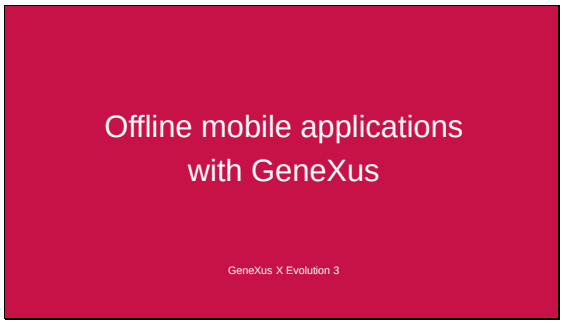
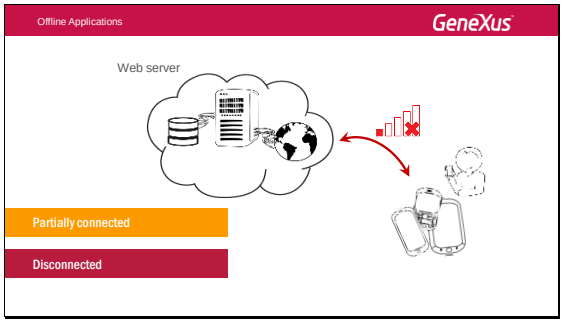
Aplicações off-line com GeneXus



Até agora desenvolvemos aplicações para dispositivos inteligentes, que sempre estavam conectadas através da Internet ao servidor web, que armazenava os serviços e os dados necessários para a aplicação.

Uma meta importante para dispositivos inteligentes é permitir que essa aplicação, ou parte dela, continue executando quando se encontre desconectada da internet.



Há tarefas que, devido à sua sensibilidade, devem ser validadas no banco de dados centralizado. Também as tarefas nas quais seus dados mudam muito frequentemente, também exigirão acesso contínuo ao servidor web.

Em nosso exemplo, o logon deverá ser com conexão, e o painel que mostra os tweets, é desejável que também seja.

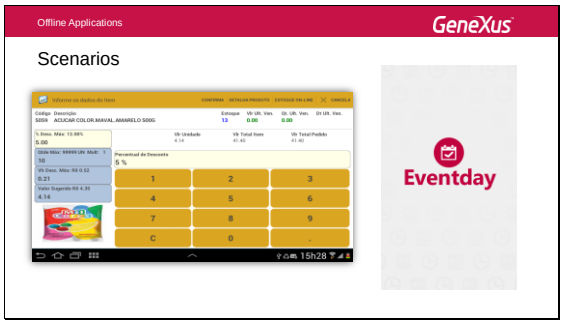
Outras tarefas podem, entretanto, executar-se sem conexão e depois sincronizar seus dados.

Portanto, devemos ser capazes de escolher quais objetos da aplicação podem ser executados off-line e quais não.

A essas aplicações chamamos PARCIALMENTE CONECTADAS, pois têm acesso a dados locais e a capacidade de executar lógica complexa no dispositivo.

Aplicações que uma vez instalada não necessita conectar ao servidor, nós as chamamos de DESCONECTADAS. É o caso de aplicações que lidam com dados pessoais e que não há nenhum interesse de enviar esses dados para nenhum servidor.

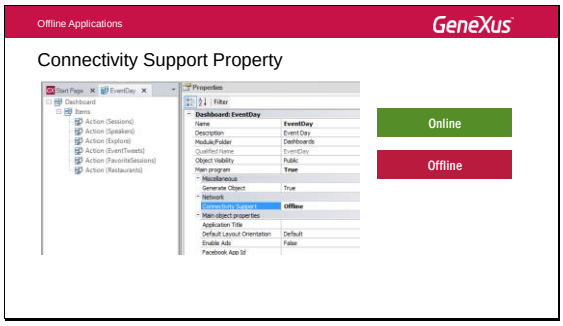
As aplicações parcialmente conectadas são as mais comuns. Como exemplos destas aplicações podemos mencionar os Pontos de Venda ou as aplicações para Eventos.



No caso dos Pontos de Venda, permite que os vendedores ou distribuidores que andam pelas ruas, possam emitir ordens de compra e faturamento, estejam ou não conectados. Então, ao obter conexão, sincronizam seus dados com os da empresa.

Ou, como no caso que estamos vendo, aplicações para eventos, onde os dados não mudam muito frequentemente, o banco de dados é carregado ao instalar a aplicação e, em seguida pode continuar usando a aplicação de forma desconectada. E por outro lado, há informações que requerem uma conexão frequente, como poder ler os tweets publicados.

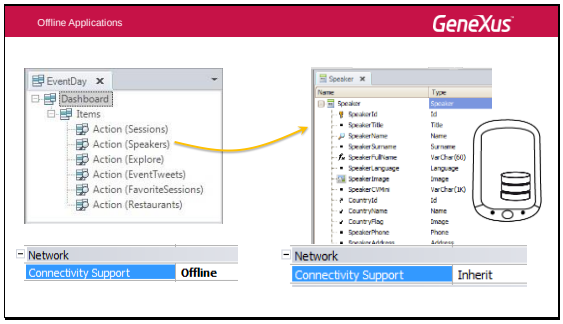
Ora bem: como se indica em GeneXus que se usará uma aplicação off-line?
Com a propriedade Connectivity Support, no nível de objetos SD que são Main.
No nosso exemplo, este objeto é o dashboard.



Esta propriedade pode ter o valor **On-line** ou **Off-line**, indicando como será a operação do objeto main.

Além disso, todos os objetos que não são main e são chamados pelo objeto main, têm a propriedade de **Inherit**, para indicar se herdam o comportamento do objeto principal que os invoca. Isto é válido para as transações que têm Business Components associados, os objetos WorkWithDevices, os painéis para SD, os Data Providers e procedimentos que estejam na árvore de invocações do objeto main.
Vejamos isto com um exemplo.

Suponha que temos uma aplicação para um evento.
O que acontece quando se configura a propriedade **Connectivity Support** do Dashboard EventDay com o valor Off-line?

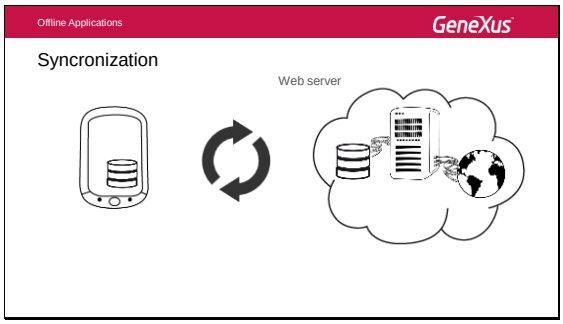


As transações associadas a business components usados em objetos WorkWithDevices chamados a partir do Main (diretamente ou indiretamente) e que tenham a propriedade Connectivity support com o valor **Inherit**, gerarão tabelas que serão criadas no banco de dados local do dispositivo. Além disso, serão criadas todas as tabelas que sejam necessárias de acordo com integridade referencial ou por atributos

mencionados nos painéis ou prompts, para garantir que a aplicação off-line funcione igual quando estava com conexão.

Naqueles objetos que queremos que estejam sempre conectados, configuramos a propriedade Connectivity support com o valor **On-line**.

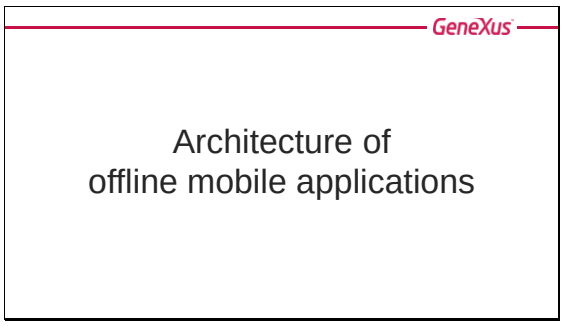
Quando se instala a aplicação no dispositivo, será criado o banco de dados no próprio dispositivo e também as tabelas que correspondam. Também se pode conseguir que a primeira vez que se estabeleça a conexão, se tragam os dados do servidor para o dispositivo, para preencher as tabelas.



Uma vez feito isso, se o dispositivo perde a conexão, o aplicativo continuará funcionando com os dados armazenados localmente. Uma vez que o dispositivo obtém conexão, a informação armazenada localmente se sincroniza com os dados que estão no servidor. Também os dados do servidor que mudaram são enviados para o dispositivo para serem atualizados, periodicamente ou sob demanda.

Depois, veremos que é possível alterar esse comportamento, por exemplo, você pode querer com você nunca se sincronize ou sincronizar sob demanda.

Agora vamos ver qual é a arquitetura das aplicações off-line.

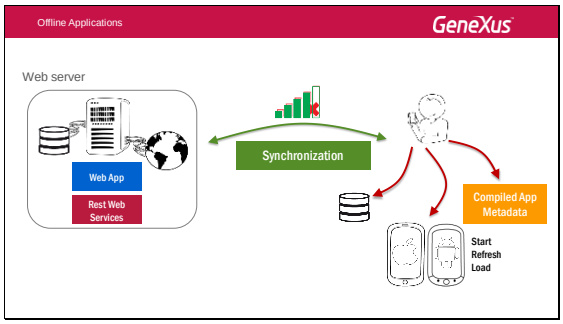


~~La arquitectura de las aplicaciones offline debe considerar dos situaciones:~~

~~Cuando la aplicación está conectada y cuando está desconectada.~~

~~Cuando la aplicación está conectada puede sincronizar los datos con el servidor mediante la invocación a los servicios REST.~~

~~Cuando la aplicación está desconectada, que ser capaz de procesar los datos e interactuar con una base de datos local, sin tener conexión con el servidor.~~



Uma aplicação off-line pode ser dividida em dois componentes:

• Um componente local que inclui um banco de dados local (normalizado) com um subconjunto das tabelas do servidor e toda a lógica necessária para ser executada localmente, e...

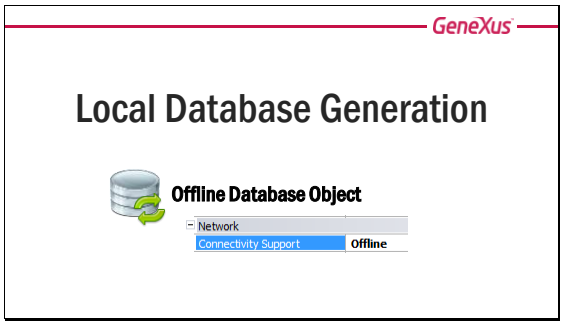
• Um componente do lado do servidor, onde se encontra o back-end da aplicação, o banco de dados e a camada de serviços para a sincronização com o dispositivo.

Ambos os componentes se comunicam via web services REST para a sincronização e para carregar o banco de dados local e, em seguida, enviar as modificações feitas localmente para o servidor.

Esta comunicação é conhecida como Sincronização.

Para que seja possível instalar o componente local no dispositivo, a aplicação deverá ser compilada, com o que incluirá toda a informação para acessar aos serviços REST.

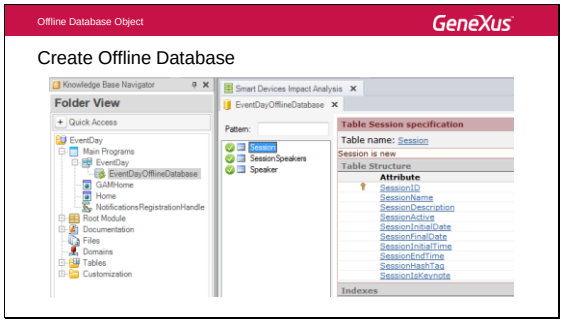
Para cada objeto main que tenha propriedade Connectivity Support em **Off-line**, será criado um objeto chamado **Off-line Database**.



Este objeto é responsável por determinar quais são as tabelas que vão para o BD local e também são quais são os dados que são transmitidos quando se sincroniza, porque obviamente o BD local não será o mesmo que o BD do servidor, mas uma versão reduzida.
O banco de dados que será criado no dispositivo é um banco de dados SQLite.

Depois de alterar a propriedade Connectivity support do objeto main de nossa aplicação para o valor off-line, deve-se executar a ação Rebuild All.

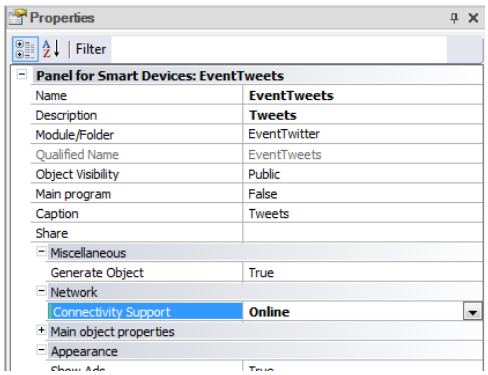
Quando se faz Build pela primeira vez do objeto main com a propriedade Connectivity support no valor **off-line**, é criado o objeto Off-line Database e é feita uma análise do impacto de quais tabelas serão criadas no dispositivo, associadas ao objeto de banco de dados off-line.



Vamos fazer isto em GeneXus.

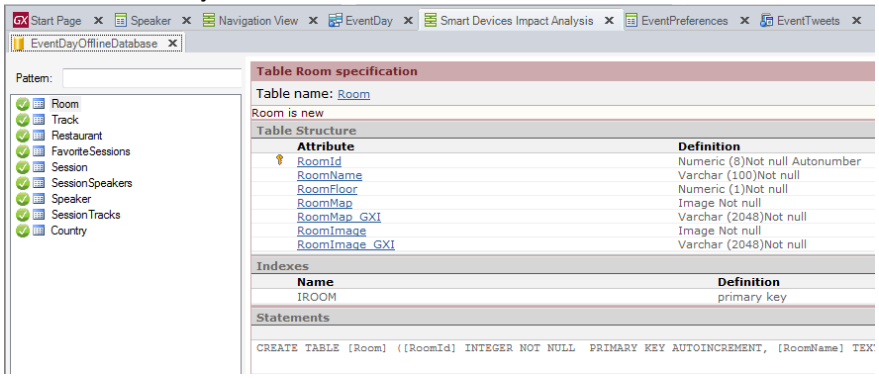
Abrimos o objeto dashboard EventDay e alteramos sua propriedade Connectivity support do valor padrão On-line para **Off-line**. Nossa intenção é que a aplicação inteira possa funcionar sem conexão à internet, exceto a tela que mostra os tweets, que queremos que funcione on-line, devido à velocidade com que essa informação é alterada.

Assim, abrimos o objeto EventTweets e modificamos a propriedade Connectivity Support, atribuindo o valor **On-line**.

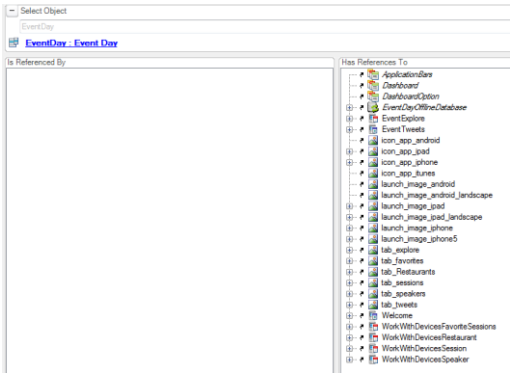


Agora clicamos no botão Rebuild All.

Vemos que nos são mostradas a análise de impacto com a criação do objeto EventDayOfflineDatabase e as tabelas a serem criadas no dispositivo, que neste caso serão todas, já que não alteramos o valor padrão da propriedade Connectivity Support de nenhuma transação e deixamos todas com o valor **Inherit**.



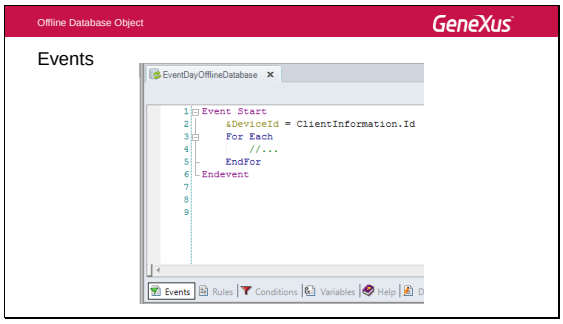
Observemos que a tabela de EventPreferences não está incluída.
Se clicarmos com o botão direito no objeto EventDay e selecionamos a opção References, vemos quais objetos chama o dashboard e não encontramos que se chame o business component EventPreferences direta ou indiretamente.



Quando se instala a aplicação no dispositivo se executa a criação de tabelas e são copiados os dados do servidor, se houver conexão.

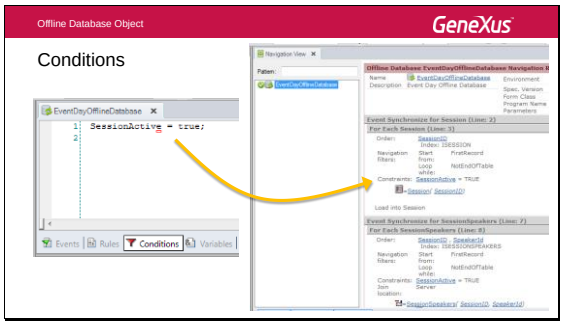
Voltemos para o objeto Database Off-line

É um objeto simples que só tem eventos, condições e propriedades.

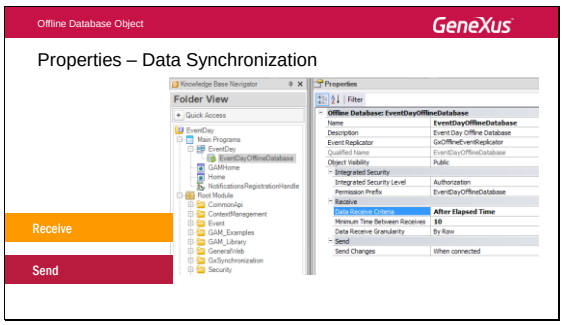


O único evento que pode ser programado é o evento de Start. Este evento é executado no servidor, antes de cada envio de dados ao cliente para sincronização.

O evento de Start destina-se à inicialização de variáveis e algum outro procedimento que deve ser feito antes da sincronização de tabelas.



Os filtros que incluímos na guia Conditions são utilizados como em qualquer outro objeto GeneXus. São filtros que se aplicam às tabelas para descobrir quais os dados são transportados ao dispositivo. São globais, portanto, aplicam-se a todas as tabelas que corresponda.
Nas condições pode-se utilizar variáveis que podem ser carregadas no evento Start.

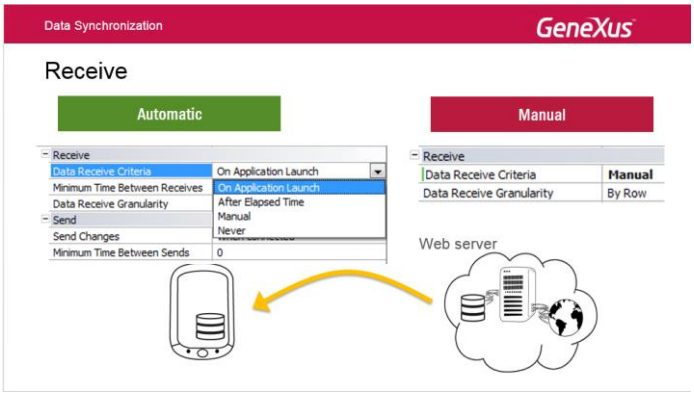


O objeto OfflineDatabase inclui propriedades que podemos utilizar para definir quando enviar ou receber dados de e para o servidor, respectivamente.

-Receber, que define como e quando receber um subconjunto de dados do servidor.

E em

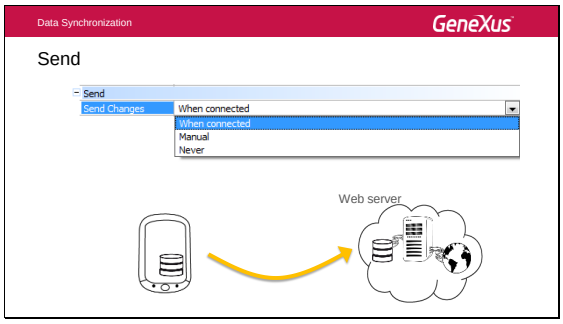
-Enviar, que define como e quando os dados são enviados do dispositivo para o servidor.



Ao receber dados do servidor, há duas maneiras para executar a sincronização:

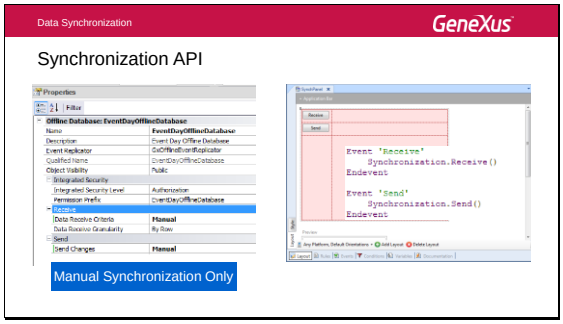
-**Automaticamente**, através do qual se pode configurar quando realizar a sincronização, se ao abrir a aplicação pela primeira vez e tendo passado certo tempo desde a anterior, ou sempre, ou a cada certo tempo, ou nunca.

- Ou de forma **manual**, na qual se deve programar por código a maneira para receber dados do servidor.



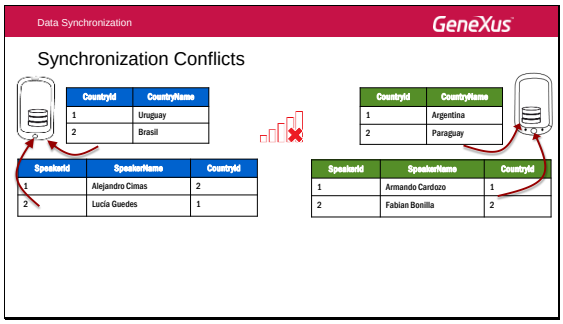
O envio de dados para o servidor também pode ser configurado se será automático, quando o dispositivo obtém a conexão, ou manual que dependerá do que programemos para fazê-lo, ou nunca, se não se deseja carregar os dados do dispositivo para o banco de dados centralizado.

No caso em que tanto o Receive quanto o Send seja feito manualmente, deve ser usada a Synchronization API para realizá-los. Esta API não se encontra como objetos dentro da pasta SmartDevicesAPI, mas é parte da gramática.



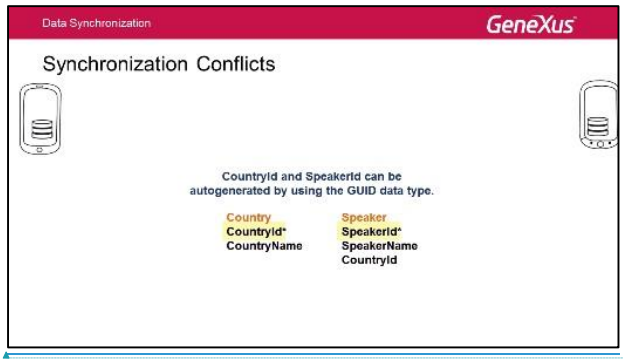
Como exemplo, pode-se fazer um painel SD que se executa a partir do dispositivo, onde se programa o send e o receive, como vemos na imagem.

Quando se enviam ou recebem dados do dispositivo, pode haver conflitos de sincronização já que a aplicação pode estar instalada em vários dispositivos e cada um deles pode inserir dados com os mesmos identificadores, mas para diferentes dados.



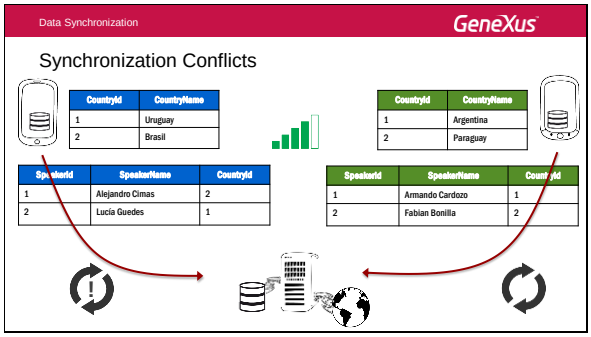
Vejamos isto em nosso exemplo.

Na KB EventDay, temos a transação Country com os atributos CountryId e a transação Speaker, que tem o atributo CountryId como chave estrangeira. ~~Tanto CountryId como SpeakerId están definidas como Autonumber.~~

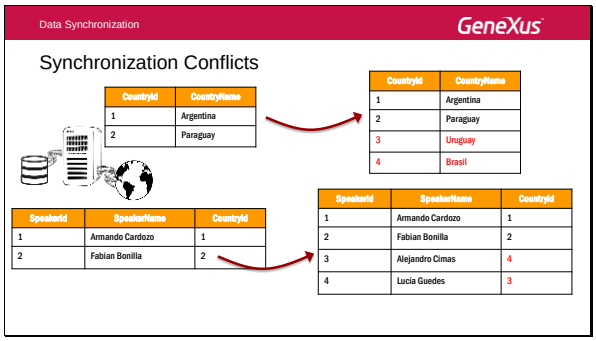


Formatted: Font: (Default) Calibri, 12 pt, Kern at 12 pt

Por exemplo, aqui vemos que um dispositivo criou os países Uruguai com Id = 1 e Brasil com Id = 2, enquanto que em um segundo dispositivo se inseriram os países Argentina com Id = 1 e Paraguai com Id = 2. Ambos os dispositivos estão trabalhando de forma off-line, sem conexão, com o que todos os dados são armazenados no banco de dados local de cada dispositivo.



Depois de conseguir a conexão, um dos dispositivos é sincronizado com o servidor, enviando todas as operações realizadas através de Business Components no banco de dados local.
Quando o segundo dispositivo tenta sincronizar dá conflito porque se repete a chave do país.
Como se usam chaves autonumeradas, GeneXus resolve o conflito.

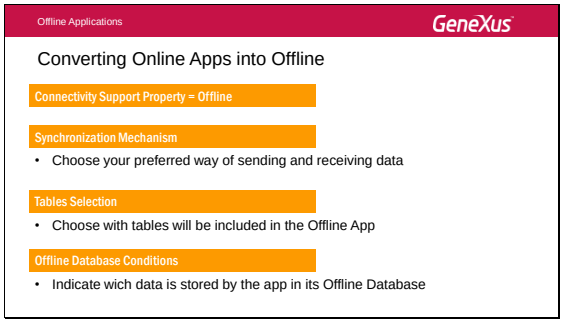


Como se faz isso? Resolvendo o conflito no servidor.

GeneXus gera uma nova chave para os países que são repetidas, e atualiza os dados da tabela servidor com esse Id gerado novamente.

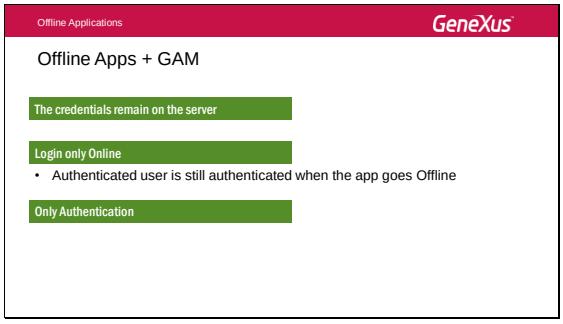
Em seguida atualiza a tabela de oradores para que as chaves estrangeiras sejam consistentes.
Por último, atualiza os dados no dispositivo, para que no dispositivo, estejam também as chaves corretas.

Resumindo o que nós vimos, para converter uma aplicação on-line em uma aplicação off-line, devemos seguir os seguintes passos:



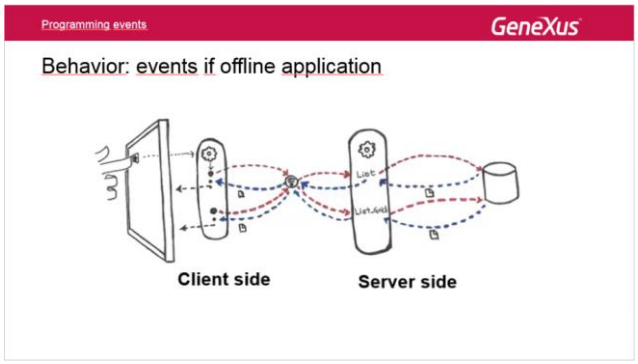
1. Configurar a propriedade Connectivity Support no valor Off-line
2. Selecionar o mecanismo de sincronização
3. Indicar quais tabelas criará no dispositivo
4. E adicionar condições para filtrar que registros vão ao dispositivo

Se tivermos uma aplicação Off-line que utiliza o GAM, deve ser levado em conta que as credenciais estarão sempre no servidor, portanto o login só pode ser feito estando On-line.



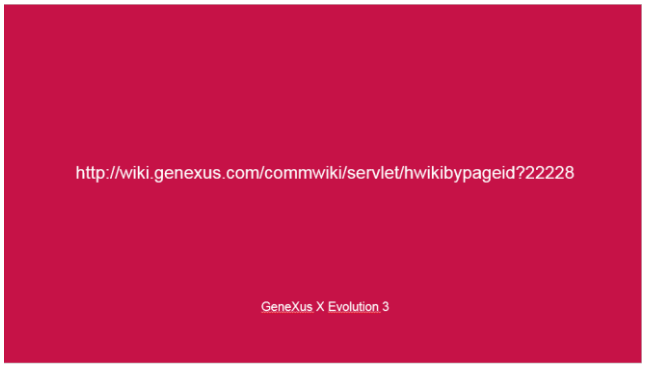
Uma vez que fizer logon, sim se pode trabalhar Off-line.

A programação dos eventos em uma aplicação off-line tem certas considerações que a diferenciam das aplicações on-line.



Veremos os eventos de uma aplicação off-line em um próximo vídeo.

Podemos obter mais informações sobre o tema Off-line, no endereço que é exibido na tela:
http://wiki.genexus.com/commwiki/servlet/hwikibypageid?22228



Video filmado con GeneXus X Evolution 3