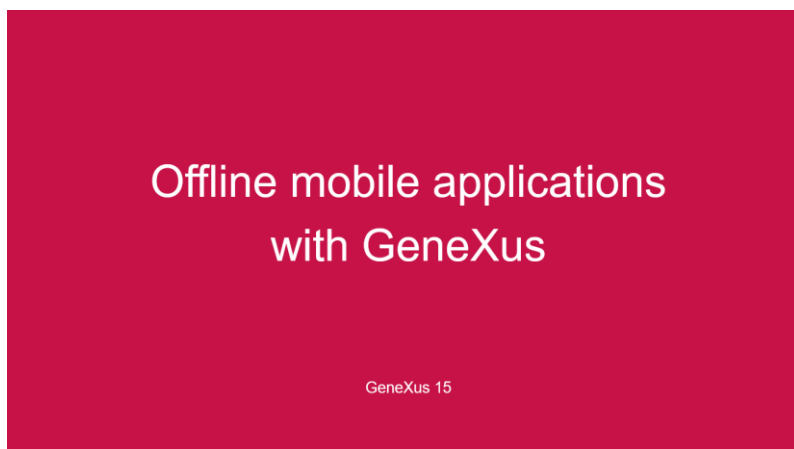


## Offline Mobile Applications with GeneXus (Part II)



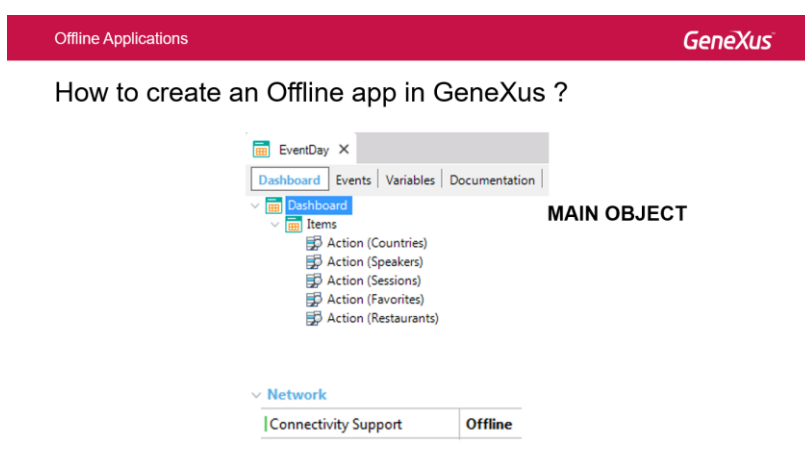
Até agora vimos o que são as aplicações conectadas, parcialmente conectadas e offline.

Porém, como fazemos para construir uma aplicação Offline, com GeneXus?

Se queremos que uma aplicação seja executada de forma Offline, devemos definir a propriedade Connectivity Support de seu objeto main, com o valor Offline. Este valor habilita a geração da aplicação de forma Offline e também o código nativo necessário para que a aplicação possa ser executada sem chamar os serviços REST do servidor web.

Depois que um objeto main tem sua propriedade Connectivity Support definida como Offline, a aplicação é instalada no dispositivo e é criado o banco de dados local juntamente com suas tabelas.

Assim, como o objeto main da nossa aplicação é o dashboard EventDay, devemos modificar sua propriedade Connectivity Support para Offline.

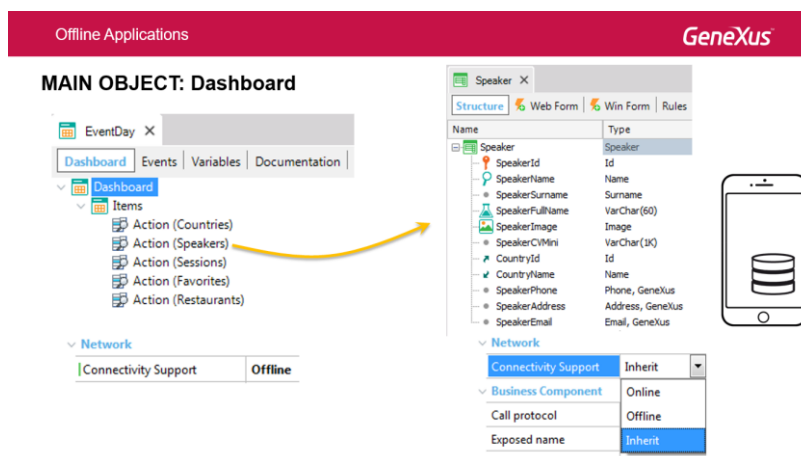


A propriedade Connectivity Support também está disponível para objetos que não são main. Seu valor pode ser Online, Offline ou Inherit.

Todas as tabelas que tiverem a propriedade Connectivity support = Inherit, utilizadas em objetos SD chamados pelo Main (direta ou indiretamente), serão criadas no banco de dados local do dispositivo. A não ser que estejam configuradas como Online. Nesse caso, não será criada essa tabela no banco de dados local porque será acessada a tabela do servidor através dos serviços REST.

O valor Inherit somente aparece nos objetos que não são Main.

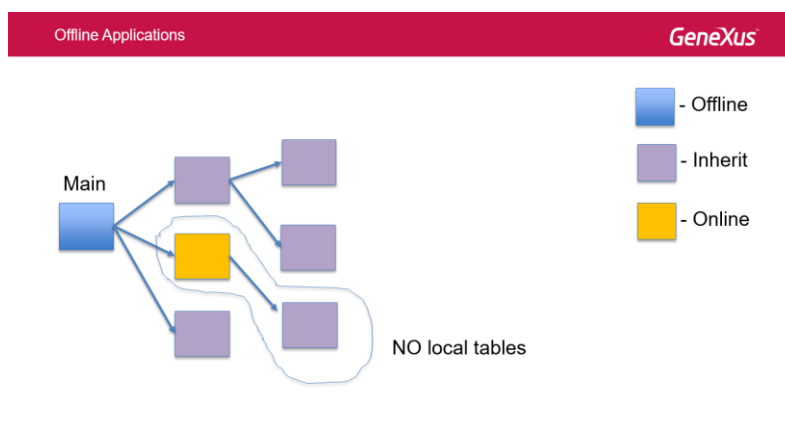
Também é possível definir a propriedade Connectivity Support de um objeto como Offline, criando assim sua tabela no banco de dados local. Nesse caso, o objeto trabalhará sem conexão, mesmo que o objeto Main esteja trabalhando Online.



Agora veja, quais tabelas efetivamente são criadas localmente quando uma aplicação passa a ser Offline?

Se observarmos a árvore de chamadas, então, as tabelas que serão criadas no dispositivo são todas as correspondentes aos objetos Offline e os que herdarem a conectividade Offline dos objetos que os chamaram.

Ainda, também serão contemplados e criados : a integridade referencial entre as tabelas e os atributos mencionados (tanto em panels como em prompts).

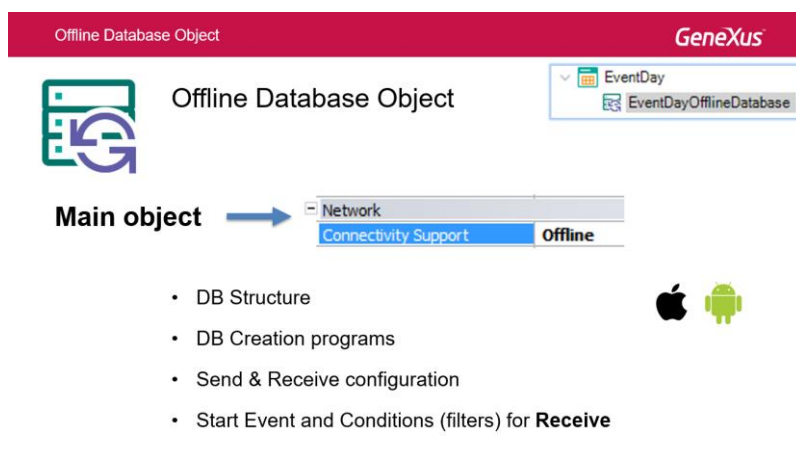


Para cada objeto main que tiver a propriedade Connectivity Support como Offline, será criado um objeto chamado Offline Database. Ele é criado na primeira vez que é feito um build sobre o respectivo objeto main, depois de alterar sua propriedade Connectivity Support para Offline.

Ao cria-lo, também se criam os programas para gerar o banco de dados local, na linguagem nativa do dispositivo.

Este objeto é o responsável por determinar, no momento da sincronização, quais são as tabelas que serão criadas no banco de dados local e também quais são os dados atualizados nas mesmas quando são sincronizados com as tabelas do servidor.

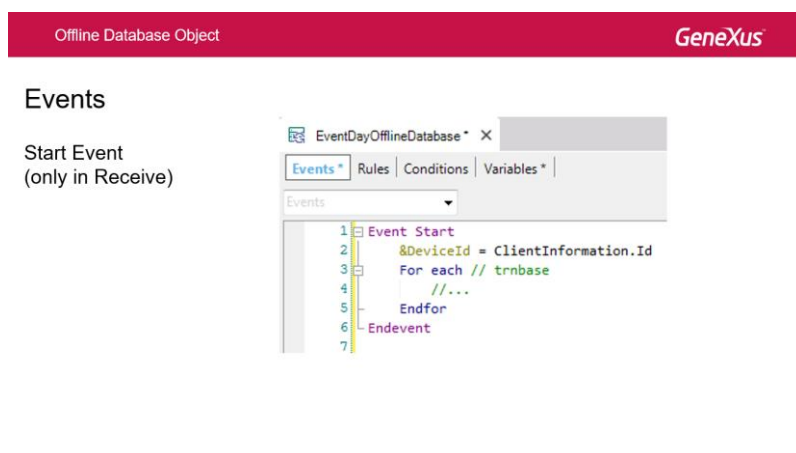
O objeto Offline Database dispõe de eventos, condições e propriedades que permitem determinar seu comportamento.



O único evento que podemos programar é o Start.

É executado no Server, antes de cada envio de dados para o cliente aconteça, numa sincronização, ou seja, unicamente no Receive.

Este evento está idealizado para inicialização de variáveis ou qualquer outro processamento que deva ser feito antes da sincronização das tabelas.



O uso das conditions é, assim como em qualquer outro objeto GeneXus, para definir filtros sobre os dados e somente são utilizados para o Receive.

Os filtros são aplicados nas tabelas do server, para definir quais dados serão levados para o dispositivo. Isso implica que os dados locais podem ser um subconjunto dos dados do server e, em casos específicos, pode acontecer que não existam registros de uma determinada tabela no dispositivo.

São independentes e se aplicam até a tabela estendida, porque GeneXus tem a inteligência de determinar em quais tabelas se aplicam. Na lista de navegação, é possível ver as condições que foram aplicadas em cada tabela.

