

GeneXus[™]
The power of doing.

Offline Database Generation

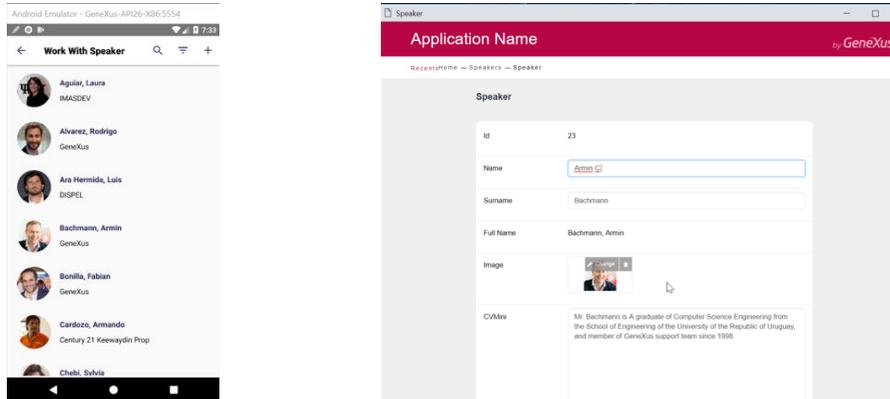
Development – Offline Applications

GeneXus™ 16

Demo: Offline Database Generation

En este video veremos como convertir la aplicación EventDay en una aplicación Offline.

Demo: Offline Database Generation



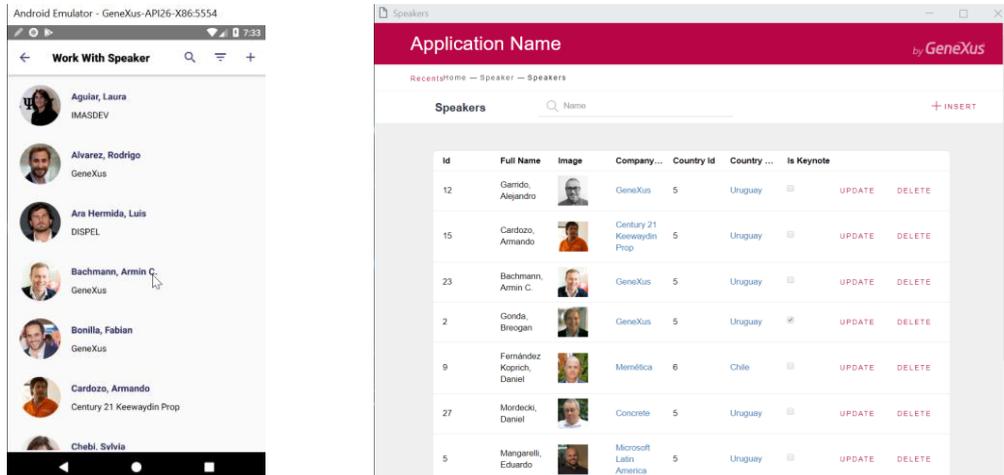
Primero vamos a ver como funciona la aplicación actualmente, aun es una aplicación Online.

Vamos a acceder a la aplicación y vamos a acceder a la aplicación web también. Vamos a tener ambas interfaces abiertas así vemos como es la interacción.

En web vamos a acceder al Work With de Oradores, y también en la aplicación SD.

Ahora en web elegimos uno, por ejemplo Armin y vamos a editarlo, vamos a agregarle una letra al final del nombre y grabamos.

