devices é um tipo de objeto específico para o desenvolvimento de aplicações para dispositivos móveis e resolve todas as funcionalidades requeridas para realizar a gestão de dados de uma entidade, como:

- Acessar uma lista,
- o acesso a um item dessa lista para ver em detalhe,
- A entrada de novos registros e sua edição,
- Além de que conta com a inteligência para gerar telas adicionais e links analisando a informação relacionada como subnível, chaves estrangeiras e relações com outras tabelas subordinadas.

Logo veremos uma Demo e usaremos este padrão para que vejam quão fácil e simples que é desenvolver com esta característica.

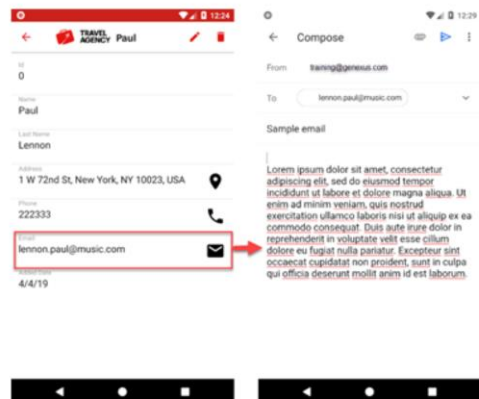
Domínio semântico: Address



Para aproveitar a Integração com o hardware do dispositivo, contamos com os domínios semânticos:

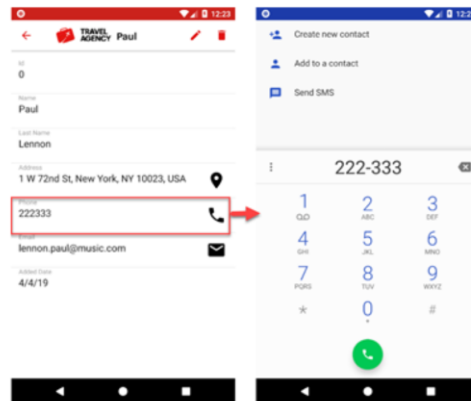
Por exemplo o domínio Address é utilizado para armazenar localizações, em tempo de execução ao dar um Tap nos leva à aplicação pré-determinada de mapas que temos em nosso dispositivo e se posiciona nessa localização.

Domínio semântico: Email



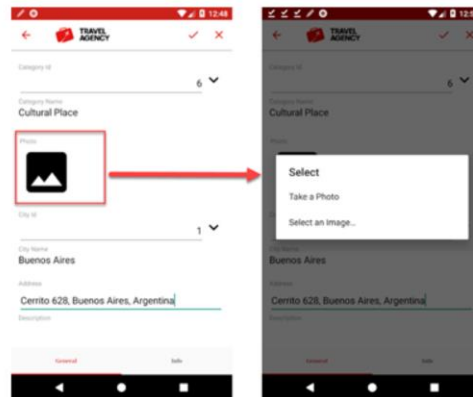
O domínio Email, é similar porém para armazenar endereços de correio eletrônico, e também ao dar um Tap abrirá a aplicação pré-determinada para enviar mensagens de correio eletrônico.

Domínio semântico: Phone



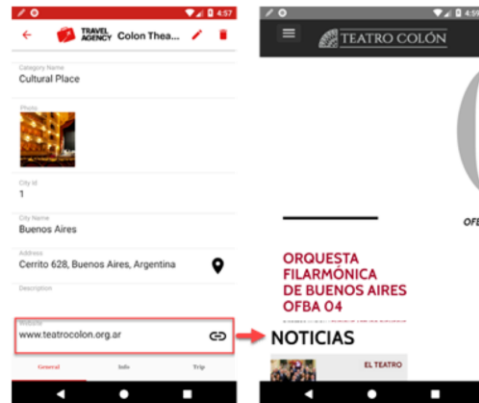
O domínio Phone é utilizado para armazenar números telefônicos e ao dar um Tap abrirá a aplicação de chamada telefônica pré-determinada.

Domínio semântico: Image



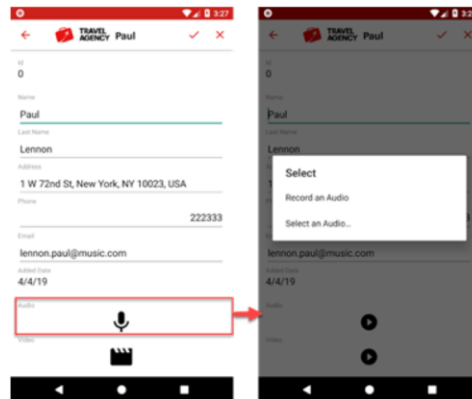
O domínio Image, é utilizado para armazenar imagens, no modo Edição é perguntado ao usuário se deseja tirar uma foto com a câmera do dispositivo ou se deseja selecionar uma imagem da galeria

Domínio semântico: URL



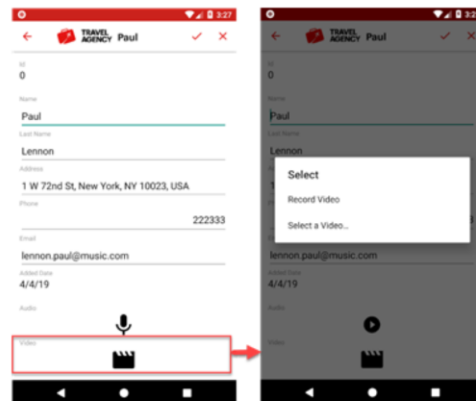
O domínio Url é utilizado para representar endereços Web, igualmente as anteriores, ao dar um Tap será aberto o navegador pré-determinado.

