



Actualización de GeneXus 9.0 a Evolution 3

Ejercicios Prácticos

Copyright 1988 - 2015 | GeneXus S.A.

Todos los derechos reservados. Este documento no puede ser reproducido en cualquier medio sin el consentimiento explícito de GeneXus S.A. La información contenida en este documento es para uso personal únicamente.

Marcas Registradas

GeneXus es una marca registrada (®) de GeneXus S.A. en múltiples países y regiones, incluyendo entre otros, América Latina, EEUU, Japón, la UE y Uruguay. Todas las demás marcas mencionadas en este documento son propiedad de sus respectivos dueños.

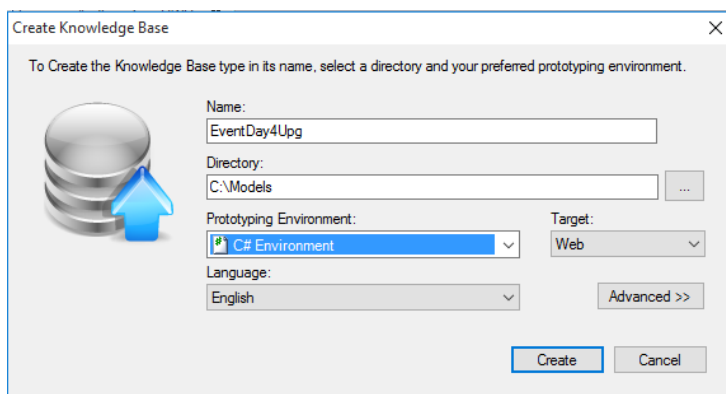
1. PRIMEROS PASOS	5
Reconocimiento del IDE.....	5
FOLDER VIEW.....	5
PREFERENCES	7
Environments	9
2. APLICACIÓN WEB.....	10
Creación de Módulos.....	10
Pattern Work With for Web	13
DEPLOY TO CLOUD	14
3. USER CONTROLS, DATA PROVIDERS Y SDTS.....	15
4. USER EXPERIENCE EN WEB FORMS.....	17
Evento de usuario: Refresh para grid relacionado	18
Evento de usuario: Refresh de línea de grid.....	20
Refresh automático en grids.....	21
Paginado automático en grids	21
5. DISEÑANDO LISTADOS PDF	22
Listado de oradores agrupados por sesión	22
Listado de oradores con la cantidad de sesiones en las que participa (cláusula Unique)	23
6. TRABAJANDO CON EL OBJETO QUERY	25
7. DATA PROVIDERS Y BUSINESS COMPONENTS	26
8. APLICACIÓN RESPONSIVE.....	28
8.2. Opcional	30
9. TRABAJANDO CON GENEXUS SERVER	35
10. OPCIONAL	37
• Versionado de la KB	37

•	Transiciones	40
11.	APLICACIÓN MÓVIL	40

1. Primeros Pasos

A lo largo de este práctico se desarrollará una aplicación para un Evento (en inglés: EventDay), que incluye países, oradores, sesiones, etc.

- Crear una nueva **Knowledge Base** (si usa la versión Trial el Environment se configurará automáticamente con generador C#, interfase Web y DataStore SQL Server)

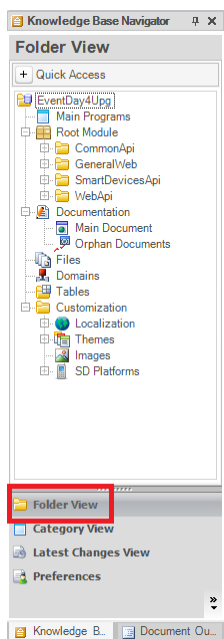


Reconocimiento del IDE

Tómese unos minutos para explorar el IDE. Empiece por explorar la ventana Knowledge Base Navigator.

FOLDER VIEW

Vea que por defecto tendrá en edición el "Folder view":



Observe el **módulo Root**. Contiene 4 folders con objetos ya inicializados:

- **CommonApi** contiene el SDT Messages, necesario para contener los mensajes de advertencia y error producto de la ejecución de todo business component.
- **GeneralWeb** contiene:
 - 2 master pages default (para los objetos web en general y para los prompts) y 2 master pages para esos mismos objetos, pero cuando la aplicación cambia su forma web por una responsiva, como veremos.
 - Además web components y SDT para mantener la lista de links visitados por el usuario en ejecución.
- **SmartDevicesApi** contiene un SDT y una api que se utilizará si una parte de la aplicación será para Smart Devices.
- **WebApi** contiene un SDT y una api que permite enviar notificaciones en una aplicación web.

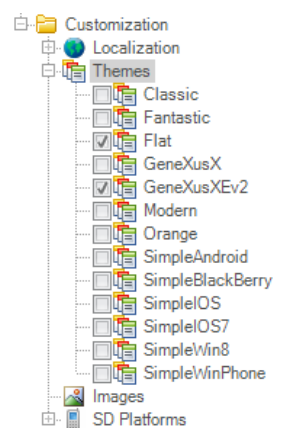
Abra el nodo **Domains**:

Haga clic sobre Name para ordenar alfabéticamente todos los dominios.

Observe, entre otros, los dominios predefinidos Address, Email, Geolocation, Phone.

Name	Type	Description
Address	VarChar(1K)	Address
APIAuthorizationStatus	Numeric(1.0)	APIAuthorization Status
CallTargetSize	Character(10)	Call Target Size
CallType	Character(20)	Call Type
Component	Url	Component
CryptoEncryptAlgorithm	Character(40)	Crypto Algorithm Encrypt
CryptoHashAlgorithm	Character(40)	Crypto Algorithm Hash
CryptoSignAlgorithm	Character(40)	Crypto Algorithm Sign
Effect	Character(20)	Effect
Email	VarChar(100)	Email
Encoding	Character(256)	IANA Encoding
Geolocation	Character(50)	Geolocation
Html	LongVarChar(2M)	Html
IMEMode	Character(40)	IMEMode property values
MessageTypes	Numeric(2.0)	Message Types
ObjectName	VarChar(256)	Object Name
Phone	Character(20)	Phone
ProgressIndicatorType	Numeric(1.0)	Progress Indicator Type
RecentLinksOptions	Numeric(4.0)	Recent Links Options
Time	DateTime	Time
Timezones	Character(60)	Timezones
Url	VarChar(1000)	Url

Bajo el folder **Customization**:



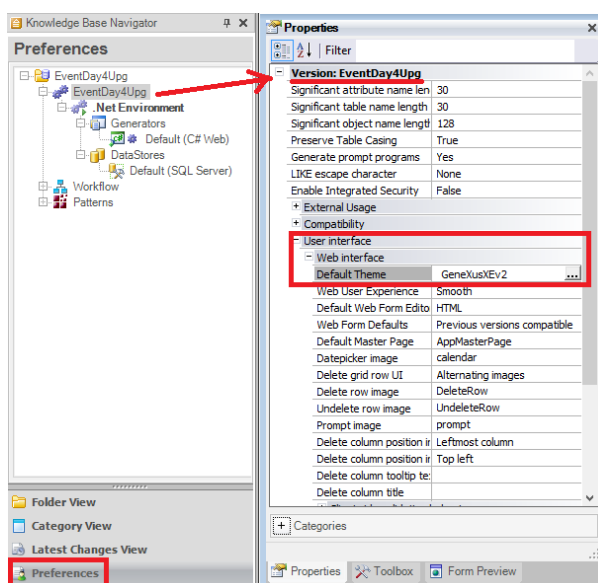
Observe los themes importados en la KB automáticamente, y los que puede importar si los marca. Abra el theme GeneXusXEv2 para recordar la definición de las clases. El diseño de las aplicaciones tanto Web como para Smart Devices estarán fuertemente enfocadas en el manejo de las clases de los themes.

Como veremos más adelante, cada KB puede versionarse. Siempre que se crea una KB, tendremos una versión cero que es conocida como Trunk (tronco), que asume el mismo nombre que la KB. En cada oportunidad, el IDE de GeneXus nos permitirá trabajar con la información de una versión a la vez. Aquella "activa", es decir, la que estemos editando. ¿Dónde vemos la información de esa versión activa?

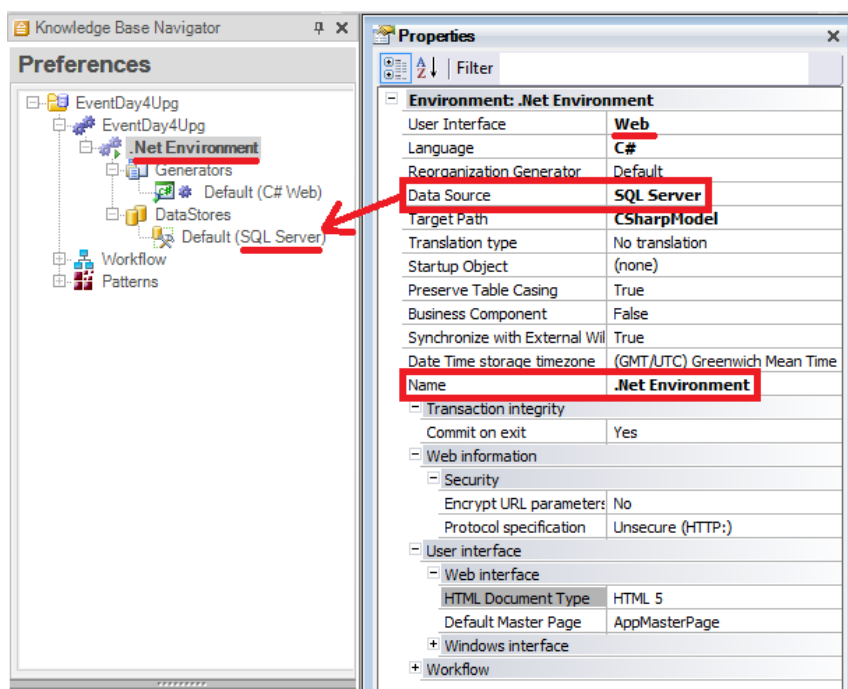
PREFERENCES

Vaya a la sección de **Preferences** en la ventana Knowledge Base Navigator:

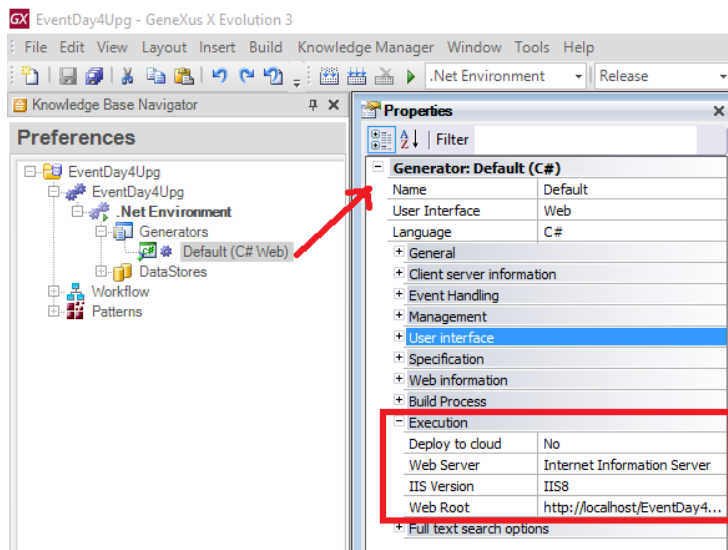
Observe las propiedades de la versión (posicionado sobre el nodo correspondiente, haciendo F4 edita la ventana de propiedades, si es que ya no la tiene abierta). Por ejemplo, bajo el grupo User Interface/Web interface verá la propiedad **Default Theme** que asume el valor **GeneXusXEv2**. Luego, cuando estudiemos las aplicaciones web “responsivas” veremos que en ese caso se cambia automáticamente al valor **Flat**.



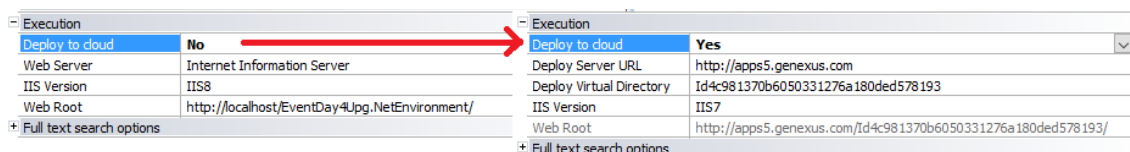
Observe que dentro de la única versión que por el momento tenemos, se ha creado automáticamente un **Environment** cuyo generador default será para interface Web y lenguaje C#, y cuyo Data Source default será SQL Server (se configuran sus propiedades en el nodo correspondiente bajo DataStores).



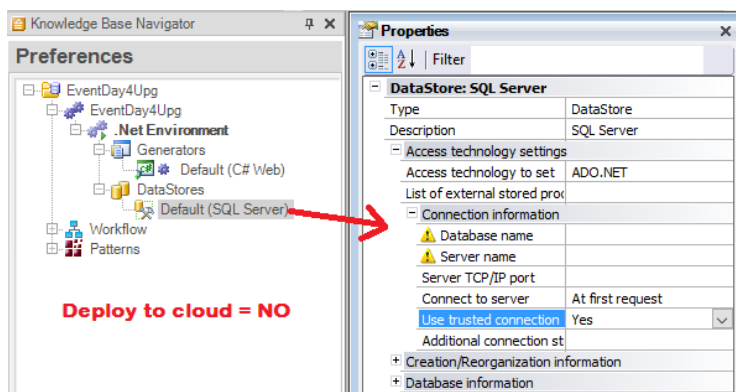
Observar las propiedades del generador Default, en particular las del grupo **Execution**. La propiedad **Deploy to Cloud** tiene el valor **YES** si está utilizando la versión Trial; en caso contrario, **No** es su valor por defecto, como vemos en la imagen:



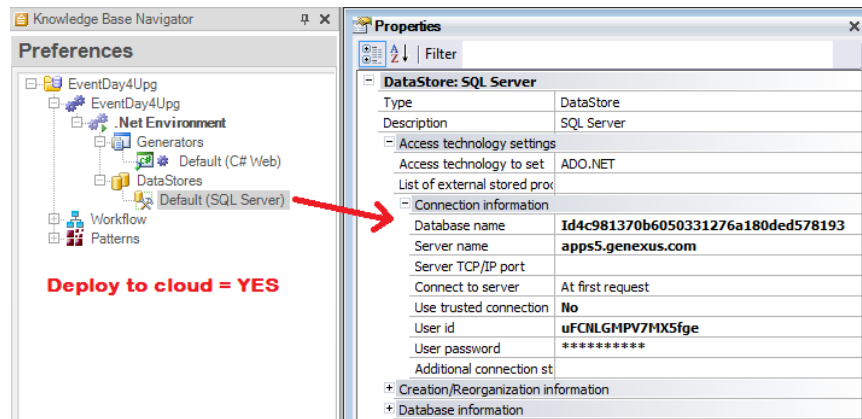
El cambio de esta propiedad de No a Yes, tiene como efecto el cambio automático en ciertas propiedades: el servidor web pasará a ser uno en la nube, se le asignará un identificador al directorio virtual en la nube, y la Web Root se compondrá de la url que apunta a ese directorio virtual.



- Observar las propiedades del **Data Store Default** que indica dónde se va a crear la aplicación —en la versión Trial esta información no está visible y la base de datos estará alojada en la nube—:



Aquí (Deploy to cloud = No) hay que dar nombre a la DB que se creará en el Server de base de datos que se indique en la propiedad Server name.



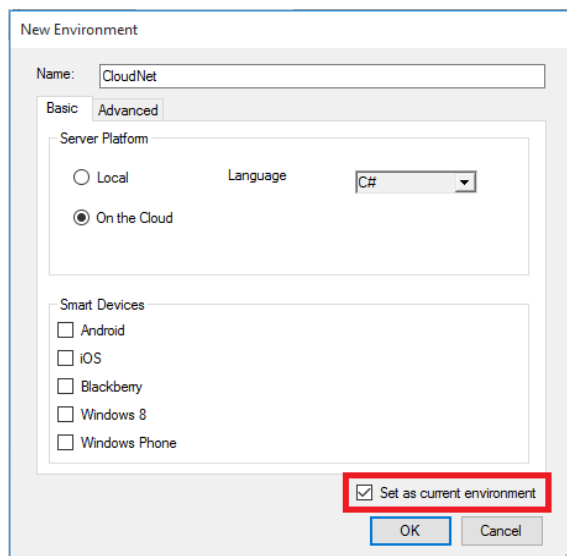
En cambio aquí el nombre de la DB corresponderá al nombre del Deploy Virtual Directory en la nube, y el Server name corresponde a la url de la nube. Observe que automáticamente se le asigna un usuario y password válidos en la nube.

Prototiparemos (a menos que se esté utilizando la versión Trial) la mayoría del práctico en forma local, y recién al final probaremos hacerlo en la nube. Consulte al instructor el nombre del servidor de base de datos.

Environments

En el punto siguiente del práctico empezaremos a construir nuestra aplicación Web. Todos los objetos que vayamos creando, serán parte de la versión Trunk (la única que tenemos por el momento). Cuando queramos ejecutar, lo que hará GeneXus será construir esa versión en el **environment** activo. En nuestro caso tenemos uno solo: el de nombre ".Net Environment". Pero podríamos querer tener otro environment para prototipar en otra base de datos y/o en otro lenguaje y/o en forma local o en la nube.

Cree otro Environment C# pero para prototipar en la nube:

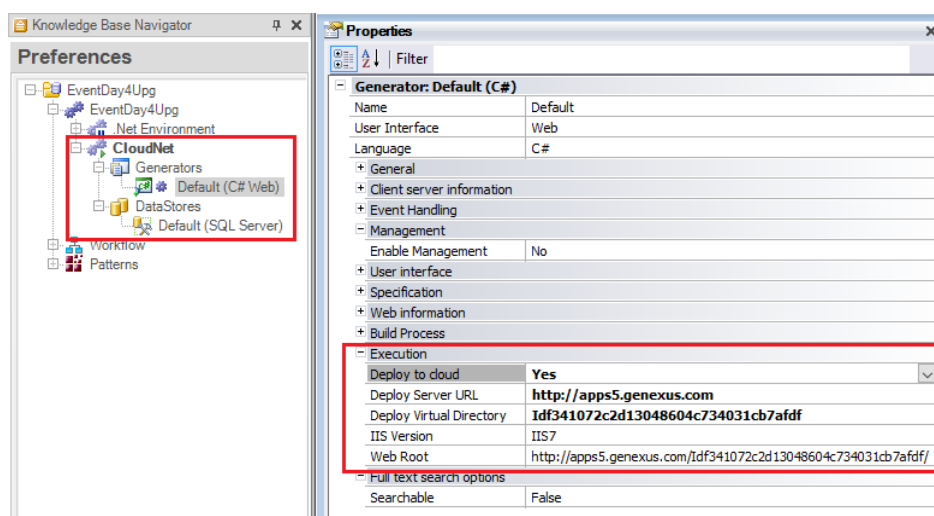
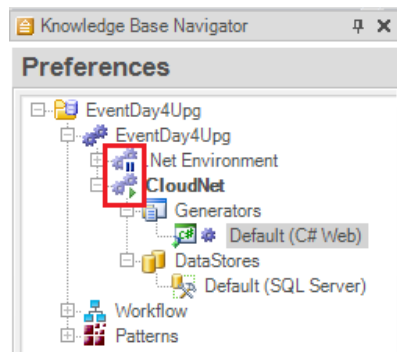


Elija la opción "On the Cloud", con lo que le cambiará el nombre a "CloudNet".

Observe que por defecto le propone que al crearlo se convierta además en el environment activo.

Una vez clique OK, verá que el environment que tenía aparece pausado, y el nuevo aparece con el ícono de play, indicando que es el que ahora está activo:

Observe las propiedades del generador Default del nuevo environment. ¿Ve que la propiedad Deploy to cloud está ahora en Yes?



Los objetos que creamos en la KB en este momento en verdad estarán siendo creados en la versión actual de desarrollo (trunk). Pero serán los mismos objetos GeneXus tanto para el environment de nombre ".Net Environment" como para el de nombre "CloudNet". La diferencia vendrá dada a la hora de generarlos. Cuando usted pida a GeneXus que construya y ejecute su prototipo, lo hará en el environment activo.

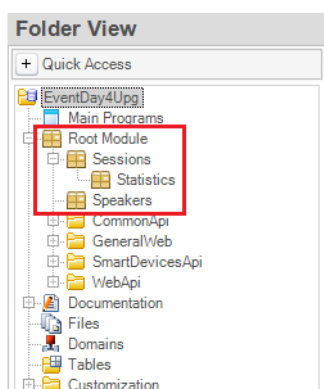
Active ahora el environment .Net (aplicación correrá local).

2. Aplicación Web

Creación de Módulos

Deseamos modularizar la KB, para encapsular todo lo que hace al trabajo con los oradores (speakers) y luego con las conferencias (sessions), para que no todos los objetos que se implementen dentro de estos módulos puedan ser visibles fuera de ellos.

- Crear la siguiente estructura de Módulos:



Defina las siguientes transacciones "Country", "Speaker" y "Session" según las imágenes que se muestran en lo que sigue.

- Las transacciones Country y Speaker deberán ubicarse dentro del **módulo Speakers**.
- La transacción Session deberá ubicarse dentro del **módulo Sessions**.

Name	Type
Country	Country
CountryId	Id
CountryName	Name
CountryFlag	Image

Name	Type
Session	Session
SessionId	Id
SessionTitle	Name
SessionDate	Date
SessionLikesQty	Numeric(4,0)
Speaker	Speaker
SpeakerId	Id
SpeakerFullName	Name

Name	Type
Speaker	Speaker
SpeakerId	Id
SpeakerName	Name
SpeakerSurname	Surname
SpeakerFullName	Name
SpeakerImage	Image
SpeakerCVMini	VarChar(1K)
CountryId	Id
CountryName	Name
SpeakerPhone	Phone
SpeakerAddress	Address
SpeakerEmail	Email

Dominios:

- Id – Numeric (8,0) con la propiedad **Autonumber = True** (a nivel del dominio, no del atributo)
- Name – VarChar(100)
- Surname -- VarChar(40)

Recuerde que los dominios Phone, Email y Address son predefinidos (se los denomina dominios semánticos, ya veremos por qué).

Tipos de datos:

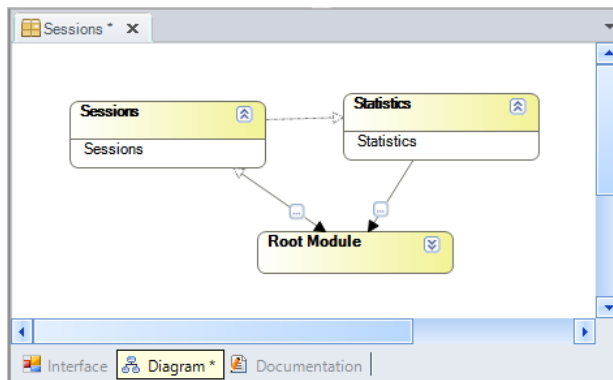
Observe que utilizamos el tipo de datos Image, en lugar del Blob. Este tipo de datos tendrá una semántica especial (permitirá abrir la cámara de fotos o la galería de imágenes cuando se está ejecutando en un dispositivo inteligente).

Fórmula global:

SpeakerFullName → `SpeakerSurname.trim()+" "+SpeakerName.trim()`

Nota: haciendo ctrl+espacio en una sección que espera que se ingrese código, se le abre un combo que le ofrece opciones posibles. Por ejemplo, en la ventana de la fórmula, al hacer ctrl+espacio puede obtener los atributos para elegir uno y no tener que tipearlo.

Abra el módulo Sessions y observe su interface (por ahora sólo contiene un objeto, la transacción Session, presentada bajo el nodo Data). Vaya a la sección Diagram y con botón derecho, elija Add Submodules y luego Add References.



Abra el módulo Speakers y vea que en su interfaz están las dos transacciones: Country y Speaker.

Observe la propiedad "Object Visibility" de cada una de las transacciones. Si pusiera a Country como Private, ¿podrá invocarlo desde la transacción Session?

Importe el archivo **UpgradeWebEv3.xpz**.

Tenga abierta la ventana "Output" antes de importar (View/Other tool Windows/Output). Observe que lo primero que se hace es crear un backup, por si tiene que volver al estado de la versión anterior a la importación.

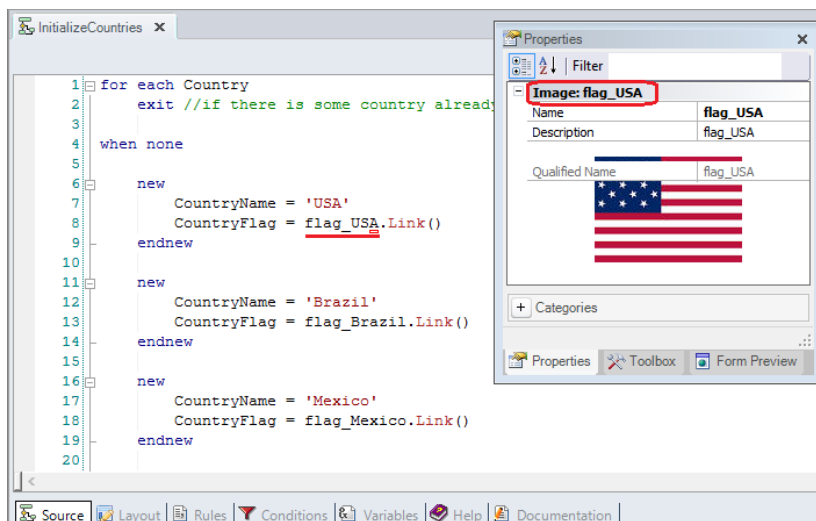
Observe que se han importado tres procedimientos para inicializar con datos las tablas de las transacciones Country, Speaker y Session:

- InitializeCountries
- InitializeSessions
- InitializeSpeakers

Mueva esos procedimientos a los módulos correspondientes.

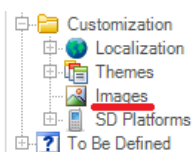
Abra el procedimiento Countries y el Speakers. Observe cómo se inicializan los valores de los atributos de tipo de datos Image (CountryFlag y SpeakerImage).

Si se posiciona sobre flag_USA ve que en las propiedades se le indica que es un objeto Image. ¿Qué es este objeto Image?



En la Evolution 3 las imágenes se guardan en la KB como un objeto GeneXus especial, para luego ser utilizadas donde se necesiten (ya sea para insertarlas en forms, como para utilizarlas como datos en la base de datos). Vea que el xpz importó todos estos objetos GeneXus Image en su KB.

Vaya al nodo Images en el Folder View y filtre por “flag” y luego por “simple”. Observe todas las imágenes que cumplen esos filtros (son las que se han importado).



La transacción Mention que se ha importado, junto con el procedimiento BestSession, será utilizada para llevar una estadística de las sesiones favoritas en cada momento. Muévalos, por tanto, al submódulo Statistics del módulo Sessions.

Asimismo mueva el procedimiento OneLike al módulo Sessions, pues se utilizará para marcar una sesión como favorita.

Abra ahora el módulo Sessions y observe su Interface. Haga lo mismo con Speakers.

Pattern Work With for Web

- Aplicar el pattern **Work With for Web** a las transacciones Country, Session y Speaker. Observe que el archivo de instancia y los objetos web panels y web components que implementan el “trabajar con” y el “view” aparecen bajo la transacción en el Folder View.
- Observe que en el folder GeneralWeb se han agregado objetos para relacionar todos los work with. En particular en el web panel **Home** creado por el pattern, codificar el evento Start para inicializar todas las tablas, invocando a los procs importados:

```
Event Start
    InitializeCountries()
    InitializeSpeakers()
    InitializeSessions()
Endevent
```

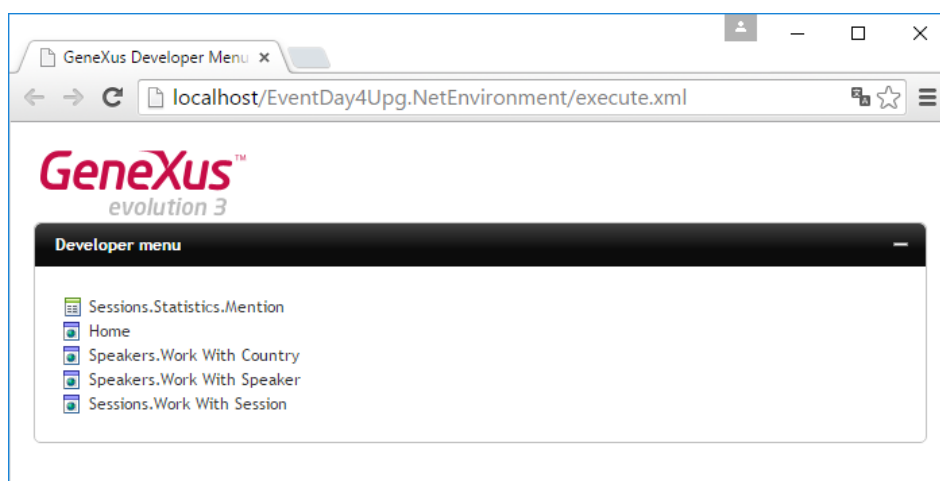
Nota: observar que para llamar a un objeto ya no es necesario escribir el método call.

Si alguno de los procedimientos anteriores tuvieran la propiedad "Object Visibility" en Private (por pertenecer a módulos), ¿podrían ser invocados desde este objeto Home? ¿Por qué no? Y si el objeto Home estuviera dentro del mismo módulo o en un submódulo, ¿se podría?

Ejecute la aplicación. F5.

Como el environment actual es el local, si no ingresó antes las propiedades del servidor de base de datos y el nombre de la base de datos, se le pedirá.

El análisis de impacto le ofrecerá crear las tablas. Al finalizar la generación, se le abrirá el Developer Menu. Observe que los objetos que están dentro de módulos, aparecerán precedidos por el nombre del módulo.



Ejectue el web panel Home (se ejecutará el Start y se cargarán los datos en las tablas).

Abra el Work with Speakers y haga clic sobre la dirección de cualquiera de los oradores. Y sobre el Email (si tiene un programa de correo electrónico instalado, se abrirá). Aquí ve qué significa que estos dominios son **semánticos**.

DEPLOY TO CLOUD

Ahora active el Environment CloudNet. Y presione F5 para construir la misma aplicación, pero en la nube. Se le pedirá un usuario válido de GXtechnical para poder registrar esta KB a su nombre.

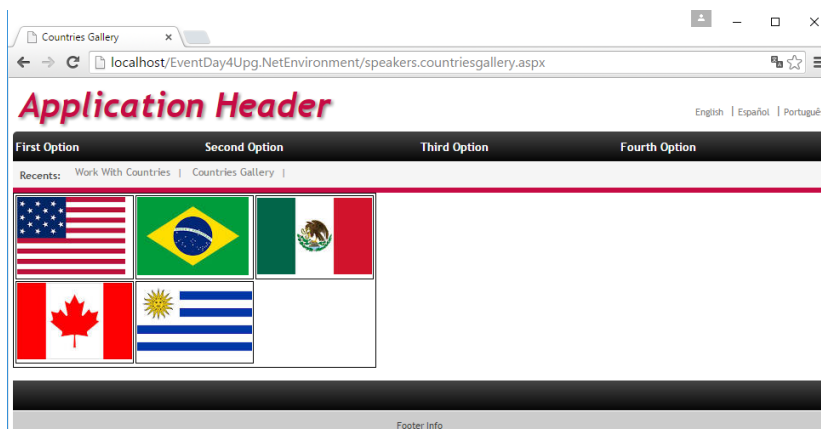
El reporte de análisis de impacto le pedirá crear las tablas, pero esta vez en la nube. Intente ejecutar Speakers.Work With Country y verá que no tiene datos. Ejecute el Home. Y ahora sí.



Vuelva a poner activo el environment local, para las siguientes ejecuciones.

3. User controls, Data Providers y SDTs

Se desea diseñar un web panel que muestre la galería de imágenes con las banderas de todos los países. ¿En qué módulo lo colocará?



Solución

- Crear un web panel de nombre CountriesGallery.
- Arrastrar sobre el form el user control (bajo el nodo "Extended controls" de la Toolbox) de nombre ImageGallery



- Observe que en el Folder View se ha creado automáticamente el SDT colección de nombre **ImagesData**. Ábralo y vea su composición.
- Observe que en el web panel se ha creado automáticamente la variable **ImagesData** de ese tipo de datos, que será la que deberá cargarse con los datos a ser mostrados en la galería. También se ha creado la variable **ImagesDataItem** del tipo de datos de los ítems, por si necesita ir agregando ítems a la colección, de a uno, con el método **Add**. No será nuestro caso. Pues usaremos un Data Provider que nos devuelva la colección ya cargada.

- c) Crear el Data Provider de nombre GetCountries (en módulo Speakers) que será el encargado de devolver el SDT ImagesData cargado con las banderas de los países.
- Arrastre el SDT ImagesData del Folder View al Source y observe las propiedades bajo el grupo "Output". (¿Por qué Collection está en False?) Rellene el Source especificando la Base Transaction del grupo repetitivo y los atributos de los que se tomará la información.

```
ImagesData from Country
{
  ImagesDataItem
  {
    Id = CountryId.ToString()
    Image = CountryFlag.ImageURI
    Thumbnail = CountryFlag.ImageURI
    Caption = CountryName
  }
}
```

Output	
Infer Structure	No
Output	ImagesData
Collection	False

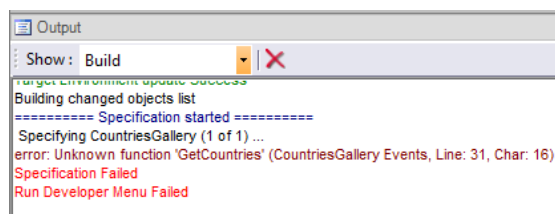
- d) Sólo resta invocar desde el web panel CountriesGallery a este data provider para que nos devuelva cargada la colección y asignársela a la variable del User control:

```
Event Start
    &imagesData = GetCountries()
Endevent
```

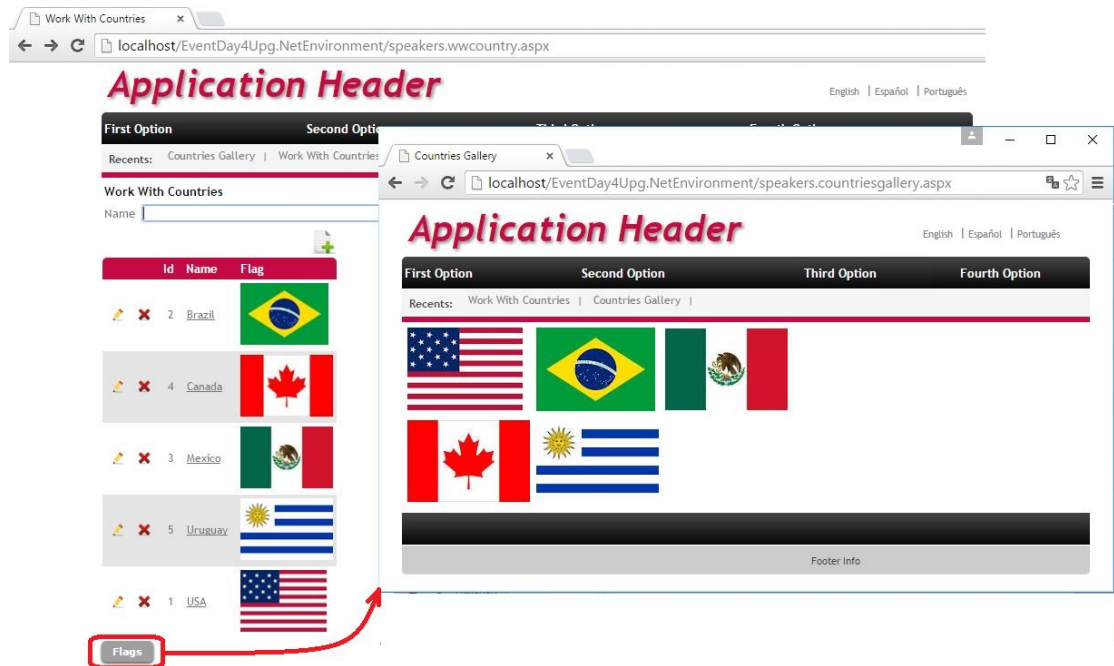
Nota: observar que ya viene programada una subrutina que, sin embargo, no es invocada. Está allí para que el desarrollador sustituya cada elemento por el que necesite. Sin embargo, al invocar al Data Provider ya estaremos haciéndolo, de forma más sencilla. Puede dejarla comentada o borrarla.

Ejecútelo desde el Developer Menu (F5).

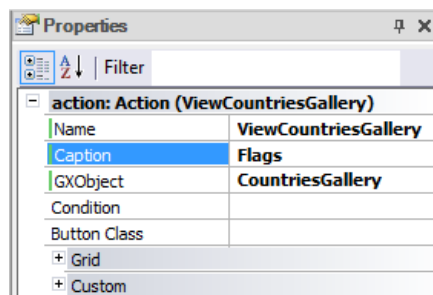
Abra el módulo Speakers y vea ahora su nodo Interface. Si el Data provider GetCountries se concibe como interno al módulo, es decir, no quiere ofrecerse como servicio a otros módulos, debería colocarlo como privado. Hágalo y vea cómo desaparece del nodo Services. Si el web panel CountriesGallery no estuviera dentro del mismo módulo, ¿podría invocarlo? Puebe mover el Web panel al módulo Root y dé F5. (luego vuelva a colocarlo en el módulo Speakers).



En la pantalla del **Work With Countries** defina un botón que invoque al web panel CountriesGallery y pruébelo en ejecución.



Solución: En la instancia del Work With de Country agregue una acción para el "Selection (Work With Countries)", a la que le asociará el objeto CountriesGallery. Aparecerá como botón bajo el grid.

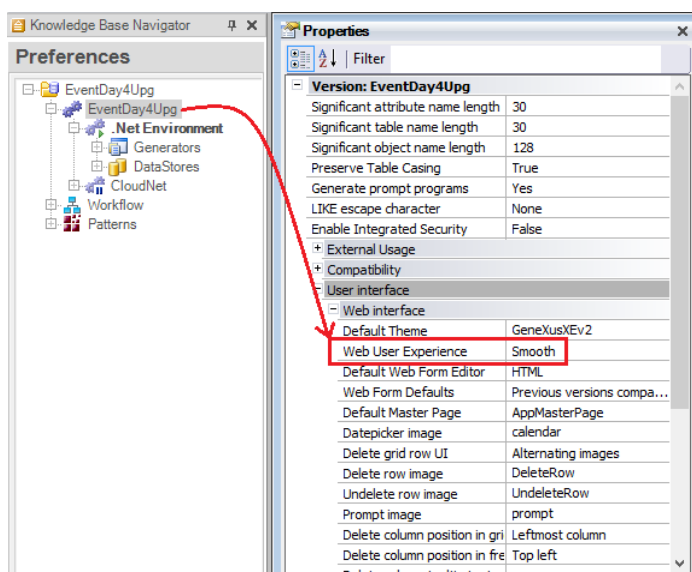


Observe el control ImageGallery en el panel CountriesGallery. Vea sus propiedades. Puede quitarle el borde negro, e incluso definir un Cell Space de 5 para dejar espacio entre las banderas. F5 y ejecute el Work With.

4. User experience en Web forms

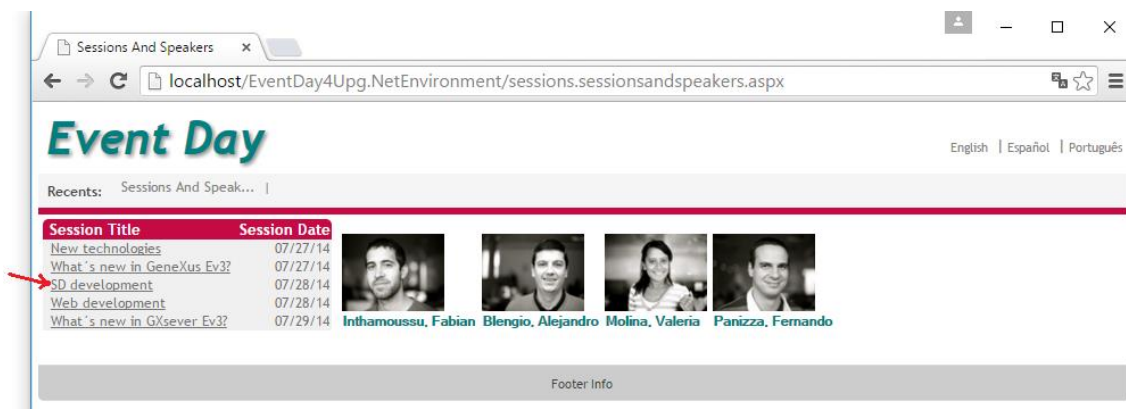
En ejecución, inserte una nueva conferencia. Observe el prompt sobre SpeakerId como ventana popup, y la manera que ofrece el grid para insertar una nueva fila, cuando las 5 default ya fueron completadas y necesita ingresar más.

Observe el valor de la propiedad **Web User Experience** a nivel de la versión actual:



Evento de usuario: Refresh para grid relacionado

Crear en el módulo Speakers un web panel de nombre **SessionsAndSpeakers** que muestre las conferencias (sessions) en una grilla y permita al usuario hacer clic sobre el título de cualquiera de ellas, mostrando en otra grilla al costado los datos de sus oradores:

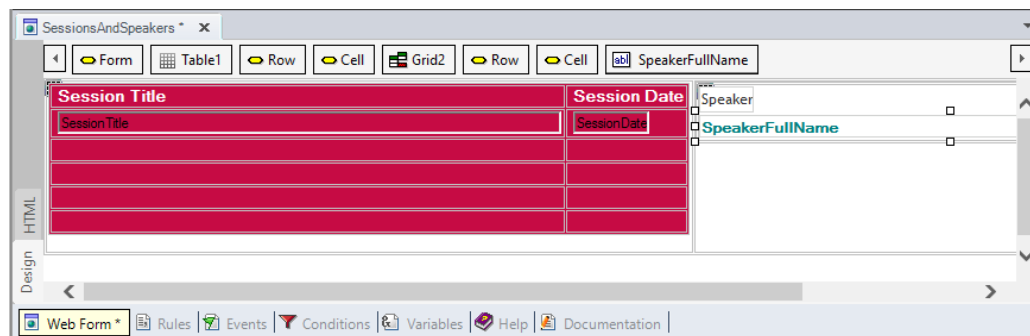


Vea que se ha personalizado la master page para que contenga el nombre de la aplicación, y se han quitado las tablas negras.

Tip: Para trabajar con los controles en el form, edite la ventana "Document Outliner" que le permitirá posicionarse dentro de las tablas con mayor facilidad.

Solución:

- Crear un web panel de nombre SessionsAndSpeakers dentro del Módulo Speakers
- Insertar una tabla de 1x2. A ambas celdas configurarles "Vertical Align = Top".
- En la primera celda, insertar un Grid con los atributos SessionId (oculto), SessionTitle y SessionDate.
- En la celda derecha insertar un Grid free style con los atributos SpeakerImage y SpeakerFullName. Recuerde editar las propiedades de SpeakerFullName para cambiar color y tipo de letra. (Observe que solamente se pueden cambiar directamente a nivel del control pocas propiedades. El resto estarán a nivel de la clase del theme). Utilice la propiedad Columns del grid free style para indicar la cantidad de columnas en las que desee que se despliegue la información. Como siempre el valor 0 corresponde a "todas".



- Declarar el siguiente evento en el web panel:

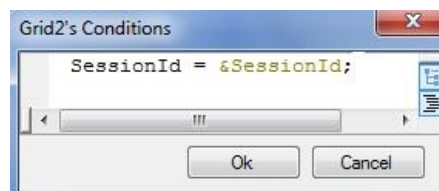
```

] Event SessionTitle.Click
    &SessionId = SessionId
    Grid2.Refresh()
- Endevent
    
```

Posicionarse sobre la variable y con botón derecho, crearla.

Al ser Smooth la UX, no se refrescará el Grid2 en forma automática. Por eso tiene que especificarlo explícitamente en el evento del cliente. Con ello se refrescará el Grid2, y nada más.

- Especificar en el Grid2 la siguiente condición: (y la Base Trn: Session.Speaker)

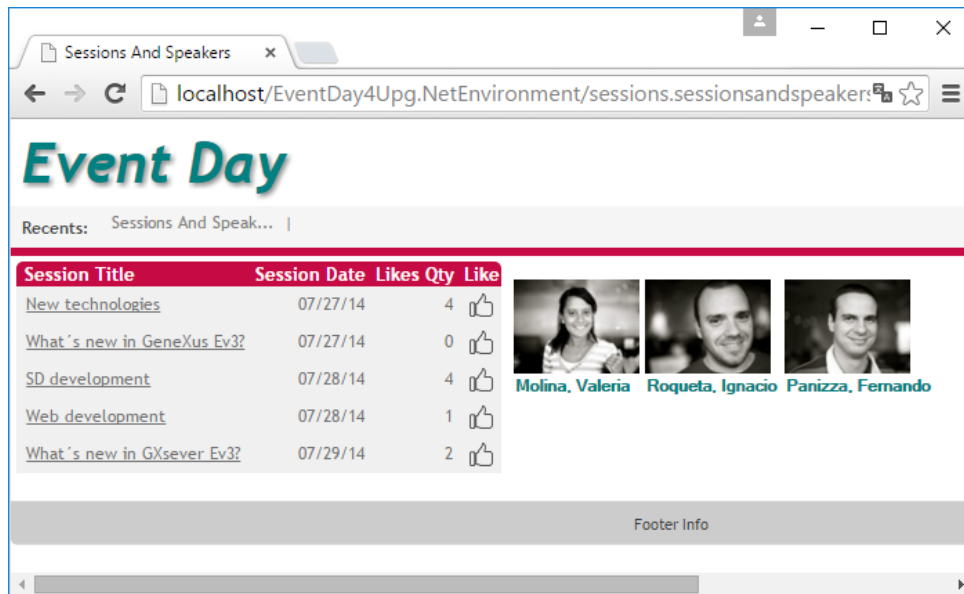


Edite la master page **AppMasterPage** que se encuentra en la carpeta GeneralWeb bajo el módulo Root, para personalizarla.

Ejecute y observe cómo al cliquear sobre un speaker diferente, solamente se refresca el grid con las fotos, y no el resto de la página contenida, ni la master page.

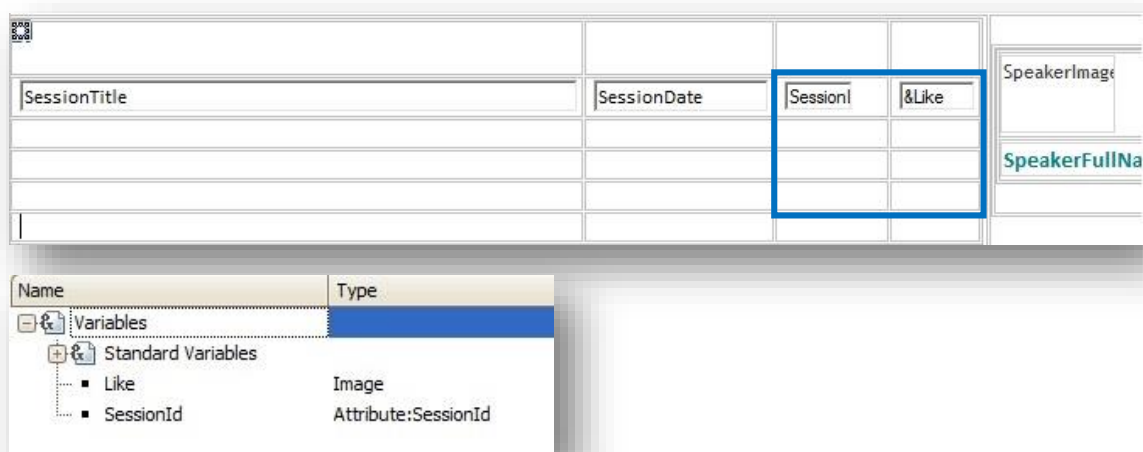
Evento de usuario: Refresh de línea de grid

- En el web panel SessionsAndSpeakers se desea ahora poder visualizar la cantidad de "Likes" que tiene cada conferencia. Además, al hacer clic sobre la imagen de "me gusta", se sumará un "like" más al atributo SessionLikesQty (que lleva la cantidad acumulada de "likes").



Solución:

- En el web panel SessionsAndSpeakers agregar dos nuevas columnas en el Grid1 (Sessions). Agregar el atributo SesstionLikesQty y una variable &Like de tipo Image.



- En el evento Start declarar la carga de la variable &Like con la imagen correspondiente (ha sido importado un objeto imagen de nombre "sample_like". Verifíquelo.):

```

]Event Start
    &Like.FromImage(sample_like)
-Endevent

```

- c) Declarar también el evento asociado a la variable &Like para que al hacer clic sobre ella se dispare el procedimiento OneLike (importado previamente) que recibe el identificador de la sesión y actualiza la cantidad de "likes".

```

]Event &Like.Click
    OneLike(SessionId)
    Grid1.Refresh()
-EndEvent

```

```

Parm(in:&SessionId);
For each Session
    where SessionId = &SessionId
        SessionLikesQty += 1
-EndFor

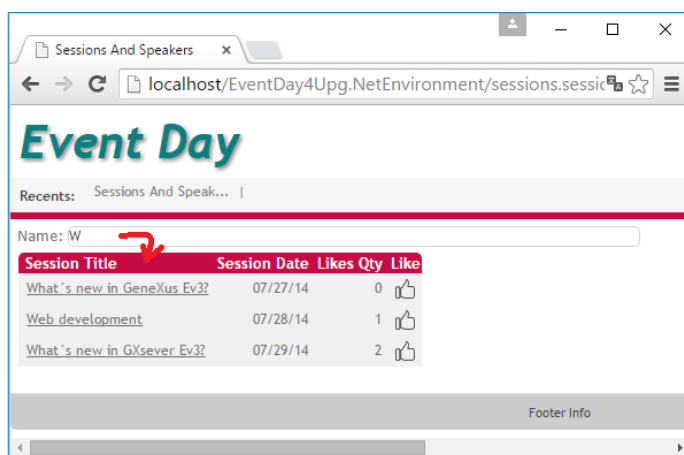
```

Ejecute la aplicación. Observe que al hacer clic sobre la imagen de "me gusta" solamente se actualiza la línea correspondiente. El resto de la página no se vuelve a cargar.

Si el usuario desea ver en ejecución las conferencias en el grid ordenadas por fecha en sentido creciente, ¿cómo hace? ¿y en sentido decreciente?

Refresh automático en grids

Agregue un filtro por título de conferencia sobre el Grid1 y vea que no necesita colocar ningún botón para que se refresque el grid en ejecución al cambiar los valores de la variable de filtro:



Paginado automático en grids

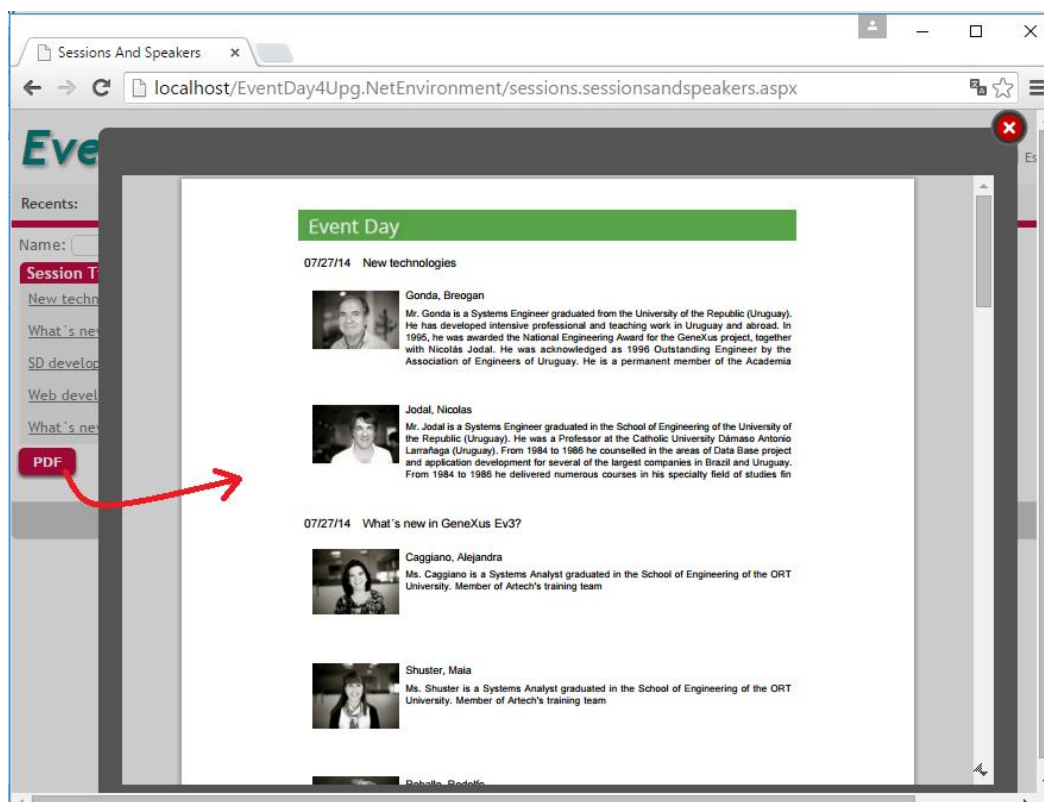
Modificar en el Web panel SessionsAndSpeakers el valor de la propiedad Rows del Grid1, para que tenga un paginado automático de 3 líneas por página. ¿Lo hizo utilizando la propiedad a nivel de diseño o de runtime?

Solución: para hacerlo en runtime, debe dejar el valor 0 a nivel de diseño y, por ejemplo, en el evento Start escribir "Grid1.Rows = 3".

5. Diseñando listados PDF

Listado de oradores agrupados por sesión

Se desea diseñar un listado pdf que se invoque desde el web panel SessionsAndSpeakers y muestre los oradores agrupados por sesión. El listado se visualizará en una ventana popup.



Solución:

- Cree el procedimiento de nombre ViewSpeakersbySession dentro del módulo Sessions.

La imagen en el printblock Title corresponde al objeto imagen "simple_event" importado en el archivo UpgradeWebEv3.xpz; el printblock Session incluye los atributos SessionDate y SessionTitle y el printblock Speakers a los atributos SpeakerImage, SpeakerFullName y SpeakerCVMini. Para el control SpeakerImage configurar Width = 100 y Height = 75.

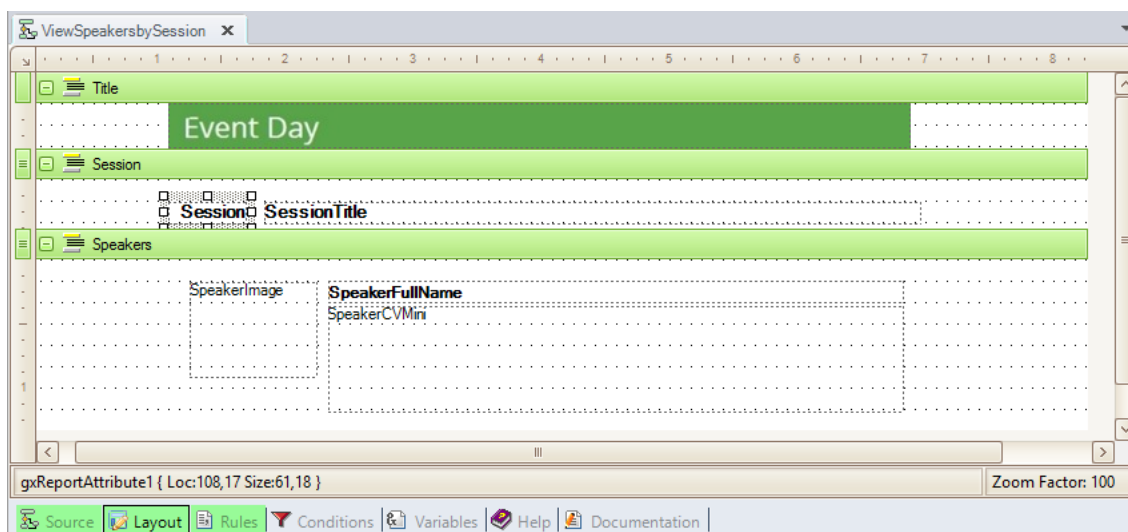
Para que el contenido del atributo SpeakerCVMini se vea en varias líneas, dar al control el tamaño deseado y especificar la propiedad Alignment = TopJustify.

Para poder ejecutar el listado pdf recuerde especificar las siguientes propiedades:

Main Program = **True**
Call protocol = **HTTP**

Y la siguiente regla:

Output_file("ViewSpeakersbySession", "pdf");



b) Codificar el source:

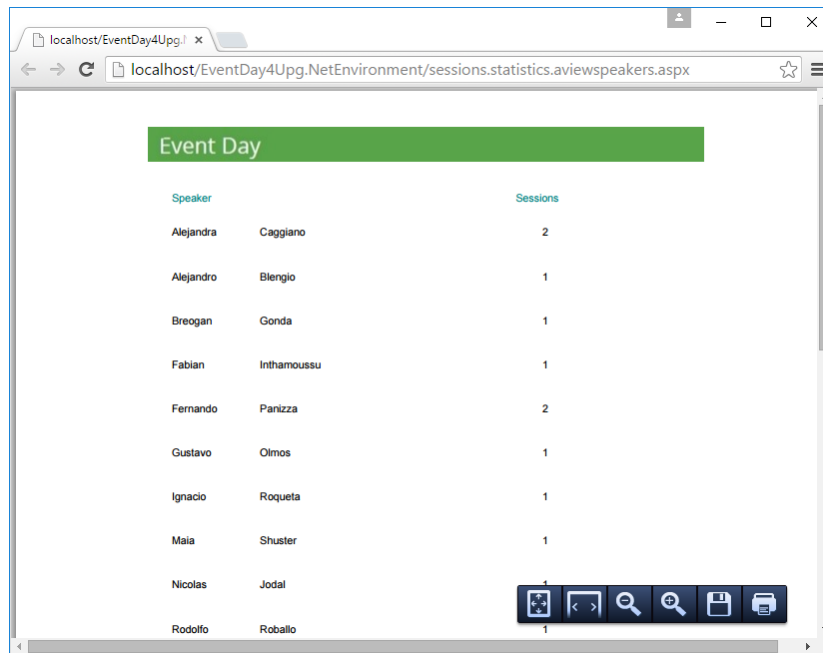
```
print Title
] For each Session
    print Session
    ] For each Session.Speaker
        print Speakers
    - Endfor
- Endfor
```

c) Volver al web panel SessionsAndSpeakers, agregar un botón de caption "PDF" asociado al evento Enter y codificar:

```
Event Enter
    ViewSpeakersbySession.Popup()
Endevent
```

Listado de oradores con la cantidad de sesiones en las que participa (cláusula Unique)

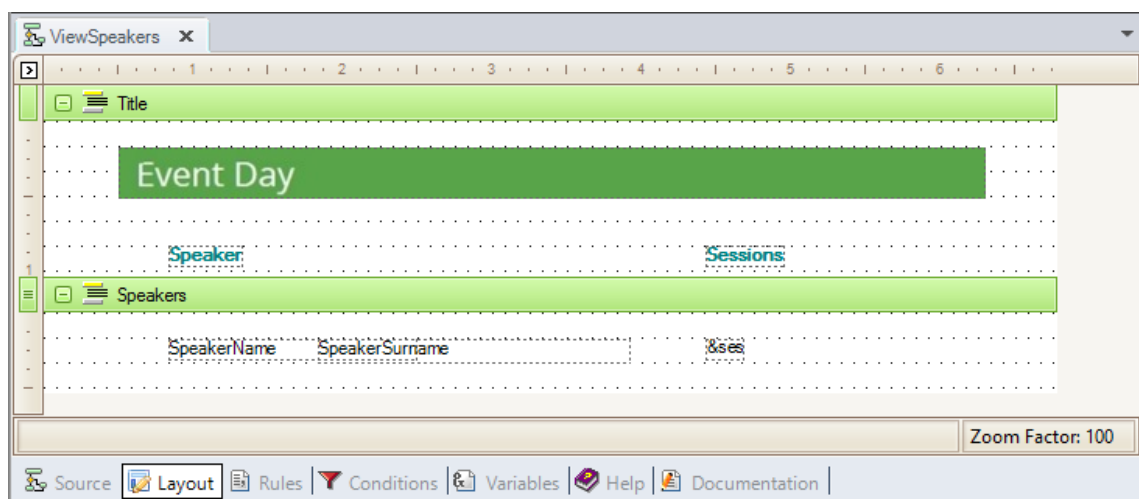
Se desea visualizar un listado pdf con todos los oradores que están registrados en alguna sesión. Junto al nombre/apellido del orador se verá la cantidad de sesiones en las que participa. Se quiere ordenar por nombre.



Speaker		Sessions
Alejandra	Caggiano	2
Alejandro	Blengio	1
Breogan	Gonda	1
Fabian	Inthamoussu	1
Fernando	Parizza	2
Gustavo	Olmos	1
Ignacio	Roqueta	1
Maia	Shuster	1
Nicolas	Jodal	1
Rodolfo	Roballo	1

Solución:

- Hacer Save As... del procedimiento anterior y salvarlo con el nombre ViewSpeakers. Asígnelo al Módulo Statistics.
- Modificar el layout como se muestra:



Codificar el source:

```
print Title
for each Session.Speaker order SpeakerName
  unique SpeakerId
  &sessions = count( SessionId )
  print speakers
endfor
```


Ejecute el listado pdf. Para ello, sobre su solapa haga clic con el botón derecho del mouse y seleccione **Run**.

6. Trabajando con el objeto Query

Se desea diseñar una tabla dinámica (pivot table) que muestra las sesiones por día, cada una con sus respectivos oradores.

Event Day

Recents: WPPivot |

Drop filters here

Session Date	Session Title	Speaker Full Name	Quantity	
07/27/14	New technologies	Gonda Breogan	1	
		Jodal Nicolas	1	
	Total of New technologies		2	
	What 's new in GeneXus Ev3?	Caggiano Alejandra	1	
		Roballo Rodolfo	1	
		Shuster Maia	1	
	Total of What 's new in GeneXus Ev3?		3	
	Total of 2014-07-27		5	
	07/28/14	SD development	Blengio Alejandro	1
			Cardozo Armando	1

10 per page

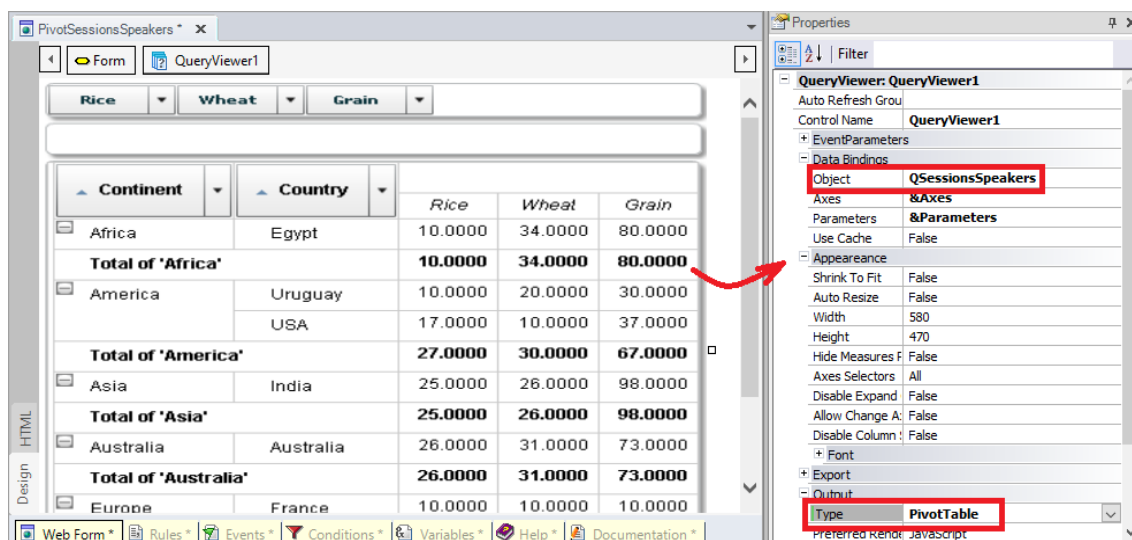
Page 1 of 3

Solución

- Crear un objeto Query de nombre QSessionsSpeakers.

Structure	Description
QSessionsSpeakers	QSessions Speakers
Attributes	
SessionDate	Session Date
SessionTitle	Session Title
SpeakerFullName	Speaker Full Name
Parameters	
Filters (AND)	
OrderBy	

- Crear en el mismo módulo un web panel de nombre PivotSessionsSpeakers, y arrastrar en el form el control QueryViewer.
- En la propiedad Object, asociar el objeto query QsessionsSpeakers.
- En la propiedad Type indicar PivotTable.



- e) Ejecutar el web panel. Mover las columnas de lugar. Exportar a pdf.

7. Data Providers y Business Components

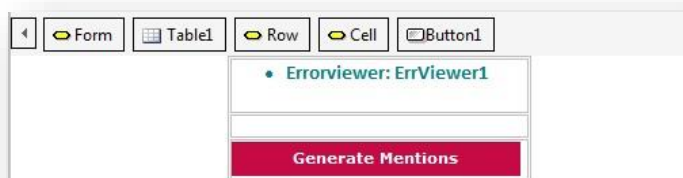
Se desea entregar una mención especial a los oradores que participaron de la sesión marcada como preferida por los usuarios (SessionLikesQty).

En el xpz importado se incorporó una transacción de nombre Mentions que ha sido definida como Business Component.

Se necesita diseñar una pantalla que simplemente ofrezca un botón que dispare el proceso de premiación. Se deberá buscar la sesión que tenga mayor valor en el atributo SessionLikesQty, y generar un registro en la tabla MENTION para cada orador participante, de acuerdo a la fecha en la que se ejecuta el proceso.

Solución:

- Crear un web panel de nombre SpecialMentions y diseñar el form como se muestra.
- Insertar un control ErrorViewer, y un botón como se muestra:



- Abra y observe el procedimiento BestSession (previamente importado) que devuelve el identificador de la sesión más votada:

```

Parm(out: &SessionId);
For each Session order (SessionLikesQty)
    &SessionId = SessionId
    exit
-Endfor


```

- d) Crear un Data Provider de nombre GetMentions (dentro del mismo Módulo Speakers) que devolverá la colección de menciones a generar para los correspondientes oradores de la sesión indicada. (Le recomendamos hacer drag & drop de la transacción Mention sobre el Source. Observe que la propiedad output asumirá el valor Mention del Business Component). ¿Por qué tiene que prender la propiedad Collection?:

```

Parm(in: &SessionId);
Mention from Session.Speaker
where SessionId = &SessionId
{
    SpeakerId
    SessionId
}

```



Nota: ¿por qué se eliminaron las asignaciones a MentionId y a MentionDate?

- e) Codificar el evento asociado al botón del web panel SpecialMentions para que reciba la salida del Data Provider y guarde físicamente cada registro en la tabla MENTION.

Name	Type	Is Collection
Variables		
Standard Variables		
Mentions	Mention, Sessions.Statistics	<input checked="" type="checkbox"/>
Mention	Mention, Sessions.Statistics	<input type="checkbox"/>
SessionId	Attribute:SessionId	<input type="checkbox"/>

```

Event Enter
    &SessionId = BestSession()
    &Mentions = GetMentions( &SessionId )
    for &Mention in &Mentions
        &Mention.Save()
    endfor
    commit
    msg( 'Mentions were successfully created' )
Endevent

```

Notas: Observe siempre la referencia a los objetos dentro de los respectivos Módulos. El mensaje automáticamente se visualiza en el control Error viewer insertado.

Si quisiera manejar posibles errores y en caso de algún error deshacer las grabaciones podría hacer lo siguiente: (donde &OkForCommit es una variable de tipo de datos Boolean; &Messages de tipo el SDT predefinido Messages y &Message el de uno de sus ítems, del tipo Messages.Message):

```

Event Enter
    &SessionId = BestSession()
    &Mentions = GetMentions(&SessionId)
    for &Mention in &Mentions

```

```

&Mention.Save()
&okForCommit = True
if &Mention.Fail()
    &Messages = &Mention.GetMessages()
    for &Message in &Messages
        msg( &Message.Description )
    endfor
    &okForCommit = False
endif
endfor
If &okForCommit
    Commit
    msg( "Mentions were successfully created" )
else
    Rollback
endif
Endevent

```

Nota3: Vale mencionar que esta implementación permite que se premie varias veces a los oradores de la misma sesión. Si lo desea, implemente algún control que no lo permita (puede utilizar un índice único, teniendo en cuenta que para fechas diferentes sí podrían repetirse sesión y speaker, pero no para el mismo día).

8. Aplicación Responsive

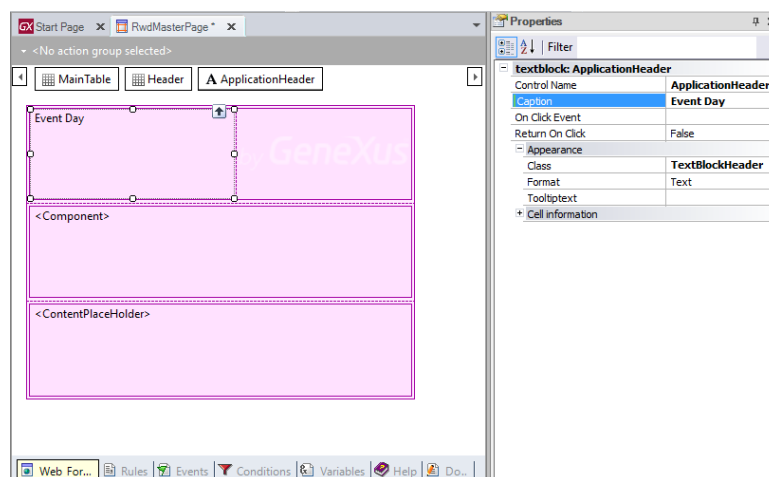
Queremos que la aplicación que venimos desarrollando sea Responsive. ¿Cómo lo logra?

Solución:

- Configurar las correspondientes propiedades de la versión:

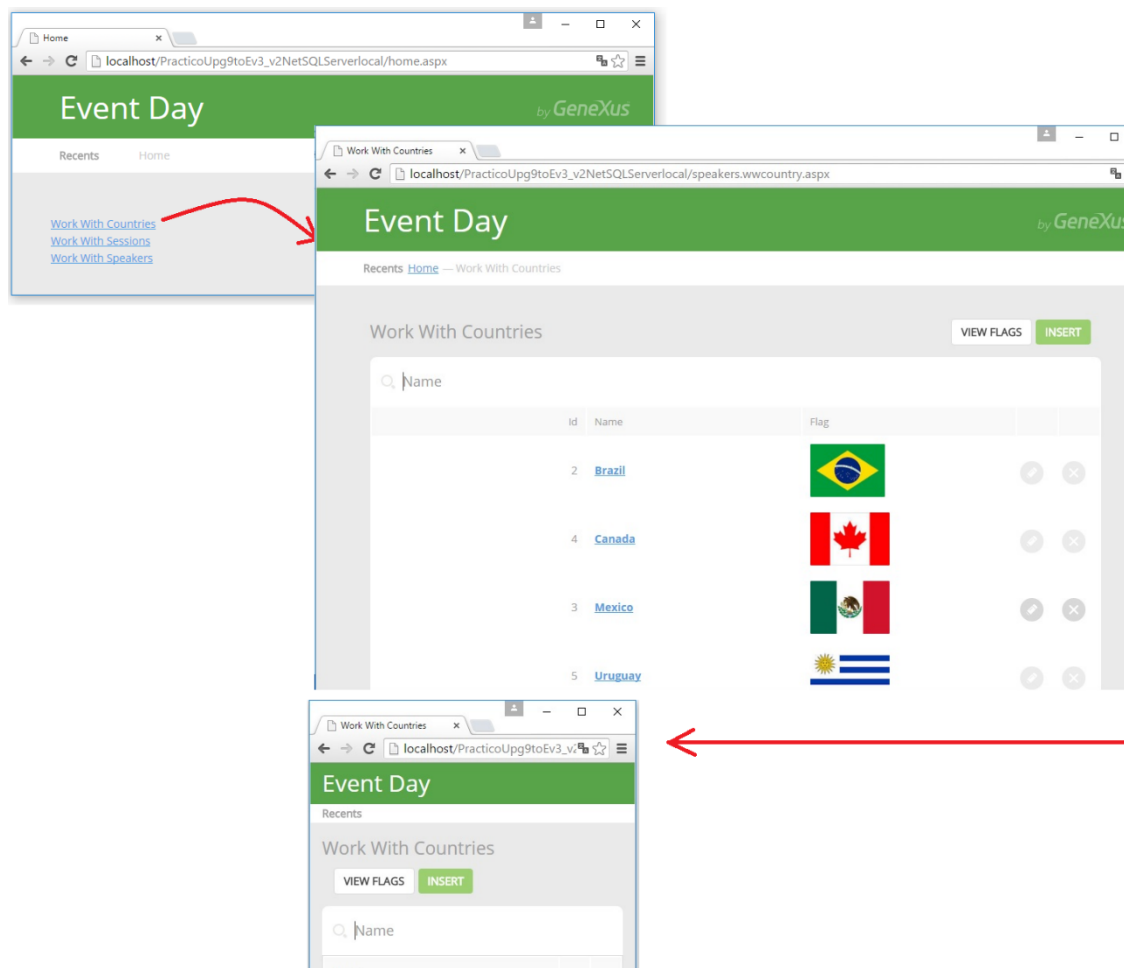
Web interface	
Default Theme	Flat
Web User Experience	Smooth
Default Web Form Editor	Abstract Layout
Web Form Defaults	Responsive Web Design
Default Master Page	RwdMasterPage
Datepicker image	calendar

- Observar que pasará a utilizarse la master page RwdMasterPage (bajo el nodo Root Module/GeneralWeb). Ver que existe una tabla responsive dentro de la que se han colocado tres controles: un textblock, un web component y el content holder.
- Modificar el Caption del textblock.



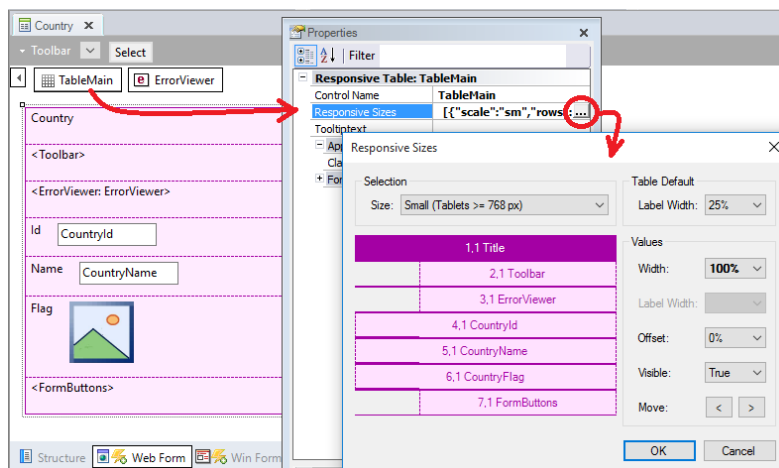
Hacer Rebuild all y luego ejecutar la aplicación presionando F5.

Ejecutar el **Web Panel Home** para acceder al Work With Countries y modifique los tamaños de la ventana para ver cómo se adaptan los controles a su tamaño.



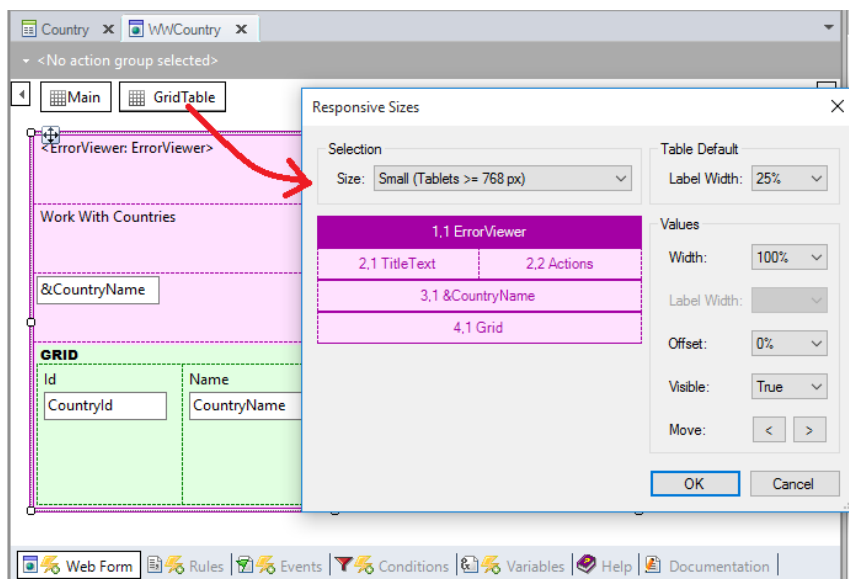
Edite un país y vea cómo se presenta ahora el form de la transacción (mueva el tamaño de la pantalla). Observe en GeneXus cómo está diseñado. Observe que <Toolbar> y <FormButtons> son controles de tipo "actiongroup".

Vea la tabla responsive sobre la que se ubican todos los controles. Vea que mediante la propiedad Responsive Sizes encuentra la ventana donde se configura cómo se muestran los controles para los cuatro tamaños Extra Small, Small, Medium y Large:



Visualice las diferencias entre la pantalla Extra Small y la Small y aprécielos en ejecución.

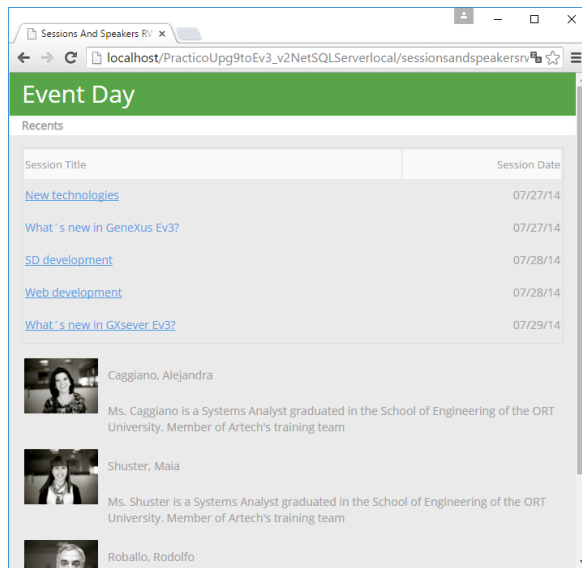
Abra el Web panel creado por el pattern: WWCountry, y observe su web form. Vea que hay una tabla responsive de nombre Main y dentro otra de nombre GridTable. Observe el diseño de esta última para los diferentes tamaños de pantalla (en particular la diferencia entre Extra Small y Small y vea su implicancia en la ejecución):



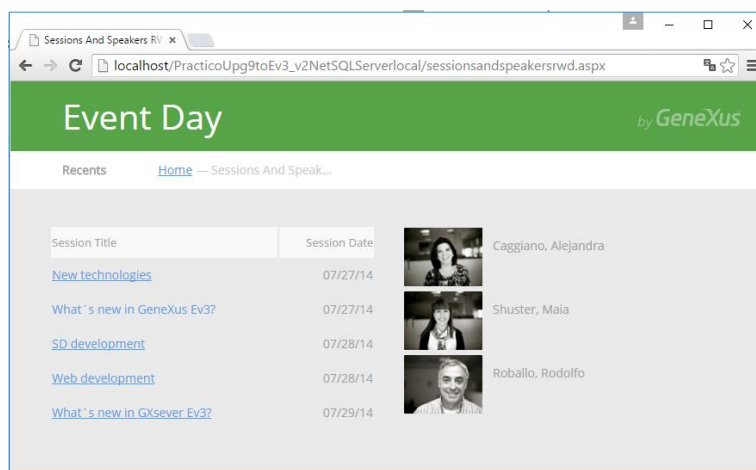
8.2. Opcional

Observe en ejecución el web panel SessionsAndSpeakers que había creado antes. Verá que al variar el tamaño de la ventana, no se adapta la información que se muestra. ¿Por qué?

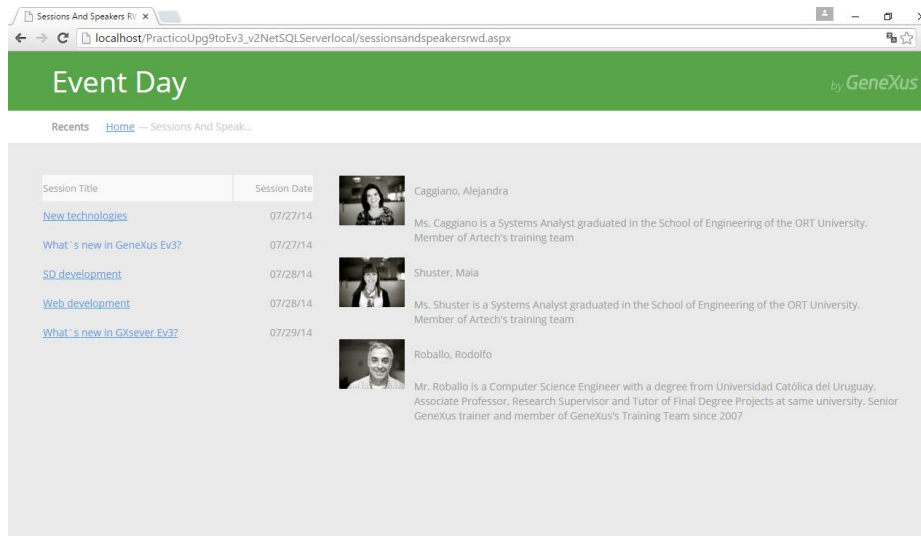
Queremos que cuando el panel se abre en una pantalla extra small, como la de un teléfono, los speakers de la sesión elegida en el primer grid se muestren debajo (foto, nombre completo y CV Mini), como vemos aquí:



Si la ventana es Small (como la de tabletas), queremos que se muestren a la derecha, y sin el CV Mini, como vemos aquí:



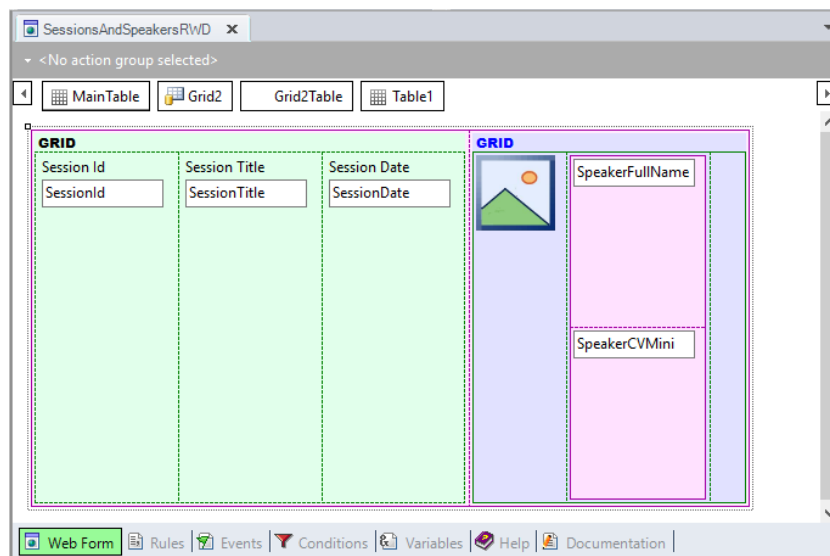
En cambio, si el tamaño es de un desktop, queremos que se visualicen a la derecha con CV, y que el grid con las sessions ocupe el 33% del ancho:



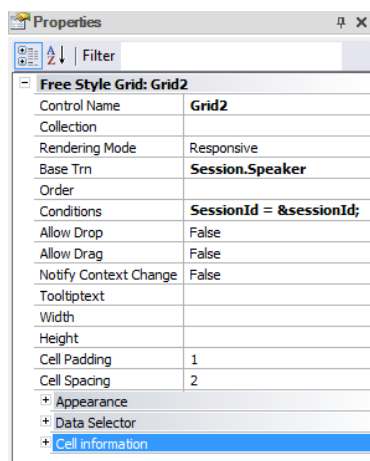
Para ello cree otro web panel: SpeakersAndSessionsRWD e impleméntelo.

Solución: Vea que por defecto aparece inicializado el form con una tabla responsive.

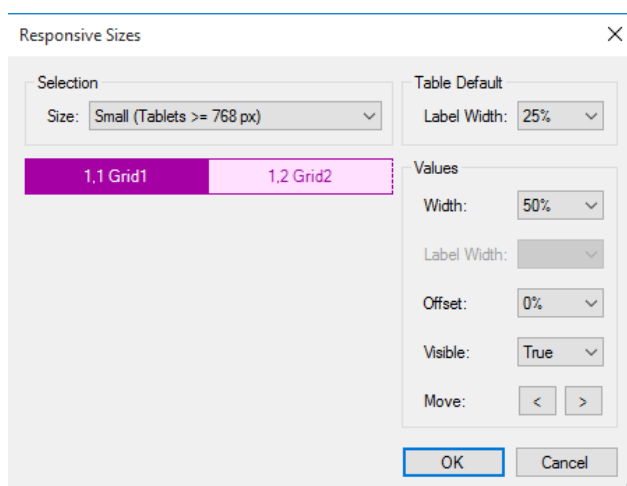
- Arrastrar un grid estándar sobre la tabla, con los atributos SessionId, SessionTitle y SessionDate. Configurar SessionId como invisible.



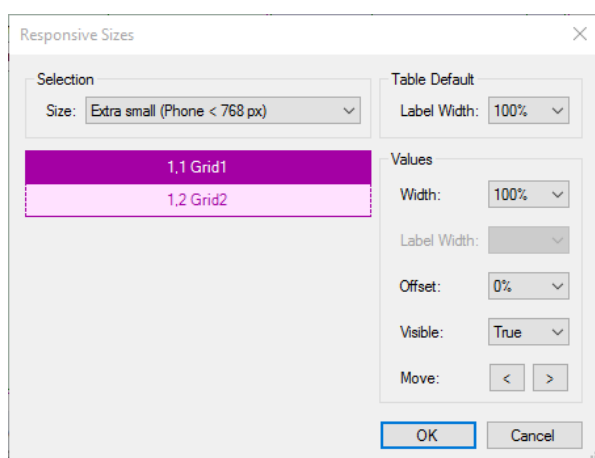
- A la derecha insertar un grid freestyle, "Grid2", e insertar el atributo SpeakerImage y a su derecha una tabla responsive dentro de la que irán los atributos SpeakerFullName y SpeakerCVMINI (a los que les quitamos las etiquetas mediante la propiedad "Label Position").
- Definir la variable &SessionId para cargarle el valor del atributo SessionId elegido por el usuario en ejecución al hacer clic sobre SessionTitle (definir el mismo evento SessionTitle.Click que en el web panel no responsive).
- En las propiedades del grid freestyle, configurar su Base Trn y las conditions para que se filtren solamente los speakers de la sesión elegida.



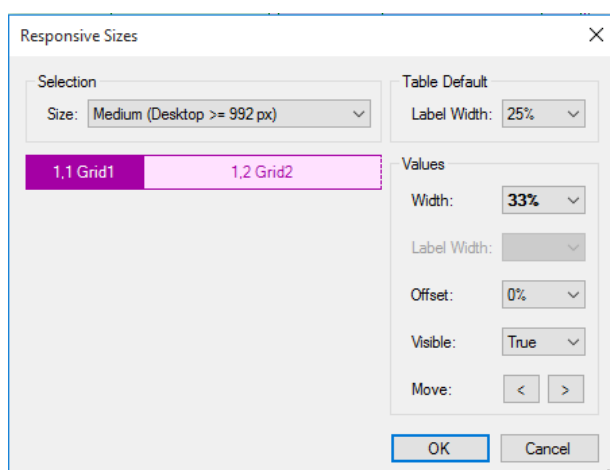
- Empecemos por definir las posiciones relativas de los dos grids de acuerdo al tamaño de pantalla. Para ello, posicóñese sobre la responsive table MainTable y entre sus propiedades edite “Responsive Sizes”:



- Cambie la selección del combo Size y vea la posición de cada grid. Vea que por defecto para el tamaño ExtraSmall los grids aparecen uno abajo del otro:

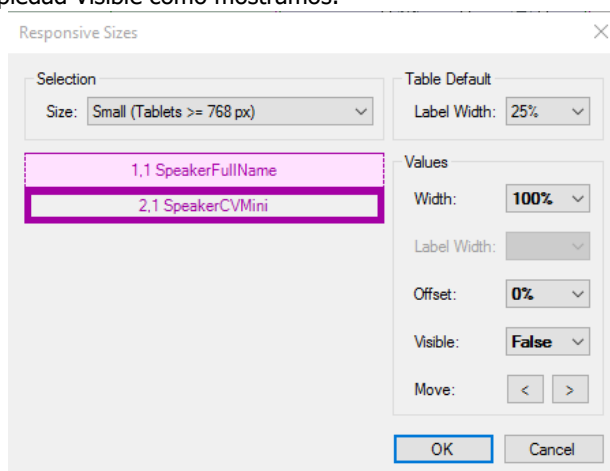


- Elija Size Small y vea que cada grid ocupa el 50% del ancho. Los tamaños siguientes heredan las propiedades del anterior.
- Elija Size: Medium. Y allí, posicóñese en el Grid1 y cambie el Width:



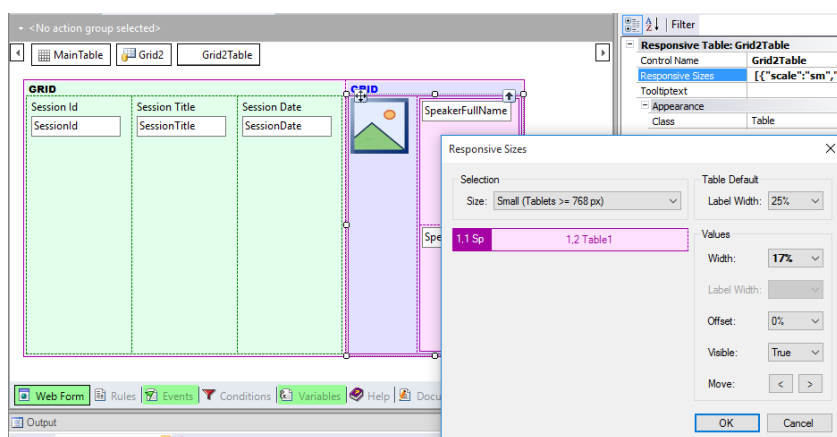
(vea que quedó cambiado también para el siguiente tamaño, Large)

- Ahora, queremos que cuando el tamaño de pantalla es Small, el CV Mini se oculte. Para ello, edite las propiedades de la otra tabla responsive, la que contiene a los atributos SpeakerFullName y SpeakerCVMini. Edite la ventana de Responsive Sizes, y eligiendo el tamaño Small, modifique la propiedad Visible como mostramos:



Dejarlo visible para los siguientes tamaños (Medium y Large).

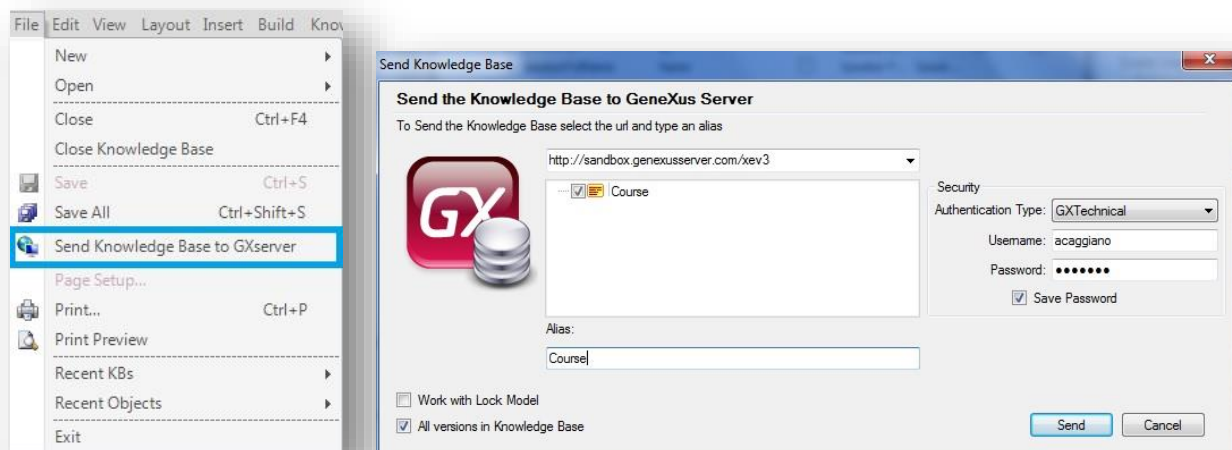
Por otro lado, para que en tamaño Small y mayores la imagen no ocupe el 50% del espacio asignado al grid freestyle edite las propiedades de la tabla responsive correspondiente y modifique su ancho por 17% o 25%.



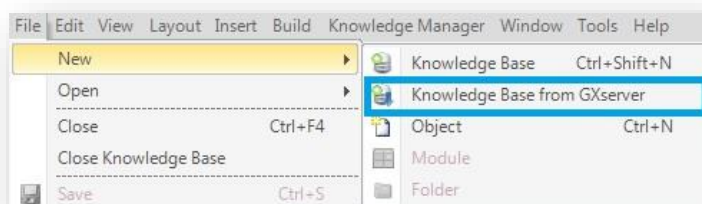
Presione F5 y pruebe en ejecución, achicando y agrandando el tamaño de la ventana del navegador.

9. Trabajando con GeneXus Server

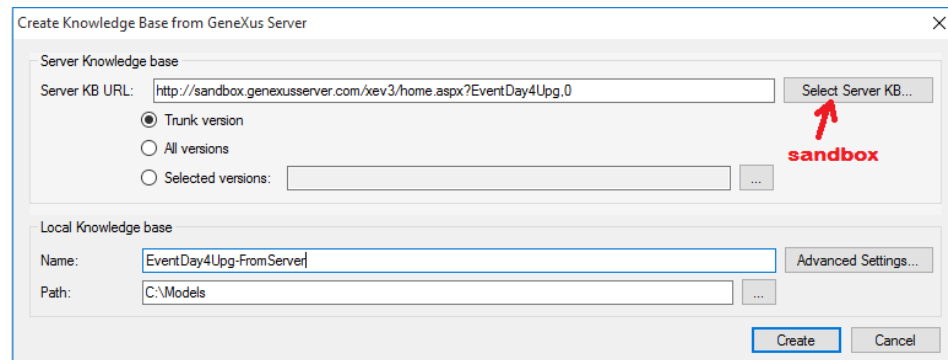
Envíe su KB al server. Para esto ejecute desde el menú File, la operación **Send Knowledge Base to Server**. Utilice la autenticación de GXTechnical.



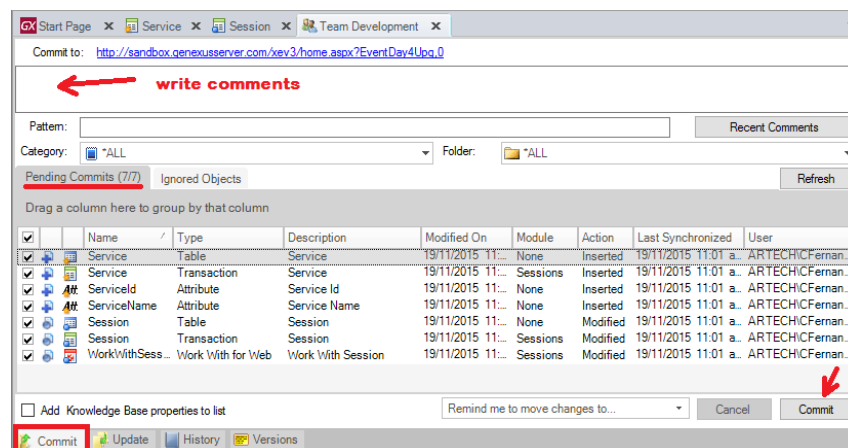
- Imagine que se ha integrado otro miembro a su equipo de desarrollo, que debe suscribirse en una KB local a la KB previamente hosteada. ¿Cómo lo hace?
 - Desde el menu File\New, ejecute la operación **Knowledge Base from GXserver**:



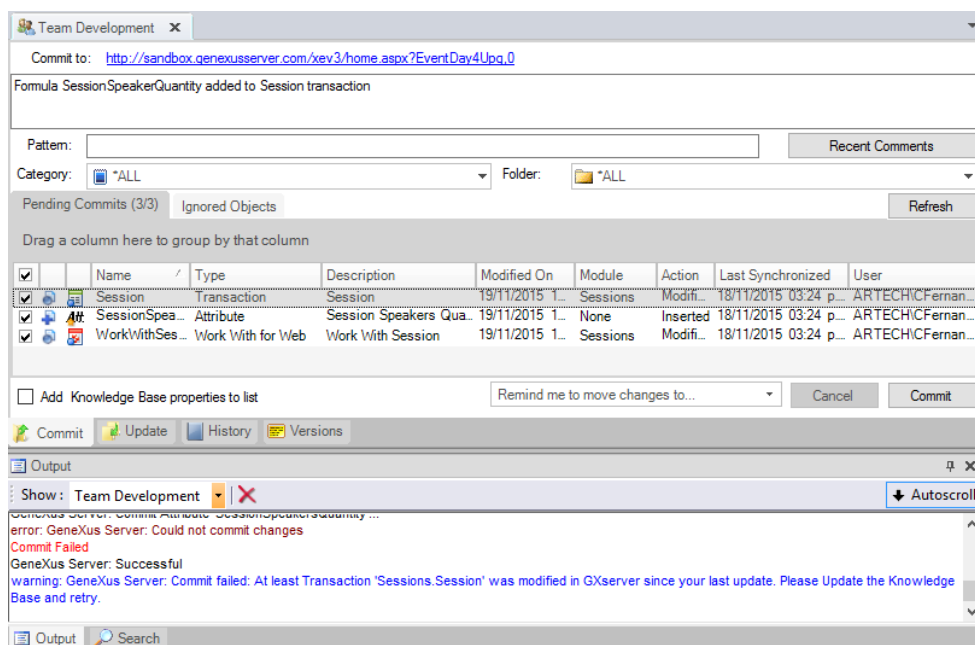
- Elija el servidor Sandbox y en la ventana que se le despliega, la KB que había subido al server en el paso anterior (en el ejemplo, la de nombre EventDay4Upg). Vea que se le permite modificar el nombre con el que será creada la KB local, así como su ubicación:



- En la KB local recién creada defina la transacción Service con los atributos ServiceId, ServiceName. Observe el circulito naranja que aparece sobre el ícono de la transacción en el Folder View. Agregue dichos atributos a la transacción Session. Vea que en el Folder View aparece la transacción marcada con un circulito azul, así como su archivo de instancia.
- Envíe estos cambios al server.
 - Para ello despliegue la ventana **Team Development** bajo el menú Knowledge Manager, y seleccione la solapa **Commit**. Observe qué objetos se despliegan entre los pendientes de commit (aparecen todos seleccionados por defecto)
 - Agregue como comentario "Transaction Service has been created and their attributes have been added to Session transaction". Ejecute la operación Commit:



- Vaya a la pestaña History y observe las entradas.
- Cierre esta KB local y abra su KB original. Agregue el atributo SessionSpeakersQuantity fórmula count(SpeakerId) a la transacción Session. Intente enviar esta modificación al server. ¿Por qué falla el commit?



Deberá ejecutar la operación de **Update** para recién luego poder hacer el **Commit**. Para esto seleccione la solapa Update del diálogo Team Development. Presione el botón Preview para poder ver todas las acciones que se realizarán sobre sus objetos. Presione el botón Cancel Preview y ahora sí, presione el botón Update. De esta forma estará recibiendo todas las modificaciones en su KB local. Observe la estructura de la transacción Session. Vea que se han integrado los cambios, es decir, que se hizo un Merge. Sobre la transacción haga botón derecho/History. Seleccione las dos últimas revisiones y mediante botón derecho, compárelas. Compare ahora la última con la anterior. Observe que siempre puede volver a dejar activa una revisión vieja.

- Ahora sí podrá realizar la operación de Commit para que en la KB administrada por el server quede la estructura final de la transacción Session.

10. Opcional

• **Versionado de la KB**

GeneXus nos permite llevar un control de los cambios realizados a los objetos de una KB. Podemos trabajar con líneas de desarrollo paralelas (Development Versions) y sacar "fotos" a la versión activa en un momento dado (Frozen Versions). A la rama principal del árbol de versiones la llamamos Trunk.

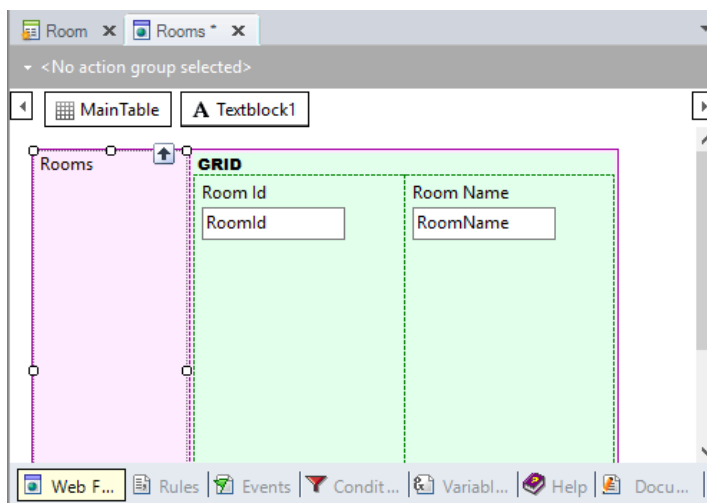
- Para probar estas nuevas funcionalidades, abra la ventana **Knowledge Base Versions** (View/Versions). Observe el nodo raíz del Trunk, que tiene el nombre de la KB y la indicación de que se está trabajando en esa versión.
- Vea que cuando importó el xpz, para conservar el estado de la KB previo a la importación, GeneXus congeló la versión en una Frozen version llamada ImportAutomaticBackup. Observe la opción de Revert que se le despliega al hacer botón derecho sobre esta Frozen version.

- Cree una Frozen Version "Version 1.0" a partir del nodo raíz (botón derecho/Freeze). Actívela y vea que ahora el nodo raíz del Folder View tiene el nombre Version 1.0. Abra un objeto cualquiera y vea cómo aparece ReadOnly.

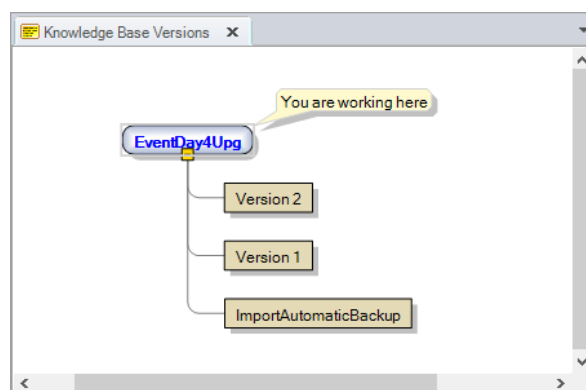
Vuelva a activar la versión trunk de desarrollo. Cree la transacción Room en el módulo Sessions, para almacenar las salas donde se realizarán las conferencias:

Name	Type	Description
Room	Room	Room
RoomId	Id	Room Id
RoomName	Name	Room Name

- Cree un web panel que contenga un grid que muestre las salas y que tenga un textblock "Rooms":

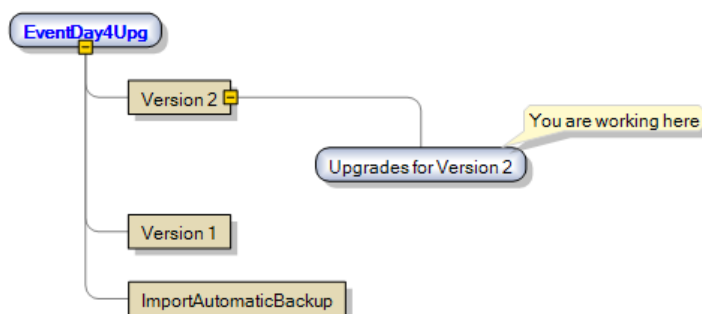


- Haga F5, cargue registros para las salas y ejecute el web panel.
- Cree otra Frozen Version "Version 2" también a partir del nodo raíz. Observe que las FV más nuevas se van a agregando hacia arriba.

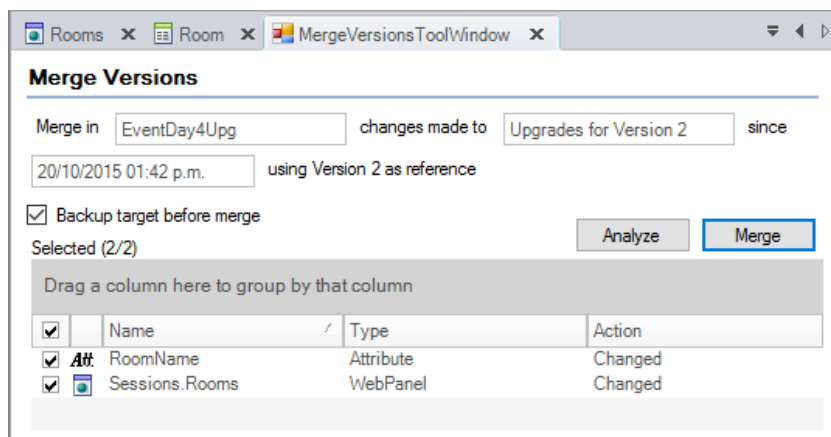


- Ponga activa la Version 1 (ReadOnly) y observe cómo desaparecen del Folder View ambos objetos (transacción Room y web panel Rooms) ¿Por qué se le deshabilitan todas las opciones del menú Build?
- Ponga activo nuevamente el Trunk, modifique el webpanel Rooms modificando el text block que había insertado (eg: "Rooms for sessions") y agregue atributo RoomFloor de tipo Numeric(1) para indicar el piso del edificio donde se encuentra esa sala.

- Cree una rama nueva (Development Version) a partir de la FV Version 2 y llámela "Upgrades for Version 2". Póngala activa.



- Abra el webpanel Rooms. ¿Qué texto espera encontrar? Abra la transacción Room. ¿Por qué no aparece al atributo RoomFloor?
- Cambie el tipo de datos del atributo RoomName por Character(10) y modifique en el web panel el text block por "Rooms 4 Sessions"
- Cambie la versión activa por la trunk y note los cambios de cada versión.
- Ahora haga un merge de la versión de desarrollo "Upgrades for Version 2" en la versión trunk.



¿Por qué al presionar Merge se le informa que el web panel Rooms tiene un conflicto? Observe que GeneXus sobrescribió el objeto en la versión destino, con el objeto en la versión origen. Ábralo y observe que el text block tiene el caption "Rooms 4 Sessions". Si le interesa conservar la versión que había antes del merge, ¿cómo la obtiene? (Solución: acceda al History del web panel y active la versión anterior).

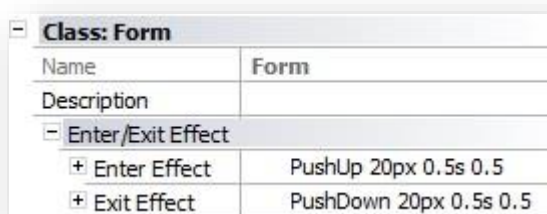
- Congele la versión de desarrollo "Upgrades for Version 2".

- **Transiciones**

Se desea cambiar el efecto de transición entre las páginas de la aplicación.

Solución

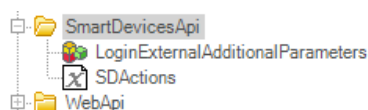
- Edite las propiedades de la clase Form del Theme Flat aplicado.
- Seleccione el efecto deseado en las propiedades Enter/Exit Effect.



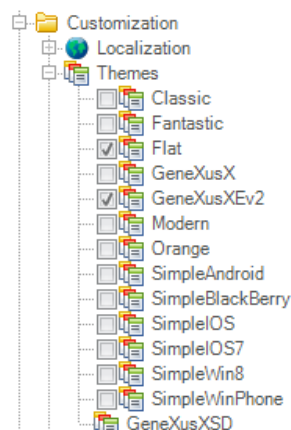
- Ejecute nuevamente la aplicación, y observe el comportamiento al pasar de una página a otra.

11. Aplicación móvil

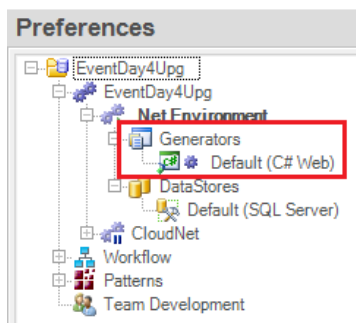
Antes de hacer nada, observe el contenido del folder SmartDevicesApi:



También observe los themes activos:

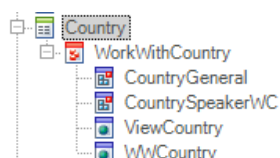


Y por último los generadores de su environment activo:

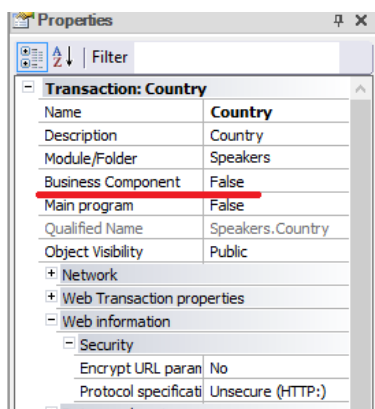


Active el Environment en la nube. Mueva las transacciones Country, Speaker y Session al módulo Root. Y haga un Rebuild all.

En el Folder View posicione sobre la transacción Country. Observe el árbol:

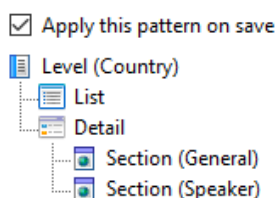


En las propiedades de la transacción observe que Business Component está en False.



Abra la transacción y aplíquela el pattern **Work With for Smart Devices**.

Investigue los distintos nodos:

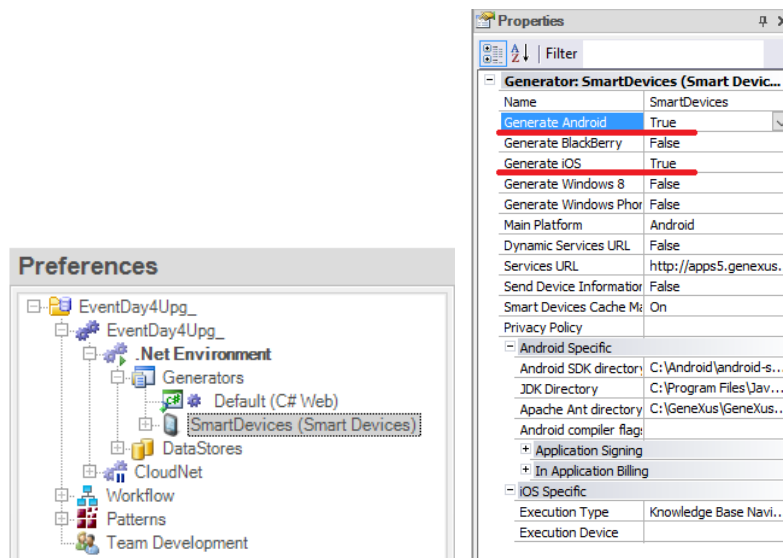


¿Por qué aparece el nodo "Section (Speaker)"?

Grabe.

Observe los cambios en:

1. el folder SmartDevicesApi
2. los themes activos
3. los generadores del Environment:

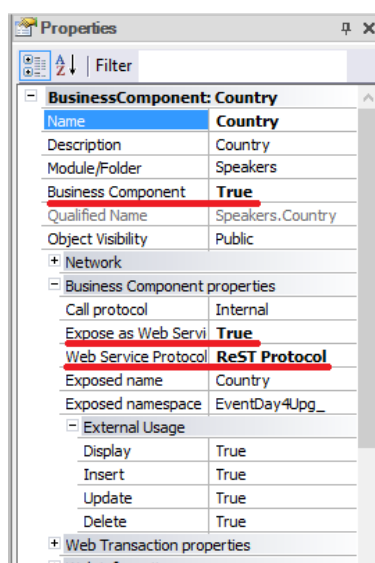


4. la transacción Country en el folder view:



¿El "WorkWithDevicesCountry" es un archivo de instancia, como el "WorkWithCountry" web?

5. Las propiedades de la transacción Country. ¿Por qué Business Component está en True?



A la transacción Speaker agréguele la siguiente regla:

```
Error( 'Speaker Surname must not be empty') if SpeakerSurname.IsEmpty();
```

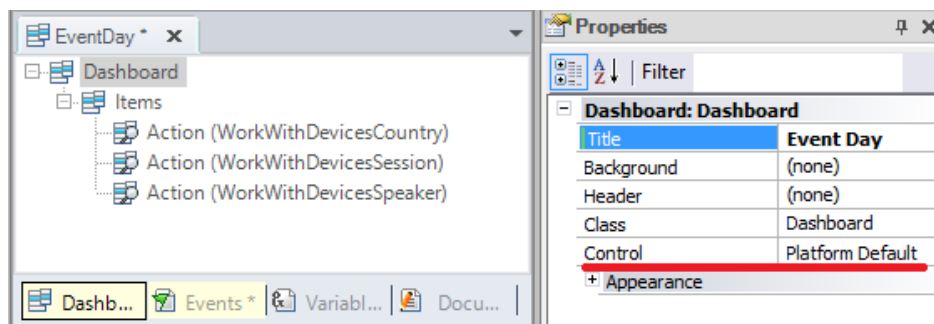
Aplique el pattern también a las transacciones Speaker y Session.

Antes de seguir, observe que se han creado

Si ahora quiere probar la parte de la aplicación móvil que acaban de implementar los patrones en el emulador de Android que viene instalado con el SDK, ¿cómo hace? Empecemos por prototipar a través del intérprete Knowledge Base Navigator y más adelante probaremos la aplicación compilada.

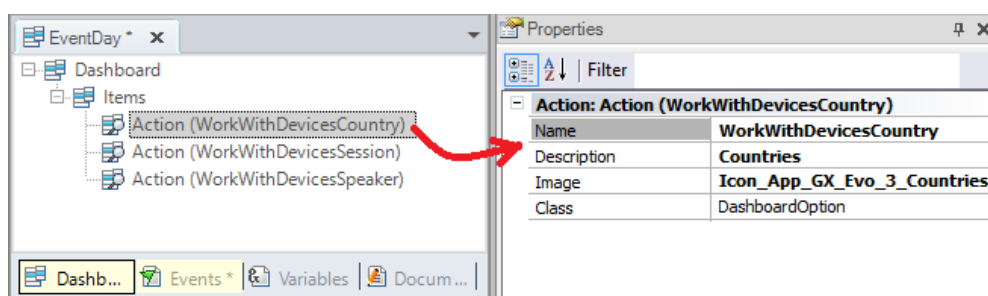
Solución:

Deberá tener algún objeto main de Smart Devices, porque el KBN ofrece ejecutar los mains. Cree un menú de entrada (objeto dashboard) que invoque a los tres work with:



Observe los valores que puede asignar a la propiedad **Control**.

Las imágenes para cada opción puede obtenerlas del folder Icons/Android icons que le brindarán los instructores. Debe insertarlas en la KB cuando vaya a asociar la propiedad Image a cada acción del dashboard:

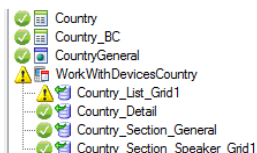


Vaya al tab Events y observe cómo hay un evento por cada acción, que invoca a cada Work With.

Observe las propiedades del objeto Dashboard. ¿Ve que por defecto la propiedad **Main program** está en True?

Ahora sí, **F5**.

Preste atención al listado de navegación. Observe cómo aparecen los Data Providers creados automáticamente por GeneXus para implementar el acceso a los datos:

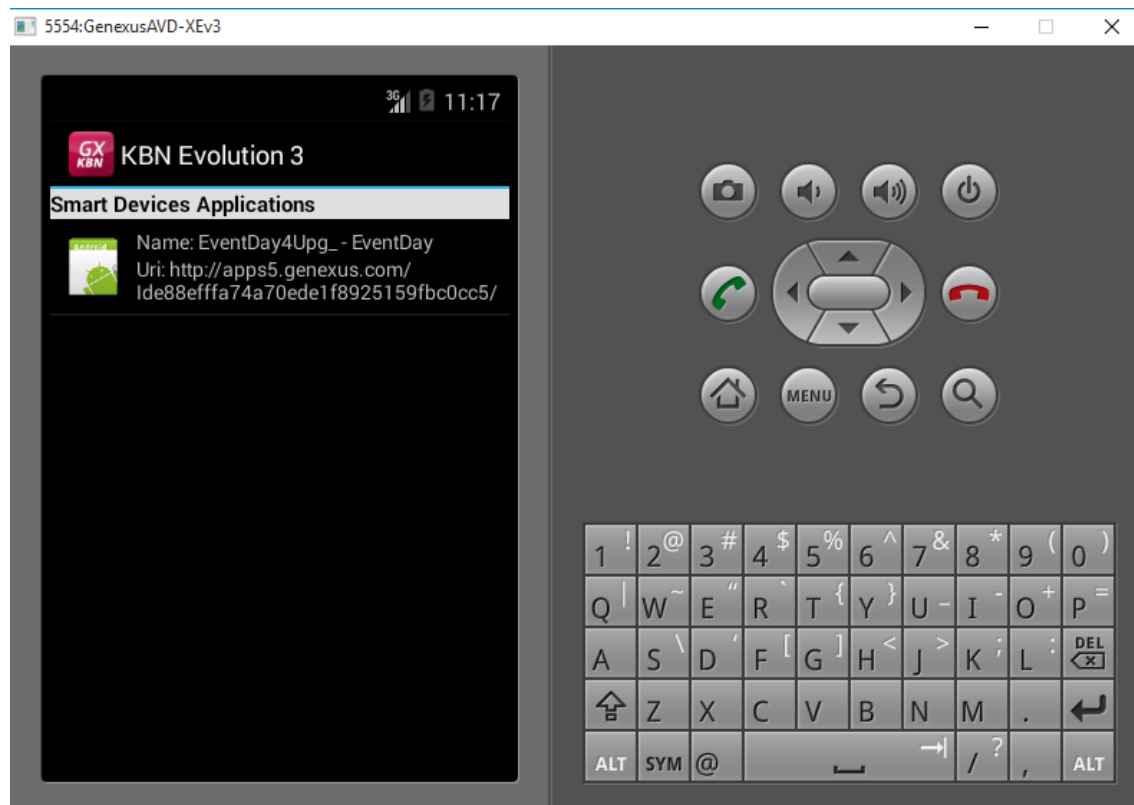


El emulador puede demorar en levantar, por lo que es posible que GeneXus arroje un error. En ese caso, vuelva a dar F5 después de que el emulador haya quedado levantado. Observe que la ventana de output mostrará casi al final:

```
\Waiting for device to become ready... (this may take a while)
Command: adb install -r "C:\GeneXus\GeneXusXEv3\Android\KBNavigator.apk"
pkg: /data/local/tmp/KBNavigator.apk
```

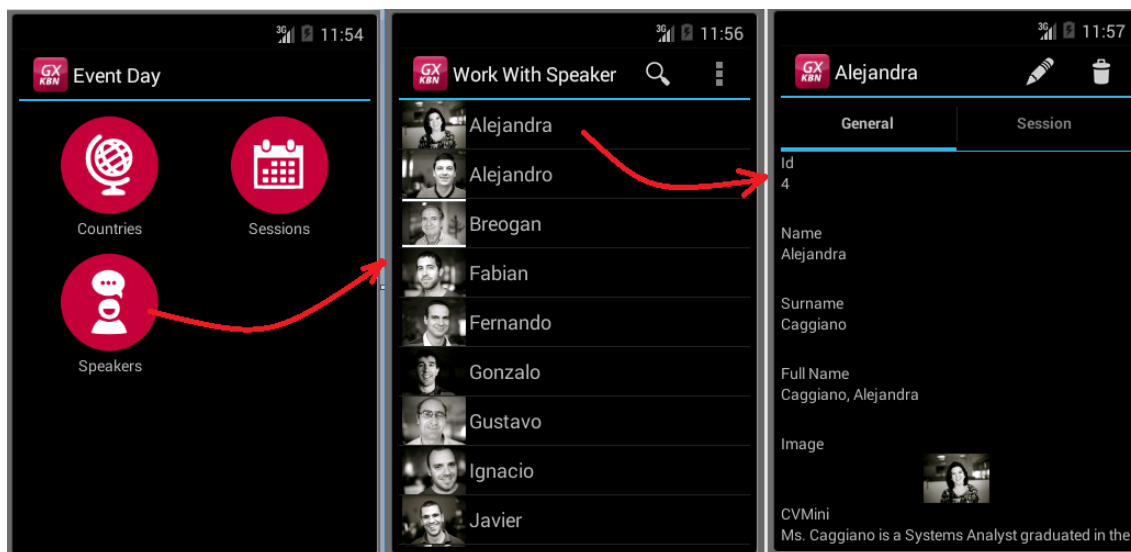
Se está intentando instalar el compilado del KBN en el emulador (KBNavigator.apk).

Al finalizar verá:



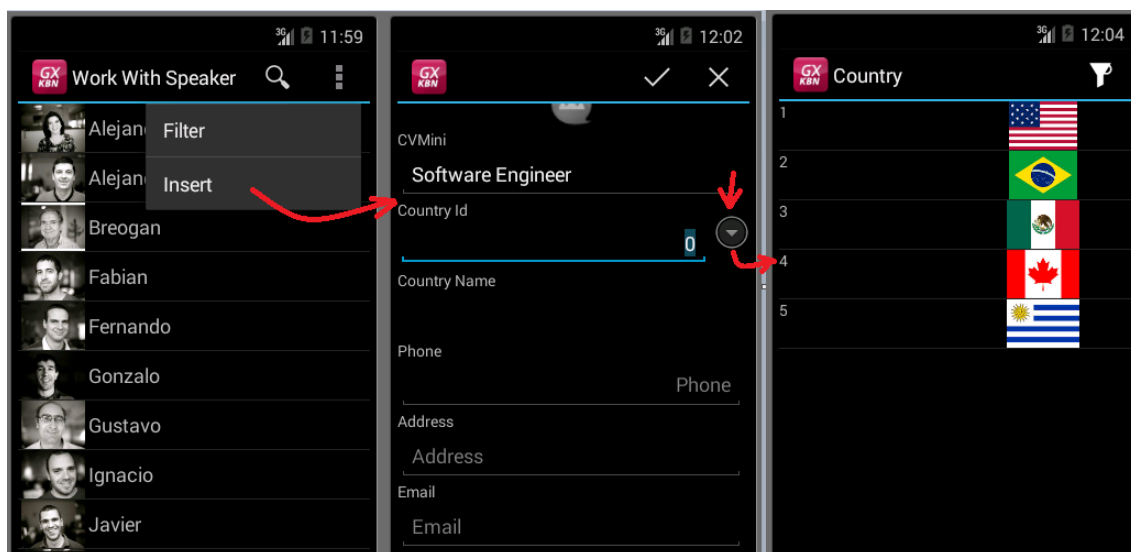
Observe que solamente se presenta una url para ejecutar: la del dashboard, único objeto main de nuestra aplicación SD. Si hubiera otro, también se ofrecería aquí.

Ejecute el dashboard. Navegue por las entidades.



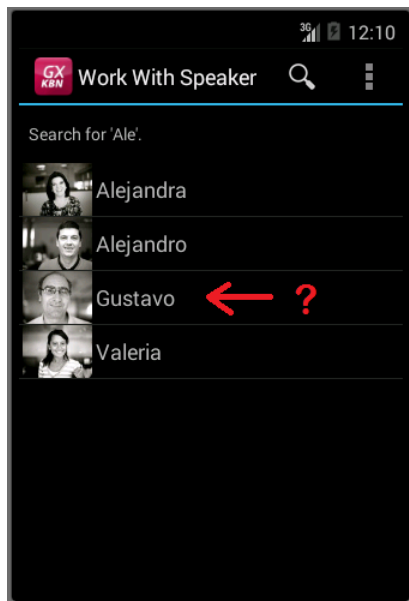
Intente crear un nuevo Speaker, dejando su apellido vacío.

Observe que cuando quiere ingresar el país del Speaker, se le ofrece un panel para elegir entre los países existentes. Ese panel es un prompt creado automáticamente por GeneXus. Vaya a GeneXus e identifíquelo. Verá que será un objeto del tipo **Panel for Smart Devices**.



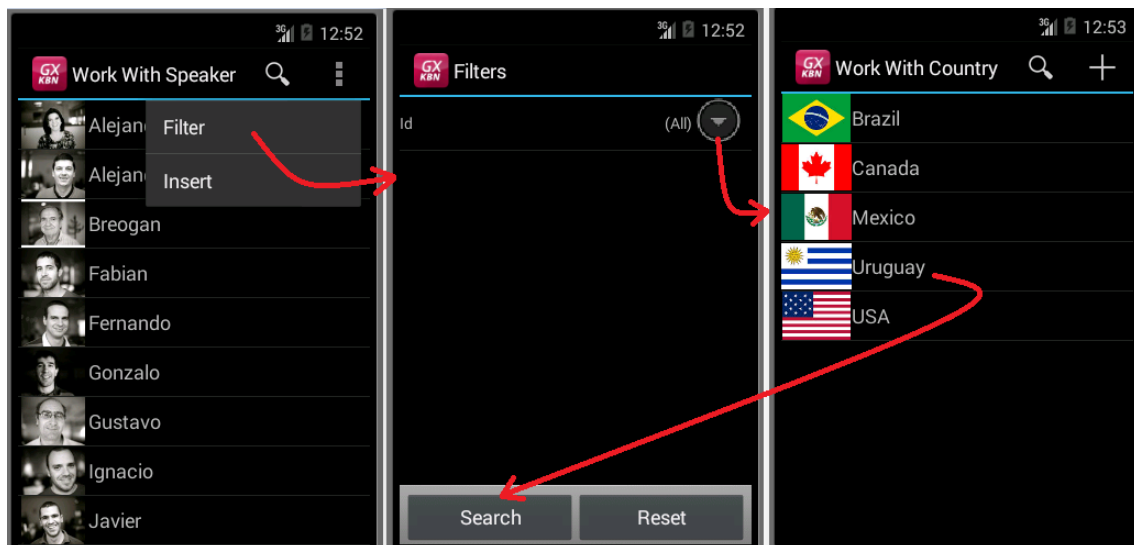
Vea que cuando quiera grabar, si dejó efectivamente vacío el Surname, se le mostrará un error. ¿Por qué? Asígnele un apellido y vuelva a intentar.

En el List de Speakers realice un Search por "Ale". ¿Por qué aparece "Gustavo"?



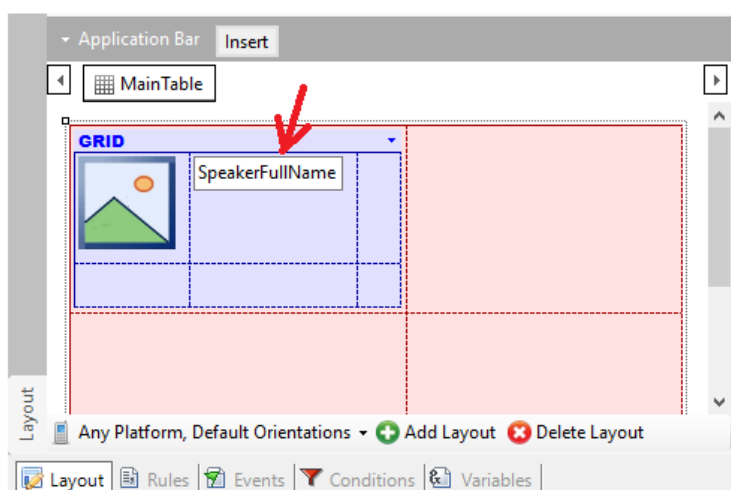
Vaya a GeneXus a observar en qué atributos se está realizando el Search. Recuerde que es a nivel del grupo Data de las propiedades del grid que se encuentra esa información.

Filtre en ejecución por país (querrá ver todos los speakers de Uruguay, por ejemplo).



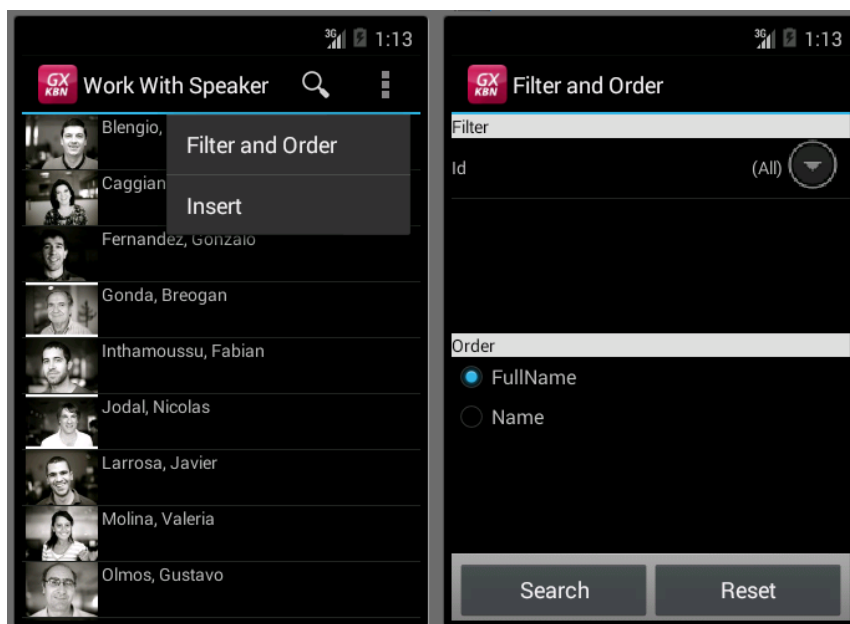
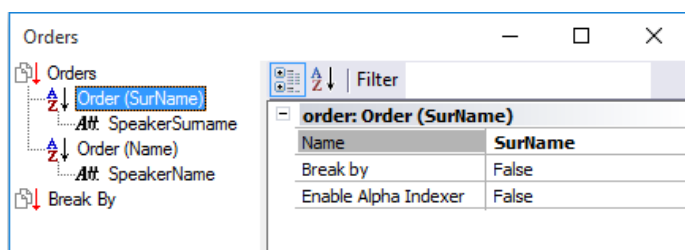
Busque dónde está definido este filtro en GeneXus.

Observe que se está ordenando la información del grid por nombre del Speaker. Vaya a GeneXus y muestre el FullName en lugar del Name:



Por defecto al insertar el control atributo lo insertará con etiqueta. Quítele la etiqueta (propiedad **Label Position**).

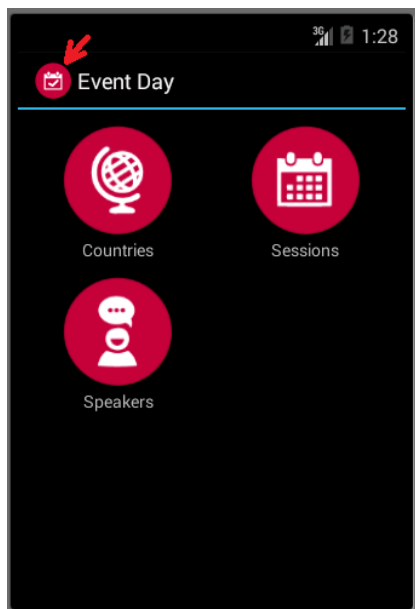
Por defecto seguirá ordenando por SpeakerName. Agregue otro orden por SpeakerSurName que sea el default y vea cómo en ejecución se le ofrece al usuario elegir el otro orden:



Ahora querrá probar en el emulador la aplicación compilada, no con el KBN. ¿Cómo lo obtiene?

Solución:

Si usted define el Dashboard como Startup object de su aplicación, el F5 compilará ese objeto. De lo contrario, si se posiciona sobre el dashboard y sobre él hace Run, también se compilará. En ambos casos no se obtendrá el Developer Menu Web.



Le hemos asignado un ícono a la aplicación, a través de la propiedad Android Application Icon del dashboard (el ícono lo obtiene del folder con los íconos que le pasaron los instructores).

Si cuenta con un dispositivo Android, puede instalarse este compilado en su dispositivo. Para ello, descárguelo leyendo el QRcode que le ofrece el Developer Menu. Para obtenerlo, en GeneXus vaya a View/Show QR Codes