Demo: Offline Database Generation



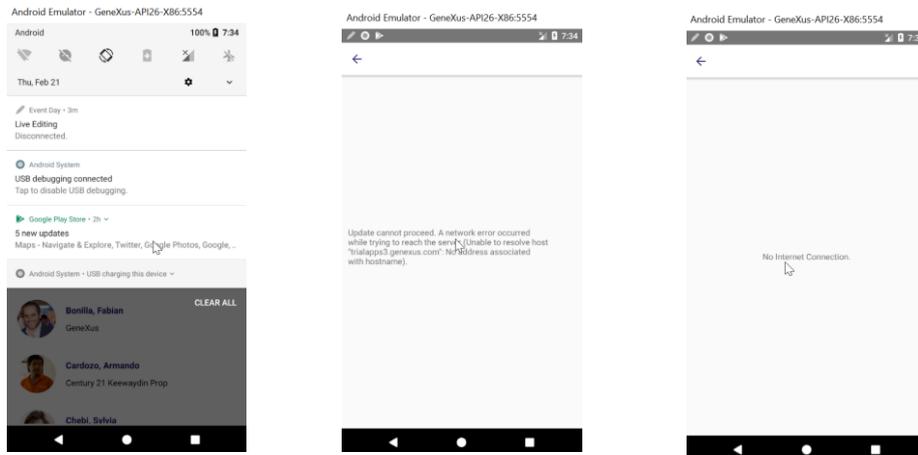
Ahora vamos a la aplicación SD, refrescamos y automáticamente vemos el cambio realizado.

Que pasa si lo editamos desde el dispositivo, vamos a cambiar la ultima letra que le agregamos, grabamos.

Refrescamos el panel web y automáticamente vemos el cambio.

Esto es porque la aplicación trabaja en forma conectada.

Demo: Offline Database Generation



Que pasa por ejemplo si ahora desconecto la aplicación, vamos a simular esto, apagamos el acceso Wifi y apagamos los datos. Ahora el emulador no posee conexión a internet, por lo que no puede acceder a la capa de servicios REST.

Si tratamos de ingresar a Armin de nuevo, vamos a poder porque estos datos están cacheados, pero si intento editarlo me da un error.

Tratemos de ingresar a otro orador que no abrimos antes.

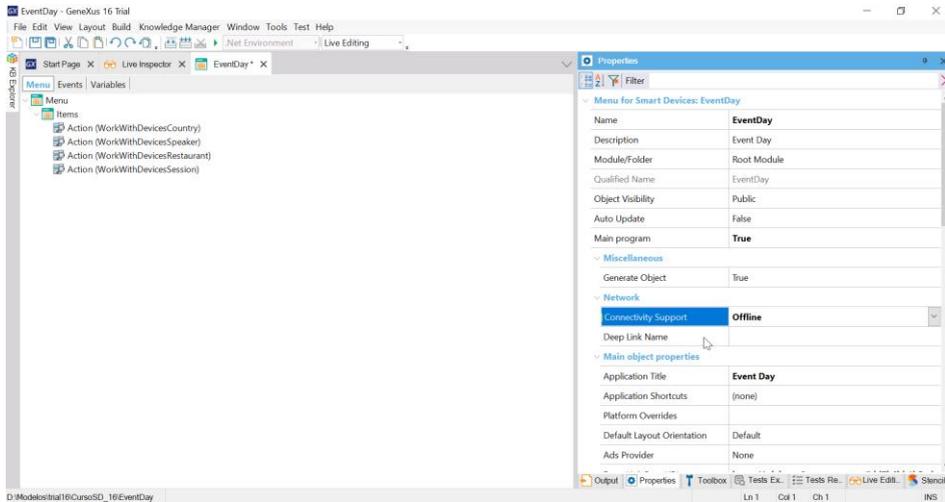
Vean que no podemos, ya que no tenemos conexión.

Incluso no puedo entrar al list de restaurants.

Cuando estamos sin conexión en una aplicación Online, solo tenemos acceso a datos están cacheados en la aplicación, pero no podemos acceder a nuevas pantallas.

Bien, vamos ahora a GeneXus así convertimos la aplicación en una aplicación Offline.

Demo: Offline Database Generation



Lo que vamos a hacer es ir al objeto EventDay, que es el objeto main de la aplicación.

En las propiedades vamos a especificar en la propiedad Connectivity Support el valor Offline. Salvamos y hacemos un build de la aplicación.