Domínio semântico: Áudio



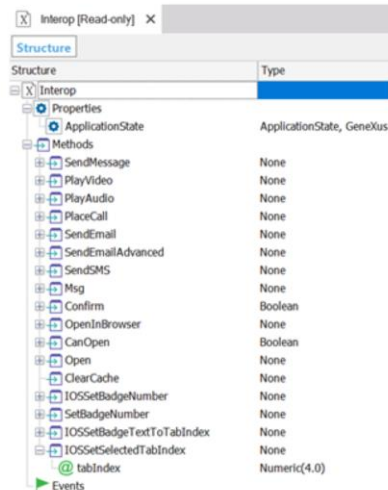
O domínio Áudio permite gerenciar conteúdo nos arquivos de áudio ou nos permite gravar um áudio utilizando o microfone do dispositivo.

Domínio semântico: Vídeo



O domínio Vídeo é similar a Imagem porém para arquivos de vídeo, podemos gravar um vídeo com a câmera ou selecionar uma da galeria.

Integração por meio de APIs: Interop



Structure	Type
Interop	
Properties	
AppState	AppState, GeneXus
Methods	
SendMessage	None
PlayVideo	None
PlayAudio	None
PlaceCall	None
SendEmail	None
SendEmailAdvanced	None
SendSMS	None
Msg	None
Confirm	Boolean
OpenInBrowser	None
CanOpen	Boolean
Open	None
ClearCache	None
IOSSetBadgeNumber	None
SetBadgeNumber	None
IOSSetBadgeTextToTabIndex	None
IOSSetSelectedTabIndex	None
TabIndex	Numeric(4,0)
Events	

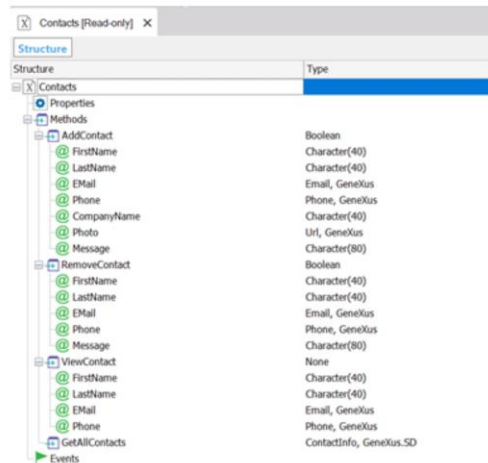
Também podemos integrar-nos através de APIs, vamos então ver algumas das mais utilizadas: Interop: esta API nos oferece toda uma série de Métodos para interagir com funções comuns como SendMessage para enviar um mensagem de texto, PlayVideo ou PlayAudio para reproduzir arquivos multimídia, PlaceCall para estabelecer uma chamada, SendEmail para enviar um mensagem de correio eletrônico, Msg para mostrar uma mensagem ao usuário, Confirm para pedir confirmação ao usuário antes de executar uma ação, etc.

Integração por meio de APIs: Calendar

Structure	Type	Is
Calendar		
Properties		
Methods		
Schedule	None	
Title	Character(30)	
StartDate	Date	
EndDate	Date	
StartTime	Time, GeneXus	
EndTime	Time, GeneXus	
Place	Character(30)	
Events		

Calendar: nos permite gerar um evento na aplicação de calendário pré-determinada do dispositivo

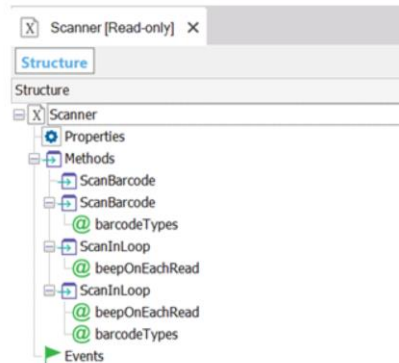
Integração por meio de APIs: Contacts



Structure	Type
Contacts	
Properties	
Methods	
AddContact	Boolean
FirstName	Character(40)
LastName	Character(40)
EMail	Email, GeneXus
Phone	Phone, GeneXus
CompanyName	Character(40)
Photo	Url, GeneXus
Message	Character(80)
RemoveContact	Boolean
FirstName	Character(40)
LastName	Character(40)
EMail	Email, GeneXus
Phone	Phone, GeneXus
Message	Character(80)
ViewContact	None
FirstName	Character(40)
LastName	Character(40)
EMail	Email, GeneXus
Phone	Phone, GeneXus
GetAllContacts	ContactInfo, GeneXus.SD
Events	

Contacts: nos permite interagir com a aplicação de contatos do dispositivo, podemos adicionar contatos, eliminá-los, ver a informação de um contato ou obter toda a lista.

Integração por meio de APIs: Scanner



Scanner: nos permite utilizar a câmera do dispositivo para escanear Códigos de barras ou QR, possui métodos para escanear somente um código, nesse caso retorna o valor em uma variável de tipo Char ou em uma coleção e ao finalizar retorna-nos uma lista de todos os códigos escaneados em uma coleção baseada no SDT ScannedBarcodes.

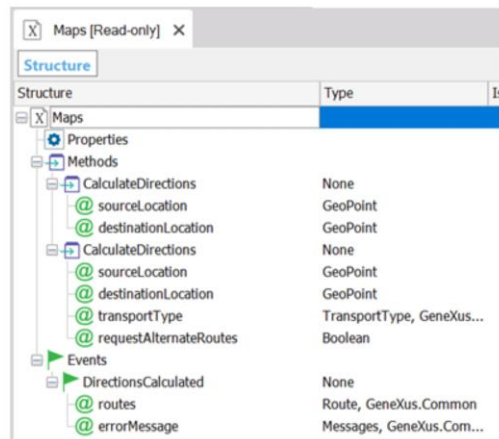
Integração por meio de APIs: SD Actions

Structure	Type
Actions	
Properties	
Methods	
Login	Boolean
Login	Boolean
LoginExternal	Boolean
LoginExternal	Boolean
Logout	None
GoHome	None
ReturnTo	None
Save	None
Cancel	None
Delete	None
TakeApplicationScreenshot	Image
Events	

SD Actions: nos prove métodos padrão para:

- Chamar o Login e fazer o logout da app.
- Ir a Home, que retorna ao objeto inicial.
- ReturnTo que nos permite retornar a um objeto específico

Integração por meio de APIs: Maps



Structure	Type	Is
Maps		
Properties		
Methods		
CalculateDirections	None	
sourceLocation	GeoPoint	
destinationLocation	GeoPoint	
CalculateDirections	None	
sourceLocation	GeoPoint	
destinationLocation	GeoPoint	
transportType	TransportType, GeneXus...	
requestAlternateRoutes	Boolean	
Events		
DirectionsCalculated	None	
routes	Route, GeneXus.Common	
errorMessage	Messages, GeneXus.Com...	

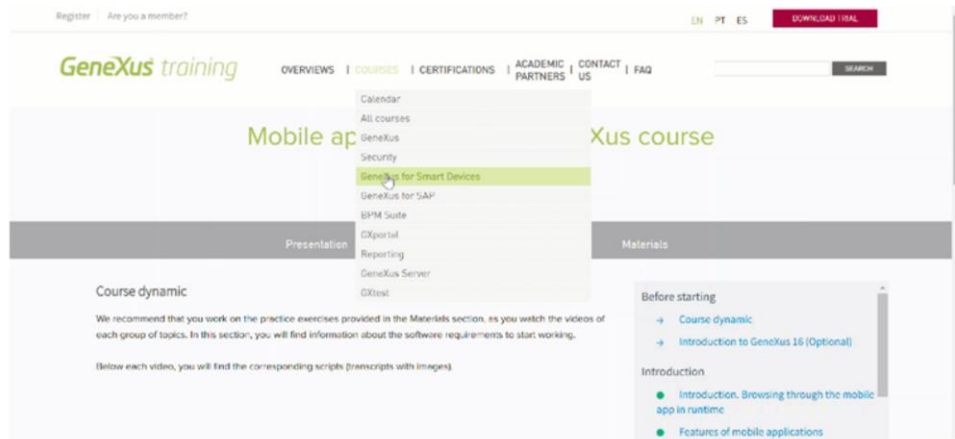
Maps API: esta pode ser utilizada na Web e em SD. Nos permite calcular caminhos e obter as rotas entre dos pontos.

Existem muitas mais APIs para utilizar, podem encontrar mais informação na wiki ou no Curso Desenvolvimento de aplicações móveis com GeneXus.

Demo: Desenvolvimento de aplicações para dispositivos móveis

Vamos para o GeneXus e ver algumas das coisas das que estivemos falando.

Curso Desenvolvimento de aplicações para dispositivos móveis



Como vocês devem ter notado, o desenvolvimento para Smart Devices é um tema muito amplo e cobre muitas outras questões que neste vídeo e na demo em particular o vimos de uma maneira simplificada ou simplesmente o mencionamos.

Os convido a seguir estudando com mais profundidade todos estes temas e muitos mais no curso de Desenvolvimento de Aplicações móveis com GeneXus no site de training.



Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications