

## Offline Mobile Applications with GeneXus (Part II)

### Offline mobile applications with GeneXus

GeneXus 15

Hasta ahora vimos qué son las aplicaciones conectadas, parcialmente conectadas y Offline.

¿Pero cómo hacemos en GeneXus para construir una aplicación Offline?

Si queremos que una aplicación se ejecute en forma Offline, debemos asignar la propiedad Connectivity Support de su objeto main, en el valor Offline. Este valor habilita la generación de la aplicación en forma Offline y se generará el código nativo necesario para que la aplicación pueda ser ejecutada sin invocar a los servicios REST del servidor web.

A partir de que a un objeto main se le asigna la propiedad Connectivity Support en Offline, se instala la aplicación en el device, se crea la base de datos local y se crean ahí esas tablas.

Así, como el dashboard EventDay de nuestra aplicación es nuestro objeto main, modificamos su propiedad Connectivity support pasándola a Offline.

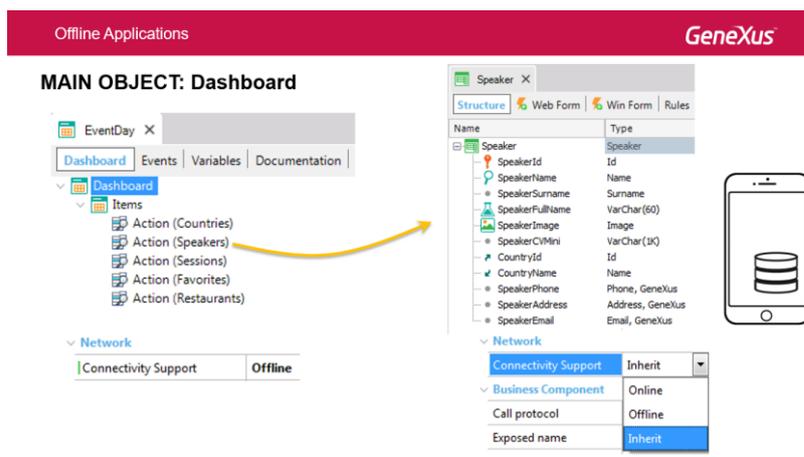
The screenshot shows the GeneXus 15 development environment. At the top, there's a red header with 'Offline Applications' on the left and the 'GeneXus' logo on the right. Below the header, the text 'How to create an Offline app in GeneXus ?' is displayed. The main area shows a tree view of the 'EventDay' project. The 'Dashboard' object is selected and labeled as the 'MAIN OBJECT'. Under 'Dashboard', there are several 'Action' objects: 'Action (Countries)', 'Action (Speakers)', 'Action (Sessions)', 'Action (Favorites)', and 'Action (Restaurants)'. At the bottom, the 'Network' section is expanded, showing the 'Connectivity Support' property set to 'Offline'.

La propiedad Connectivity Support está también a nivel de objetos que no son main. Su valor puede ser Online, Offline o Inherit.

Todas las tablas que tengan la propiedad Connectivity support = Inherit, utilizadas en objetos SD invocados desde el Main (directa o indirectamente), se van a crear en la base de datos local del dispositivo. A no ser que se les configure que sean Online, en cuyo caso no se creará la tabla en la base de datos local porque se accederá a la tabla del servidor a través de los servicios REST.

El valor Inherit solamente aparece en objetos que no son main.

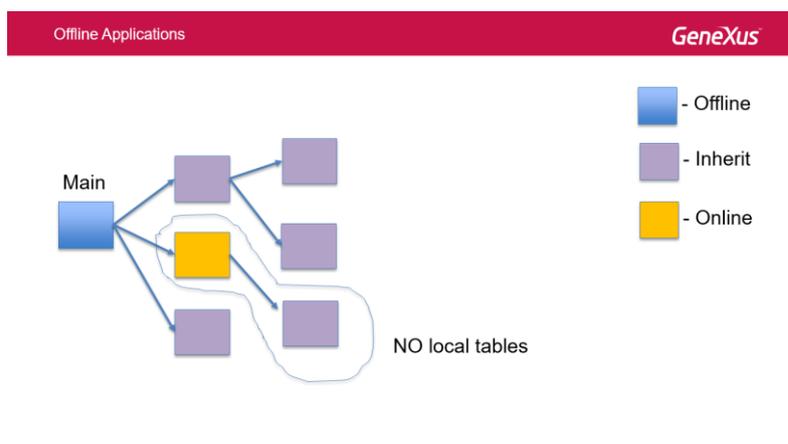
También es posible asignar a un objeto la propiedad Connectivity Support en Offline, en cuyo caso se crearán tablas en la base de datos local y el objeto trabajará sin conexión, aún en el caso de que el objeto main trabaje Online.



Ahora bien, ¿qué tablas efectivamente se llevan cuando una aplicación pasa a ser Offline?

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).

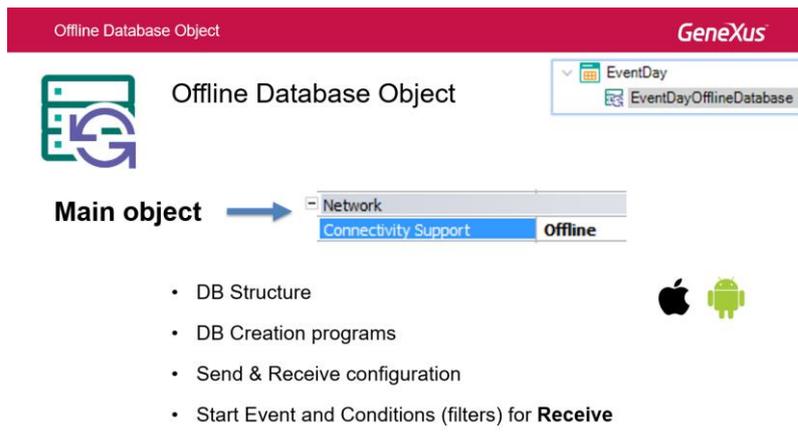


Para cada objeto main que tenga la propiedad Connectivity Support en el valor Offline, se crea un objeto llamado Offline Database. Éste se crea la primera vez que se hace un build sobre el dicho objeto main luego de cambiar la propiedad Connectivity Support al valor Offline.

Al crearse, se crean también en el lenguaje nativo del dispositivo, los programas para crear la 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.

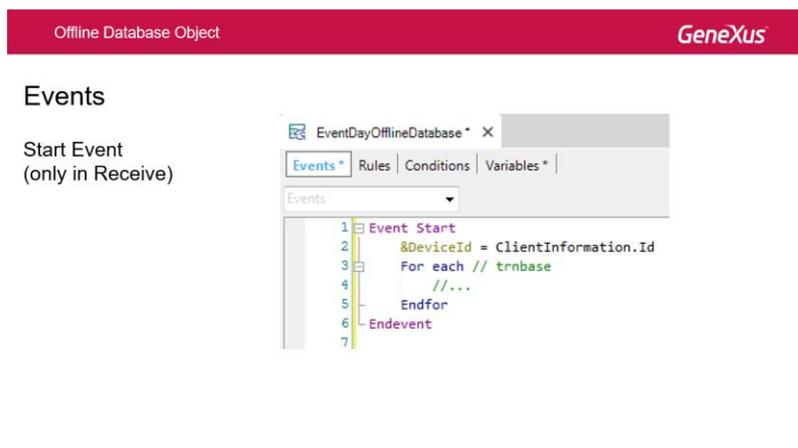
El objeto Offline Database dispone de eventos, condiciones y propiedades que permiten determinar su comportamiento.



El único evento que se puede programar es el Start Event.

Se ejecuta en el Server, antes de que cada envío de datos al cliente suceda, para una sincronización, es decir únicamente en el Receive.

Este evento está pensado para inicialización de variables y algún otro procesamiento que se debe hacer antes de la sincronización de tablas.



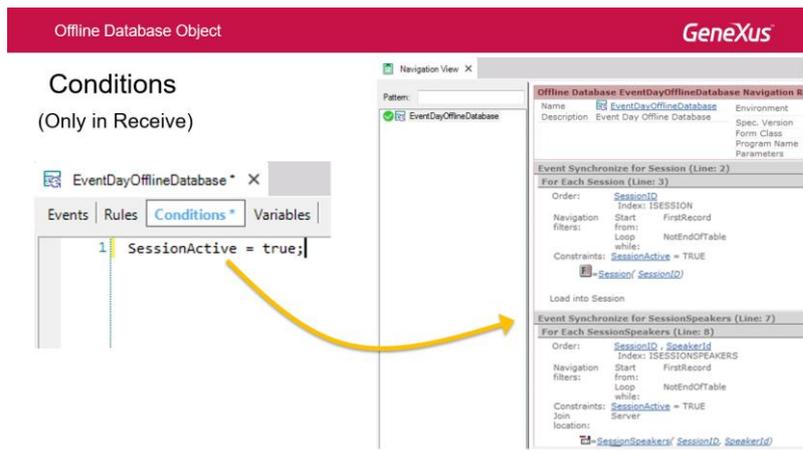
El uso de las Conditions es, al igual que en cualquier otro objeto GeneXus, para definir filtros sobre los datos y solamente se utilizan para el Receive.

Los filtros se aplican a las tablas del server, para saber qué datos se llevan al dispositivo. Esto implica que los datos en la base de datos local pueden ser un subconjunto de los datos de la

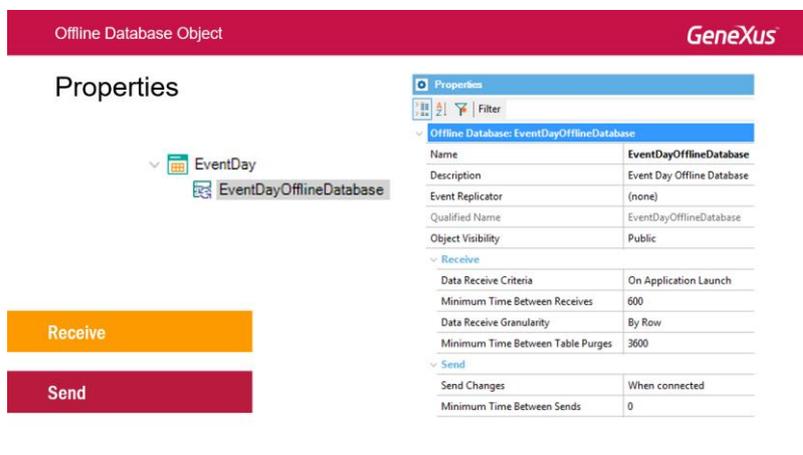
base de datos del server y como caso particular, que no hayan registros de una tabla específica en el dispositivo.

Son independientes y se aplican a la tabla extendida, por lo que GeneXus tiene la inteligencia de determinar sobre qué tablas se aplican. En el listado de navegación se pueden ver las condiciones que se aplican a cada tabla.

Para las expresiones de las condiciones se pueden utilizar variables predefinidas o definidas por el desarrollador, en cuyo caso deben asignarse en el evento Start.



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



Para finalizar algunas puntualizaciones....

Si se tiene una aplicación Offline que usa GAM, hay que tener en cuenta que las credenciales siempre estarán en el server, por lo tanto el login sólo se puede hacer estando Online.

Una vez que se loguea, sí se puede trabajar en forma Offline.

En aplicaciones móviles no está implementada la autorización, sino solamente la autenticación.

## Offline Apps + GAM

The credentials remain on the server

Login only Online

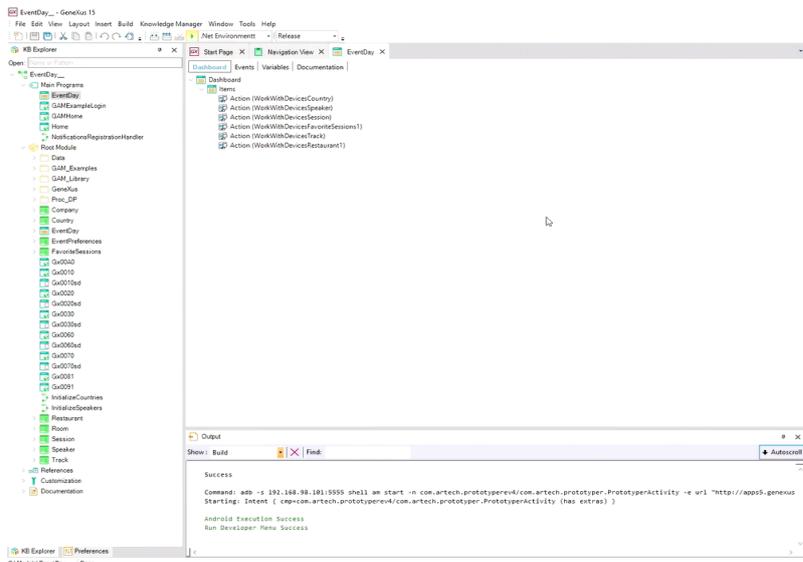
- Authenticated user is still authenticated when the app goes Offline

Smart Devices: Only Authentication

A continuación, veremos una breve demo de cómo construir una aplicación Offline en GeneXus.

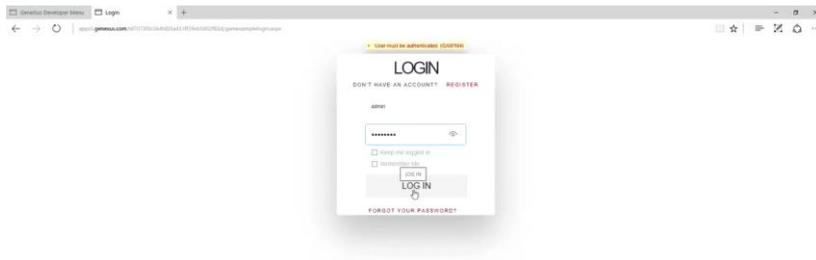
Lo que voy a hacer ahora es ejecutar la aplicación. En definitiva, ¿qué es lo que voy a mostrar?

Que como tengo la aplicación todavía online, cuando hago un por la interface web voy a ver el cambio en el dispositivo, y si cambio algo en el dispositivo lo voy a ver actualizado en la interface web; porque la aplicación está online, o sea que el dispositivo está sincronizado con los servicios REST, o sea que está viendo la base de datos de la aplicación.

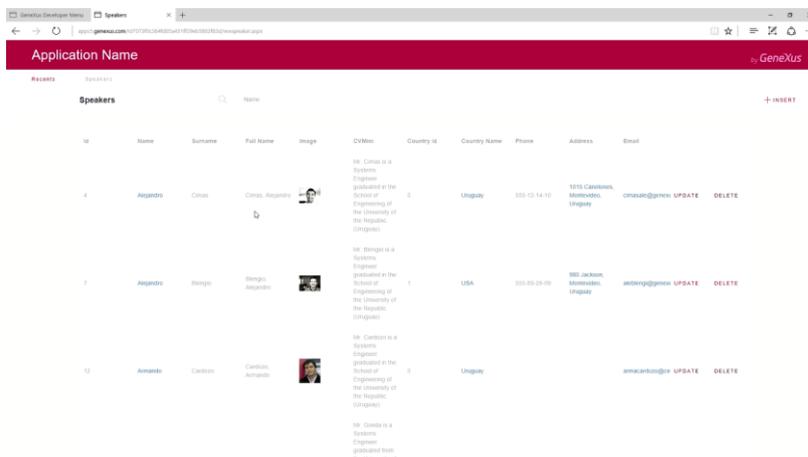


Después lo que voy a hacer es convertirla a una aplicación online.

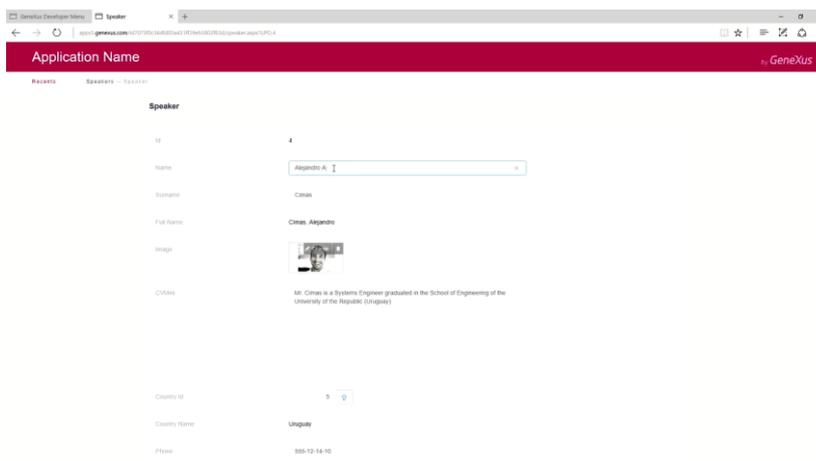
Bueno, me estoy logueando en la aplicación web...