Demo: Offline Database Generation

Table Session specification

Table name: Session

Session is new

Table Structure

Attribute	Definition	Previous values	Takes value from
SessionId	Numeric (8), Not null		
SessionName	Varchar (160)		
SessionDescription	Long varchar		
SessionActive	Character (1)		
RoomId	Numeric (8)		
SessionInitialDate	Date, Not null		
SessionFinalDate	Date, Not null		
SessionInitialTime	Datetime		
SessionDuration	Numeric (3)		
SessionHashtag	Varchar (40)		
SessionIsKeyNote	Boolean, Not null		

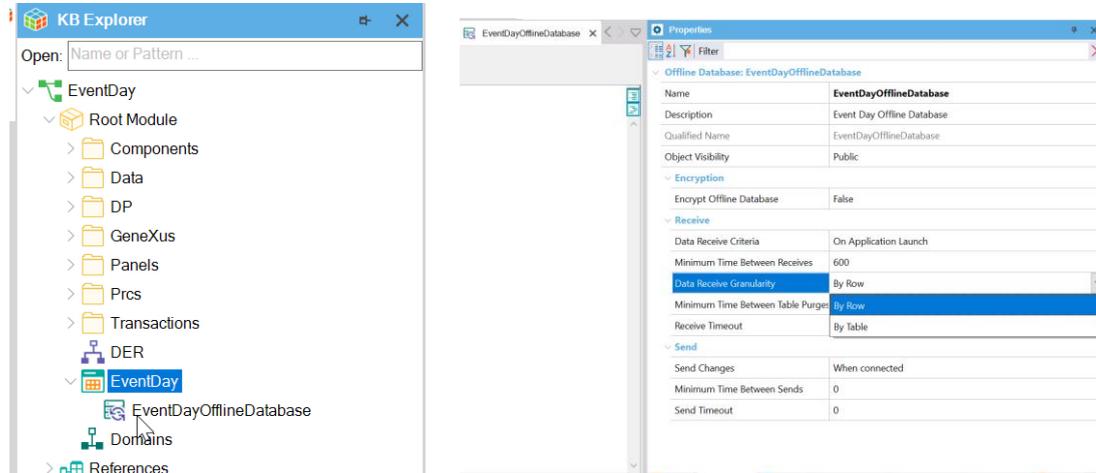
Indexes

Name	Definition	Composition
ICHARLA	primary key	SessionId
USESSION	unique	SessionName

0 Errors 0 Warnings 11 Success

Una de las cosas que veremos a continuación es un análisis de impacto de la base de datos Offline, estas serán las tablas que se llevarán y se crearán en el dispositivo. Ya veremos cuáles son y cómo se determinan.

Demo: Offline Database Generation



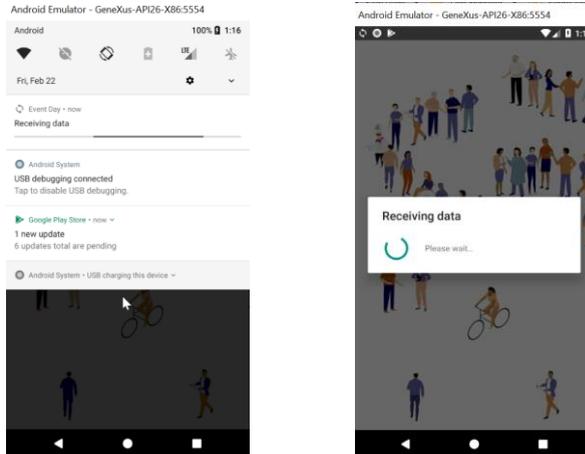
Vamos a ver que ahora tendremos un objeto nuevo, EventDayOfflineDatabase que queda asociado al objeto EventDay.

Este objeto nos permitirá configurar ciertas cuestiones relativas a la base de datos Offline, la que se generará en el dispositivo, como la granularidad.

Luego estudiaremos estas propiedades mas detalladamente.

Además este objeto tiene Eventos, Reglas, Conditions y variables, luego veremos como se utilizan, por ahora dejaremos todo default.

Demo: Offline Database Generation



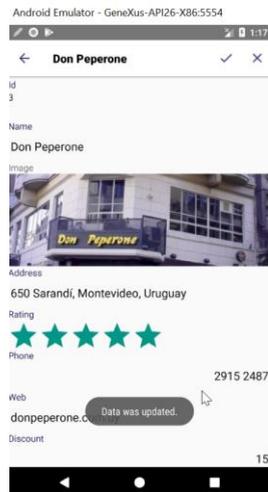
Bien. Ya tenemos la aplicación en el emulador.

Vamos a ver que ahora cuando se lanza la aplicación, podemos ver en la barra de estado que se esta ejecutando la sincronización.

Y en el panel también nos va a indicar esta situación.

Una vez que termina el proceso, podremos utilizar la aplicación normalmente.

Demo: Offline Database Generation



Por ejemplo si accedemos a los oradores, veremos la lista.
Vamos a desconectar el dispositivo.

Y ahora intentemos ingresar a restaurants, y no tenemos el problema que teníamos antes.
Porque los datos están en el dispositivo.
Incluso puedo editar un registro, vamos a poner una estrella mas y grabamos.

Para el usuario no hay ninguna diferencia.

Demo: Offline Database Generation



Speakers

Application Name by GeneXus

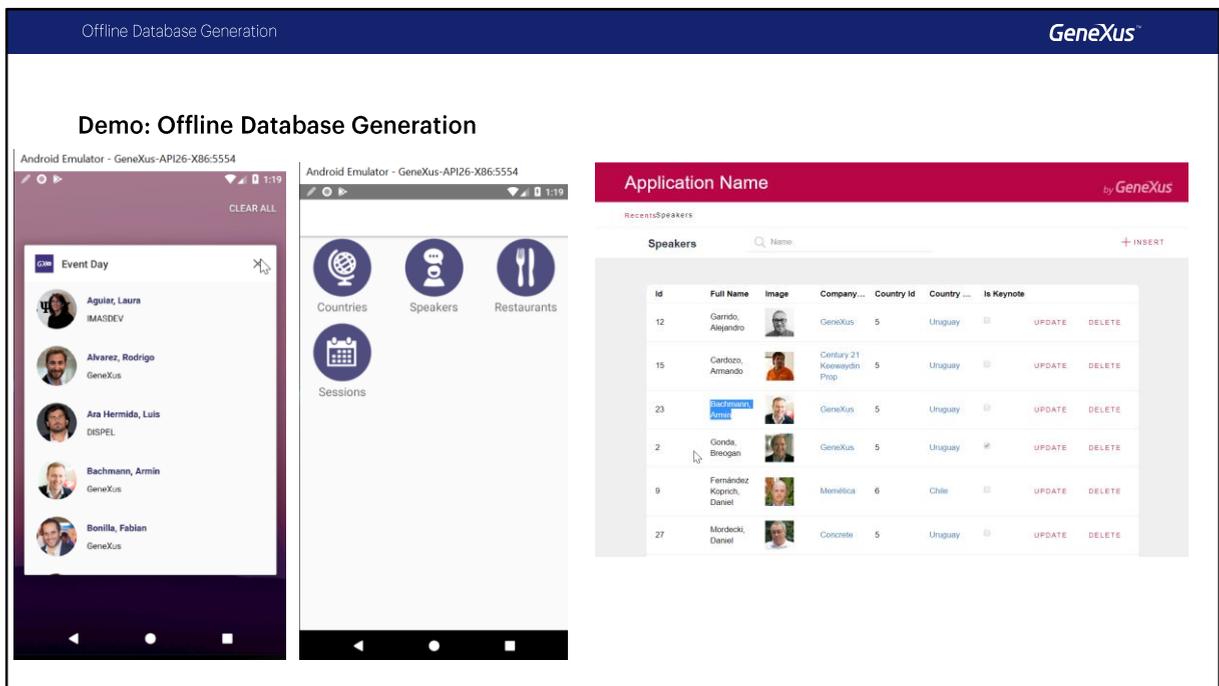
RecentHome - Speaker - Speakers

Speakers 🔍 Name + INSERT

Id	Full Name	Image	Company...	Country Id	Country ...	Is Keynote
12	Garido, Alejandro		GeneXus	5	Uruguay	UPDATE DELETE
15	Cardozo, Armando		Century 21 Kowwaydin Prop	5	Uruguay	UPDATE DELETE
23	Bachmann, Armin C.		GeneXus	5	Uruguay	UPDATE DELETE
2	Gonda, Breogan		GeneXus	5	Uruguay	UPDATE DELETE
9	Fernández Koprich, Daniel		Memética	6	Chile	UPDATE DELETE
27	Mordecki, Daniel		Concrete	5	Uruguay	UPDATE DELETE
5	Mangarelli, Eduardo		Microsoft Latin America	5	Uruguay	UPDATE DELETE

Vamos a editar un orador, vamos a ir a Speakers, seleccionamos a Armin de vuelta, vamos a editarlo y vamos a quitar la letra que habíamos agregado antes en el nombre y grabamos. Noten que no tenemos ningún problema por estar offline. los datos son grabados siempre en el dispositivo.

Y si vamos a la interfaz web y refrescamos no veremos ese cambio. Porque el dispositivo esta offline.



Vamos a volver a conectar el emulador a la red.

Y para forzar la sincronización voy a cerrar y abrir la aplicación de nuevo, esto es porque no configuramos el objeto OfflineDatabase, así no esperamos.

Abrimos de nuevo la aplicación. Esto ejecuta la sincronización.

Y ahora si, vamos a la web y actualizamos y ahí vemos el cambio.

Terminamos aquí esta Demo básica, donde vimos lo simple y fácil que es crear una aplicación Offline en GeneXus.

Además hemos visto el proceso de sincronización de los datos entre el dispositivo trabajando en forma Offline y el servidor web y las operaciones de CRUD que siempre se realizaran sobre la base de datos local.

volvamos a la presentación.

Offline Database Generation

The screenshot shows the GeneXus IDE interface. On the left, a project tree under 'Network' shows 'EventDayOfflineDatabase' selected. A blue arrow points from this object to the 'Table Session specification' dialog on the right. The dialog displays the following table structure:

Attribute	Definition	Previous values	Takes value from
SessionId	Numeric (8), Not null		
SessionName	Varchar (160)		
SessionDescription	Long varchar		
SessionActive	Character (1)		
RoomId	Numeric (8)		
SessionInitialDate	Date, Not null		
SessionFinalDate	Date, Not null		
SessionInitialTime	Datetime		
SessionDuration	Numeric (3)		
SessionHashtag	Varchar (40)		
SessionIsKeyNote	Boolean, Not null		

Below the table structure, the 'Indexes' section shows:

Name	Definition	Composition
ICHARLA	primary key	SessionId
USESSION	unique	SessionName

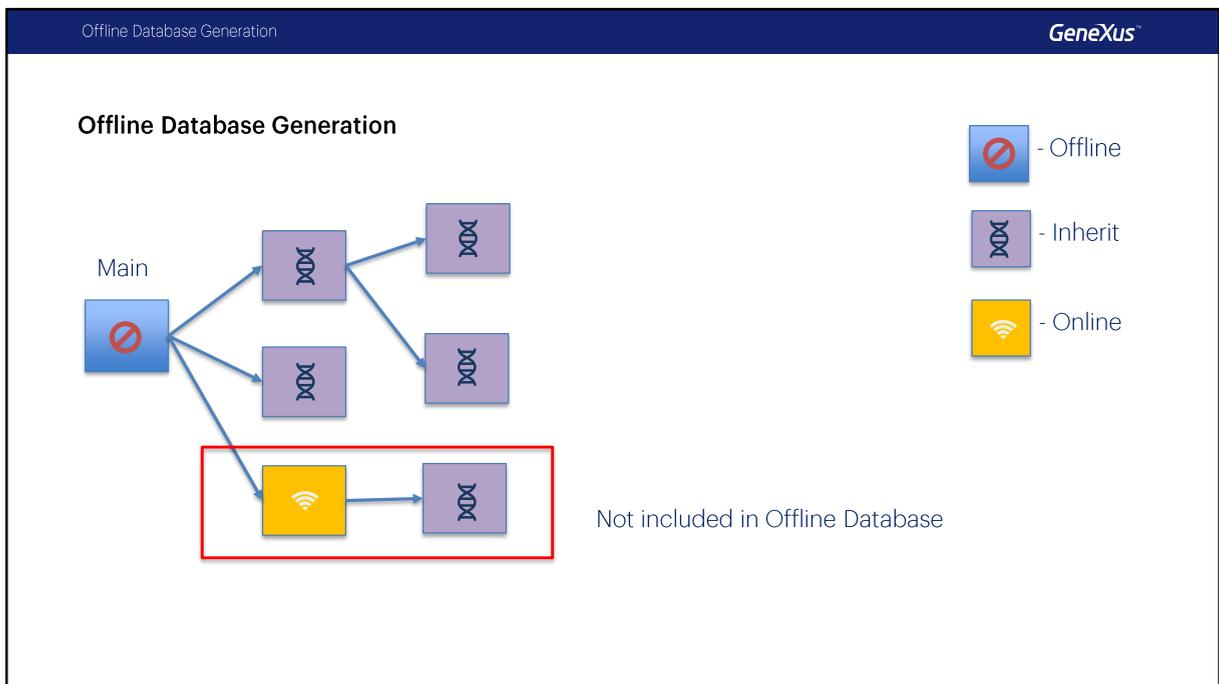
Como pudimos ver recién en la demo, cuando indicamos que el objeto main era Offline y realizamos el Build, se genero automáticamente un objeto OfflineDatabase asociado al objeto Main.