Nas expressões das condições, é possível utilizar variáveis pré-definidas ou definidas pelo desenvolvedor, sendo que, nesse caso, essas variáveis devem ser assinaladas no evento Start.

The screenshot shows the 'Offline Database Object' window for 'EventDayOfflineDatabase'. The 'Conditions' tab is active, showing a single condition: '1 SessionActive = true;'. Below the conditions, there are tabs for 'Events', 'Rules', 'Conditions', and 'Variables'. To the right, the 'Navigation View' shows the 'Offline Database: EventDayOfflineDatabase Navigation' section. It details the 'Event Synchronize for Session (Line: 2)' with a 'For Each Session (Line: 3)' loop. The loop's constraints include 'SessionActive = TRUE'. A yellow arrow highlights the connection between the 'SessionActive' condition and its use in the navigation logic.

No objeto OfflineDatabase são configuradas as propriedades que vimos para o Send e o Receive.

The screenshot shows the 'Offline Database Object' window for 'EventDayOfflineDatabase'. The 'Properties' tab is active. On the left, there are buttons for 'Receive' (orange) and 'Send' (red). The main area shows the 'Properties' window for 'Offline Database: EventDayOfflineDatabase'. The 'Receive' section is expanded, showing settings: 'Data Receive Criteria' (On Application Launch), 'Minimum Time Between Receives' (600), 'Data Receive Granularity' (By Row), and 'Minimum Time Between Table Purges' (3600). The 'Send' section is also expanded, showing: 'Send Changes' (When connected) and 'Minimum Time Between Sends' (0).

Para finalizar, alguns pontos adicionais....

Se a aplicação Offline utiliza o GAM, deve-se levar em conta que as credenciais sempre estarão no server, portanto o login só pode ser feito estando Online.

Depois que estiver logado, aí sim pode-se trabalhar Offline.

Nas aplicações mobile, não está implementada a segurança por autorização, somente por autenticação.

Offline Applications

GeneXus

Offline Apps + GAM

The credentials remain on the server

Login only Online

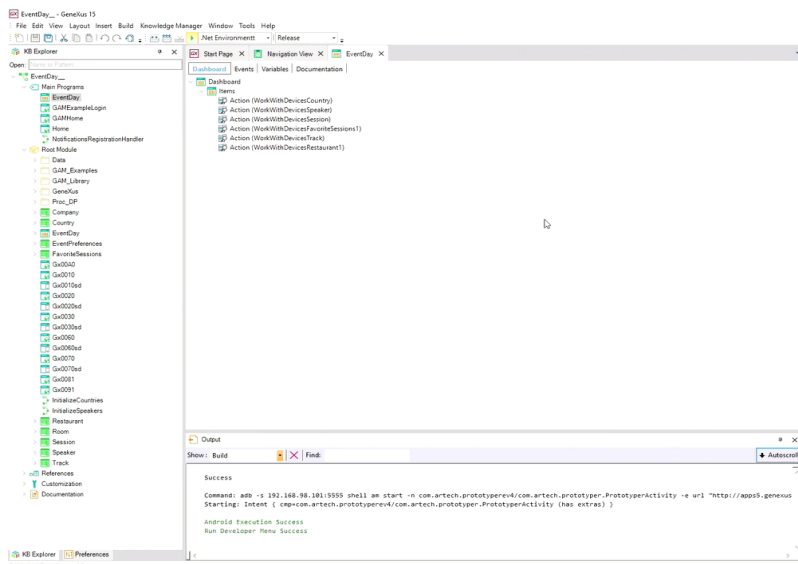
- Authenticated user is still authenticated when the app goes Offline

Smart Devices: Only Authentication

Em seguida, veremos uma breve demonstração de como construir uma aplicação Offline com GeneXus.

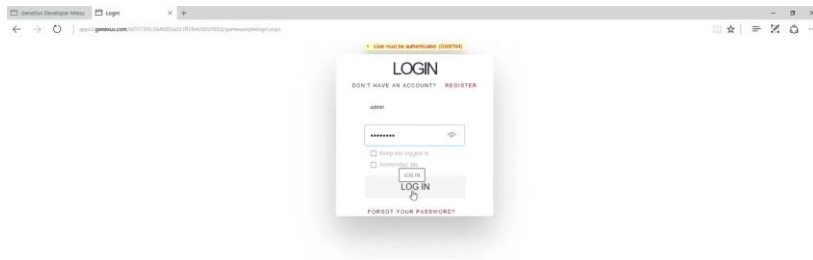
O que vou fazer agora é executar a aplicação. O que é que vou mostrar?

Como a aplicação é online, quando faço alterações pela interface web já consigo ver as alterações no dispositivo, e, se altero algo no dispositivo, já me mostra isso na interface web; Isso acontece porque a aplicação está online, ou seja, o dispositivo está sincronizado com os serviços REST, que estão conectados no banco de dados da aplicação.