Vamos a ver algún dato...

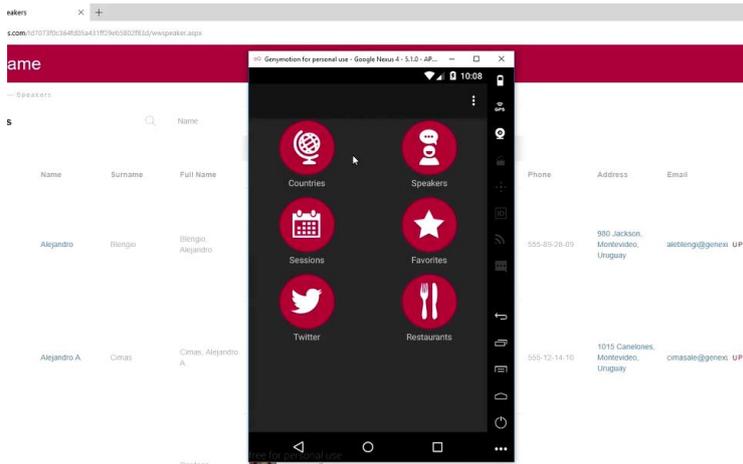


Acá por ejemplo Alejandro Cimas... y vamos a cambiarle Alejandro; vamos a suponer que su segundo nombre empieza con A...

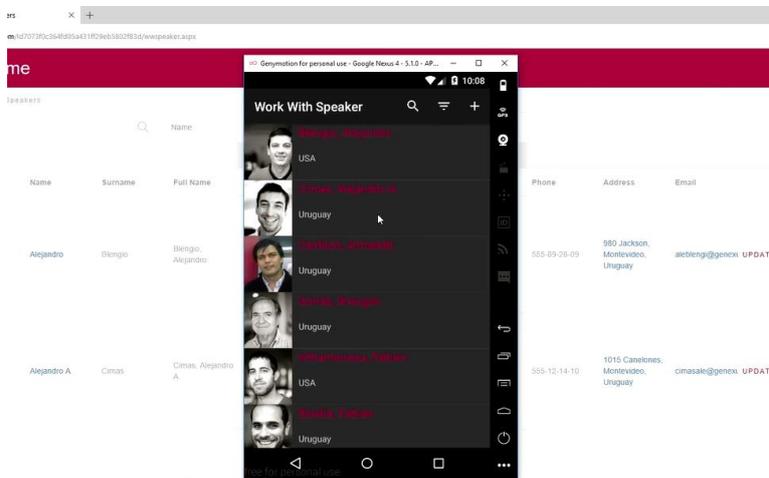


Y salvamos.

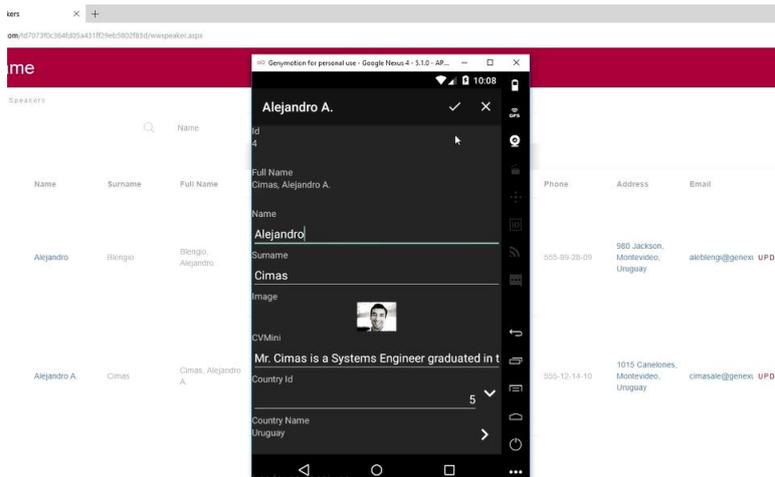
Ahora vamos a la aplicación Android...que estoy ejecutando con el KBN...



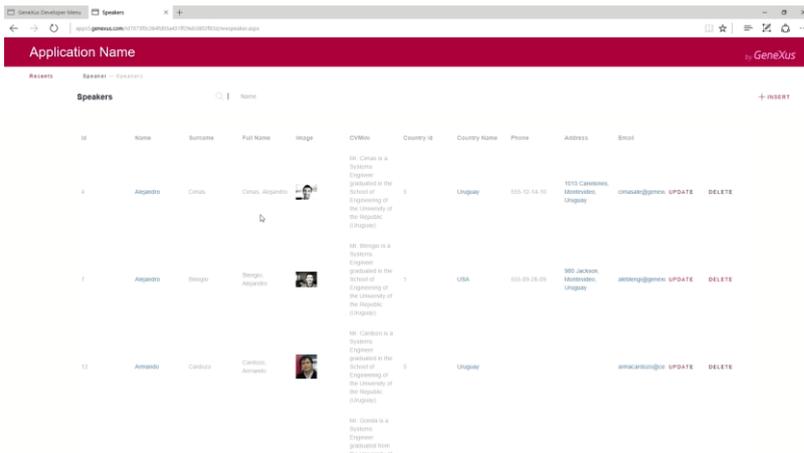
Pero ven que acá apareció Alejandro A. Cimas, o sea que el cambio fue automático:



Si yo voy aquí, y edito el valor...

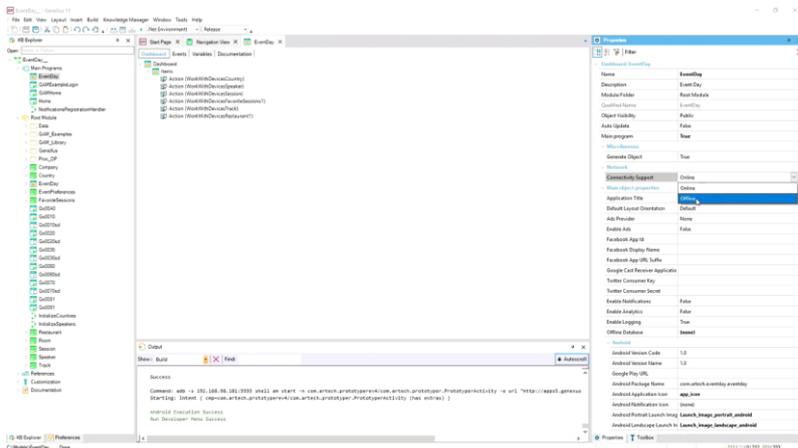


Y luego vamos aquí, al web, vemos que automáticamente quedó hecho el cambio:



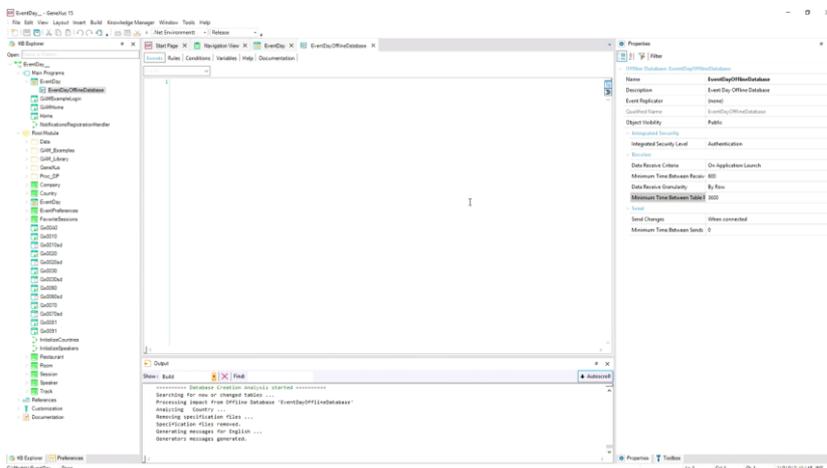
Lo que voy a hacer ahora es convertir esta aplicación en una aplicación Offline.

Para eso, vamos a GeneXus, y en mi objeto Main, que en este caso es este menú EventDay, voy a cambiar la propiedad Connectivity Support al valor Offline.



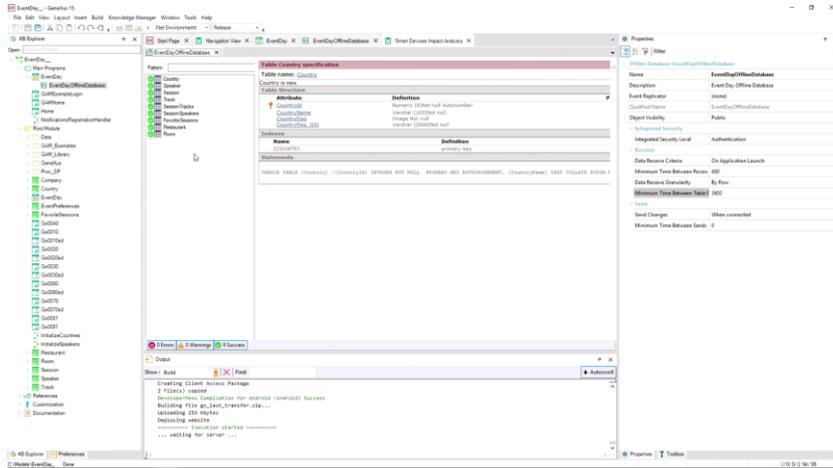
Y después voy a hacer un Run.

Vemos que se creó, bajo el EventDay, bajo mi Dashboard, el objeto EventDayOfflineDatabase, que es el objeto en el cual yo puedo cambiar todos los criterios para hacer la recepción o hacer el envío de datos hacia IDEs del server.

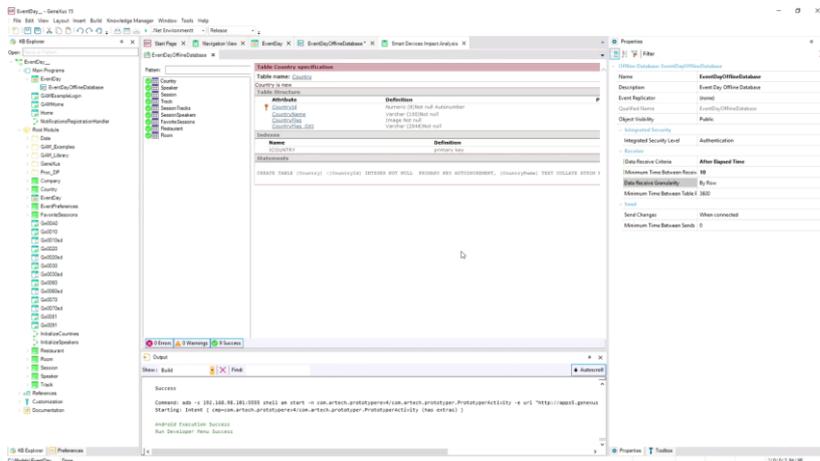


Además, también veo en el Análisis de Impacto que todas estas tablas se van a crear en forma local en el dispositivo.

Ya que yo no modifiqué ninguno de estos Work Withs, por lo tanto todos tienen la propiedad Inherit, y por lo tanto se van a crear estas tablas en el dispositivo.

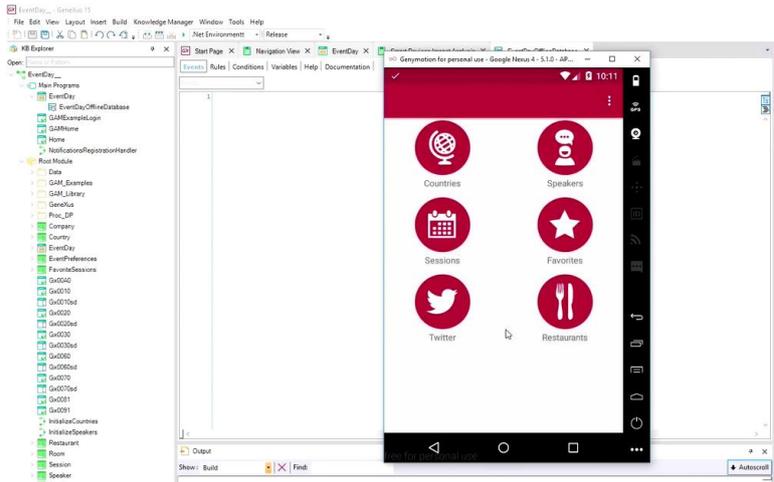


Lo que sí voy a hacer es cambiar estos valores: en lugar de que el criterio de actualización sea cuando la aplicación inicia, le voy a poner cada cierto tiempo, y que la actualización, o sea la sincronización, sea cada 10 segundos.



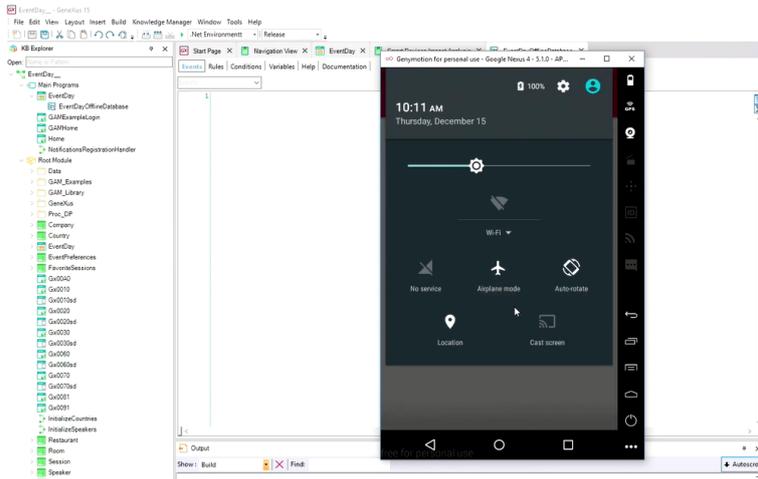
Perfecto.

Ahora lo que voy a hacer es correr la aplicación nuevamente. Ahora ya tenemos las tablas generadas en el dispositivo, y toda la lógica necesaria de sincronización, y yo lo que quiero es que esa sincronización sea cada 10 segundos; por eso puse en la propiedad Minimum Time Between Receives que el tiempo máximo entre recepciones sean 10 segundos.



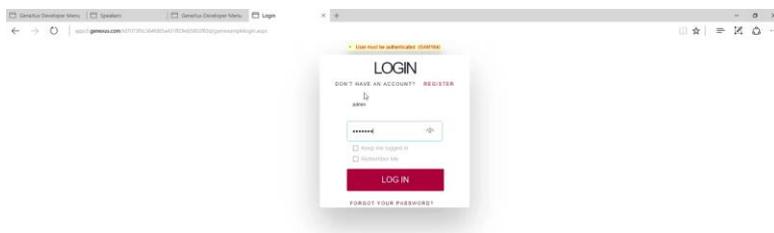
Aunque tenemos aquí nuestra aplicación, que es Offline, todavía está conectada, así que lo que vamos a hacer es desconectarla de Internet.

Para eso, voy a prender el modo avión...

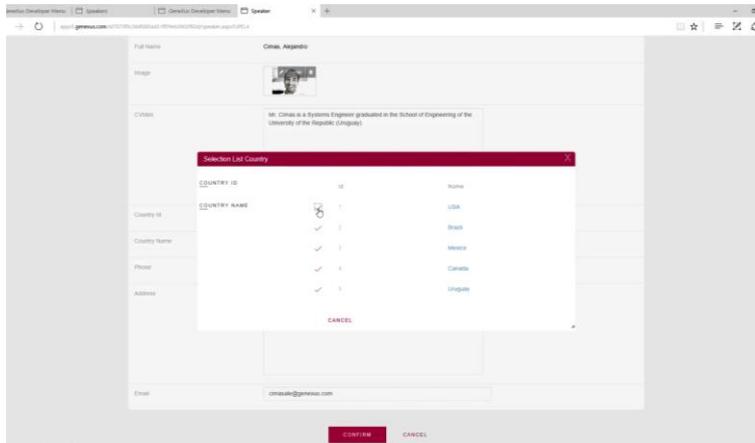


Ahora sí, la aplicación está totalmente desconectada.

Vamos por la interface web, y vamos a hacer algún cambio. Nos logueamos...

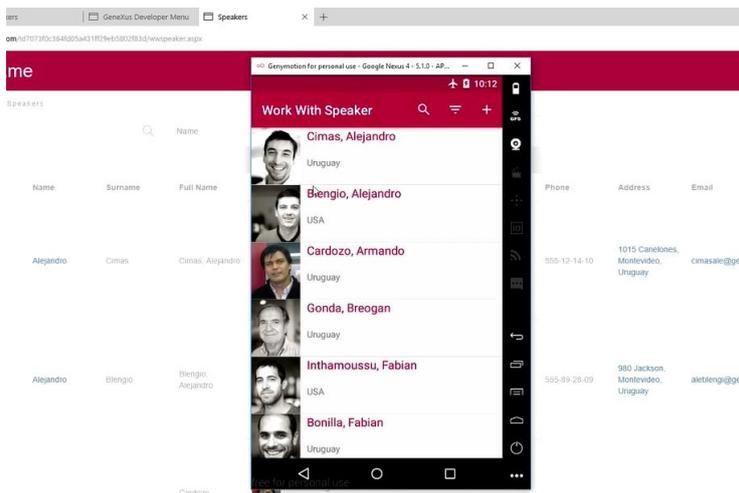


Bueno, vamos a suponer que ahora a Alejandro Cimas le decimos que no es de Uruguay, sino que es de Estados Unidos.

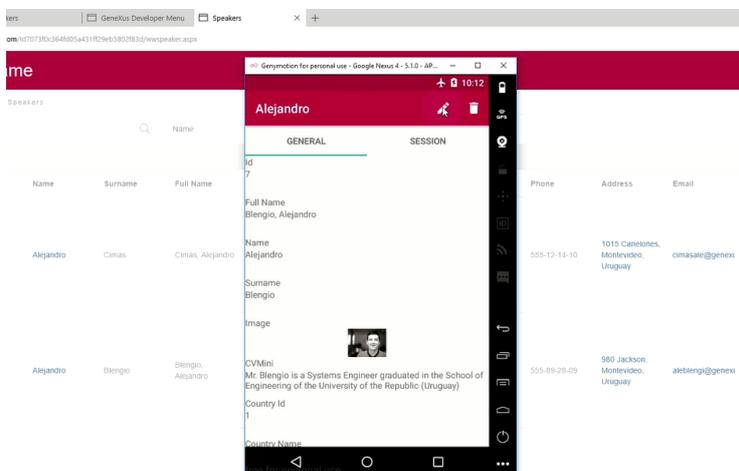


Y salvamos.

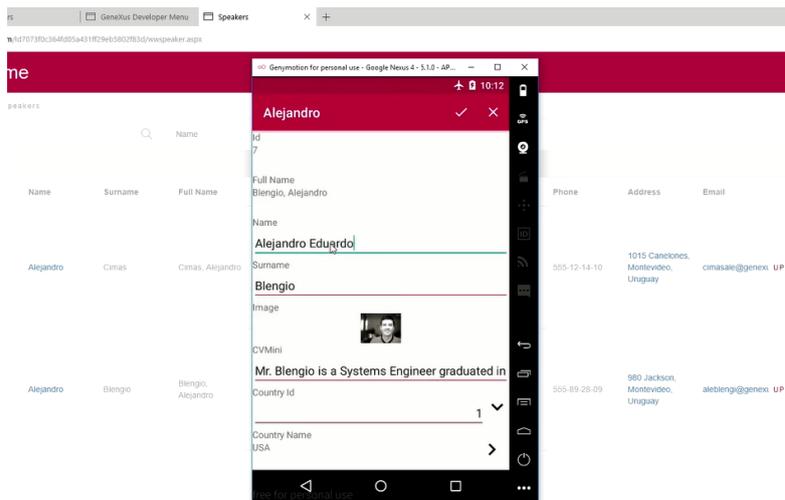
Vamos ahora al emulador, y vemos que Alejandro Cimas sigue diciendo Uruguay...



¿Por qué? Porque esto no se ha sincronizado todavía. Vamos a aprovechar y vamos a cambiar a otro de los oradores...

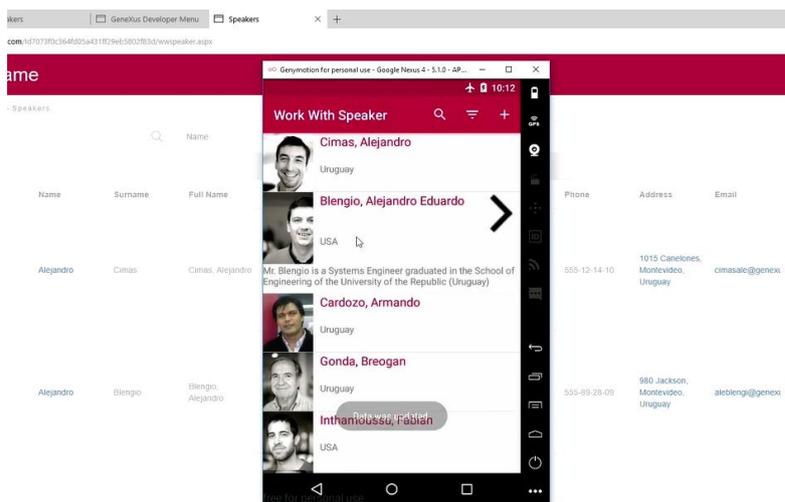


Y por ejemplo, vamos a suponer que Alejandro se llama Eduardo también.

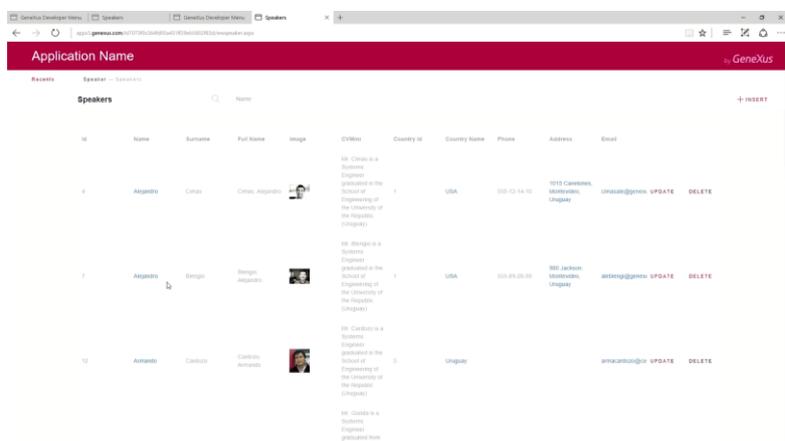


Y vamos a salvar.

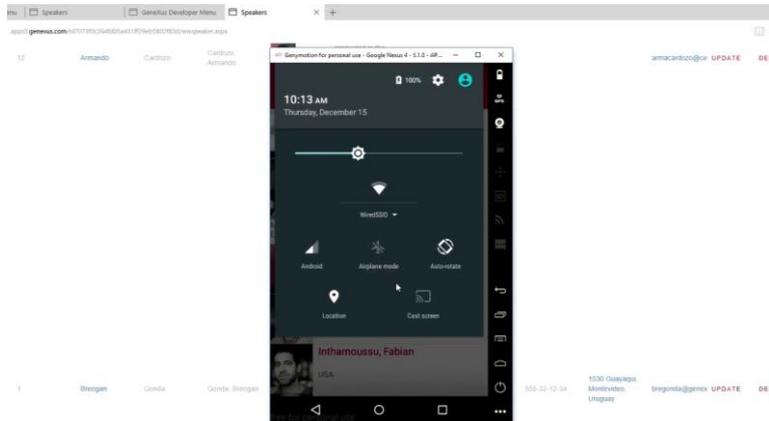
Aquí tenemos el cambio...



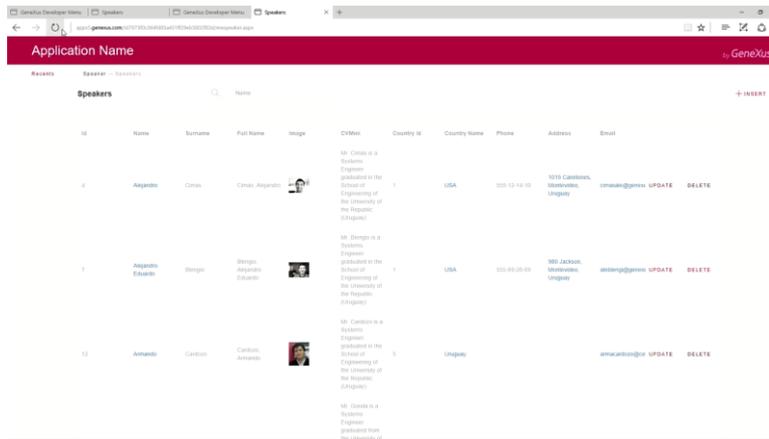
Vamos a actualizar el navegador... obviamente no están los cambios porque no está sincronizado, o sea Alejandro Cimas es de USA y Alejandro Blengio no se llama Eduardo.



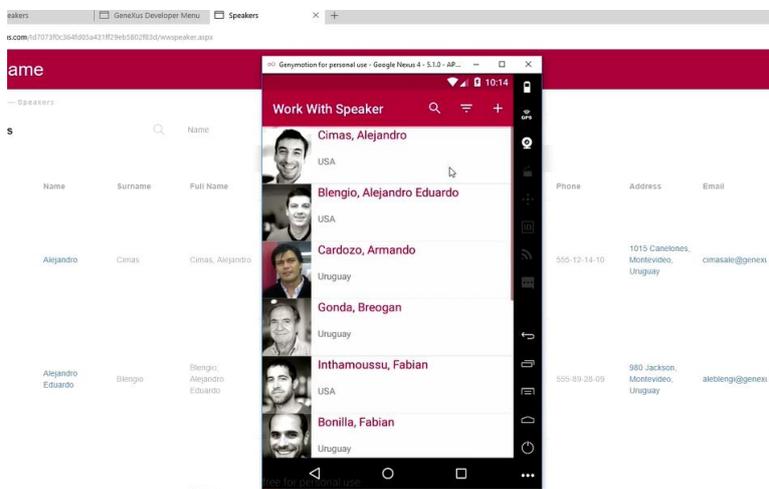
Así que lo que voy a hacer ahora es sincronizar los cambios del dispositivo. O sea que voy a volver a poner el dispositivo online para que se produzca la sincronización.



Muy bien. Ahora ya está conectado nuevamente. Volvemos al navegador... vamos a actualizarlo... Y vemos que ahora aparece Alejandro Eduardo Blengio, es decir que lo que se había cambiado en el dispositivo se actualizó en la base de datos del server.



Y vamos a ver ahora si los cambios estos fueron para el dispositivo. Efectivamente Alejandro Cimas dice que ahora es de Estados Unidos, o sea que también se sincronizaron los datos del dispositivo.



O sea que mientras estuvo sin conexión, yo pude estar trabajando con el dispositivo como si hubiera tenido conexión, y en el momento que se reestableció la conexión por WiFi, ahí se produjo la sincronización. O sea que puedo perfectamente trabajar en una aplicación Offline, y después en el momento que logro la conexión nuevamente, se sincroniza.

Terminamos aquí esta Demo básica, era simplemente para dar una idea de qué tan fácil es poder crear una aplicación Offline en GeneXus.

En este video hemos visto como con GeneXus podemos implementar aplicaciones móviles conectadas, parcialmente conectadas, o totalmente desconectadas, así como también el proceso de sincronización de los datos entre el dispositivo trabajando en forma Offline o parcialmente conectado, y el servidor web.

Si desea más información sobre este tema, visite el link que se muestra en pantalla.



The screenshot shows a web page with a red header bar. On the left side of the header, it says "Offline Applications" and on the right side, it says "GeneXus". Below the header, the text "More info:" is followed by a URL: <http://wiki.genexus.com/commwiki/servlet/wiki?22228>.