Vimos también que nos presento un análisis de impacto, similar a cuando hacemos el créate de la base de datos, pero estas tablas que se muestran son las tablas que analizo GeneXus que debe llevar a la base de datos SQLite en el dispositivo.

Al crearse el objeto OfflineDatabase, se crean también en el lenguaje nativo del dispositivo, los programas para crear esta base de datos local.

Este objeto es el encargado de determinar cuándo se produce la sincronización, cuáles son las tablas que se crearán en la base de datos local y también cuáles son los datos que se llevan a las mismas cuando se sincronizan con las tablas del server.

En el objeto OfflineDatabase además es donde se configuran las propiedades que vimos para el Send y el Receive.



Ahora bien, ¿Qué tablas efectivamente se llevan cuando una aplicación pasa a ser offline?

Supongamos que tenemos este árbol de llamadas desde el objeto Main, podemos tener Objetos con la propiedad Connectivity Support Offline, como el Main, otros con el valor default que es Inherit, que hereda el tipo de acuerdo al objeto que lo llame y otros que pueden ser Online.

Si tenemos este árbol de invocaciones, entonces las tablas que se van a llevar al dispositivo son todas las de los objetos Offline y los que heredan la conectividad offline del que los invocó.

Pero también se va a contemplar la integridad referencial y los atributos mencionados (tanto en paneles como en prompts).

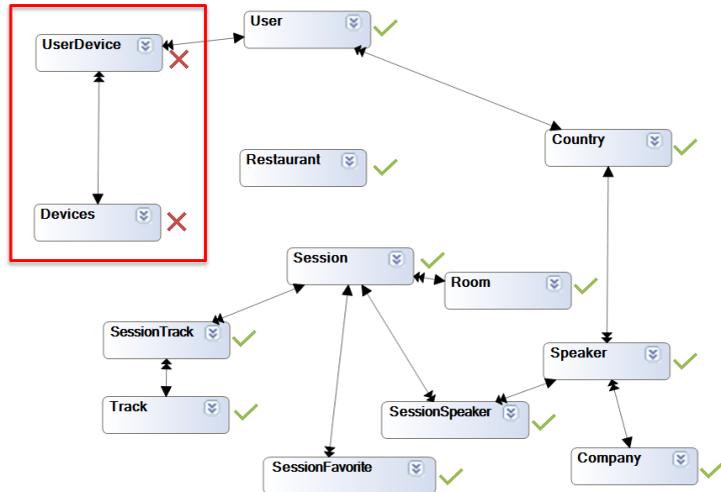
Este objeto que es Online que llama a este otro que hereda el tipo de conexión no se incluirían, van a ser online.

Offline Database Generation

EventDayOfflineDatabase

Pattern:

- Session
- Country
- Speaker
- User
- Restaurant
- SessionTrack
- SessionSpeaker
- Room
- Track
- Company
- SessionFavorite



En nuestro caso se van a llevar todas las tablas menos UserDevice y Devices que son tablas que aun no utilizamos desde la aplicación.

Con esto terminamos el tema.

En el próximo video veremos mas detalles sobre la sincronización y las propiedades del objeto OfflineDatabase

GeneXus™

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications