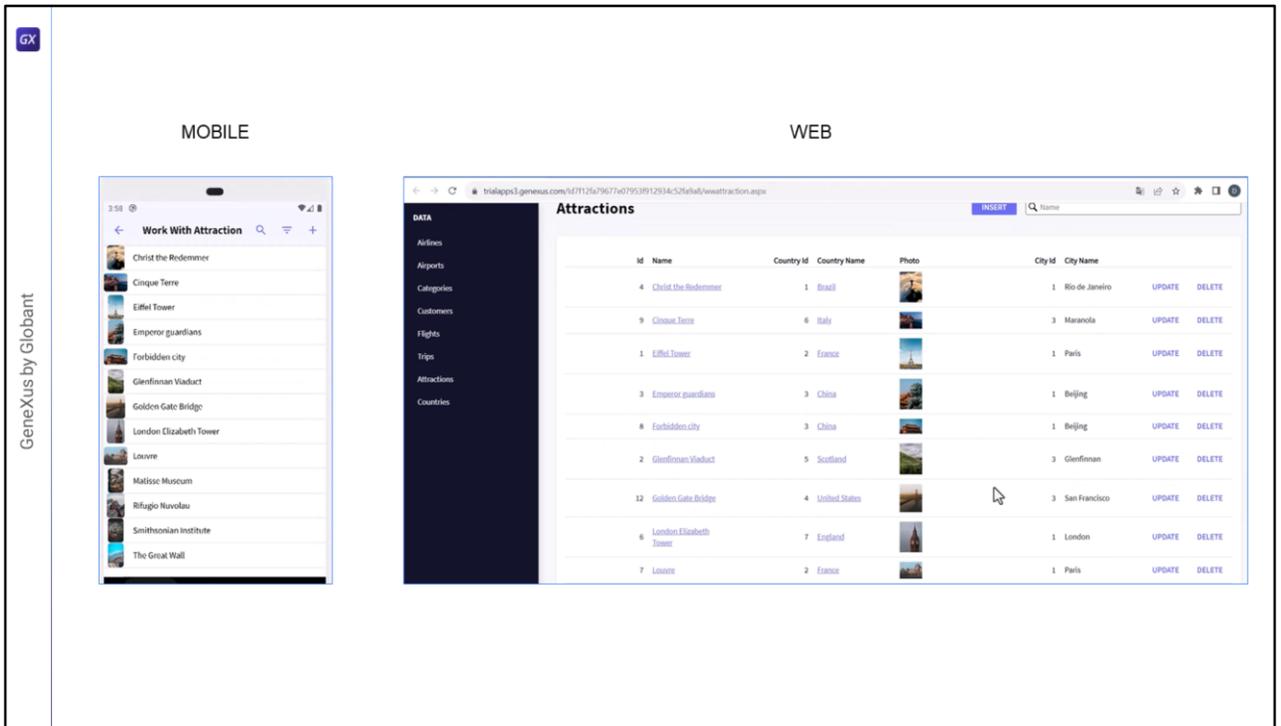


# Offline Database Generation



Diego Marranghello



Neste vídeo veremos como converter uma aplicação Online em uma aplicação Offline.

Primeiro vamos ver como funciona atualmente nossa aplicação Travel Agency, que atualmente é uma aplicação Online.

Vamos acessar a aplicação mobile e a aplicação Web ao mesmo tempo, e teremos ambas as interfaces abertas para ver como é a interação.

Para mostrar um exemplo claro e simples, à transação Attraction aplicamos ambos os padrões Work With: para web e para mobile.

Entramos primeiro pela web, selecionamos uma atração, por exemplo Louvre, editamos seu nome e gravamos.

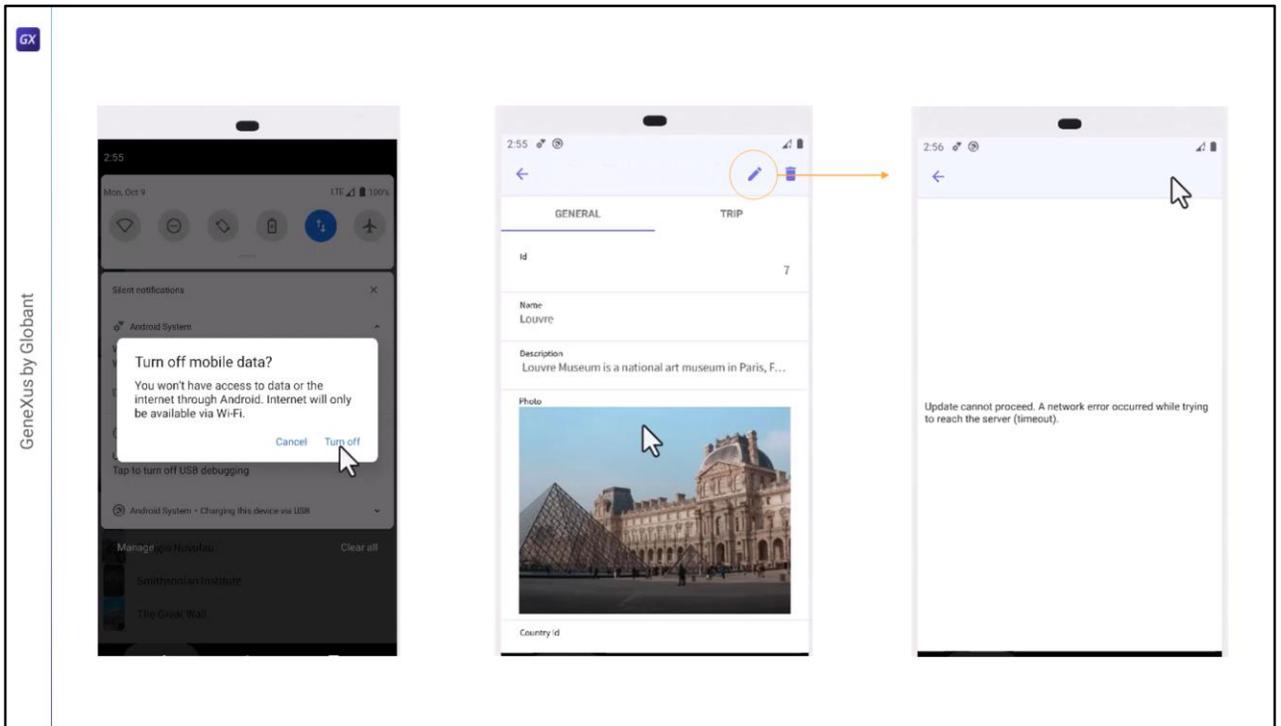
Agora vamos para a aplicação mobile, atualizamos e automaticamente vemos a alteração realizada.

O que acontece se o editarmos a partir do dispositivo?

Vamos novamente editar o nome para deixá-lo como estava inicialmente e gravamos.

Atualizamos no panel web e automaticamente vemos a mudança.

Isso ocorre porque a aplicação trabalha de forma conectada.



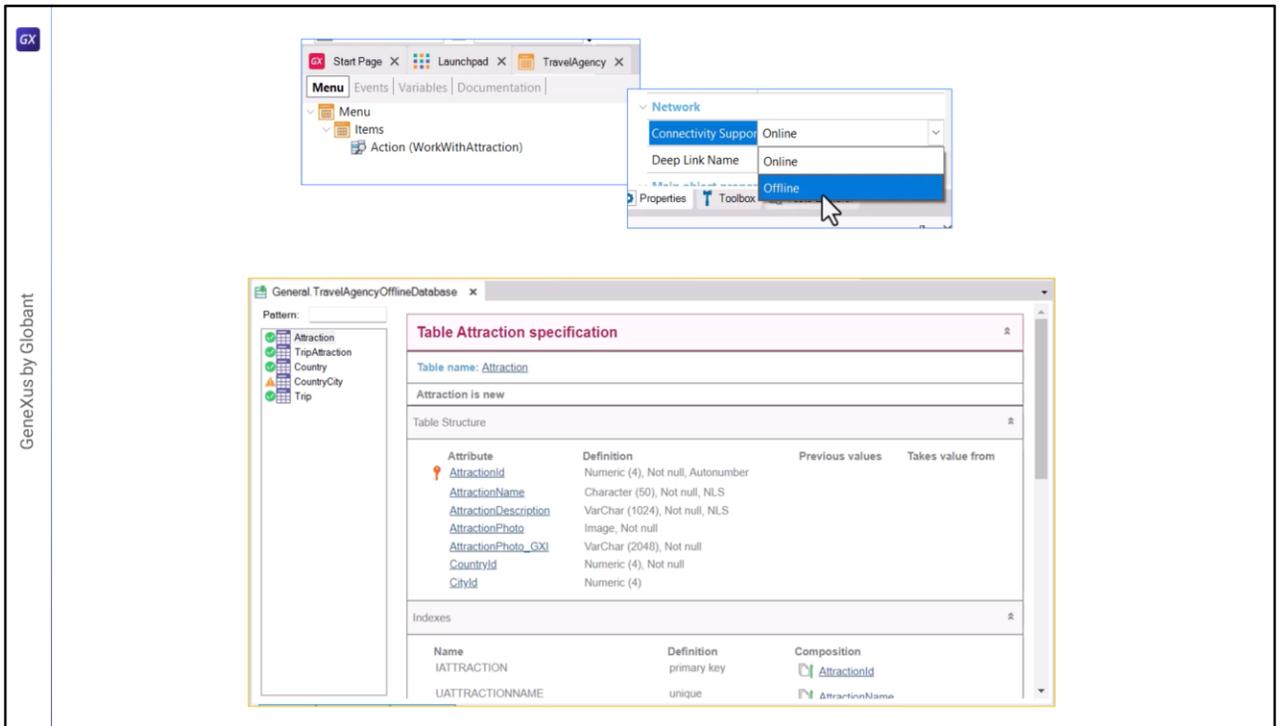
O que acontece, por exemplo, se agora desconectamos a aplicação? Vamos simular isto, desligamos o acesso Wifi e desligamos os dados móveis. Agora o emulador não tem conexão com a internet, portanto não pode acessar a camada de serviços REST.

Se tentarmos entrar na atração conseguiremos porque esses dados estão armazenados em cache do acesso anterior, mas se tentamos editar dá um erro.

Se tentarmos uma atração que não tenhamos acessado antes, não poderemos, pois não temos conexão e também não temos seus dados salvos.

Quando estamos sem conexão em uma aplicação Online, temos acesso apenas aos dados que estão armazenados em cache na aplicação, mas não podemos acessar novas telas.

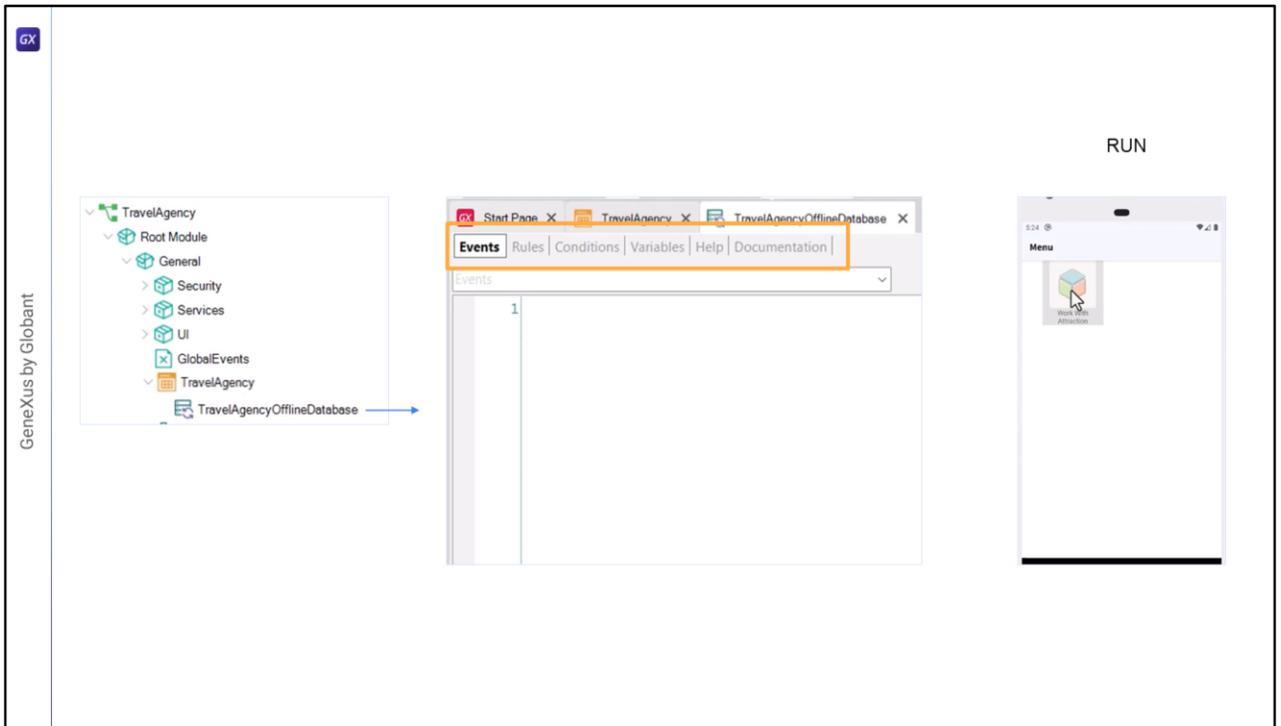
Bem, vamos agora ao GeneXus para podermos converter a aplicação em uma aplicação Offline.



Em nossa aplicação temos um objeto Menu denominado TravelAgency, que é o objeto main da aplicação. E dentro, como Item, só temos no momento, o work with de atração.

Ao ir para as propriedades do objeto menu, vemos uma chamada Connectivity Support que por padrão tem o valor Online, e vamos alterar para Offline. Salvamos as alterações e fazemos um build da aplicação.

Uma das coisas que veremos a seguir é uma análise de impacto da base de dados Offline, estas serão as tabelas que serão criadas no dispositivo.



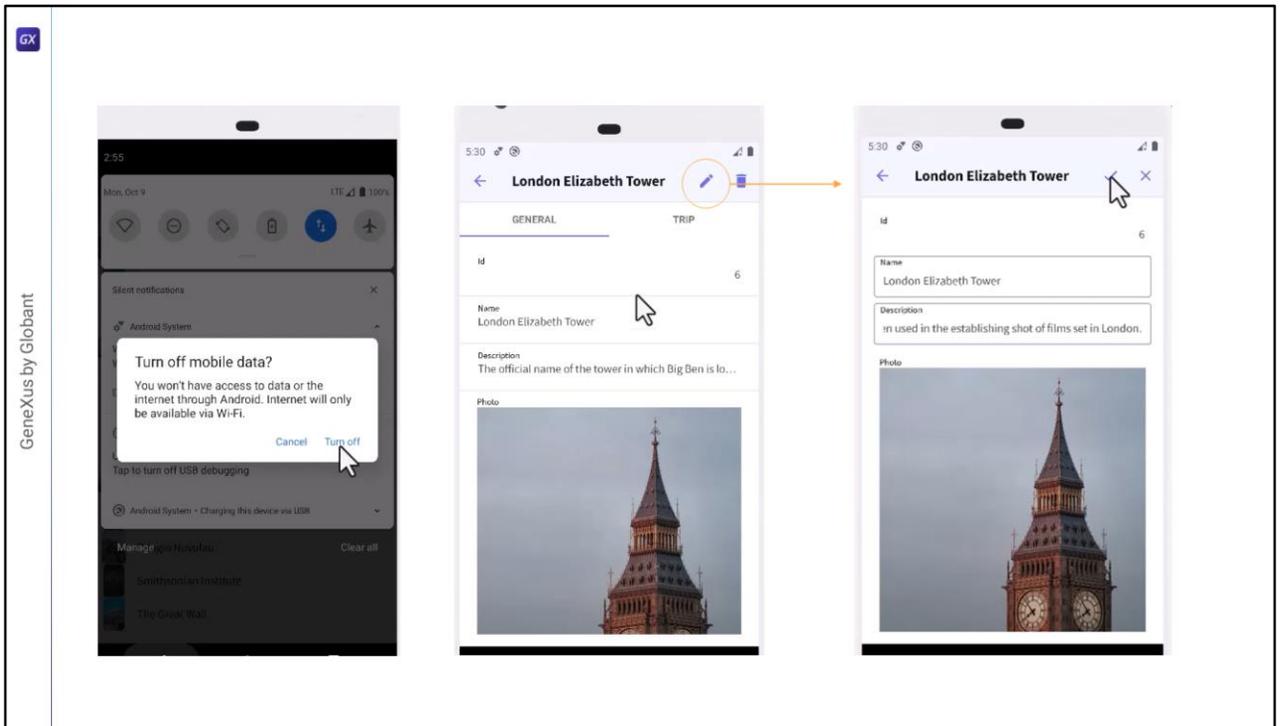
Vamos ver que agora temos um objeto novo, `TravelAgencyOfflineDatabase` que está associado ao objeto menu `TravelAgency`.

Este objeto nos permitirá configurar determinadas questões relacionadas à base de dados Offline, que será gerada no dispositivo, como por exemplo a granularidade. Posteriormente estudaremos essas propriedades com mais detalhes.

Além disso, este objeto possui Eventos, Regras, Condições e variáveis, logo veremos como são utilizadas, por enquanto deixaremos tudo default.

Vejamos que agora, ao iniciar a aplicação mobile, podemos ver que está sendo executada a sincronização.

Uma vez que termina o processo, podemos utilizar a aplicação normalmente.

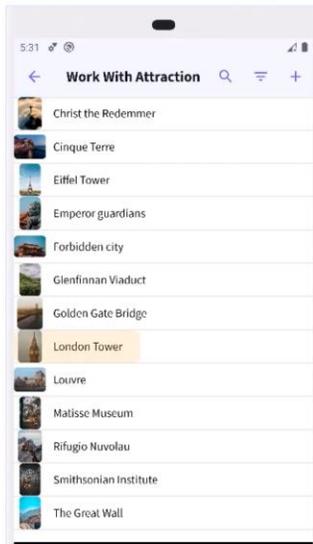


Entramos para ver as atrações, veremos a lista.  
Vamos desconectar o dispositivo.

E agora vamos tentar entrar em uma atração, e não temos o erro que dava antes, quando estávamos conectados e perdíamos a conexão. Isso ocorre porque agora os dados estão no dispositivo.

Posso até editar os registros.

Para o usuário não há diferença em relação a estar conectado.



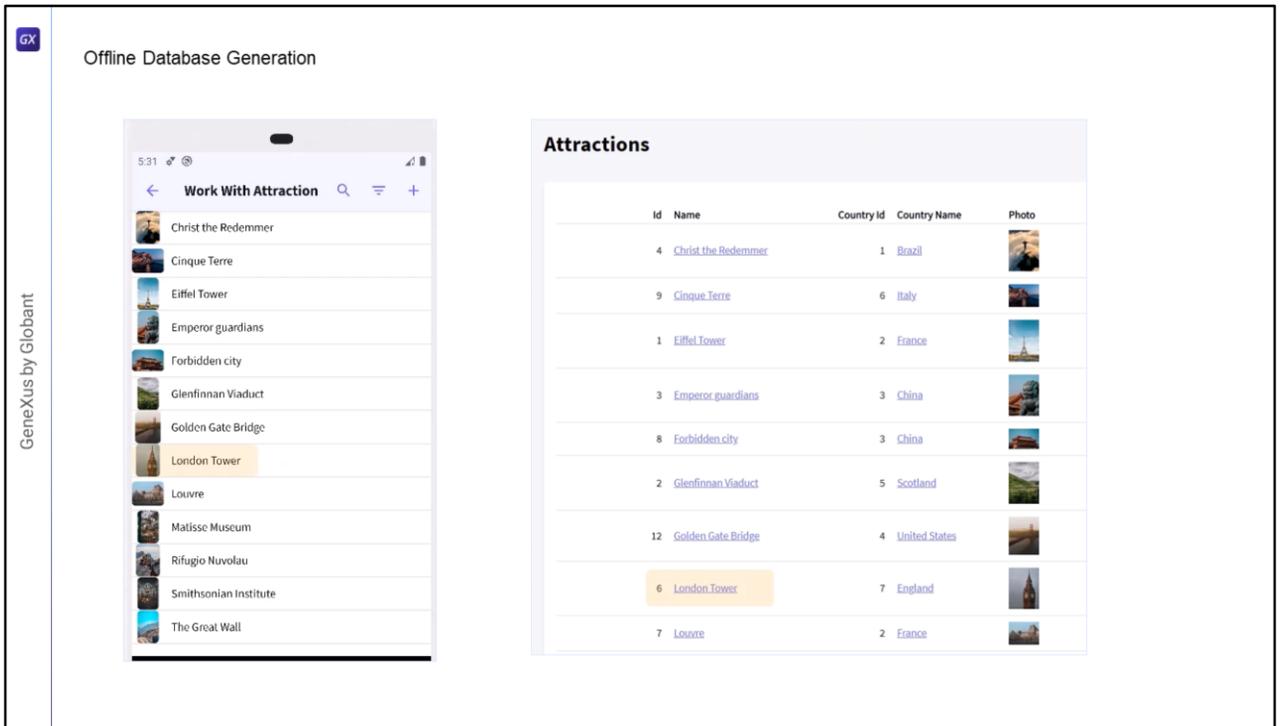
### Attractions

Id	Name	Country Id	Country Name	Photo
4	<a href="#">Christ the Redemmer</a>	1	<a href="#">Brazil</a>	
9	<a href="#">Cinque Terre</a>	6	<a href="#">Italy</a>	
1	<a href="#">Eiffel Tower</a>	2	<a href="#">France</a>	
3	<a href="#">Emperor guardians</a>	3	<a href="#">China</a>	
8	<a href="#">Forbidden city</a>	3	<a href="#">China</a>	
2	<a href="#">Glenfinnan Viaduct</a>	5	<a href="#">Scotland</a>	
12	<a href="#">Golden Gate Bridge</a>	4	<a href="#">United States</a>	
6	<a href="#">London Elizabeth Tower</a>	7	<a href="#">England</a>	
7	<a href="#">Louvre</a>	2	<a href="#">France</a>	

Vamos voltar a editar o nome de uma atração, gravamos.

Notem que não temos nenhum problema por estar offline. Os dados são gravados sempre no dispositivo.

E se formos para a interface web e atualizarmos não veremos essa mudança. Porque o dispositivo está offline.



Vamos reconectar o emulador à rede.

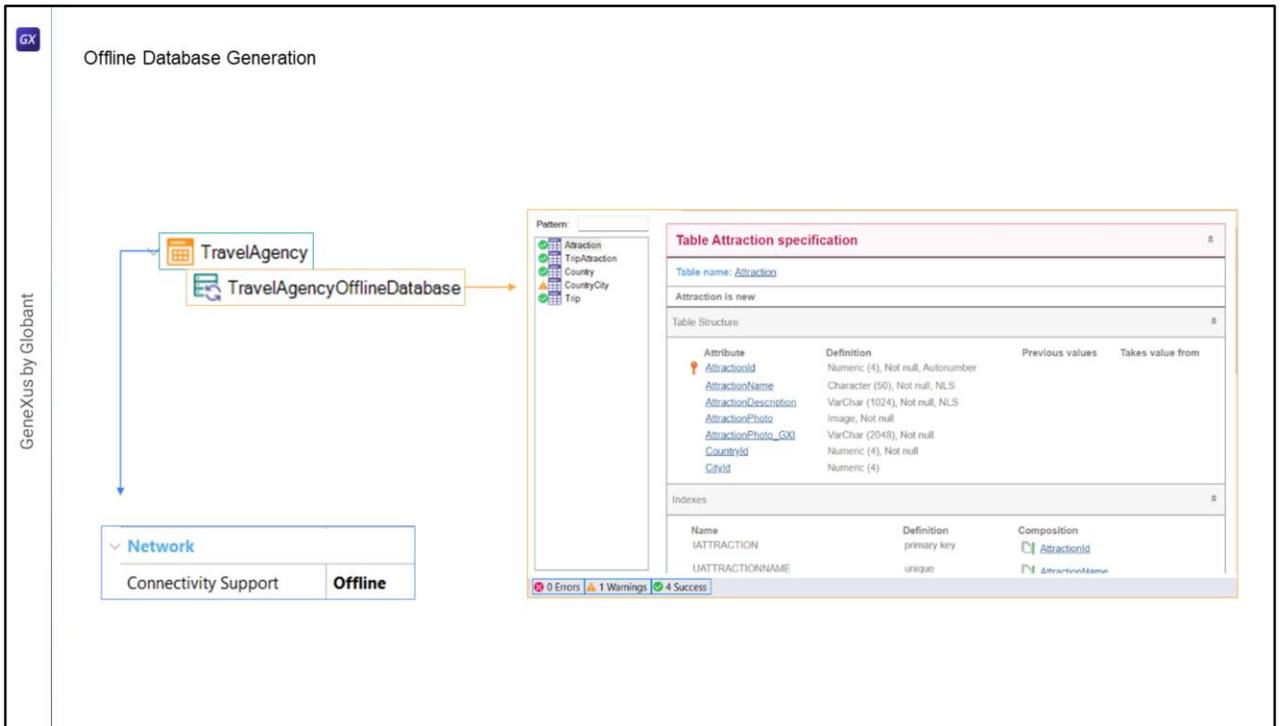
E para forçar a sincronização vamos fechar e abrir a aplicação novamente, isso porque ainda não configuramos nada do objeto OfflineDatabase.

Ao abrir novamente executa a sincronização.

E agora sim, vamos para a web e atualizamos, e ali vemos a mudança.

Terminamos aqui esta Demo básica, onde vimos como é simples e fácil criar uma aplicação Offline em GeneXus.

Também vimos o processo de sincronização dos dados entre o dispositivo trabalhando de forma Offline e o servidor web, e as operações de CRUD que sempre serão realizadas sobre a base de dados local.



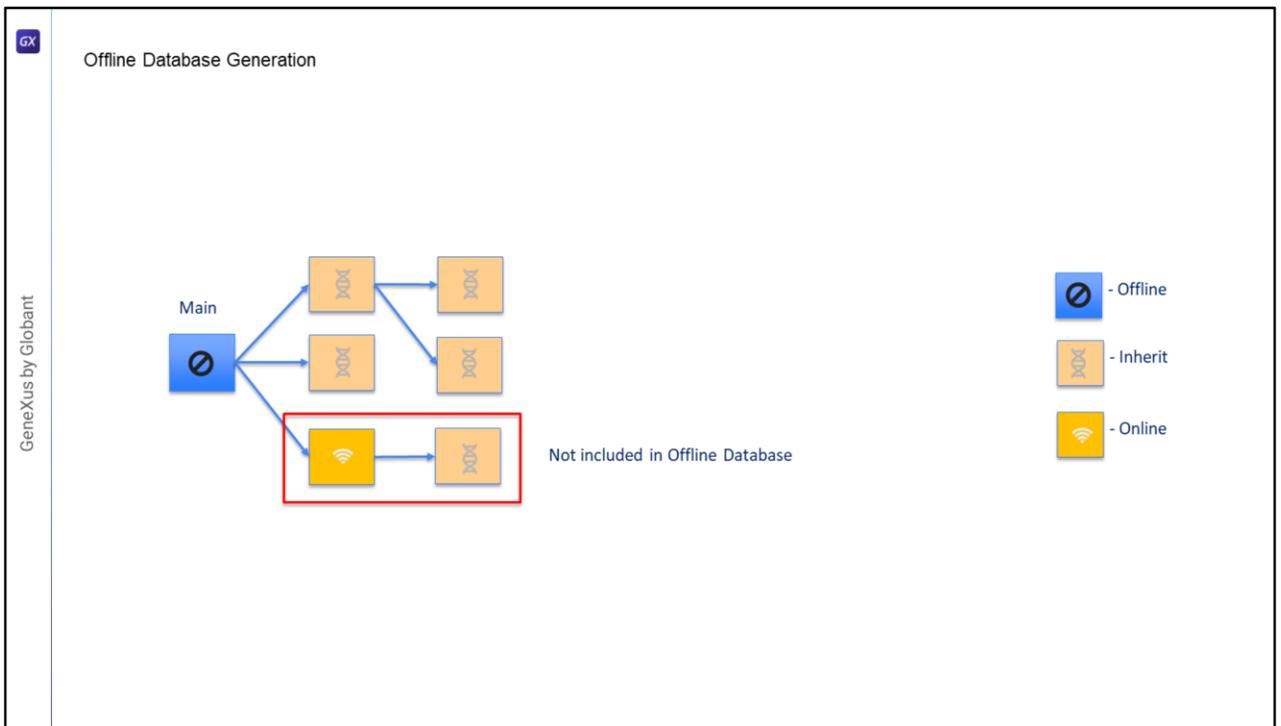
Como acabamos de ver, quando indicamos que o objeto main era Offline e realizamos o Build, foi gerado automaticamente um objeto OfflineDatabase associado ao objeto Main.

Vimos também que nos foi apresentada uma análise de impacto, semelhante a quando fazemos o create da base de dados, mas essas tabelas que são mostradas são as tabelas que GeneXus analisou e que deve levar para a base de dados SQLite no dispositivo.

Ao criar o objeto OfflineDatabase, são criados também na linguagem nativa do dispositivo, os programas para criar esta base de dados local.

Este objeto é responsável por determinar quando ocorre a sincronização, quais são as tabelas que serão criadas na base de dados local e também quais são os dados que são levados para elas quando forem sincronizados com as tabelas do server.

O objeto OfflineDatabase também é onde são configuradas as propriedades que vimos para o Send e o Receive.



Agora, quais tabelas efetivamente são levadas quando uma aplicação torna-se offline?

Suponhamos que temos essa árvore de chamadas a partir do objeto Main, podemos ter objetos com a propriedade Connectivity Support Offline, como o Main, outros com o valor default que é Inherit, que herda o tipo de acordo com o objeto que o chama e outros que podem ser Online.

Se tivermos esta árvore de invocações, então as tabelas que serão levadas ao dispositivo serão todas aquelas dos objetos Offline e aqueles que herdaram a conectividade offline daquele que os invocou.

Mas também vai ser considerada a integridade referencial e os atributos mencionados, tanto em panels quanto em prompts.

Esse objeto que é Online que chama esse outro que herda o tipo de conexão, não seriam incluídos, serão online.

Nos próximos vídeos veremos mais detalhes sobre a sincronização e as propriedades do objeto OfflineDatabase

GX

GeneXus by Globant

**GeneXus**<sup>™</sup>  
by Globant

[training.genexus.com](https://training.genexus.com)