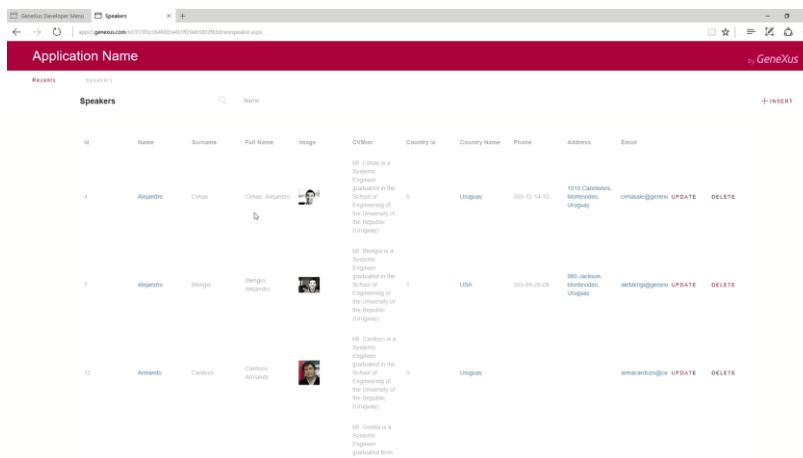


Depois, o que vou fazer é converte-la em uma aplicação online.

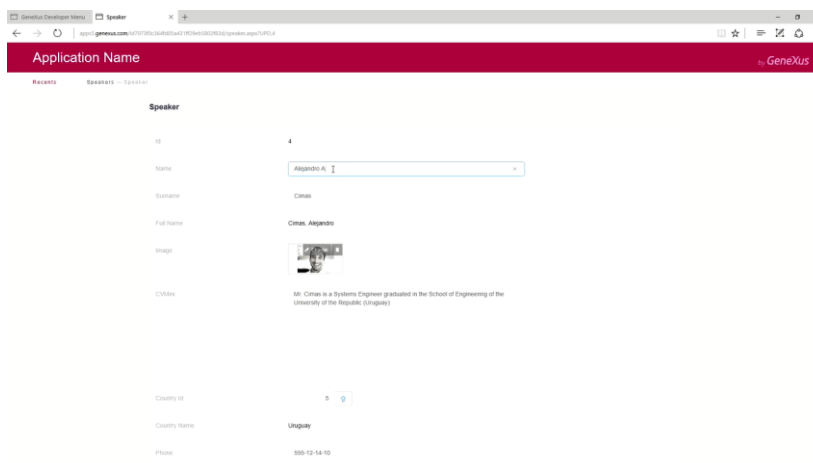
Bom, vou me logar na aplicação web...



Vamos visualizar alguma informação...

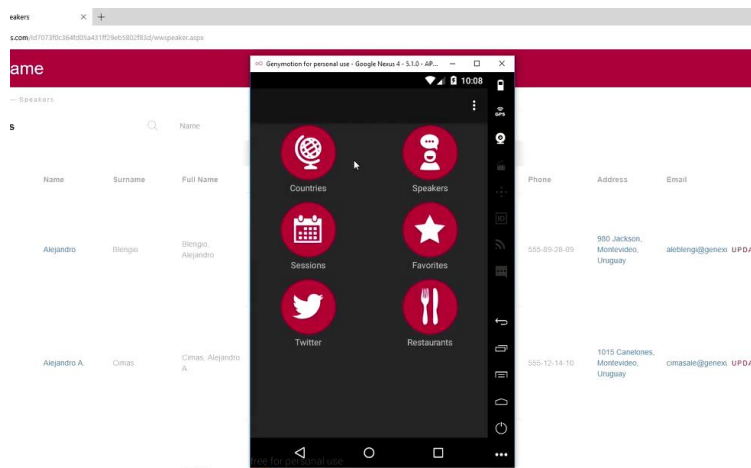


Aqui, por exemplo, temos Alejandro Cimas... vamos alterar Alejandro; vamos supor que seu segundo nome comece com A...

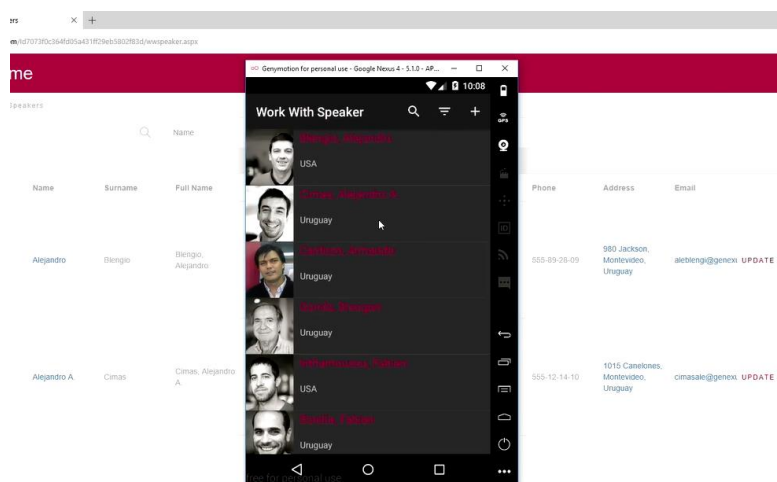


E salvamos.

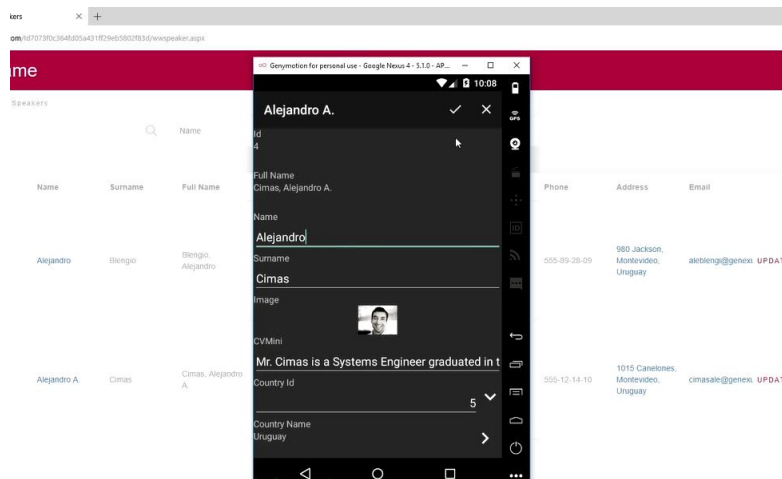
Agora vamos na aplicação Android...que estou executando com a KBN...



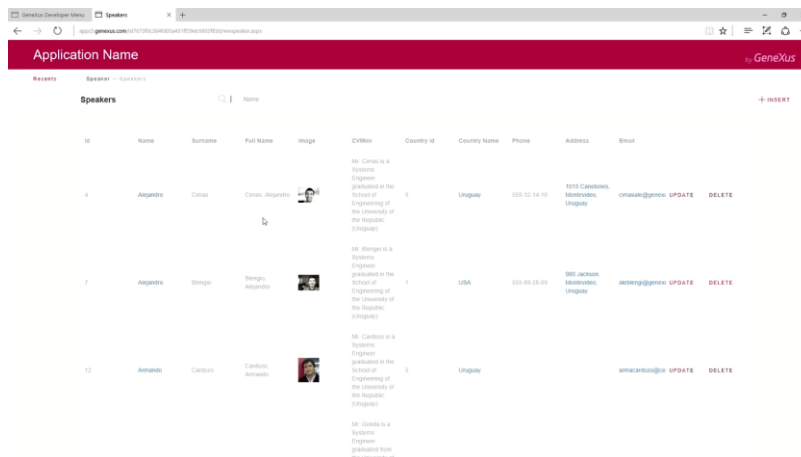
Vejam que aqui já apareceu Alejandro A. Cimas, ou seja, que a alteração foi automática:



Se entro por aqui, e edito o valor...

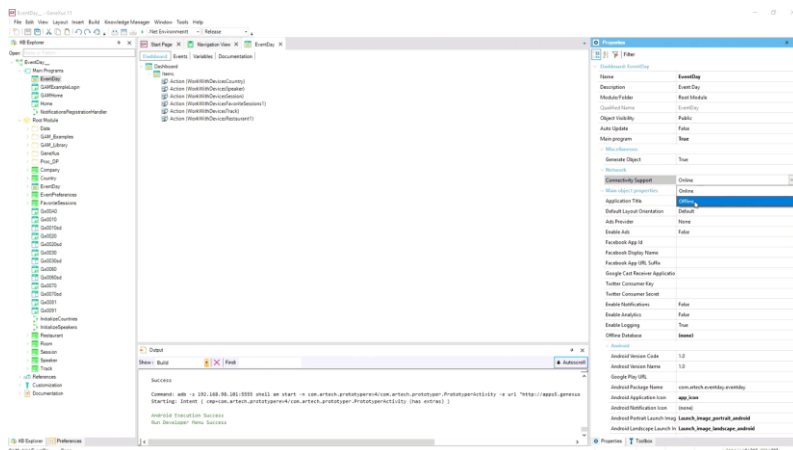


E depois, vamos na web, vemos que as alterações foram atualizadas automaticamente.



O que vou fazer agora é converter essa aplicação para Offline.

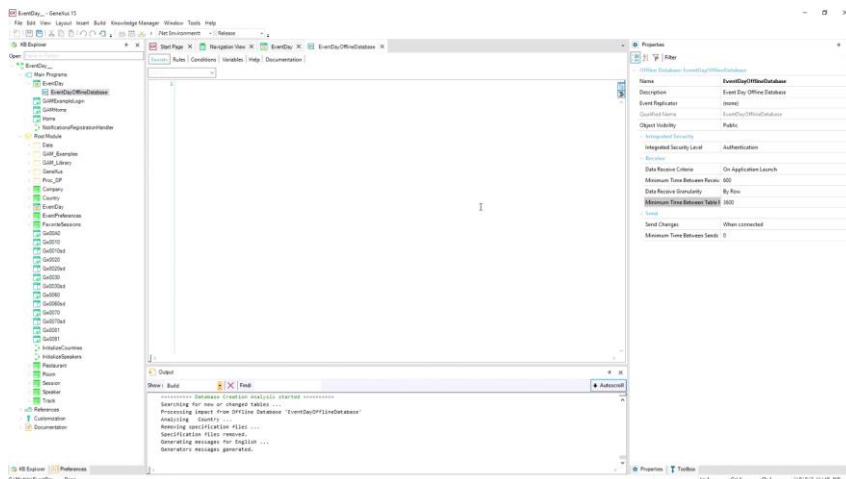
Para isso, vou ao GeneXus, no objeto Main (que neste caso é o menu EventDay), e vou alterar a propriedade Connectivity Support para Offline.



E depois vou executar (Run).

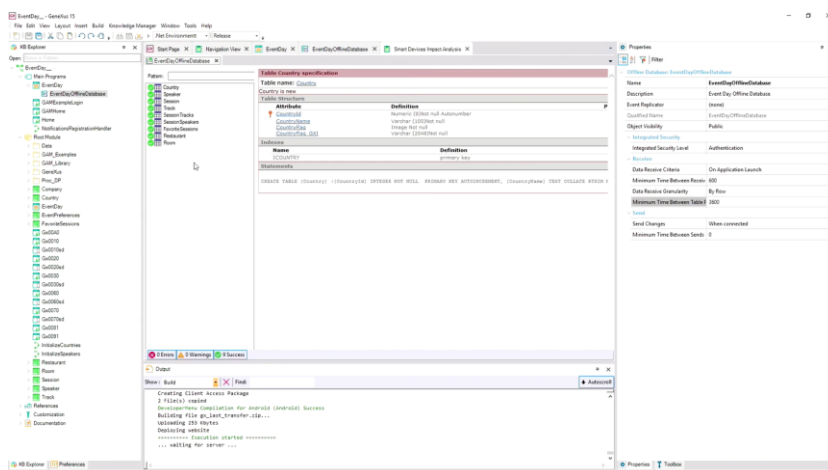
Vemos que foi criado, abaixo de EventDay, abaixo de meu Dashboard, o objeto EventDayOfflineDatabase, que é o objeto no qual vou poder alterar todos os critérios para fazer o recebimento e o envio de dados nas IDEs do server.



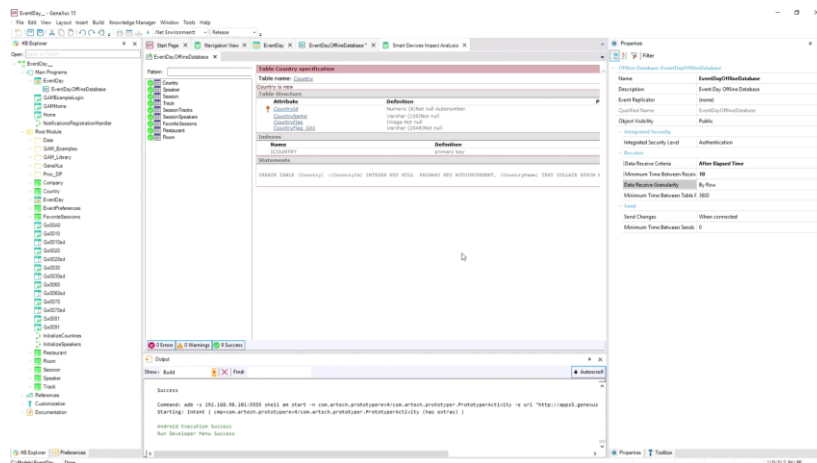


Além disso, também observamos na Análise de Impacto, que todas estas tabelas serão criadas de forma local no dispositivo.

Já que eu não modifiquei nenhum destes Work Withs, todos tem a propriedade Inherit, e, portanto, estas tabelas serão criadas no dispositivo.

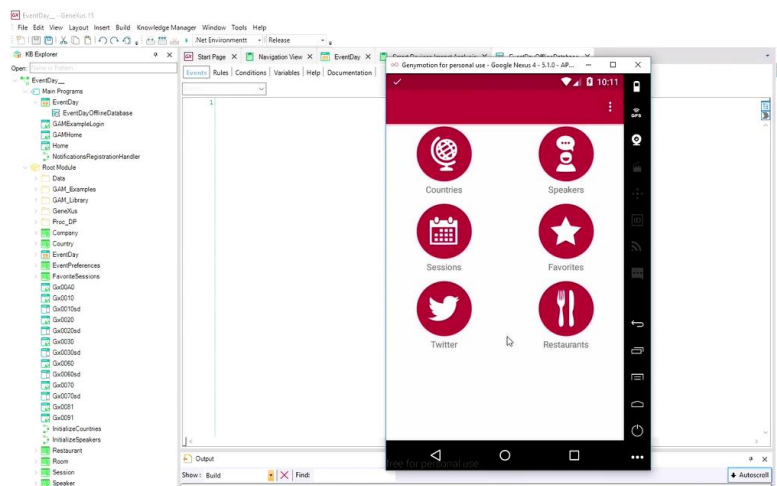


O que vou fazer é alterar estes valores: para o critério de atualização, em vez de deixar para quando a aplicação inicia, vou colocar a cada certo tempo, e que a atualização, ou seja, a sincronização, seja a cada 10 segundos.



Perfeito.

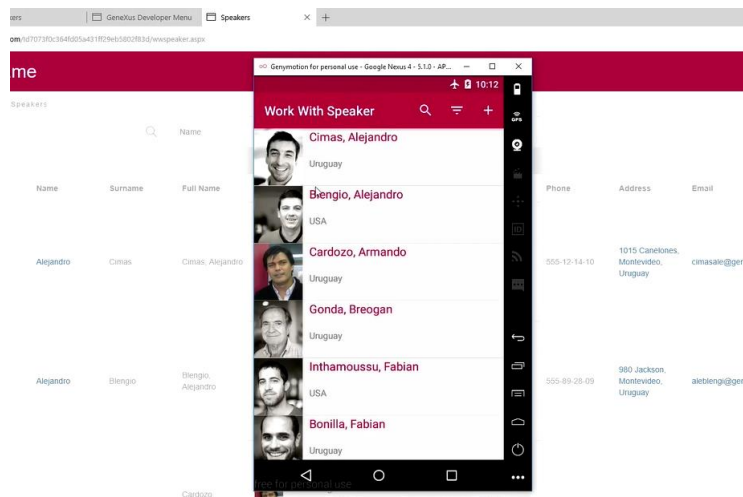
Agora, o que vou fazer é conferir a aplicação novamente. Já temos as tabelas geradas no dispositivo, e toda a lógica necessária de sincronização, e o que quero é que essa sincronização seja a cada 10 segundos; por isso defini a propriedade Minimum Time Between Receives como 10 segundos.



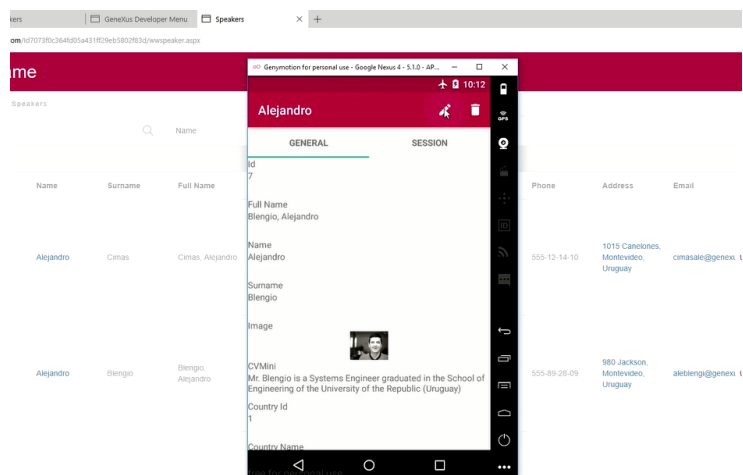
Mesmo sabendo que nossa aplicação é Offline, a internet está conectada. Então, o que vou fazer é desconectá-la da Internet.

Para isso, vou habilitar o modo avião...

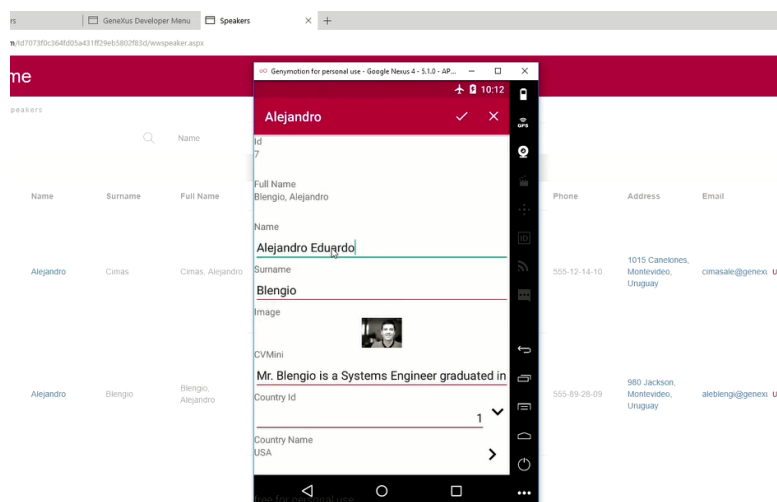




Por que? Porque as informações não foram sincronizadas. Vamos aproveitar e realizar uma alteração em outro dos oradores...

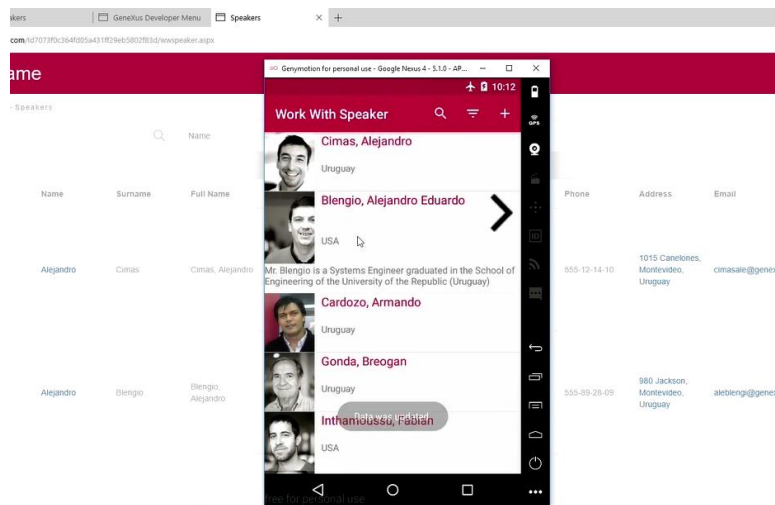


E, por exemplo, vamos supor que Alejandro se chama Eduardo também.

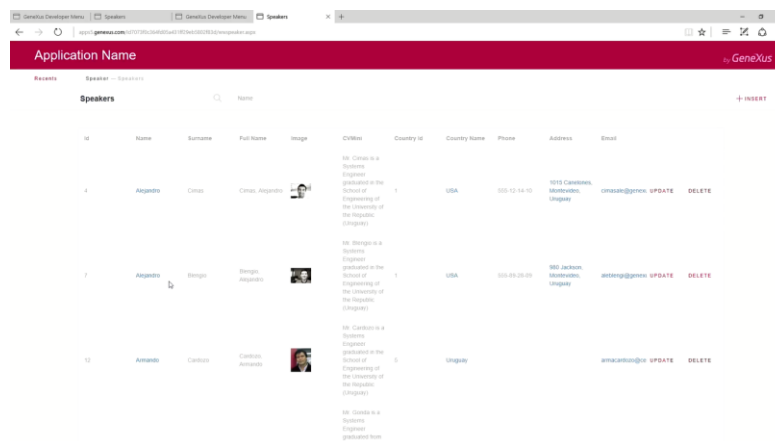


E vamos salvar.

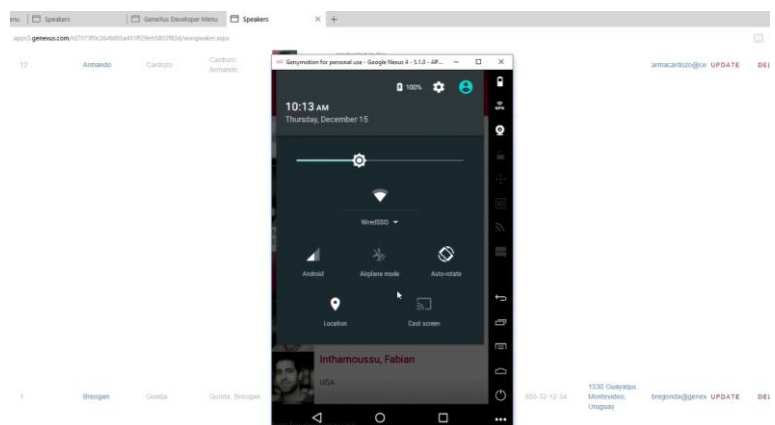
Aqui temos a alteração...



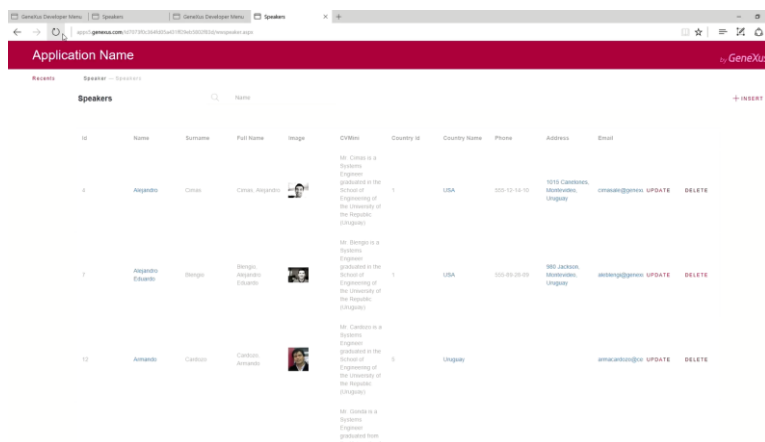
Vamos atualizar o navegador... obviamente as alterações não estão atualizadas ainda porque não está sincronizado, ou seja, Alejandro Cimas é dos USA e Alejandro Blengio não se chama Eduardo.



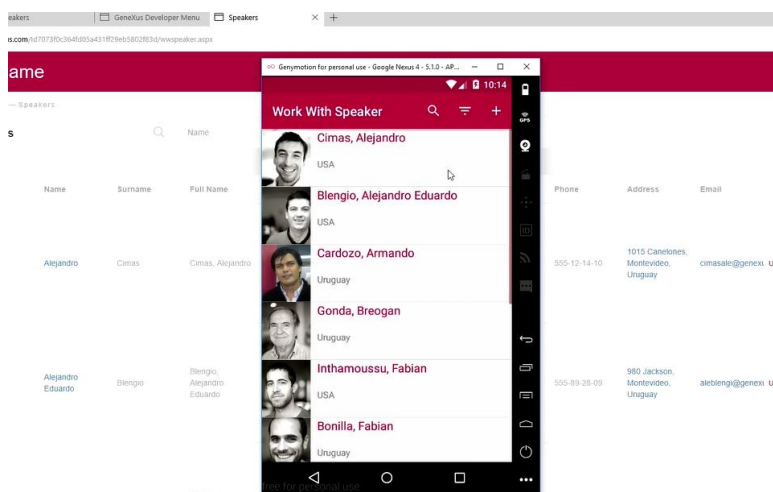
Assim, o que vou fazer agora é sincronizar as alterações do dispositivo. Ou seja, vou voltar a colocar o dispositivo conectado à internet para que aconteça a sincronização.



Muito bem. Agora já está conectado novamente. Voltemos ao navegador... vamos atualiza-lo... E vemos que agora aparece Alejandro Eduardo Blengio, ou seja, aquilo que havia sido alterado no dispositivo foi atualizado no banco de dados do server.



E vamos ver agora se as alterações feitas no navegador foram para o dispositivo. Realmente, mostra que Alejandro Cimas agora é dos Estados Unidos, provando que também foram sincronizados os dados no dispositivo.



Ou seja, enquanto estive sem conexão, eu pude trabalhar com o dispositivo como se tivesse online, e, no momento que se reestabeleceu a conexão por WiFi, então aconteceu a sincronização. Posso perfeitamente trabalhar numa aplicação Offline, e depois, no momento que estabeleço a conexão novamente, tudo é sincronizado.

Terminamos aqui essa Demonstração básica. Era simplesmente para dar uma ideia de como é fácil poder criar uma aplicação Offline com GeneXus.

Neste video, vimos como é fácil, com GeneXus, poder implementar aplicações mobile conectadas, parcialmente conectadas, ou totalmente desconectadas, assim como também o processo de sincronização dos dados entre um dispositivo trabalhando Offline ou parcialmente conectado, e o servidor web.

Se desejar mais informações sobre este tema, visite o link exibido na tela.

Offline Applications

GeneXus<sup>tm</sup>

More info:

<http://wiki.genexus.com/commwiki/servlet/wiki?22228>