

# Curso práctico de actualización a

# **GeneXus™ 15**

Julio 2019 - Práctico adaptado a GeneXus 16 - u4

*Copyright © GeneXus S.A. 1988-2019.*

*All rights reserved. This document may not be reproduced by any means without the express permission of GeneXus S.A. The information contained herein is intended for personal use only.*

**Registered Trademarks:**

*GeneXus is trademark or registered trademark of GeneXus S.A. All other trademarks mentioned herein are the property of their respective owners.*

OBJETIVO.....	4
COMENCEMOS.....	4
MÓDULOS.....	4
BUSINESS COMPONENTS.....	17
POBLADO DE TRANSACCIONES.....	23
TRANSACCIONES DINÁMICAS .....	24
WEB – USANDO LAS TRANSACCIONES DINÁMICAS .....	28
WEB - CARMINE Y NUEVO DISEÑO DEL WORK WITH FOR WEB.....	30
WEB - NAVIGATION BAR .....	34
<b>Opcional: menú dinámico .....</b>	<b>39</b>
WEB - CONTROL HORIZONTAL GRID .....	40
WEB - INFINITE SCROLLING.....	42
WEB - TAB CONTROL Y COMPONENTES .....	44
WEB - CALLOPTIONS TARGET .....	50
WEB - GLOBAL EVENTS .....	54
SD - APLICACIÓN ANDROID .....	57
SD - LIVE EDITING .....	59

AGREGANDO DISEÑO CON LIVE EDITING.....	66
SD - CONTROL HORIZONTAL GRID, SMART GRID, FLEX GRID .....	69
SD COMPONENTS Y GLOBAL EVENTS .....	71
SD – COMPARTIR INFO MEDIANTE ALGUNA DE LAS APPS INSTALADAS EN EL DISPOSITIVO (SHARE EO).....	74
SD - AUDIO CONTROLLER Y COLAS DE REPRODUCCIÓN.....	75
SD – PULL TO REFRESH .....	82
SD – DESPLEGAR PANTALLA A LA DERECHA: CALLOPTIONS Y NAVIGATION EO .....	82
REPORTING (OPCIONAL).....	84
<b>Card: Qry Conferences .....</b>	<b>84</b>
<b>Card: QryLikes .....</b>	<b>85</b>
<b>Chart: QryLikesChart .....</b>	<b>87</b>
<b>Pivot: QryLikesPivot .....</b>	<b>89</b>
<b>TimeLine: QryLikesTimeline .....</b>	<b>89</b>

## OBJETIVO

Conocer las principales nuevas funcionalidades de **GeneXus 15** para desarrollar **aplicaciones Web y Smart Devices (SD)**. Opcionalmente se verá lo nuevo en Reporting. Se ha modificado lo necesario para que este práctico pueda realizarse con GeneXus 16.

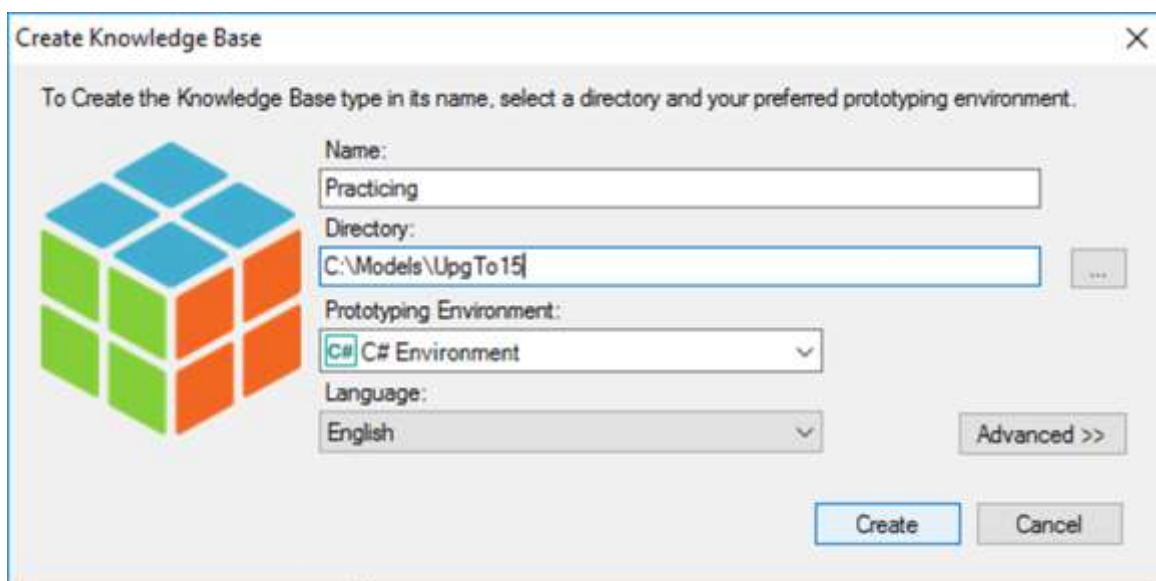
La aplicación sobre la que trabajaremos es **EventDay**, utilizada para el seguimiento de las charlas (sessions) y oradores (speakers) de un evento (tal como el encuentro GeneXus). Se partirá de una KB en GeneXus Server, ya inicializada con objetos que implementan app Web y SD.

## COMENCEMOS

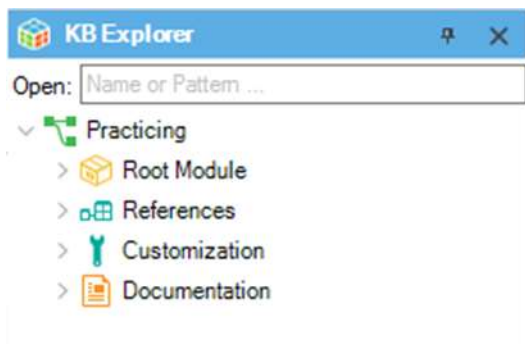
Utilizaremos la versión **GeneXus 16**, en su último upgrade liberado, el browser **Chrome** y el Emulador de **Android SDK**. Opcionalmente usted puede contar con otro emulador, como el genymotion, o incluso podrá probar directamente en su dispositivo.

## MÓDULOS

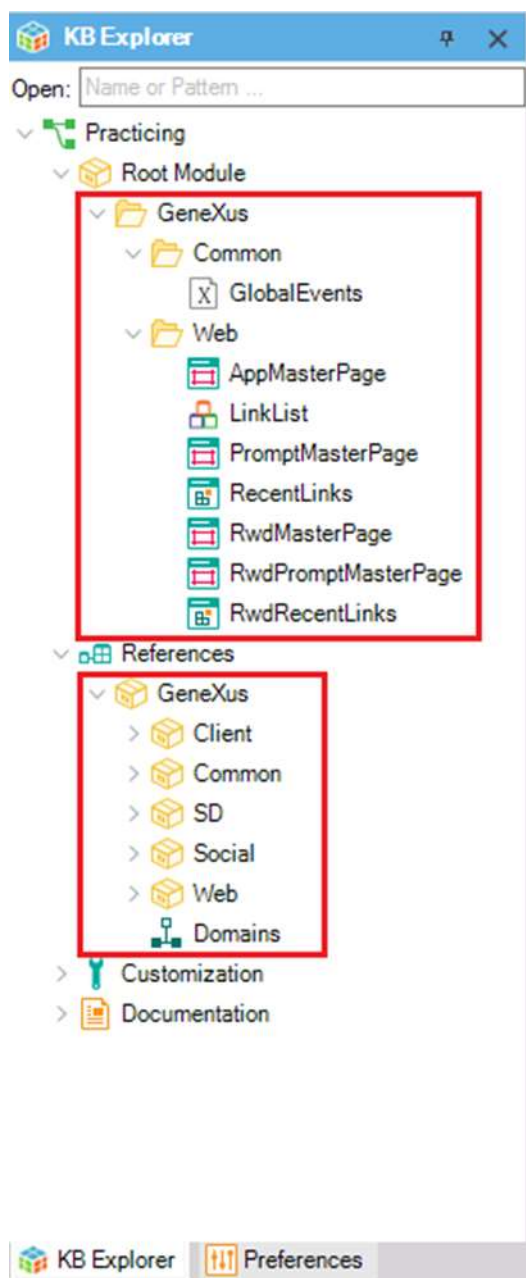
Comience por crear una KB desde cero. La intención es observar todo lo que viene inicializado por defecto.



Observar el IDE. Existe una nueva sección **KB Explorer** (sustituyendo a Folder View). La primera cosa que podemos ver es un módulo Root, y a continuación un nodo References:



Si expandimos ambos nodos:



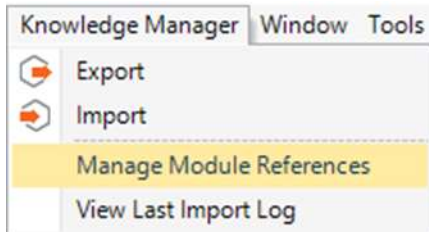
Podemos ver que el módulo Root viene con un **folder** GeneXus, y el nodo References con un **módulo** GeneXus, que contiene submódulos y Domains. ¿Qué es este **módulo** GeneXus? Para descubrirlo, acceda a la propiedad **Module Description** de este módulo. Dice: “GeneXus Core Module is a basic set of interfaces and implementations of data structures and algorithms to solve common programming use cases.”

¿Por qué todos los objetos y submódulos contenidos son read-only?

Observe que dentro de la propiedad **Module Manager Information** del módulo se le muestra la versión.

Abra alguno de los procedimientos del módulo (por ejemplo el de nombre AddDeviceGroups del módulo Notifications). Puede hacerlo a través del buscador de objetos en el KB Explorer, "Open:". ¿Por qué no puede ver el Source del procedimiento?

Si quisiera instalarse otra versión de este módulo, ¿cómo lo hace?



Cree un módulo Lib donde desarrollará funciones utilitarias para la aplicación, como un procedimiento para acortar un string y uno para formatear un DateTime.

Cree ambos procedimientos dentro del módulo, con las siguientes Rules, Variables y Source:

**Procedure TruncateString:**

```
parm( in:&LongString, in:&CharCount, out:&ShorterString );
```

**Variables:**

```
&LongString: Varchar(1024)
```

```
&CharCount: SmallInt = Numeric(4.0) /* con la asignación se definirá el nuevo dominio SmallInt. Observe que en la definición del tipo de datos aparecerá "SmallInt, Lib" */
```

```
&ShorterString: Varchar(256)
```

**Source:**

```
&Diff = &LongString.Length() - &CharCount
if &Diff > 0
&TextLen      = &CharCount - 3
&ShorterString = substr(&LongString,1,&TextLen) + "... "
else
&ShorterString = &LongString.Trim()
endif
```

## Procedure: FormatDateTime

```
parm(in:&DateTime, in:&FormatName, out:&DateString);
```

## Variables:

```
&DateTime: DateTime
```

```
&FormatName: Character(100)
```

```
&DateString: String = VarChar(255) /*con la asignación se definirá el nuevo dominio String.
Observe que en la definición del tipo de datos aparecerá "String, Lib" */
```

```
&Date: Date
```

## Source:

```
do case
```

```
case &FormatName = '!ddd hh:mm'
```

```
    &DateString = &DateTime.DayOfWeekName() + ' ' +
                padl(&DateTime.Hour().ToString().Trim(), 2, '0') + ':' +
                padl(&DateTime.Minute().ToString().Trim(), 2, '0')
```

```
case &FormatName = '!hh:mm'
```

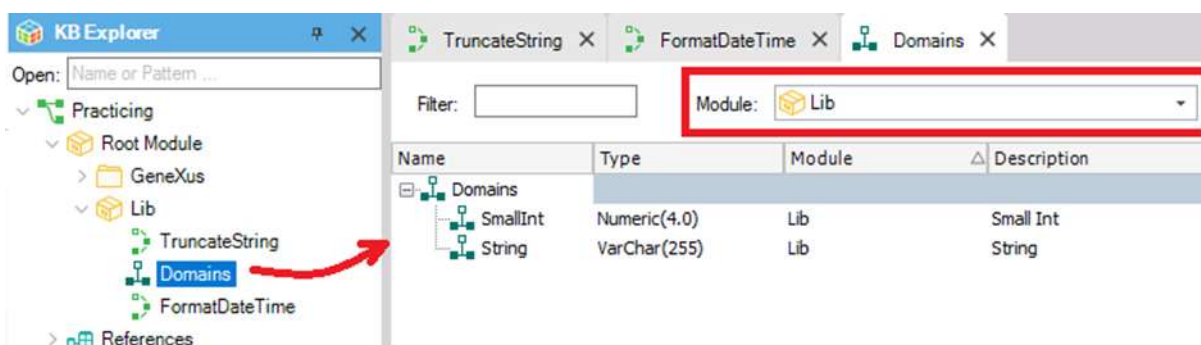
```
    &DateString = padl(&DateTime.Hour().ToString().Trim(), 2, '0') + ':' +
                padl(&DateTime.Minute().ToString().Trim(), 2, '0')
```

```
otherwise
```

```
    &DateString = &DateTime.ToFormattedString()
```

```
endcase
```

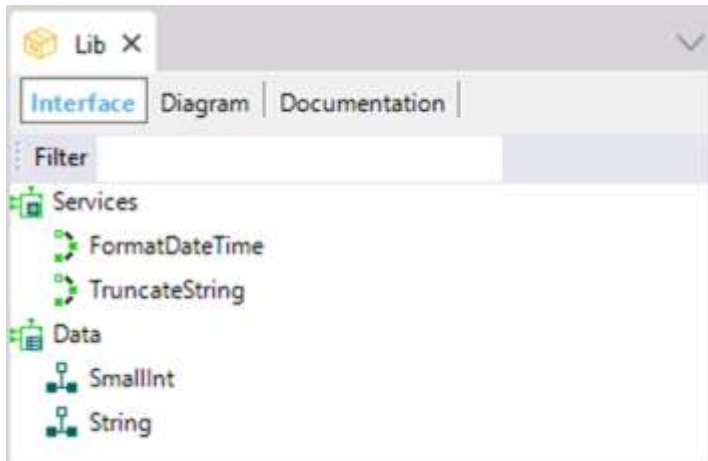
Observe que al definir los dominios se creó un nodo Domains dentro del módulo Lib.





La ventana Domains permite visualizar los dominios de otro módulo, así como de todos los módulos (elija \*ALL en el combo). Observe que hasta el momento los únicos dominios editables son los que acaba de crear.

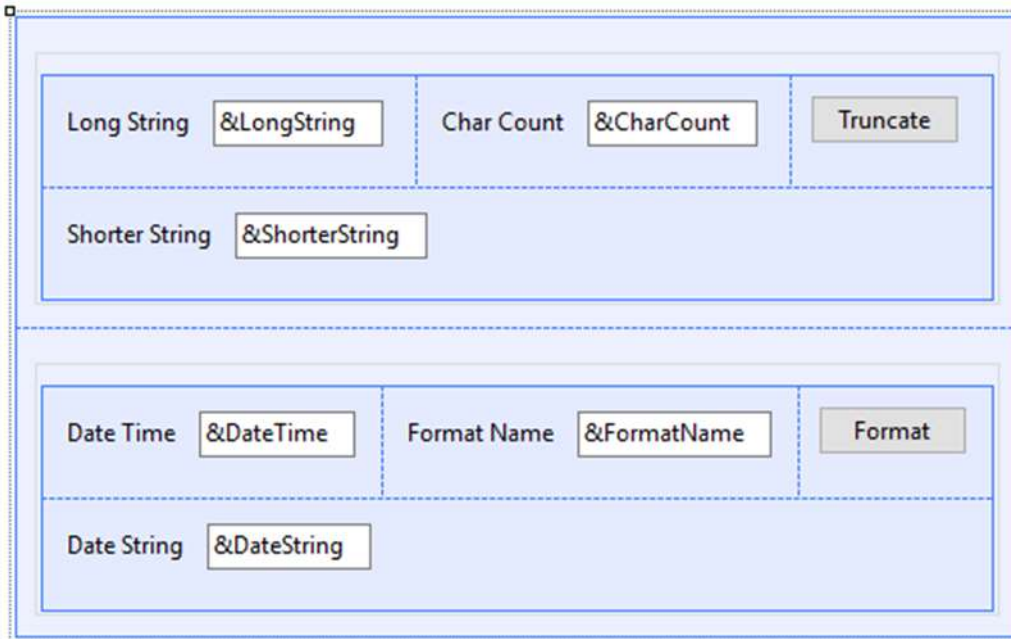
Posiciónese en el nodo del módulo Lib, botón derecho, Open:



Pruebe cambiarle la propiedad Object Visibility a FormatDateTime, por Private. Grabe, vuelva a abrir la Interface del módulo Lib y observe cómo ya no aparece como servicio. Vuelva a dejar la propiedad como estaba (con Object Visibility Public).

Creemos un web panel “Test” para testear los procedimientos:

Name	Type
Variables	
Standard Variables	
LongString	VarChar(1K)
CharCount	SmallInt, Lib
ShorterString	VarChar(256)
DateTime	DateTime
FormatName	Character(100)
DateString	String, Lib



```

Event 'Truncate'
  &ShorterString = TruncateString( &LongString, &CharCount)
Endevent

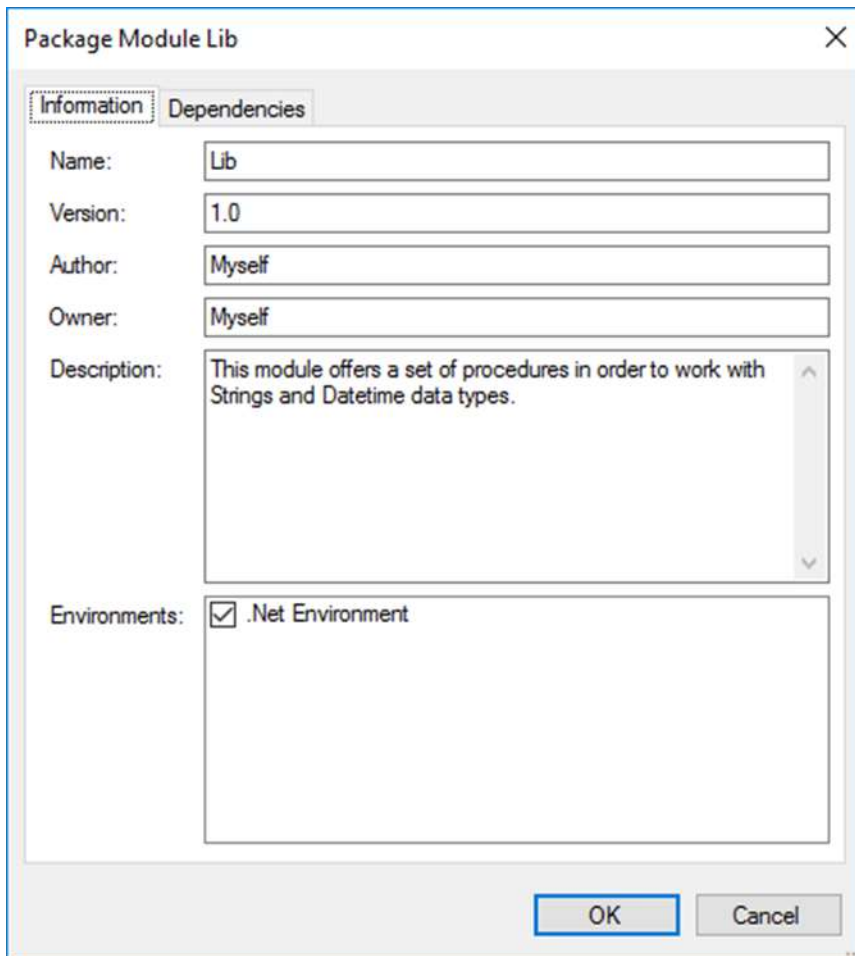
Event 'Format'
  &DateString = FormatDateTime( &DateTime, &FormatName)
Endevent

```

Pruébalo (consulte al instructor sobre si utilizar la opción Deploy to Cloud o prototipar local. Si está realizando el práctico en su propia máquina, elija la opción que prefiera). Si lo creó dentro del módulo, entonces deberá lograr que su visibilidad no sea “Public”.

Querremos empaquetar el módulo para distribuirlo a otras KBs. Como ese web panel tiene master page, quítesela para poder empaquetarlo. De lo contrario, tendrá que colocar también la Master Page dentro del módulo.

Ahora querremos empaquetar el módulo para poder instalarlo en otras KBs sin que los desarrolladores de esas KBs puedan modificar los objetos. Lo ideal sería probar las funcionalidades en todos los environments para los que desee que el módulo pueda utilizarse. Nosotros hemos probado únicamente con un environment .Net. Pero debería definir todos los environments para los que quiera obtener binarios y marcarlos (con los check boxes) para distribuirlos. No lo haremos en este práctico.



Al presionar OK, se hará un rebuild de todos los objetos para cada environment seleccionado, y luego creará un paquete que contendrá los binarios. Observe que quedará en:

"<Knowledge Base Directory>\modules\Lib\_1.0.opc"

opc: Open Packaging Convention. Es un archivo zip. Si lo extrae, podrá ver que dentro del folder Platforms/CSharpWeb contiene Lib.dll.

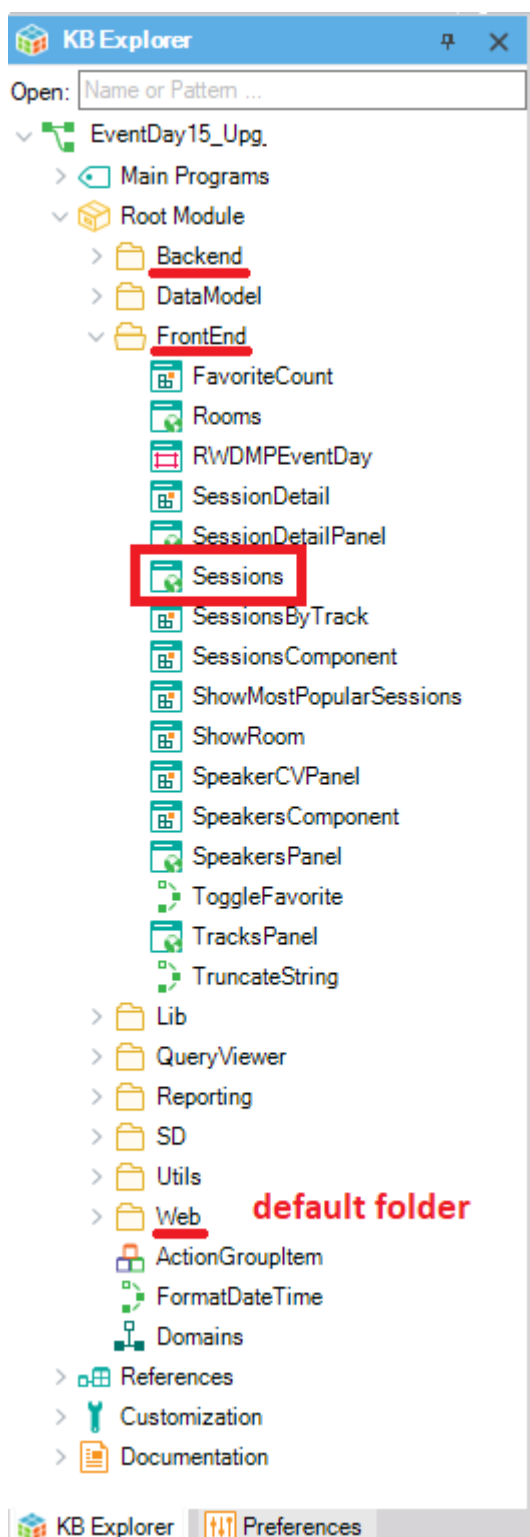
## CREATE KNOWLEDGE BASE FROM GENEXUS SERVER

Ahora cree desde GeneXus Server una Knowledge Base a partir de la KB de nombre **EventDay15** del server de samples ([samples.genexusserver.com/v16/main.aspx](http://samples.genexusserver.com/v16/main.aspx)).

Observe en las Preferences que tiene dos environments. Setee como activo el **CloudNet**. Consulte al instructor acerca de si prototipará en la nube o en forma local. Configure lo que el instructor le indique (o lo que usted prefiera si está trabajando por su cuenta). Si va a realizar dentro de este curso el módulo de Smart Devices, entonces más adelante deberá habilitar para el generador Smart Devices la generación para Android (observe que la propiedad viene con valor False del

server). Realice un Rebuild all para que se actualice el módulo GeneXus en caso de ser necesario. Mientras GeneXus trabaja, siga leyendo este documento.

En esta KB se crearon folders y objetos para el desarrollo de la app web, incluyendo queries para reporting, así como folders y objetos para el desarrollo de la app móvil nativa.



Empezaremos con la aplicación web. Esta consiste tanto en el desarrollo de un backend, como de un frontend.

Observe que bajo el Root Module se organizaron los objetos en folders.

- En el **DataModel** encontrará las transacciones, los objetos resultantes de la aplicación del pattern Work with (tanto para Web como para Smart Devices), Data Providers asociados a las transacciones y algunos procedimientos.
- En el folder **Backend** encontrará la master page default, y otros objetos, pero no el principal, pues éste será el Home que viene por defecto en el folder **Web** (cuando se le aplica el patrón Work With for Web a las transacciones). Lo hemos personalizado, como apreciará si lo abre.
- En el folder **FrontEnd** encontrará una master page diferente, creada para los objetos de la app para los usuarios finales. El objeto principal de la app del FrontEnd será el panel Sessions.

Antes de seguir, asegúrese de tener el environment .Net correctamente configurado como environment activo. Observe que tiene creado también el generador de Smart Devices. Solo prototiparemos para Android, haciendo uso del emulador. Si en su curso no se va a realizar la parte de Smart Devices, entonces asegúrese de que esté deshabilitada la generación tanto para Android como para iOS. Realice un Ctrl+F5 cuando el Rebuild all haya concluido.

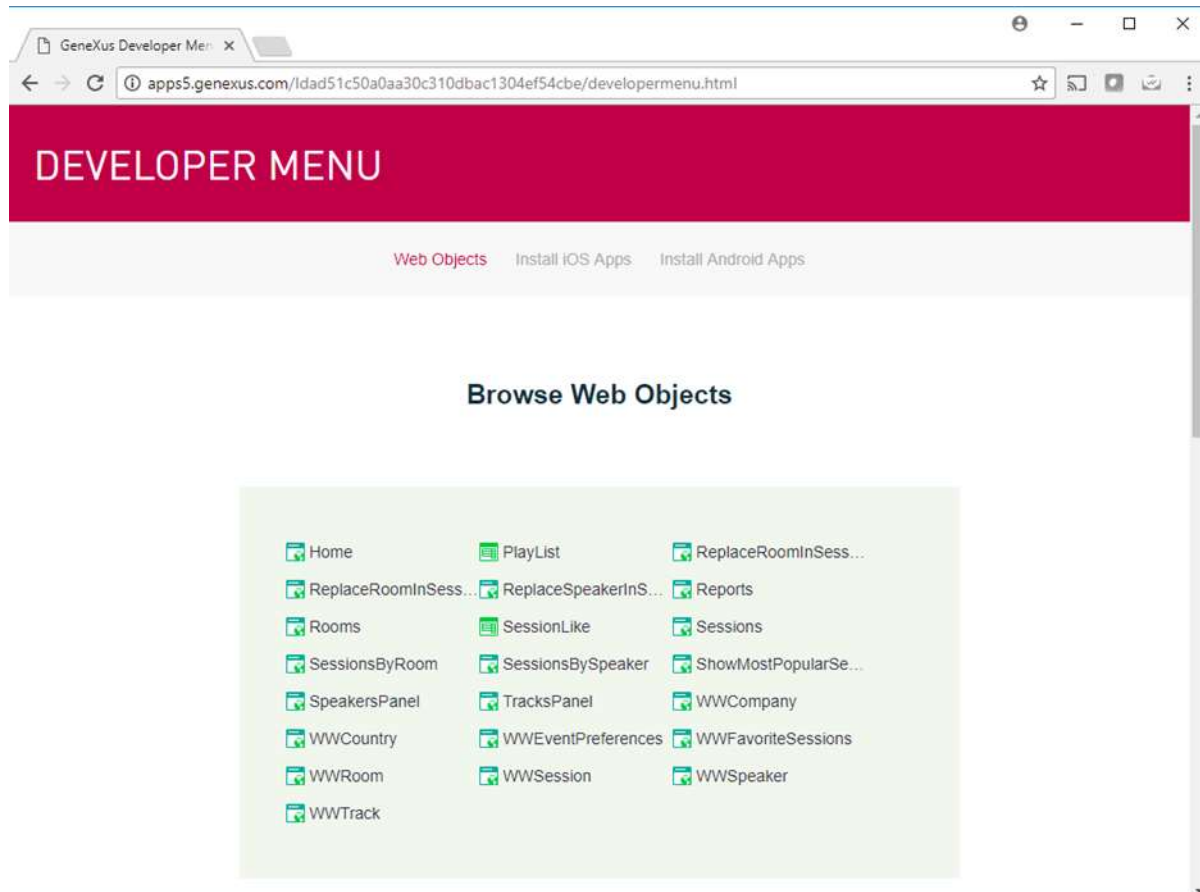
Observe que se ha mejorado la ventana **Output** de GeneXus con nueva font y un nuevo “find” incorporado, que muestra los resultados de la búsqueda resaltados.

Como no hemos configurado a nivel de la versión un Startup Object, la ventana de Output mostrará un error para la ejecución de Android (si es que habilitó su generación).

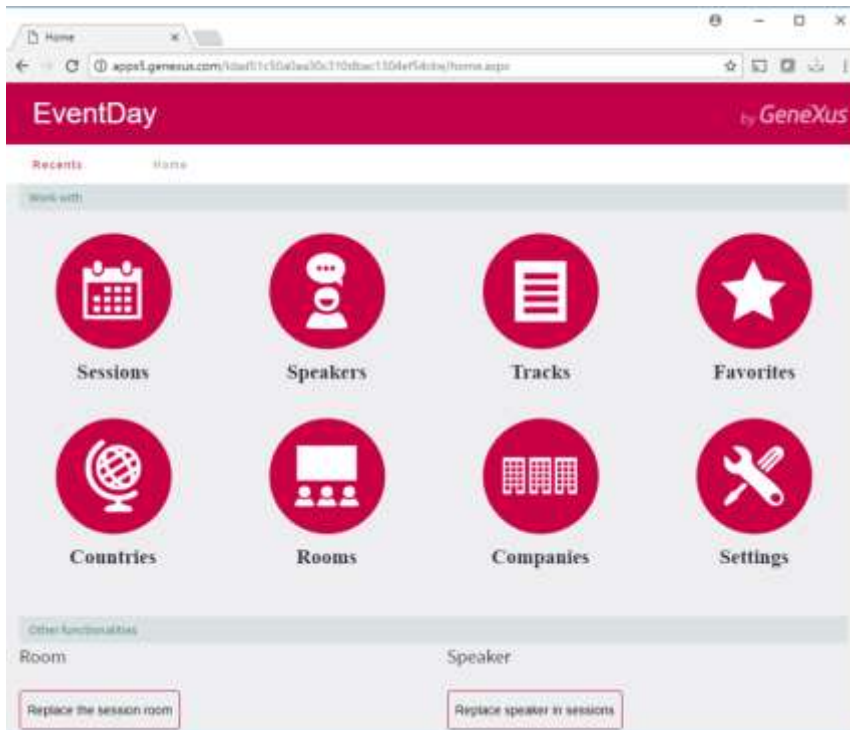
```
==== Android Execution started =====  
error: A startup object must be selected  
Android Execution Failed
```

No le prestaremos atención mientras desarrollemos la parte web. Como vamos a estar alternando entre backend y frontend, no configuraremos Startup Object.

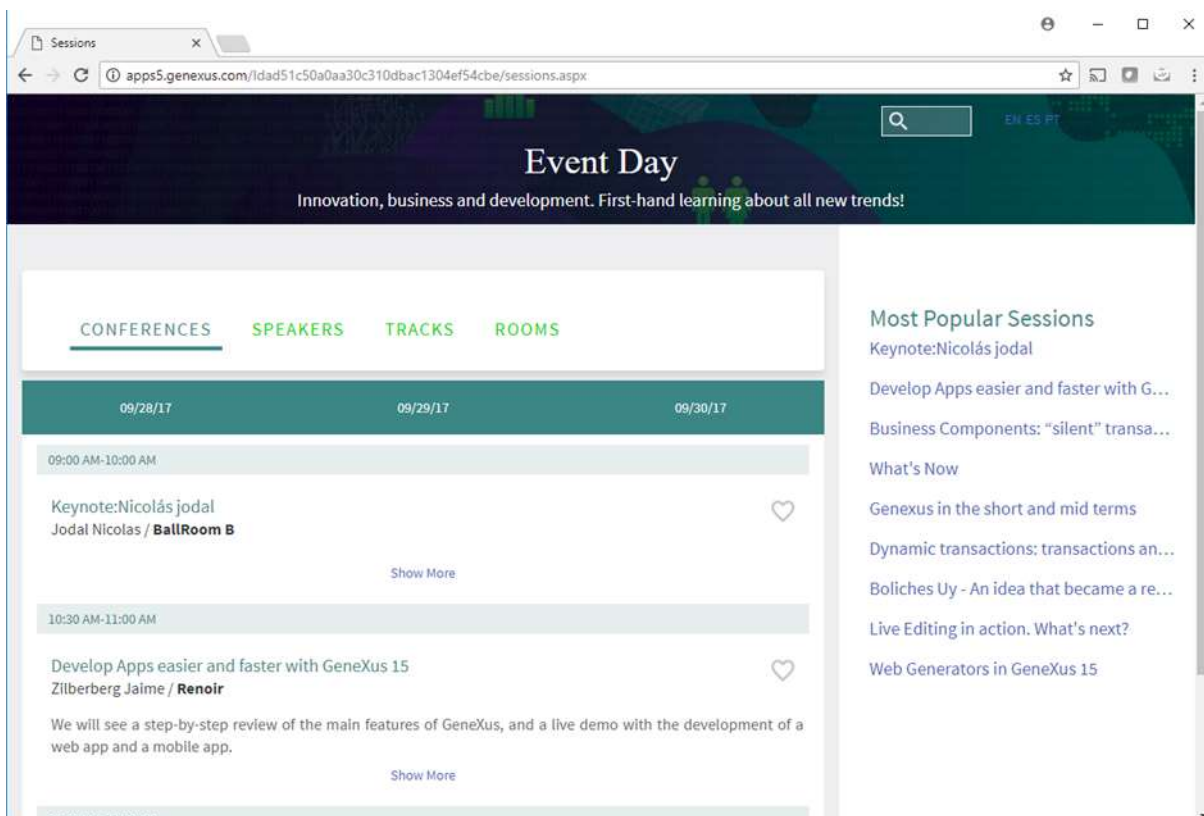
Se abrirá el DeveloperMenu:



**Backend:** Ejecute el panel **Home** para acceder al backend. Observe todos los “trabajar con” (sessions, speakers, tracks, favorites, countries, rooms, companies, eventpreferences –settings-). Además aparecen invocaciones a otras pantallas (bajo “Other functionalities”) que veremos luego.



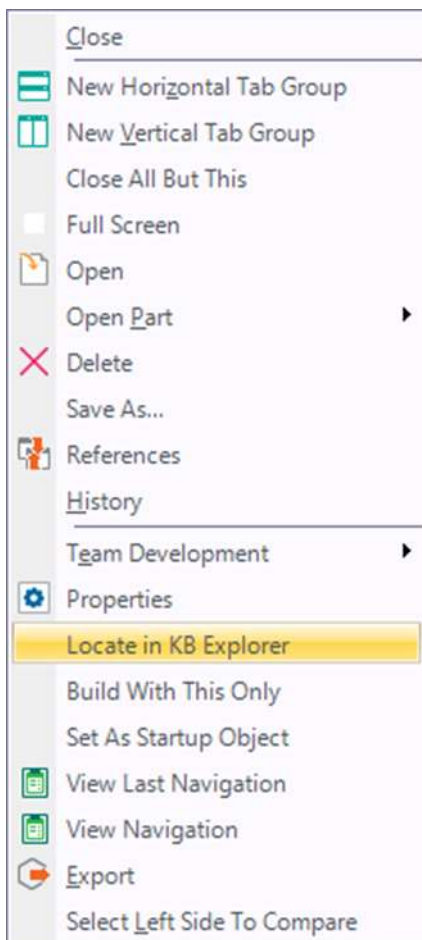
**Frontend:** Luego ejecute el main del Frontend (el panel Sessions).



Navegue un poco por la app para familiarizarse con lo implementado.

Vuelva a GeneXus y observe el folder **Lib** bajo **Root Module**. No tiene nada que ver con el módulo que usted ha creado y empaquetado. Es un folder que los desarrolladores de esta KB han creado por su cuenta. Pero si busca en el KB Explorer a través del campo Open un objeto de nombre “TruncateString”, lo encontrará. ¡Los desarrolladores de esta KB lo habían implementado! Ábralo. Está programado casi idéntico al que usted creó en el módulo que empaquetó.

Ubíquelo en el árbol del KB Explorer. Para ello recuerde hacer botón derecho sobre el objeto abierto / Locate in KB Explorer:



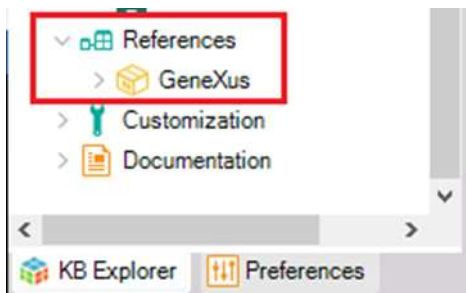
Luego, haciendo botón derecho / References, observe qué objetos están utilizando la función. Será el web component SessionsComponent, que a su vez es parte del objeto Sessions y además el objeto de Smart Devices WorkWithDevicesSession.

Ubique el componente web **SessionsComponent** dentro de la app del frontend en ejecución.

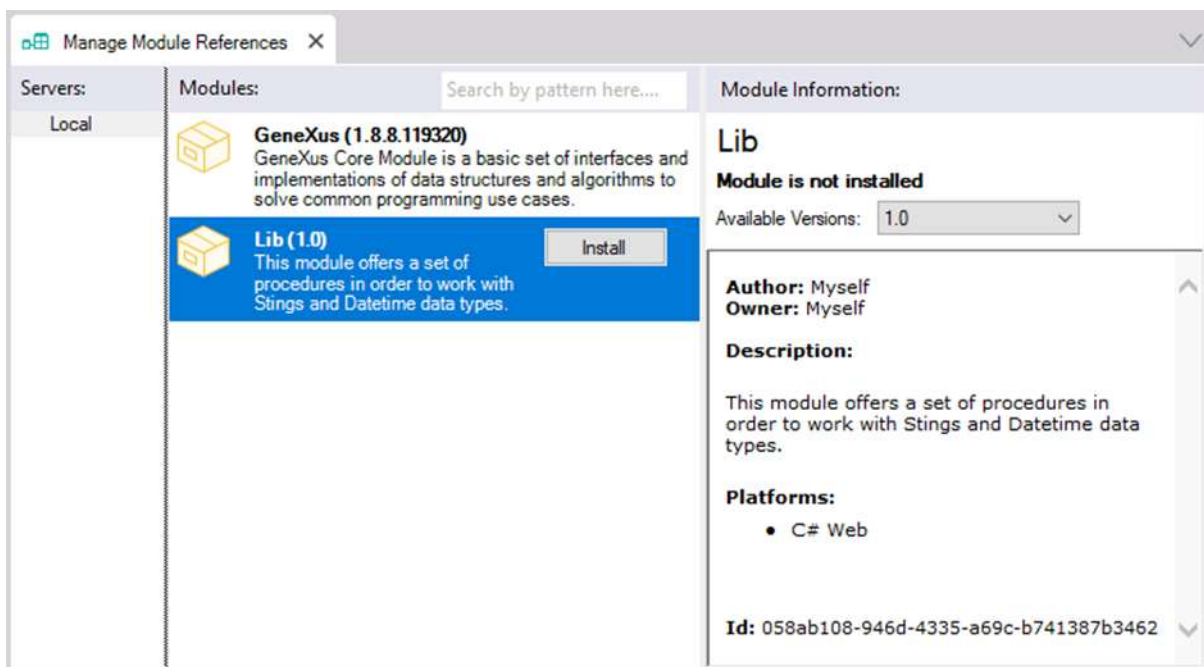
Abra en GeneXus **SessionsComponent** y encuentre entre los eventos la invocación al procedimiento TruncateString.



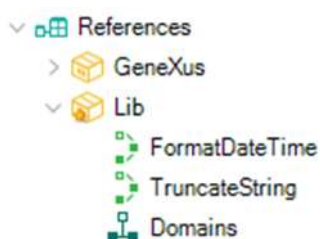
Antes de seguir, observe que bajo el nodo References solo se encuentra el módulo GeneXus:



Ahora importe el módulo que empaquetó hace un rato en la otra KB.



¿Dónde debe quedar ubicado en el KB Explorer?



Abra el procedimiento TruncateString de este módulo y observe que es read-only y que no aparece la sección Source, ni Layout, ni Conditions.

**Nota:** los binarios del módulo serán copiados al directorio del target environment en tiempo de build.



Recuerde que no habrá conflicto con las funciones del mismo nombre de esta KB, pues los objetos de los módulos tienen un qualified name.

Si ahora usted quiere pasar a utilizar las funciones de este módulo y no las que se definieron especialmente en esta KB, ¿cómo debe modificar las invocaciones?

Respuesta: en SessionsComponent cambie:

```
TextDesc.Caption = TruncateString(SessionDescription, 145)
```

Por:

```
TextDesc.Caption = Lib.TruncateString(SessionDescription, 145)
```

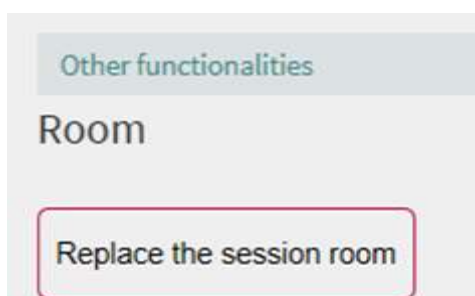
Ejecute (haciendo **Run** sobre el web panel **Sessions**). Puede cambiar la cantidad de caracteres para truncar el string.

**Nota:** Podría hacer lo mismo con el otro objeto que utilizaba la función TruncateString, el WorkWithDevicesSession, nodo List. Pero todavía no ejecutaremos la app SD, así que por el momento no lo haremos.

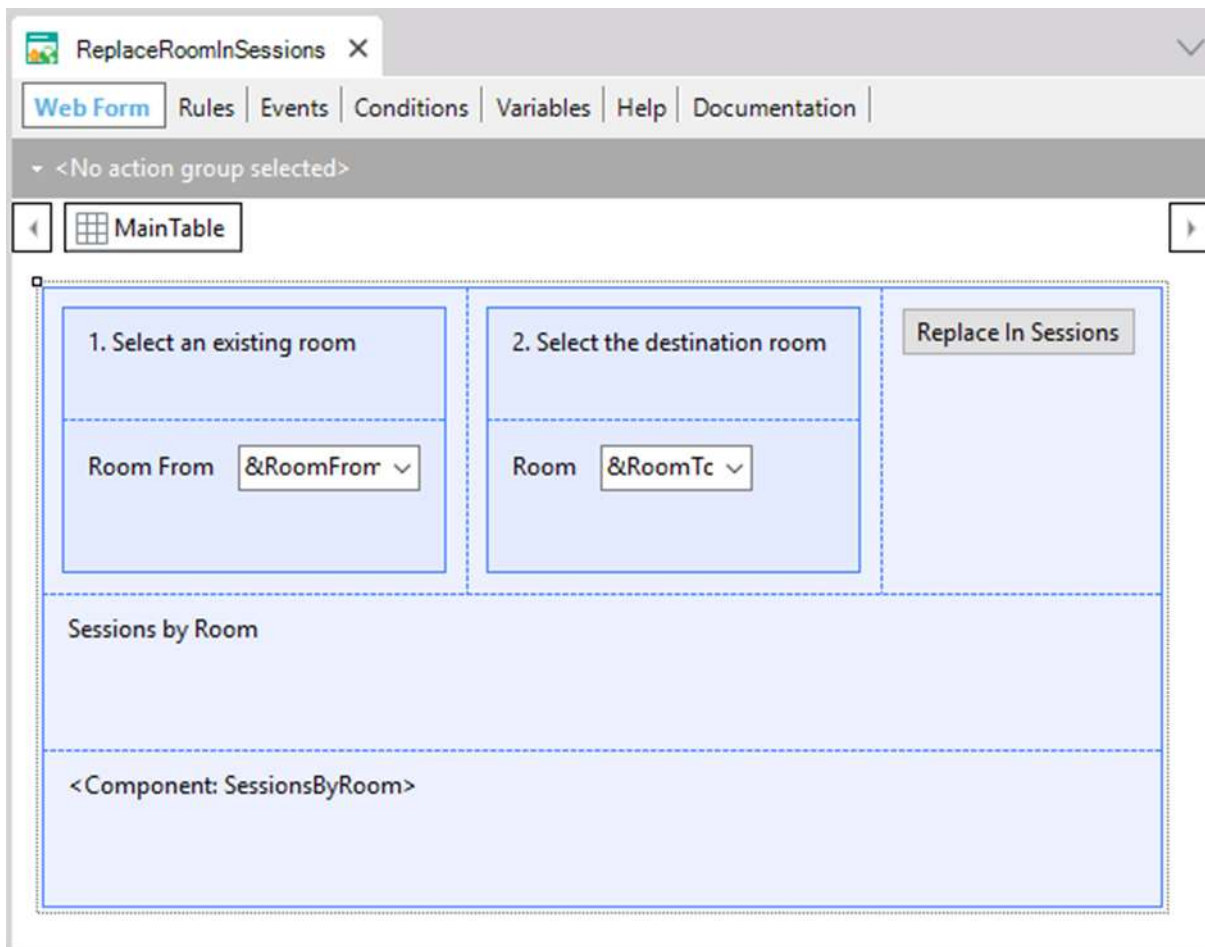
## BUSINESS COMPONENTS

Se desea implementar un web panel para el **backend** que permita modificar todas las conferencias (sessions) que iban a realizarse en una sala, de modo que se realicen en otra.

Observe que ya se ha creado con el nombre **ReplaceRoomInSessions**, y es invocado desde el web panel Home:

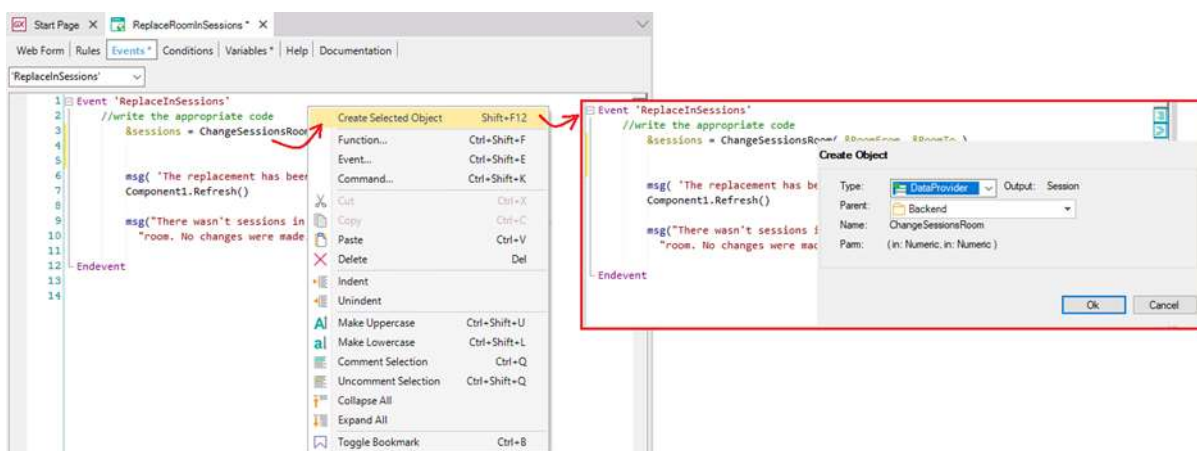


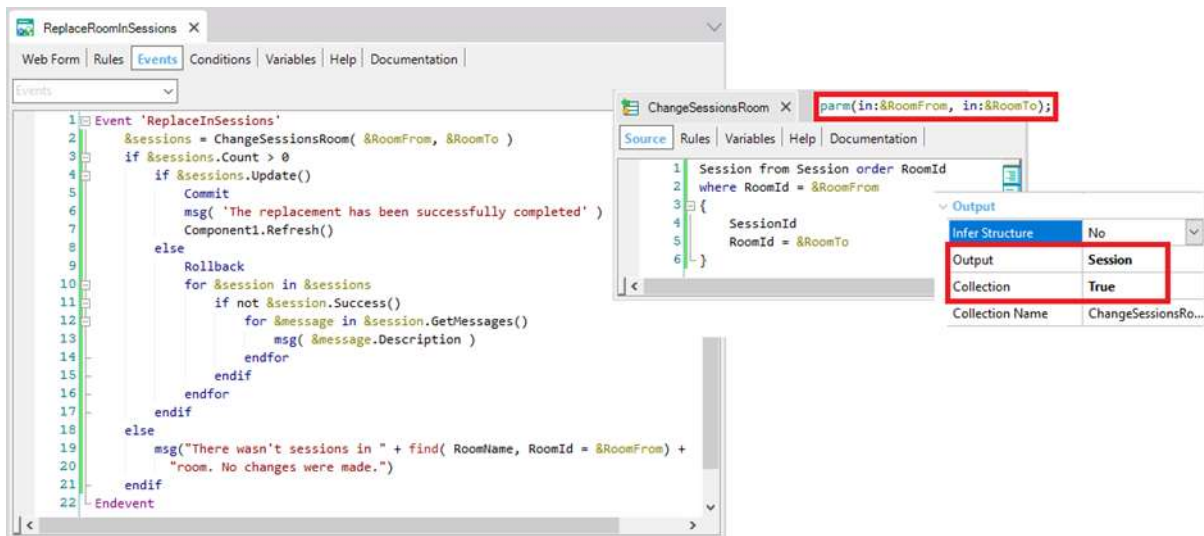
Se ha diseñado su Web Form con dos combos dinámicos para las variables &RoomFrom y &RoomTo, based on RoomId. Usted deberá programar el evento asociado al botón. Observe que además se ha agregado un componente que muestra todas las sessions por room de acuerdo a los datos actuales. Dentro del evento deberá refrescar ese componente.



Una posible solución:

Definir un Data Provider ChangeSessionsRoom en el folder Backend, que devuelva una colección de Business Component Session con RoomId cambiado por el nuevo. Pruebe no definirlo desde el principio, sino hacerlo desde el código del evento 'ReplacelnSessions':





### Event 'ReplacelnSessions'

```

&sessions = ChangeSessionsRoom( &RoomFrom, &RoomTo )
if &sessions.Count > 0
  if &sessions.Update()
    Commit
    msg( 'The replacement has been successfully completed' )
    Component1.Refresh()
  else
    Rollback
    for &session in &sessions
      if not &session.Success()
        for &message in &session.GetMessages()
          msg( &message.Description )
        endfor
      endif
    endfor
  endif
else
  msg("There wasn't sessions in " + find( RoomName, RoomId = &RoomFrom ) +
    "room. No changes were made.")
endif

```

### Endevent

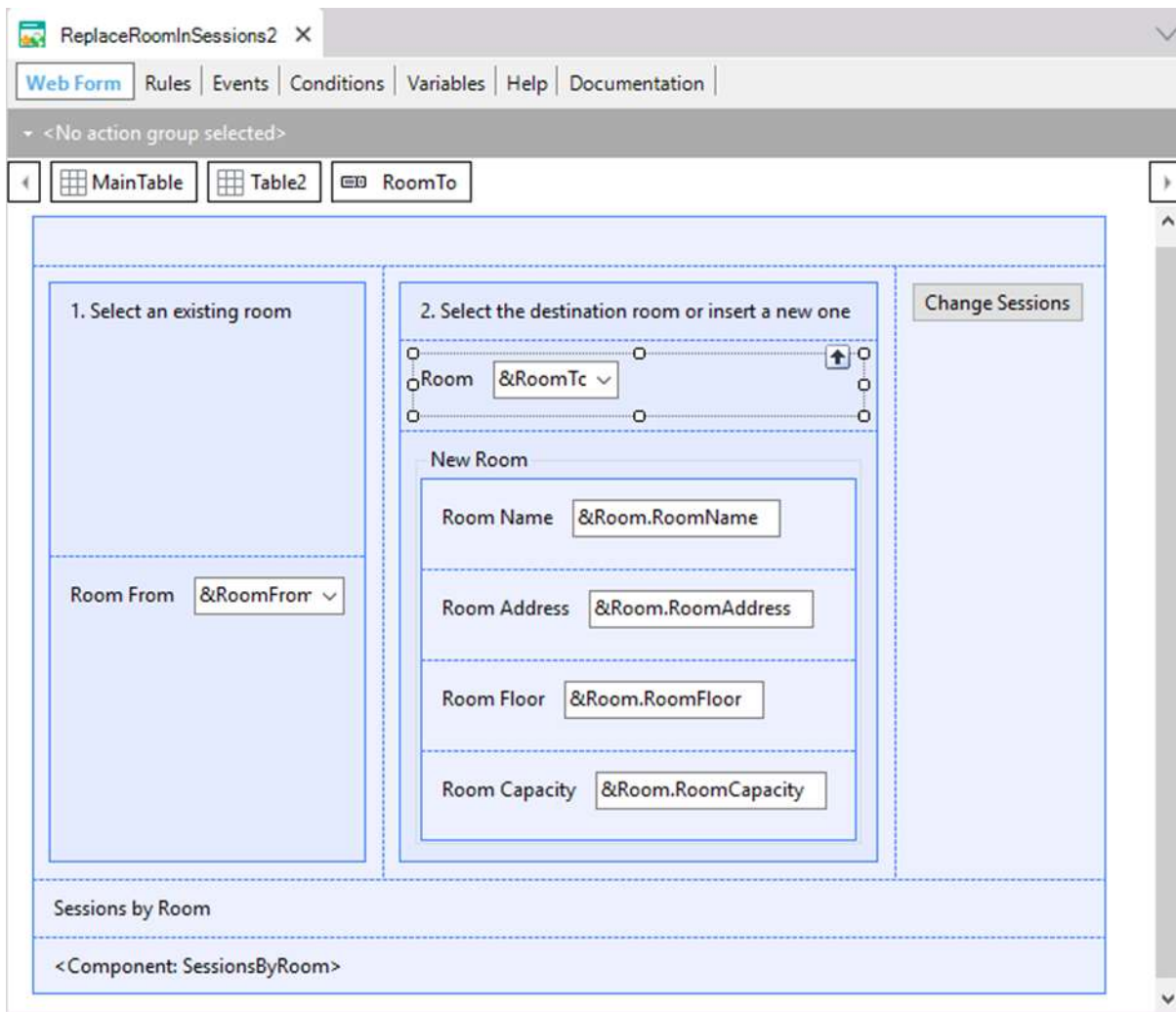
Observar que el Data Provider ChangeSessionsRoom solo devuelve cargados SessionId y RoomId. El resto de los campos vacíos, dado que utilizaremos el método Update, que únicamente necesita el Id (para internamente hacer un Load en variable auxiliar) y los campos a ser modificados. Los demás quedarán como estaban.

Pruebe su solución en ejecución. Hay más soluciones. Si usted pensó otra, discuta en qué se diferencia con la aquí presentada. ¿Utilizó el método Update o el Save? ¿Utilizó el método Load?

También puede brindar la alternativa de, si la nueva sala no estaba ingresada al sistema, ingresarla antes. Para ello se ha creado el web panel **ReplaceRoomInSessions2**.

En el Form se insertaron, además de los combos, una variable &Room de tipo de datos el Business Component Room.

Programa el código asociado al botón y desde Home invoque a este web panel en lugar de al anterior.



Observe que la transacción Room tiene regla de error para que no pueda quedar una sala sin nombre en el sistema. Testee en ejecución intentando dejar vacíos los campos &RoomTo y &Room.RoomName.

Pruébalo en ejecución.

Una posible solución:

```
Event 'ChangeSessions'
  &continue = True
  if &RoomTo.IsEmpty()
    &continue = &Room.Insert()
  if &continue
    &RoomTo = &Room.RoomId
```

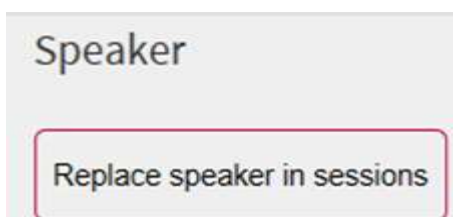
```

        &RoomFrom.Refresh()
    else
        msg( &Room.GetMessages().ToJson())
    endif
endif
If &continue
    &sessions = ChangeSessionsRoom( &RoomFrom, &RoomTo )
    if &sessions.Count > 0
        if &sessions.Update()
            Commit
            msg( 'The update has been successfully completed' )
            Component1.Refresh()
        else //some update for a session has failed
            for &session in &sessions
                if not &session.Success()
                    msg( &session.GetMessages().ToJson())
                endif
            endfor
            Rollback
        endif
    else
        msg("There wasn't sessions in " + find( RoomName, RoomId = &RoomFrom) + " room. No
changes were made")
    endif
endif
Endevent

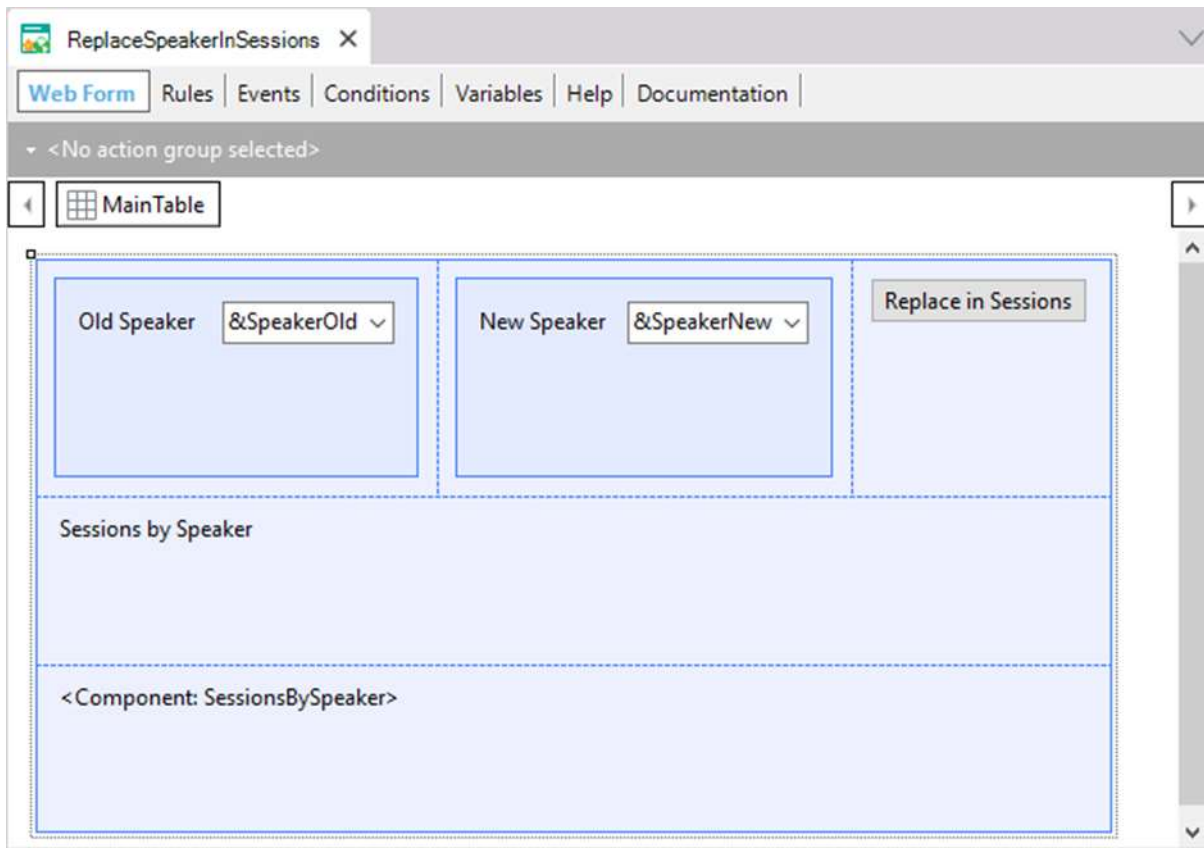
```

¿Y si ahora lo que queremos es sustituir a un orador por otro en todas las conferencias que el primero tuviera asignadas?

Observe que el Home permite invocar a un panel **ReplaceSpeakerInSessions**, para cambiar un orador.



Deberá, como antes, programar el evento asociado al botón.



Una posible solución:

Event 'Replace in Sessions'

```

&ok = True
for each Session.Speaker
  where SpeakerId = &SpeakerOld
  &Session.Load(SessionId)
  &Session.Speaker.RemoveByKey(&SpeakerOld)
  &SessionSpeaker = new()
  &SessionSpeaker.SpeakerId = &SpeakerNew
  &Session.Speaker.Add(&SessionSpeaker)
  If not &Session.Update()
    for &message in &Session.GetMessages()
      msg( &message.Description)
    endfor
    &Ok = False
  endif
when none
  msg( 'The Speaker to be replaced was not in any Session')
  &ok = False
endifor
if &ok
  msg( 'The replacement has been successfully completed' )
  commit
  Component1.Refresh()
else
  rollback
endif

```

## Endevent

**Observaciones:** Como SpeakerId es parte de la clave primaria del segundo nivel del Session, no puede cambiarse. Hay que eliminar ese speaker para la session e ingresar el nuevo. Observe que para eliminar líneas no tenemos más alternativa que utilizar el método Load para cargar la Session y luego el método de las líneas RemoveByKey. En este caso da lo mismo utilizar el método Save o el Update, porque el Update de todos modos hará un Save, y no creará una variable auxiliar, dado que ya sabe que la variable &session está en update, por el uso del método Load.

## POBLADO DE TRANSACCIONES

Observe que bajo el folder DataModel se encuentran las transacciones que se están utilizando. Si observa cualquiera de ellas, por ejemplo, Country, verá entre sus propiedades que las del grupo Data se modificaron preñdiendo la propiedad Data Provider:

Data	
Data Provider	True
Used to	Populate data
Update Policy	Updatable

Y que esto hizo que bajo el nodo Country en el KB Explorer apareciera un Data Provider:



Si lo abre verá que allí se especificaron los datos con los que se quiso inicializar la tabla asociada. Esto fue todo lo que tuvo que hacer el desarrollador.

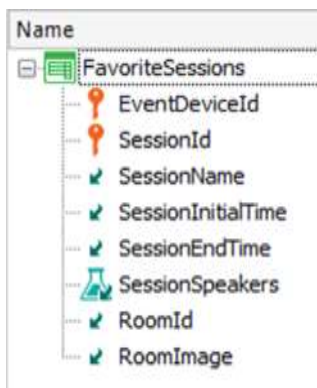
Gracias a estos Data Providers es que usted ya vio la app con datos desde el principio.



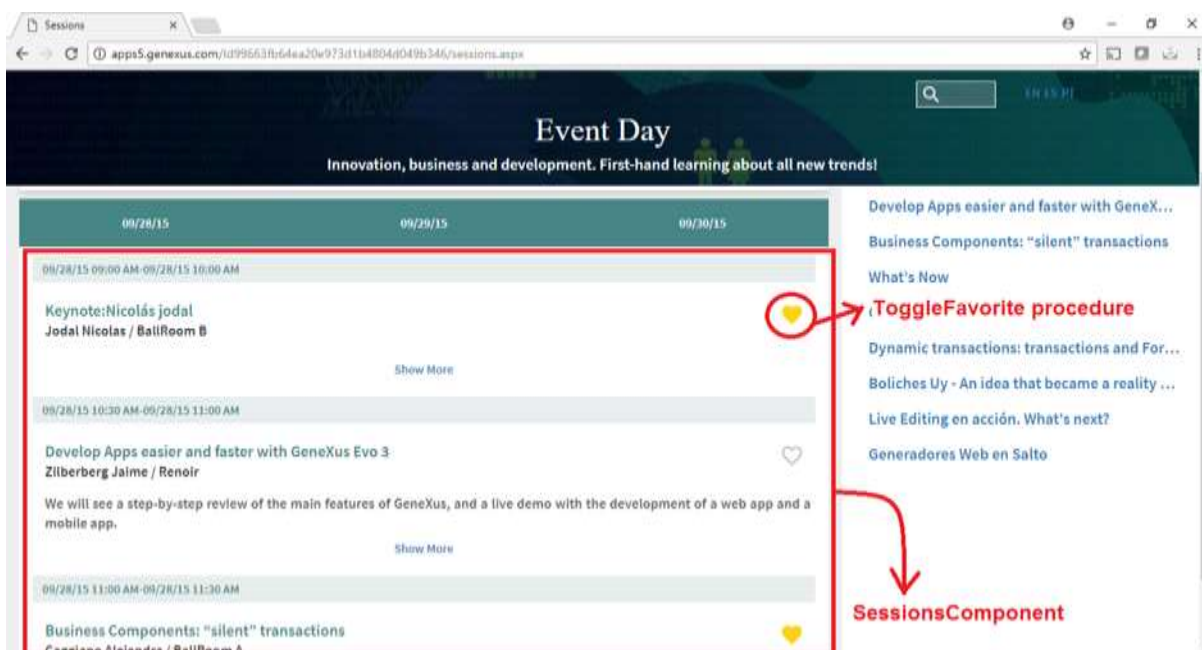
## TRANSACCIONES DINÁMICAS

La particularidad de las Transacciones Dinámicas es que no provocarán la creación de tablas físicas sino vistas para recuperar los datos en tiempo de ejecución a partir de lo que definamos en un Data Provider. Esto nos permite resolver muchos y muy variados escenarios de consultas.

Veámoslo en un ejemplo práctico. Tenemos la transacción FavoriteSessions que registra por dispositivo las conferencias marcadas como favoritas:

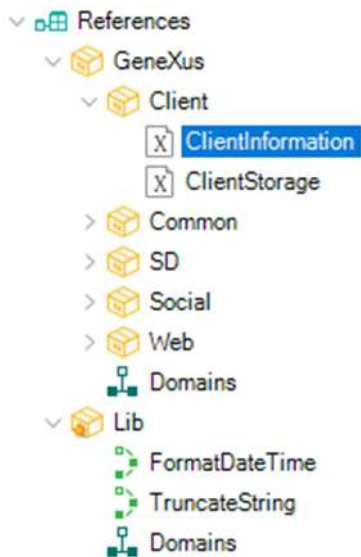


El usuario marca en la app del frontend las conferencias favoritas cuando desde las sessions hace clic sobre el ícono de corazón. Allí se invoca al procedimiento ToggleFavorite que es el que efectivamente inserta esa conferencia en la tabla de FavoriteSessions:



Ábralo y observe cómo lo hace. Observe que se utiliza el objeto externo ClientInformation (read only) del módulo GeneXus para obtener el dispositivo web (navegador) o SD que está ejecutando. Por más información: <https://wiki.genexus.com/commwiki/servlet/wiki?20198>



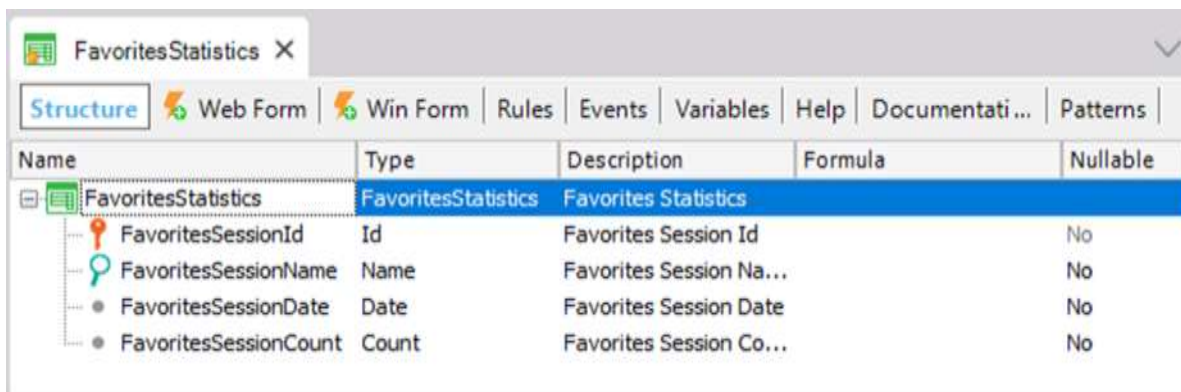


Bien, una vez comprendido esto, lo que ahora se desea es consultar el ranking de conferencias que más han sido seleccionadas como favoritas (desde los distintos dispositivos).

La información que necesitamos está en la tabla asociada a la transacción FavoriteSessions. Aquí queremos contar por cada session marcada como favorita, cuántos dispositivos la seleccionaron.

¿Por qué no tener una transacción dinámica que corresponda a esta información?

Para ello creamos la transacción **FavoritesStatistics**, que registrará las conferencias marcadas como favoritas, registrando cuántas veces cada una ha sido favorecida:

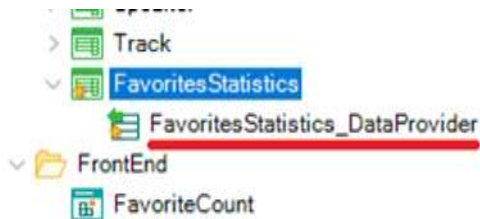


Para indicar que la transacción es dinámica, debemos configurar las siguientes propiedades:



Como no queremos actualizar información a través de la transacción, dejamos la propiedad Update Policy con su valor default para **Retrieve data** que es “Read Only”.

Al grabar el objeto, si lo observamos en el KB Explorer (podemos hacer botón derecho → Locate in KB Explorer, y mover la transacción para el folder DataModel si ya no la habíamos creado allí), vemos que se ha creado automáticamente un data provider asociado a la transacción dinámica:



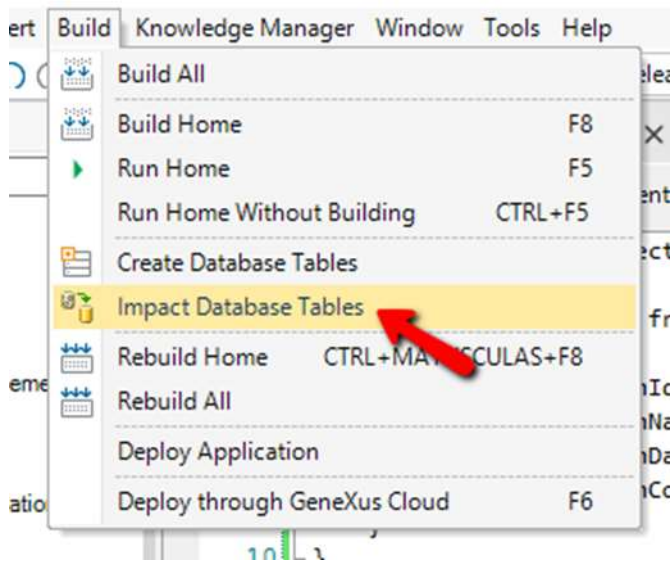
Modifiquemos el data provider para personalizar cómo deseamos cargar nuestra transacción dinámica. Descomentemos el código. Ahora, ¿cómo programa lo deseado?

Aquí una solución.

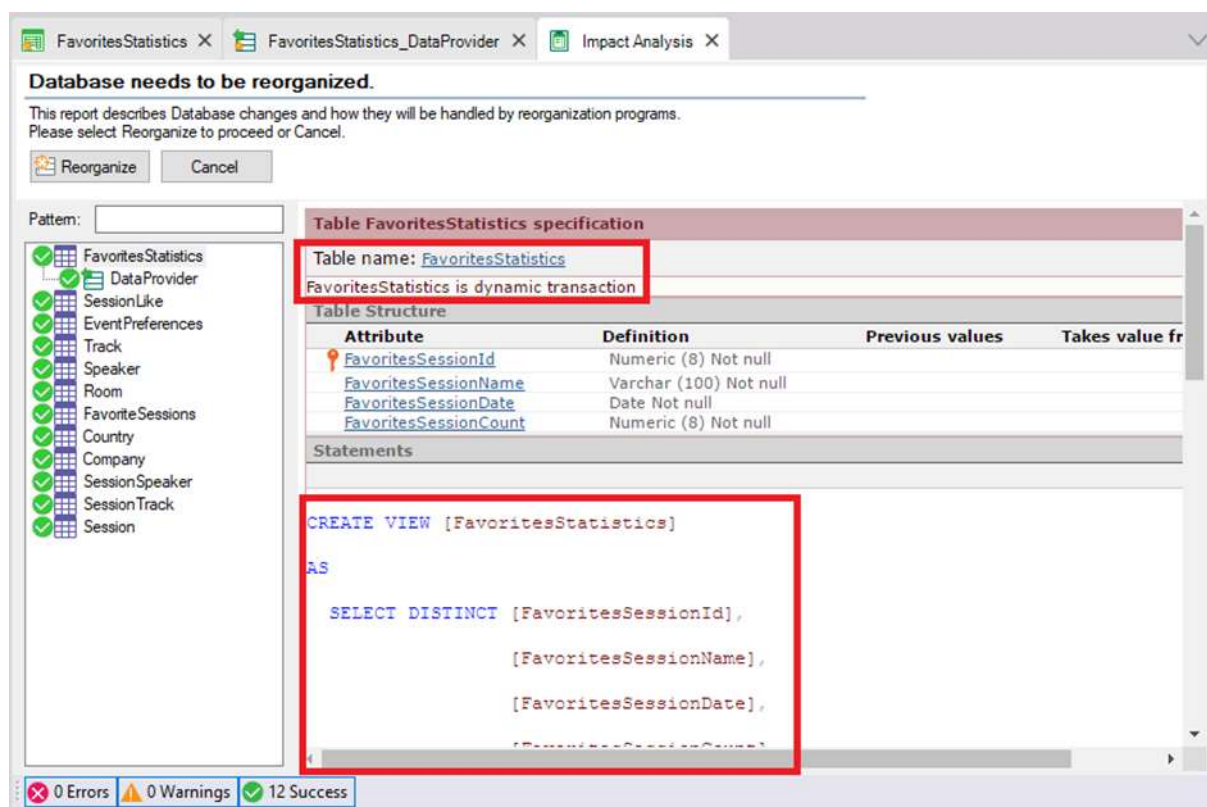
El Source quedaría:

```
FavoritesStatisticsCollection
{
    FavoritesStatistics from FavoriteSessions unique SessionId
    {
        FavoritesSessionId = SessionId
        FavoritesSessionName = SessionName
        FavoritesSessionDate = SessionInitialDate
        FavoritesSessionCount = count(EventDeviceID)
    }
}
```

Grabemos el data provider, y hagamos un impacto (Build → Impact Database Tables):



Observemos el diagrama de impacto, donde se indica que la transacción es dinámica, y la sentencia para crear la vista en el SQL Server en vez de crear una tabla:

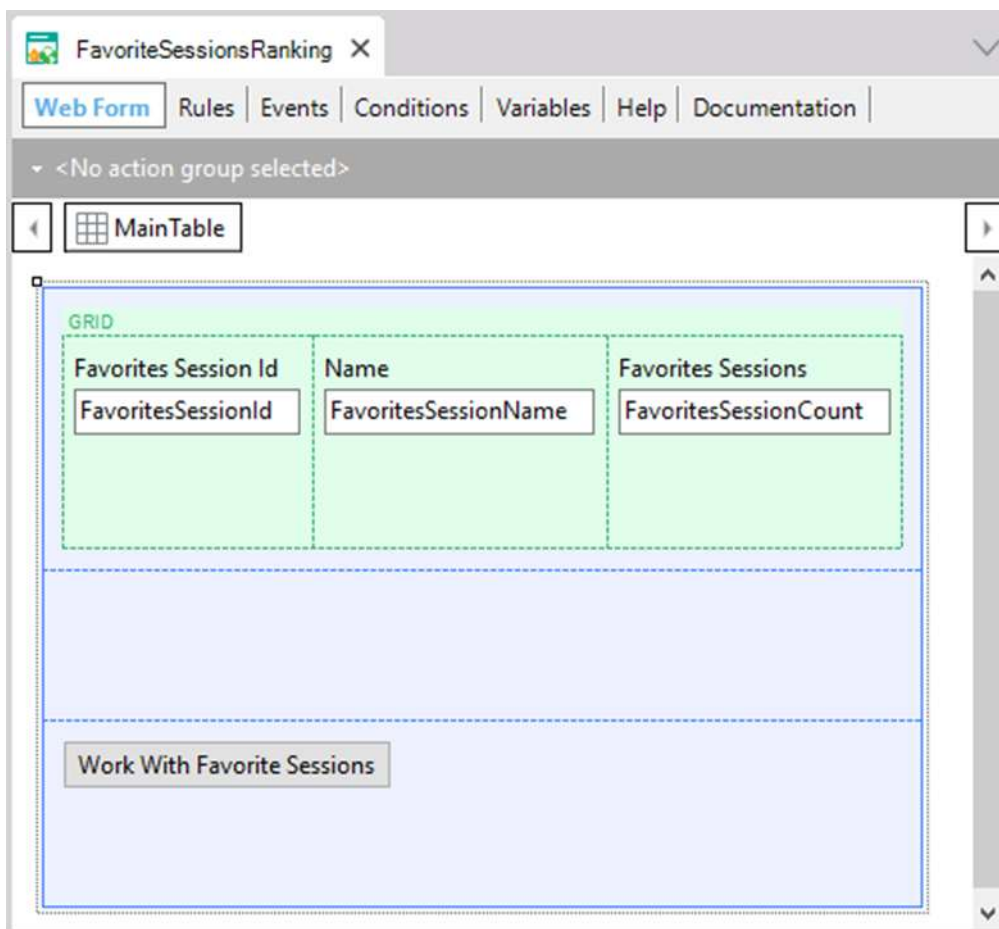


**Nota:** en el upgrade en el que se probó este práctico y se capturó la pantalla están apareciendo las demás tablas porque informa que debe ejecutar los data providers para poblar las tablas. No debería hacerlo, pues esas tablas no se han tocado, pero internamente GeneXus entiende que sí.

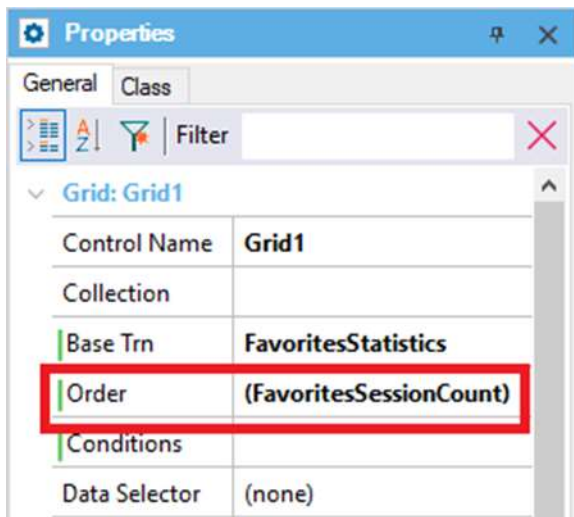
Haga clic en "Reorganize" para que la vista sea creada en la base de datos.

## WEB – USANDO LAS TRANSACCIONES DINÁMICAS

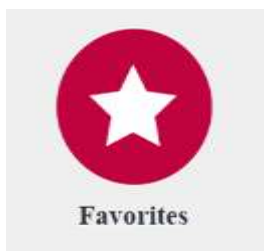
Ahora, para completar el requerimiento pedido –consultar el ranking de conferencias más *favoriteadas*–, simplemente utilizaremos los atributos de la transacción como lo solemos hacer. Crearemos en el folder Backend un web panel **FavoriteSessionsRanking**, con un grid que contenga los atributos de nuestra transacción dinámica:



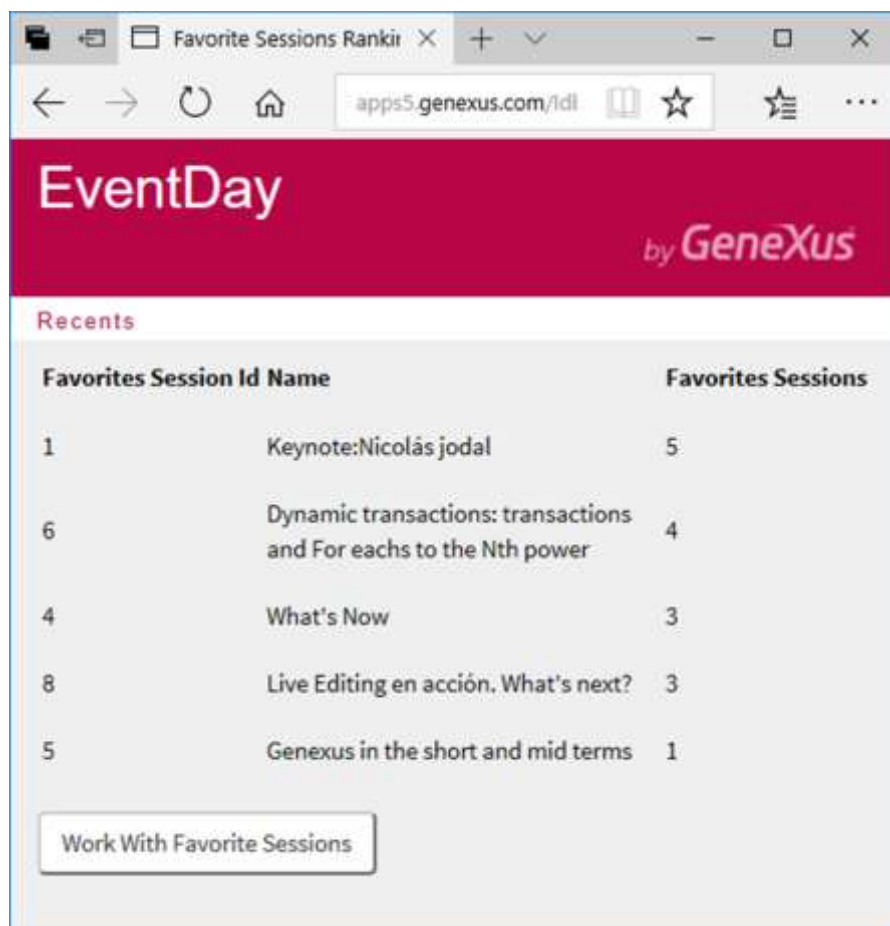
Asignaremos el valor “(FavoritesSessionCount)” a la propiedad **Order** del grid, para indicar que queremos ver primero las conferencias con más votos:



Desde el botón “Work With Favorite Sessions” invocamos al `WWFavoriteSessions()` que muestra los favoritos de cada dispositivo. Y desde el **Home** del backend cambiamos la invocación de Favorites, para que ahora invoque al web panel que acabamos de crear, `FavoriteSessionsRanking`, en lugar de invocar al Work with.



Ejecutemos el Home y accedamos a Favorites:



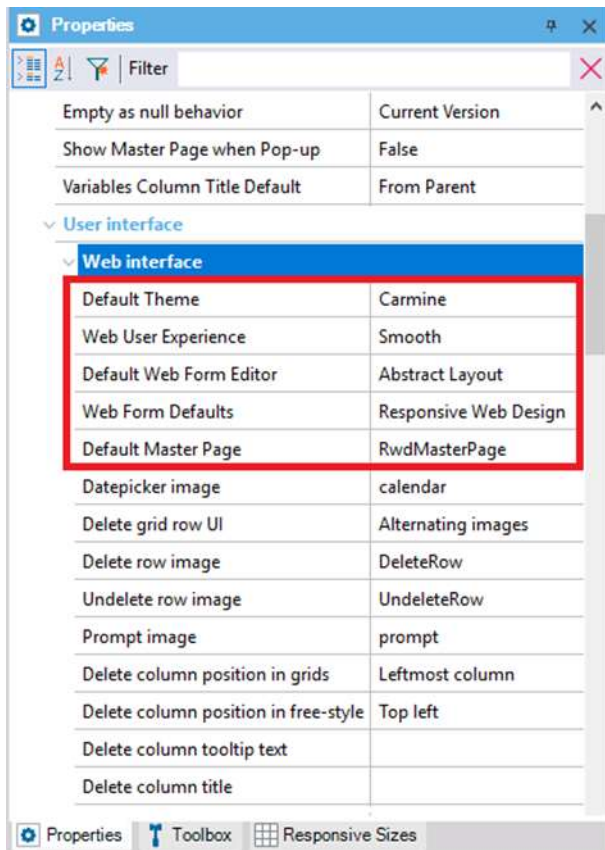
Si va a realizar la parte de Smart Devices más adelante:

En el Panel for Smart Devices **PopularSessions** se utiliza de manera similar la transacción dinámica para mostrar las sessions más favoriteadas. Ábralo y descomente el código del Load, pues ahora ya existen los atributos FavoritesSessionId y FavoritesSessionName. Más adelante probará este panel en ejecución.

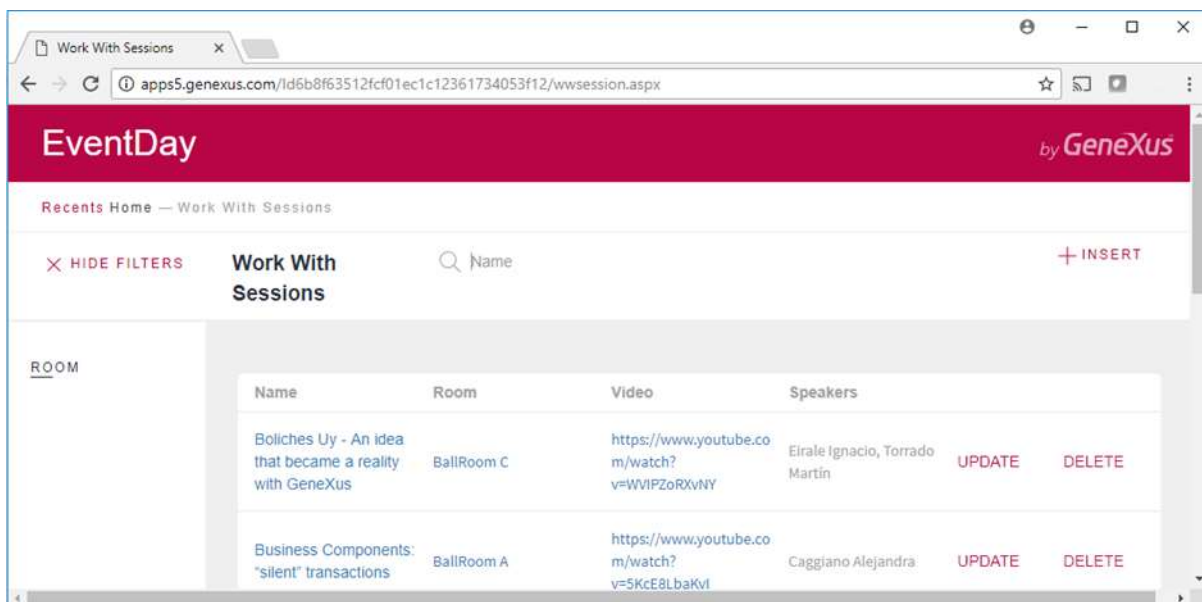
## WEB - CARMINE Y NUEVO DISEÑO DEL WORK WITH FOR WEB

**Carmine** es el nuevo theme utilizado por defecto en las KBs creadas con la versión 15 de GeneXus para aplicaciones **Responsive (RWD)**. Con el uso de Carmine, además de utilizar nuevos colores en tonos de rojos, se cuenta con un **nuevo y más moderno diseño** de los objetos generados por el **pattern Work With for Web**.

Observaremos estas novedades en ejecución. Pero antes acceda a las propiedades de la versión (en Preferences) y observe bajo el grupo User interface / Web interface los defaults:

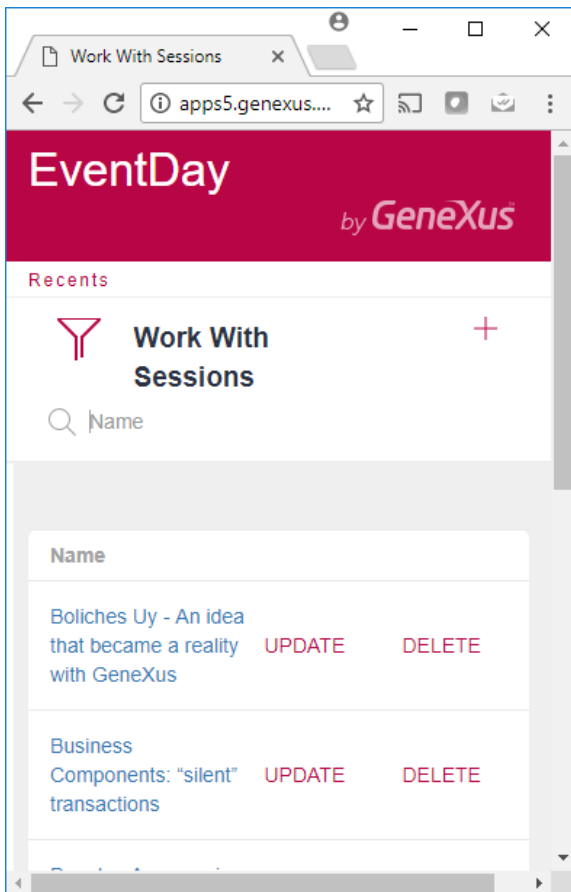


Ejecute el Home de la aplicación del backend (que como verá tiene configurado como theme Carmine), opción “Sessions”.




Pruebe achicar el tamaño de la ventana del browser.



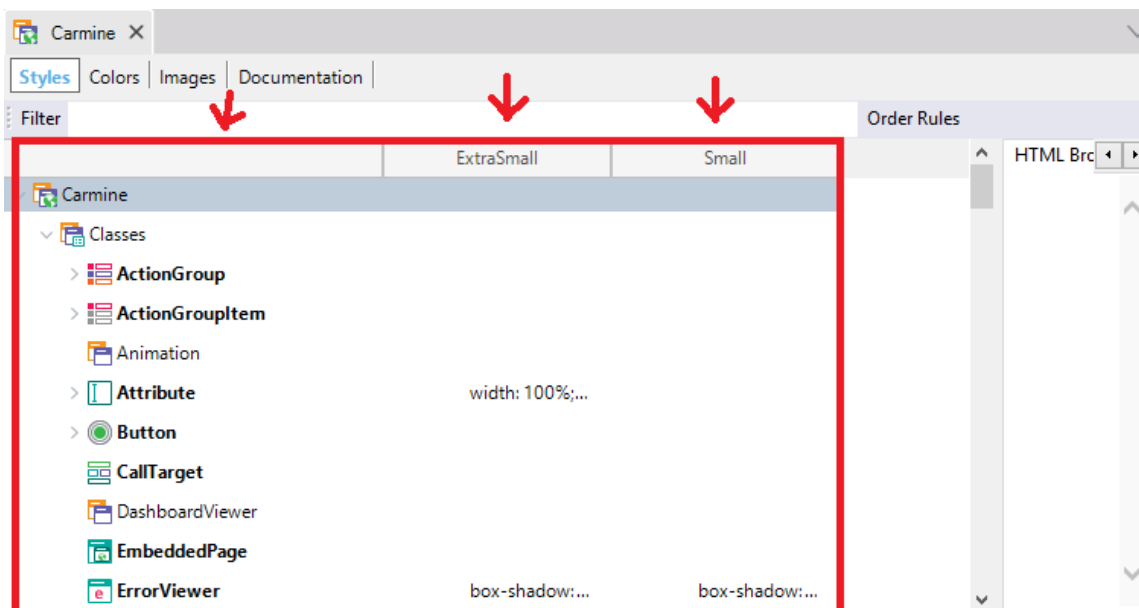


Observar cómo automáticamente se van ‘eliminando’ columnas a medida que se va haciendo menos ancha la ventana. Y vuelven a aparecer a medida que se agranda.

Observar cómo se genera un nuevo botón para acceder a los filtros 

Responsive sizes:

El Carmine por defecto viene sólo configurado para tres tamaños (no considera el Medium):



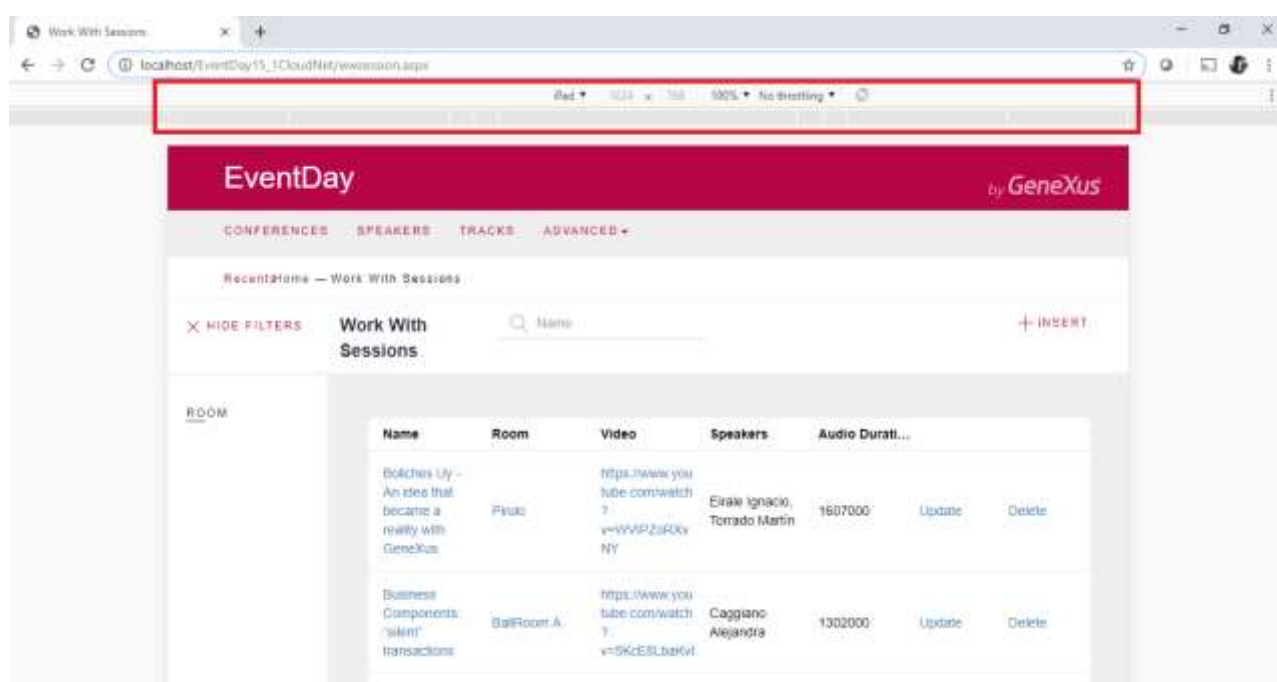


Por tanto si quisiéramos considerar el Medium, es necesario agregar la regla y configurar los tamaños respectivos. Hágalo en el theme. Obsérvelo. Así quedarán las reglas:

	Min Width	Max Width
Extra Small		767
Small	768	991
Medium	992	1199
(Large queda automático)	1200 & up	

Con estos tamaños, si queremos ver rápidamente en el browser los diferentes tipos de pantalla podemos usar la 'Device Toolbar' de chrome (o equivalente en otros browsers) y elegir Iphone 5 (Extra Small), Ipad Portrait (Small), Ipad Landscape (Medium) y volviendo al desktop normal (Large)

Ejecute el Work With Sessions y observe que en tamaño Medium están apareciendo las columnas relativas a los atributos Name, Room, Video, Speakers y Audio Duration.

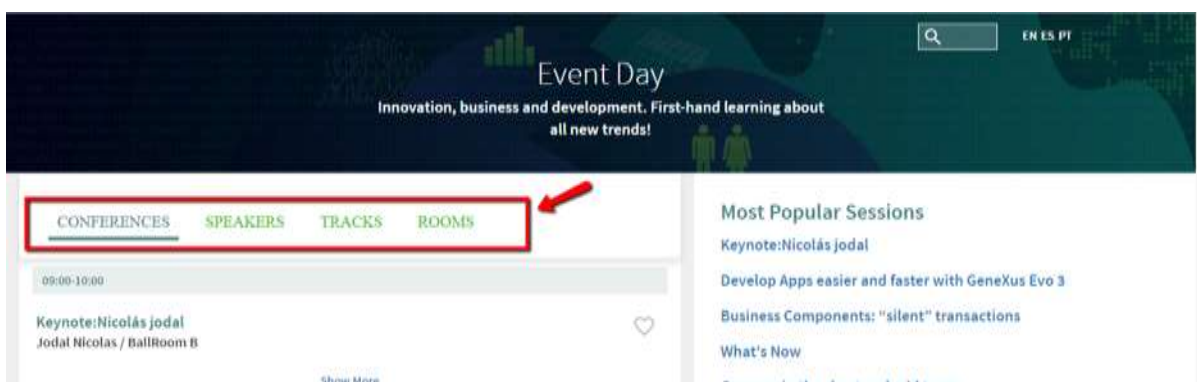


Si no queremos que aparezca Room Name para ese tamaño, ¿cómo lo resuelve?

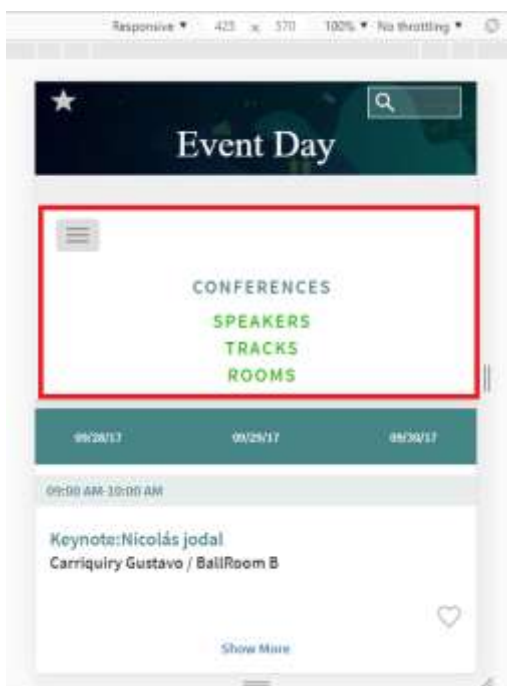
Observe qué valor asume la clase `WWSecondaryColumn` para el tamaño `Medium` que acaba de crear, en cuanto a la propiedad `Display`.

## WEB - NAVIGATION BAR

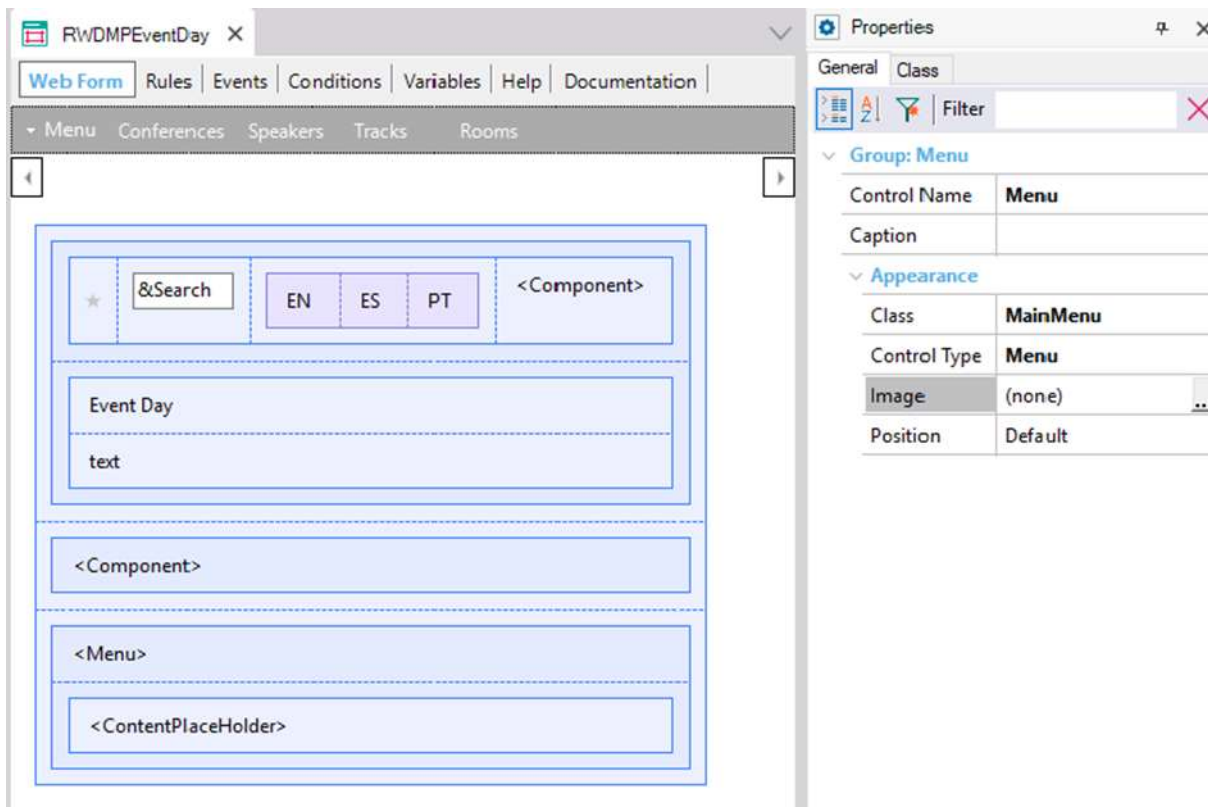
En la versión 15, los **Action Groups** permiten crear fácilmente una “navigation bar” en Web. Para verlo en ejecución, ejecute el web panel **Sessions** (botón derecho sobre él → Run).



Observe también cómo al reducir el tamaño de pantalla, se muestra como el clásico menú hamburguesa:



Abra la master page de este objeto (RWDMPEventDay). Y observe el menú:



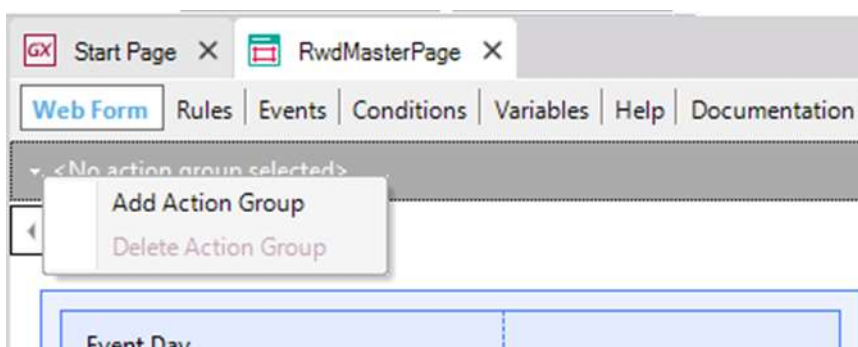
Observe en el evento Start cómo se programaron los links de cada opción.

Veamos ahora cómo crear un menú como ese, pero para la página Home del Backend, que permitirá invocar al Work With de Sessions, Speakers y Tracks.

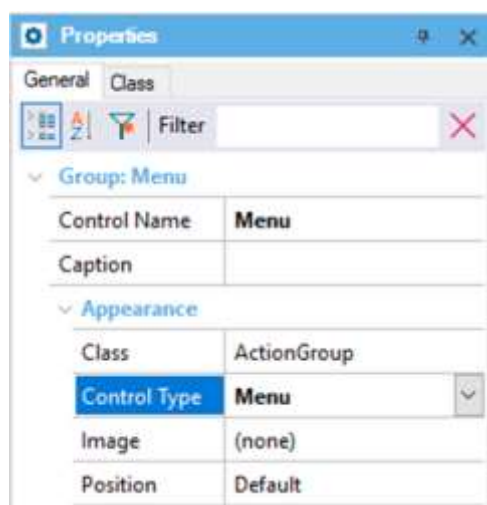
Abrir la master page del backend (**RwdMasterPage**). Nosotros ya hemos creado un menú (dinámico) que le mostraremos luego, pero como podrá observar en el form, no lo estamos insertando, por lo que no se está utilizando (observe que no aparece en ejecución).

Cree usted uno nuevo, que será estático.

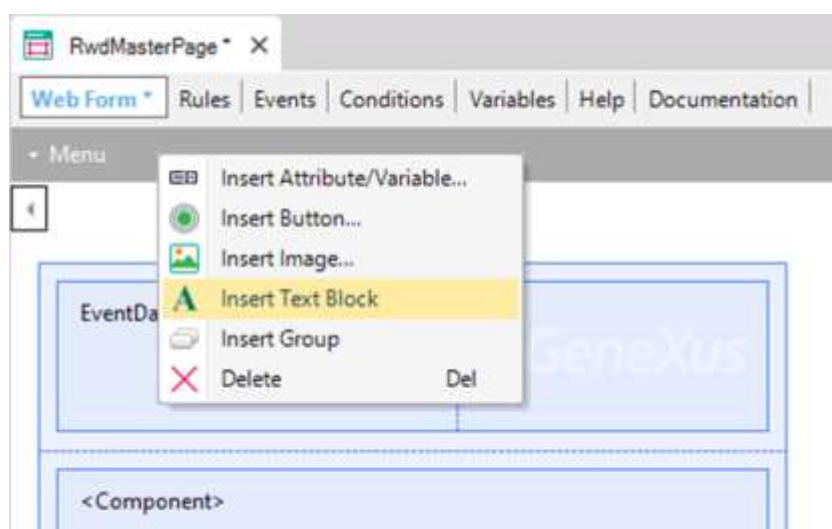
Hacer clic en la barra gris, en la flechita a la izquierda y seleccionar "Add Action Group":



Configuremos las propiedades Control Name = Menu, Caption = <vacío> y Control Type = Menu:



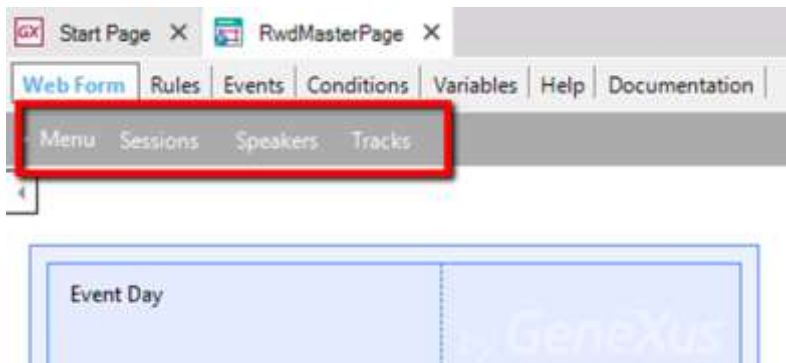
Insertemos un Textblock en el action group, haciendo botón derecho:



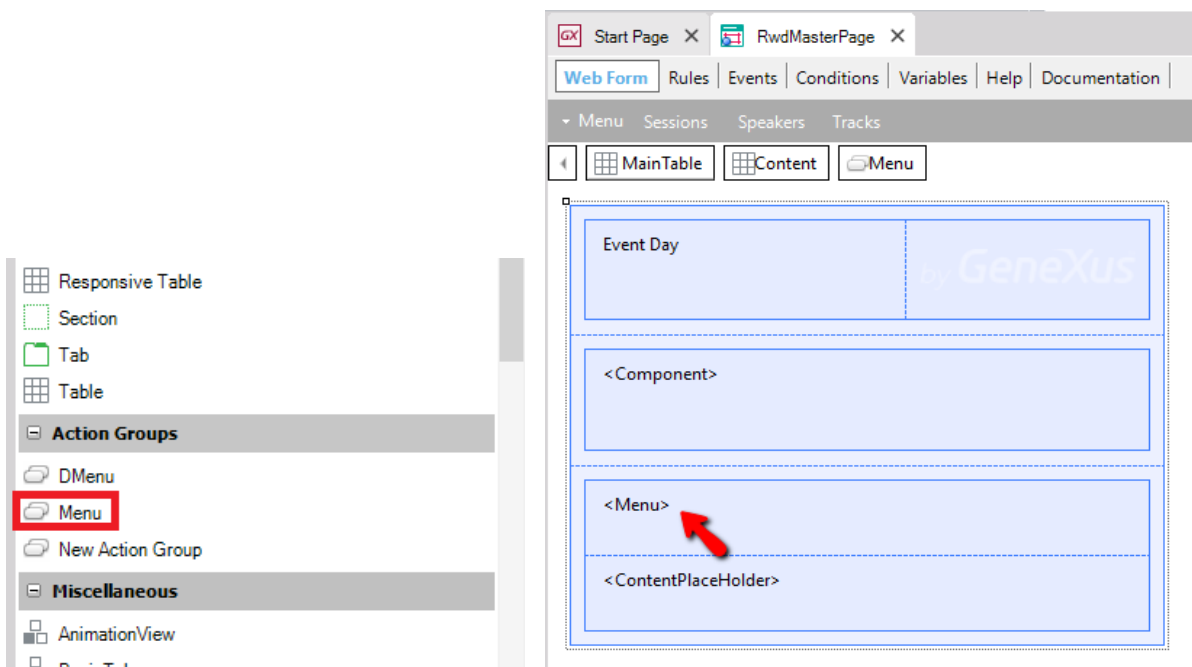
Configuremos las siguientes propiedades del Textblock:



Repitamos la operación para los Textblocks Speakers y Tracks, de forma que el action group quede de la siguiente forma:



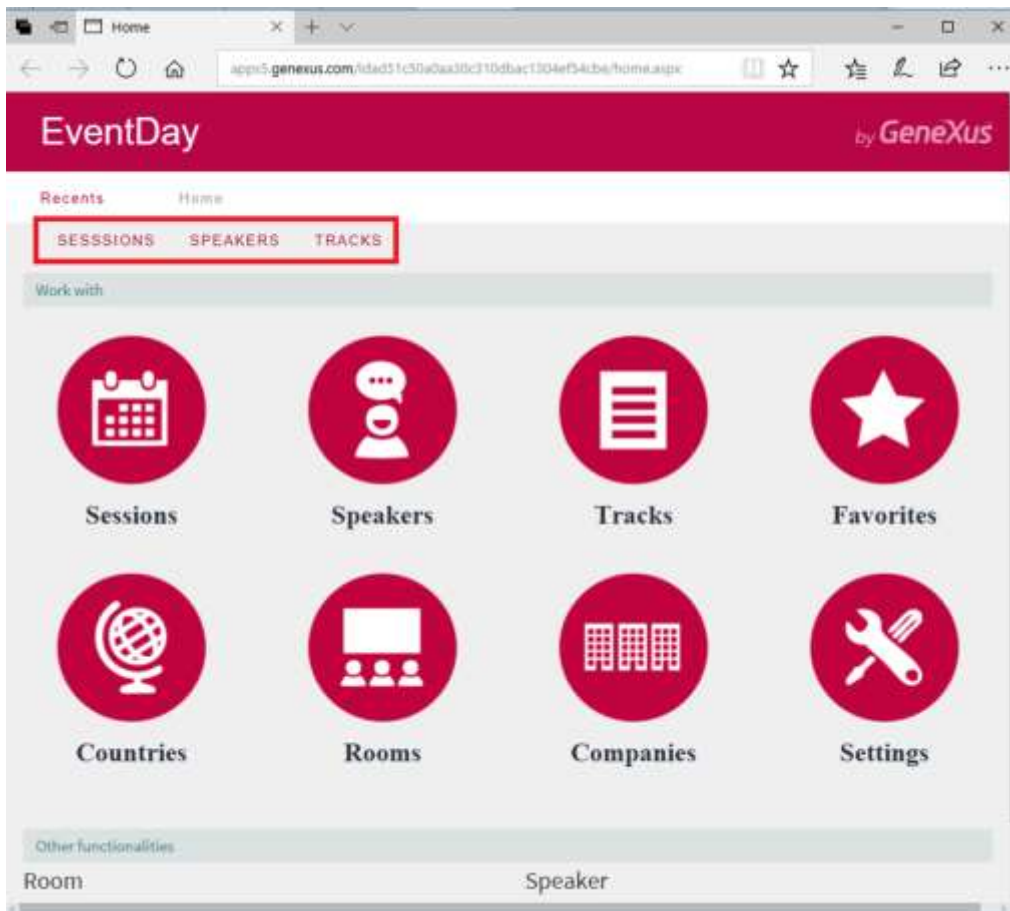
Grabemos el objeto, y hagamos clic en el form, en, por ejemplo, el <ContentPlaceHolder>. Luego desde la Toolbox (donde vemos que aparece ahora un grupo **Action Groups**), arrastremos el action group **Menu** a nuestro form, ubicándolo en la misma tabla que el <ContentPlaceHolder>:



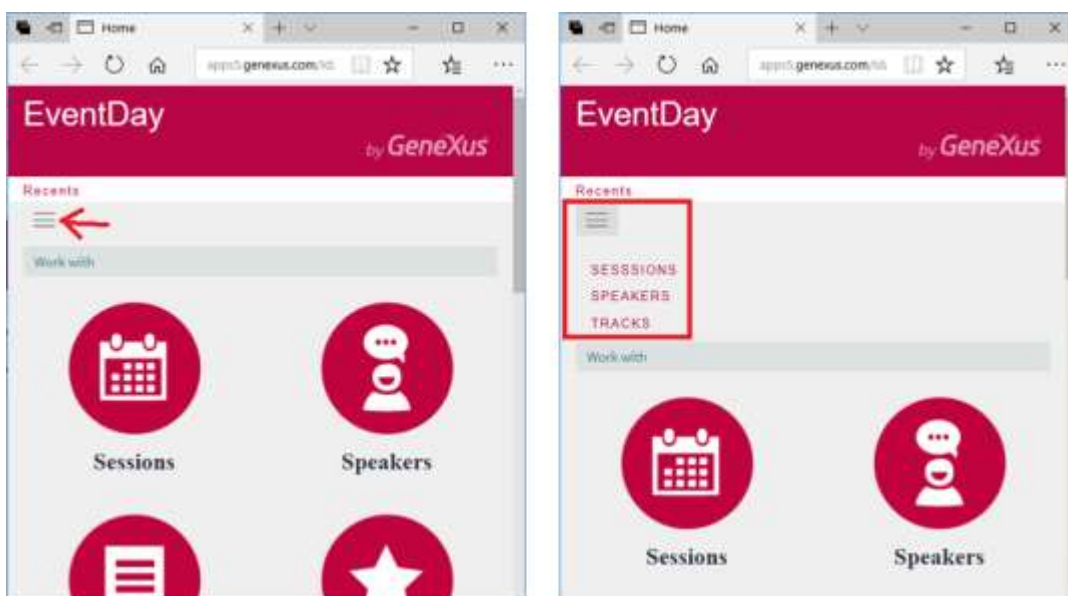
En el evento Start de la master page, agregaremos los links de cada text block (puede hacerlo dentro de la subrutina 'Load Menu'):

```
Sessions.Link = WWSession.Link()
Speakers.Link = WWSpeaker.Link()
Tracks.Link = WWTrack.Link()
```

Ejecutemos el web panel **Home** y observemos cómo aparece una barra de navegación con los ítems que agregamos:



Cambiamos las dimensiones del browser, y observemos cómo cuando la pantalla del dispositivo es pequeña, la barra de navegación aparece colapsada, y cómo al hacer clic sobre ella aparecen las opciones en forma vertical:



Como el menú se configuró a nivel de la Master Page, entonces lo verá en todas las páginas del backend, que tienen a ésta como su Master Page. Obsérvelo invocando a Sessions. Verá que aparece el mismo menú.

Observe que los textos están en mayúscula. Si ahora quiere que aparezcan en minúscula, ¿cómo lo consigue?

Observe qué sucede si cambia la clase del menú, pasándola de ActionGroup a MainMenu:



Y que al seleccionar una de los ítems aparece con esa barra rosada debajo:



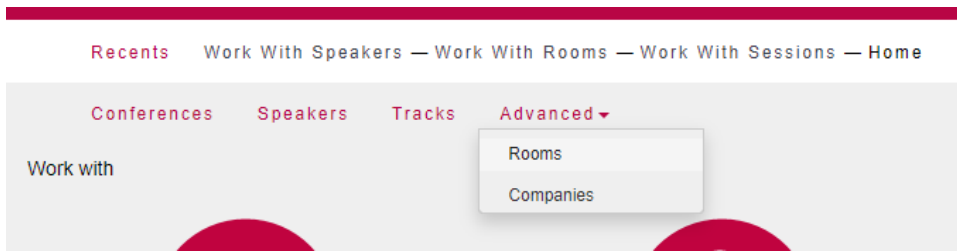
Busque en el theme dónde está esto configurado.

## OPCIONAL: MENÚ DINÁMICO

Pruebe ahora eliminar del form el Action Group Menu que ha creado (y comente en los eventos las asignaciones de links a sus ítems) e inserte, en su lugar, el de nombre DMenu, que ya venía creado pero no insertado en el form. Se trata de un menú dinámico.

Descomente el código de la subrutina 'Load Menu' que es el que carga las opciones del menú (observe el Data Provider BackendMenu utilizado para aportar la información de los ítems). Ejecute.





## WEB - CONTROL HORIZONTAL GRID

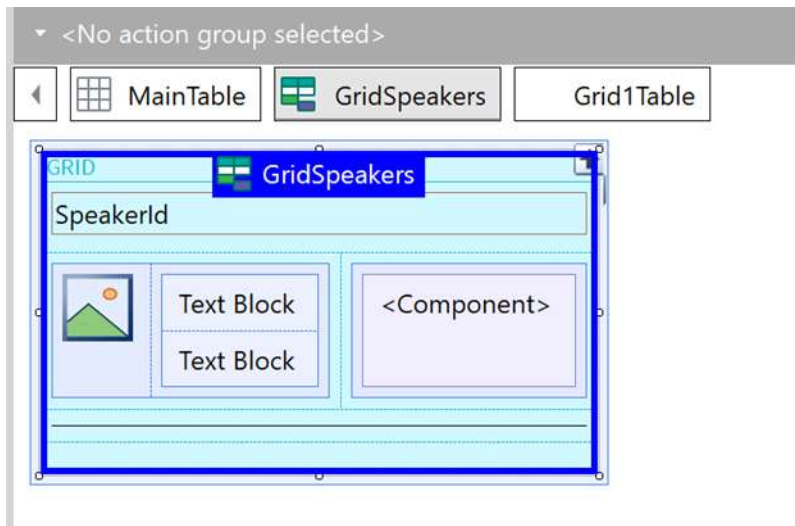
El control Horizontal Grid nos permite visualizar los elementos de una lista de forma horizontal en vez del clásico grid con filas.

Actualmente los Speakers se visualizan en una grilla vertical, pero queremos lograr que se visualicen de la siguiente manera:

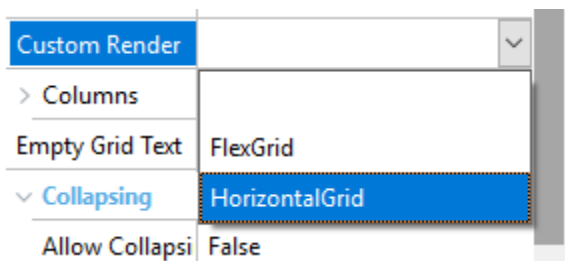


Para ello abra el web panel `SpeakersPanel`, y seleccione el grid `GridSpeakers`:

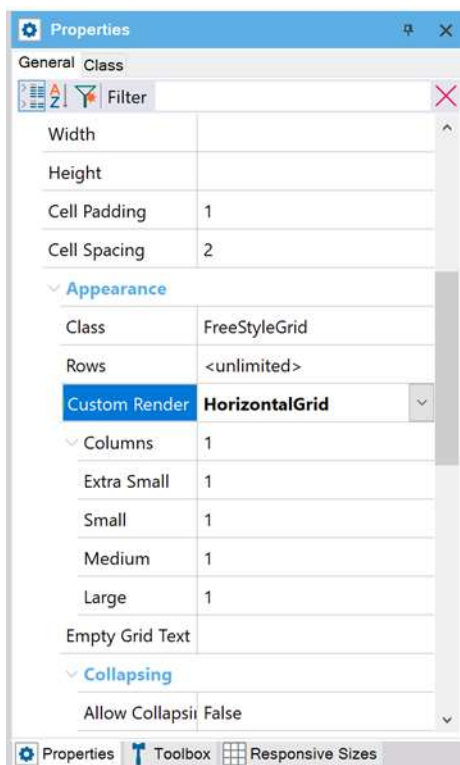




Cambie el valor de la propiedad “Custom Render” a “HorizontalGrid”:



Nota: observe que con GeneXus 16 aparece la nueva opción FlexGrid para el render de un grid.



Grabe el objeto, y vuelva a ejecutar el web panel **Sessions**

Veremos que en ejecución ahora la lista de oradores es mostrada de forma horizontal, y vamos cambiando la información desplegada utilizando el controlador de página, ubicado en la parte inferior.

Puede probar cambiar el color del controlador, pasándolo a color Teal, por ejemplo, y también cambiar el color de las flechas de navegación (propiedad "Page Controller Class") Pruébalo en ejecución:



## WEB - INFINITE SCROLLING

Cuando se tienen varios registros en un grid, una opción a la paginación es el "Infinite Scrolling". Esto es que a medida que el usuario va haciendo scroll van apareciendo nuevos registros en la grilla.

Veámoslo en el objeto **WWSpeaker** (backend). En ejecución actualmente se están viendo 10 filas por página para la grilla:








Image	Full Name	Company	Country	Actions
	Caggiano Alejandra	GeneXus	Uruguay	UPDATE DELETE
	Muriato Claudia	GeneXus	Uruguay	UPDATE DELETE
	Shuster Claudia	GeneXus	Uruguay	UPDATE DELETE
	Azzato Federico	GeneXus	Uruguay	UPDATE DELETE
	Milano Gastón	GeneXus	Uruguay	UPDATE DELETE
	Listón Graciela	GeneXus	Uruguay	UPDATE DELETE
	Carrquiry Gustavo	GeneXus	Uruguay	UPDATE DELETE
	Eirale Ignacio	GeneXus	Uruguay	UPDATE DELETE

Abrirlo y observar que el valor de la propiedad **Rows** es 10. La bajamos a 5. Gracias a que su valor no es 0, se muestra la propiedad **Paging** con el valor “One page at a time”. La cambiamos a:



The screenshot shows the GeneXus IDE interface. On the left, a grid control is visible with columns: Id, SpeakerId, Full Name, SpeakerFullName, Language, SpeakerLanguage, Image, Country Id, CountryId, Country Name, CountryName, Company Id, CompanyId, and Cor. A red arrow points to the grid control. On the right, the Properties window is open, showing the configuration for the 'Grid' control. The 'Rows' property is set to 5, and the 'PagingMode' is set to 'Infinite scrolling'. A red box highlights these two properties.

Observar en ejecución que aparece una barra de scroll vertical en la grilla de los oradores, y que a medida que se va haciendo scroll hacia abajo, van apareciendo más registros en la grilla:

Work With Speakers  + INSERT

Full Name	Image	Country Name	Company Name		
Carrquiry Gustavo		Uruguay	GENEXUS	UPDATE	DELETE
Eirale Ignacio		Uruguay	GENEXUS	UPDATE	DELETE
Zerberberg Jaime		Uruguay	GENEXUS	UPDATE	DELETE
Torrado Martin		Uruguay	GENEXUS	UPDATE	DELETE
Bartram Ursula		Uruguay	GENEXUS	UPDATE	DELETE
Keymetlian Silvia		Uruguay	GENEXUS	UPDATE	DELETE
Bengio Alejandro		Uruguay	GENEXUS	UPDATE	DELETE

Observe que abajo, antes de scrollear en el grid, aparece qué página está actualmente cargada:

	Carrquiry Gustavo	GeneXus	Uruguay	Update	Delete
	Eirale Ignacio	GeneXus	Uruguay	Update	Delete

PAGE 1 OF 2

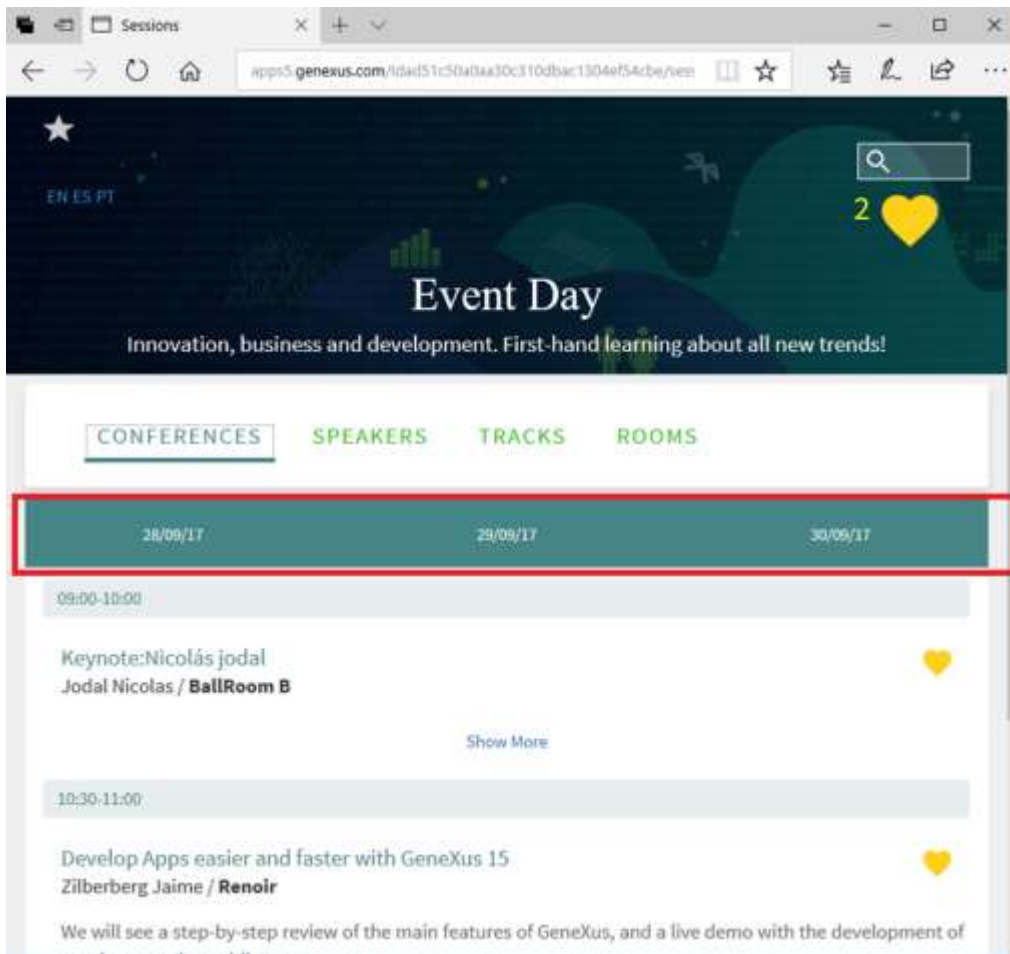
Observe qué pasa el hacer scroll.

¿Y si queremos que la barra de scroll no aparezca en el grilla sino en el form?

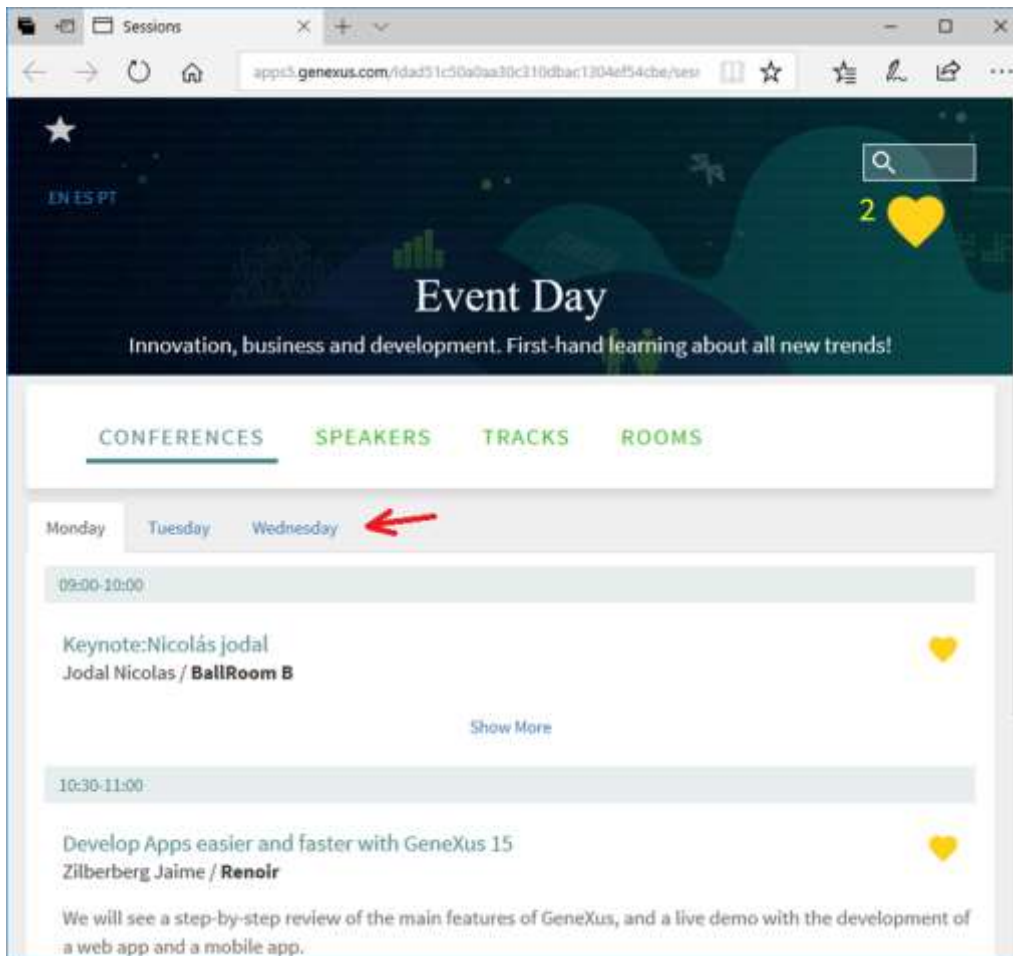
## WEB - TAB CONTROL Y COMPONENTES

Veremos a continuación un nuevo control para aplicaciones Web, que permite desplegar un conjunto de controles agrupados en diferentes tabs.

Actualmente estamos mostrando las conferencias por día, con una grilla que carga los diferentes días:

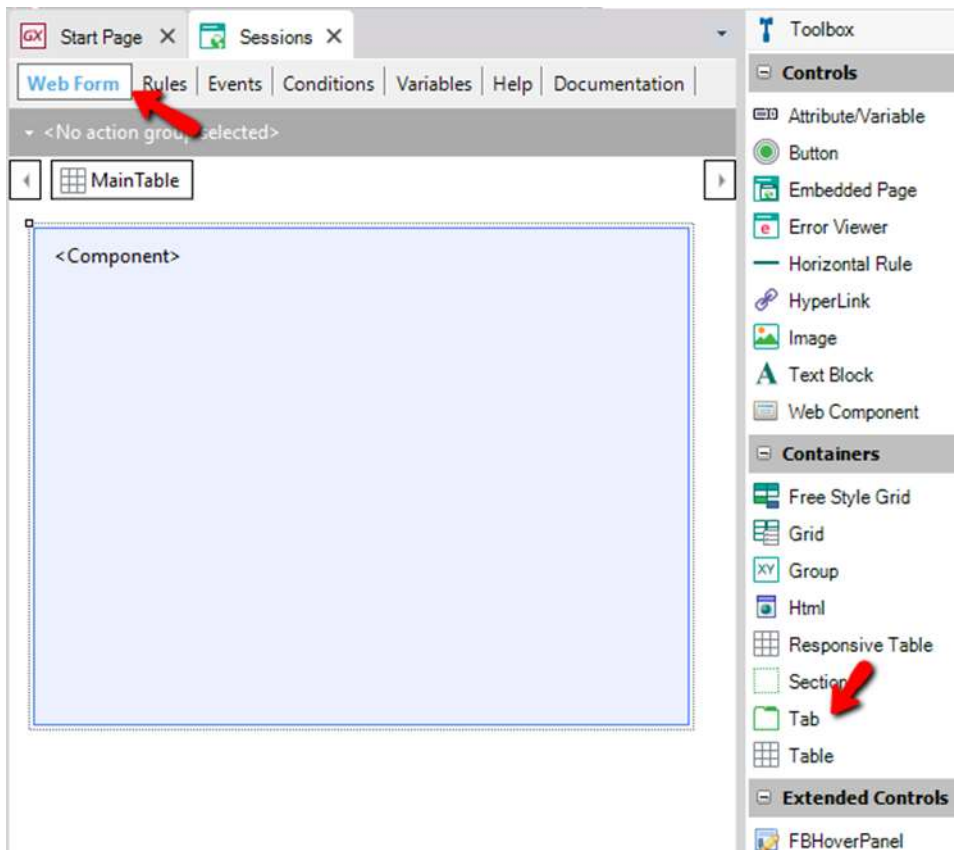


Pero queremos lograr que los días se visualicen de la siguiente forma, sabiendo que el evento siempre se desarrollará en tres días, más allá de sus fechas: lunes, martes y miércoles:

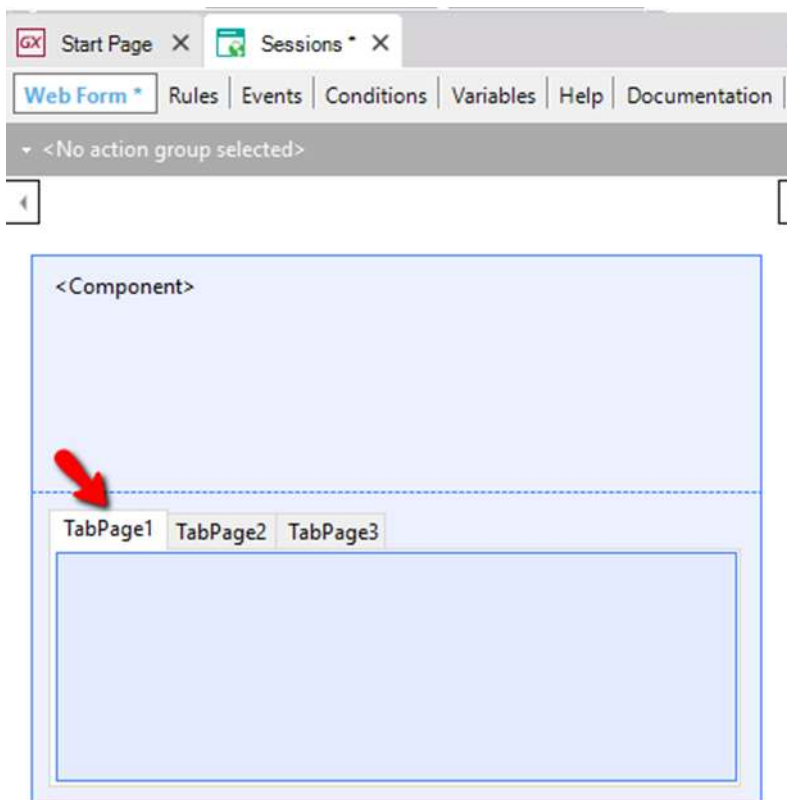


¿Cómo lo implementa?

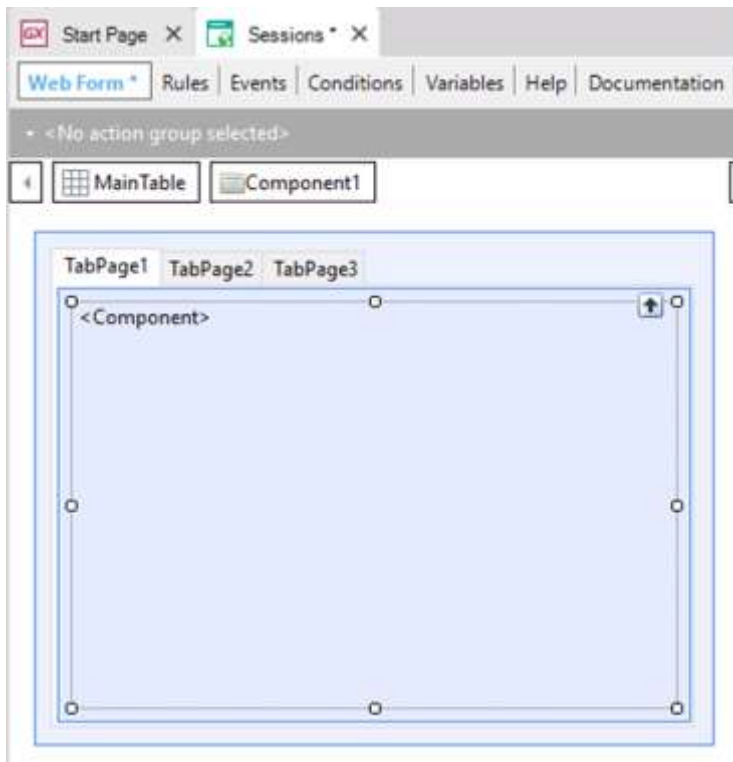
Abra el web panel **Sessions**, elimine la grilla y arrastre un control Tab desde la Toolbox al Web Form:



Observe el control que aparece ahora por debajo del Componente:



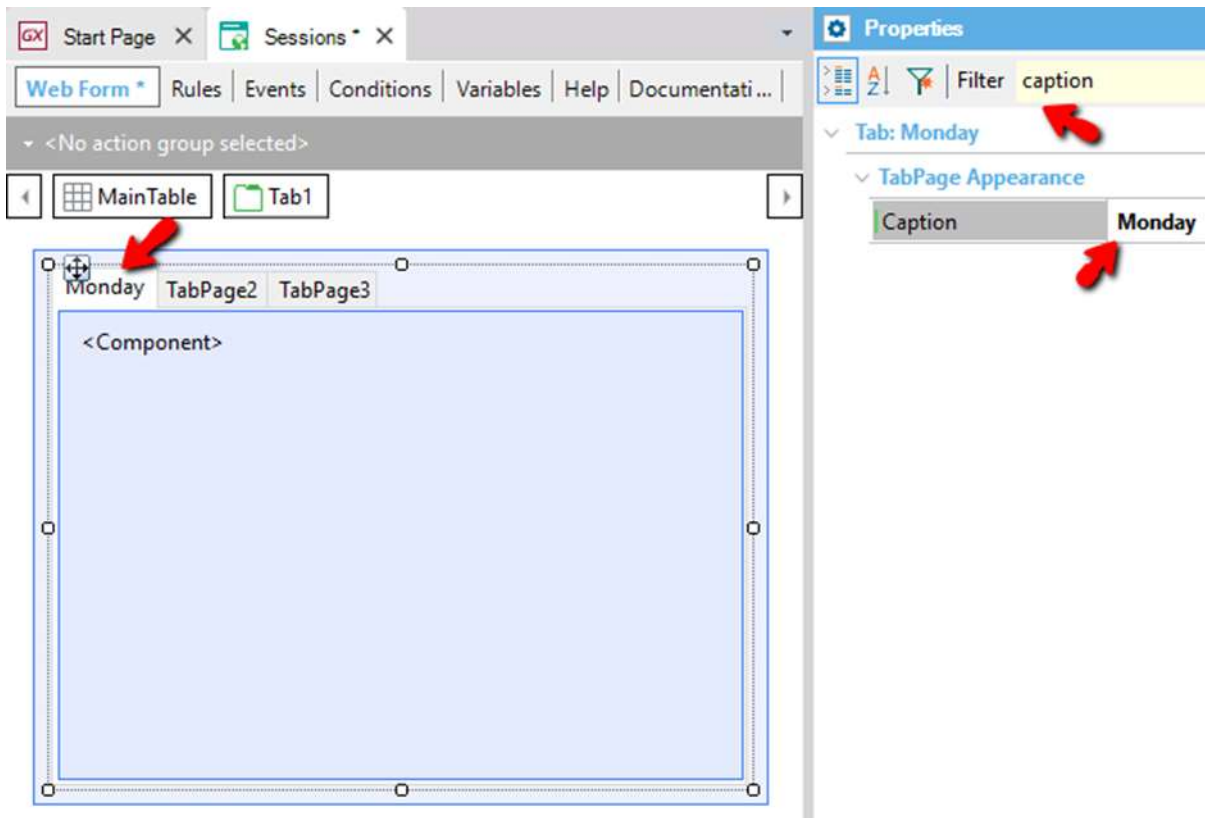
Arrastre ahora el control Component sobre la página TabPage1 del control Tab:



Seleccione la página TabPage2 del control, y arrastre desde la Toolbox un control Web Component sobre el control Tab. Verifique que la propiedad Control Name del Web Component quede con el valor Component2. Repita la operación para página TabPage3, verificando que el Web Component quede con nombre Component3.

Seleccione la página TabPage1 y asigne el valor Monday a la propiedad Caption:





Repita la operación para las otras dos páginas con los valores Tuesday y Wednesday.

¿Cómo modifica los eventos para que cada tab page cargue el componente con la fecha deseada?

**Una posible implementación** es eliminar los eventos que teníamos definidos y definir tres variables &date1, &date2 y &date3 para obtener las fechas correspondientes a lunes, martes y miércoles del evento, y una variable &i Numeric(1,0), y hacer:

#### Event Start

```

&i = 1
for each Session order SessionInitialDate
  unique SessionInitialDate
  do case
    case &i = 1
      &date1 = SessionInitialDate
    case &i = 2
      &date2 = SessionInitialDate
    case &i = 3
      &date3 = SessionInitialDate
  endcase
  &i+=1
endfor
Component1.Object = SessionsComponent.Create(&date1)

```

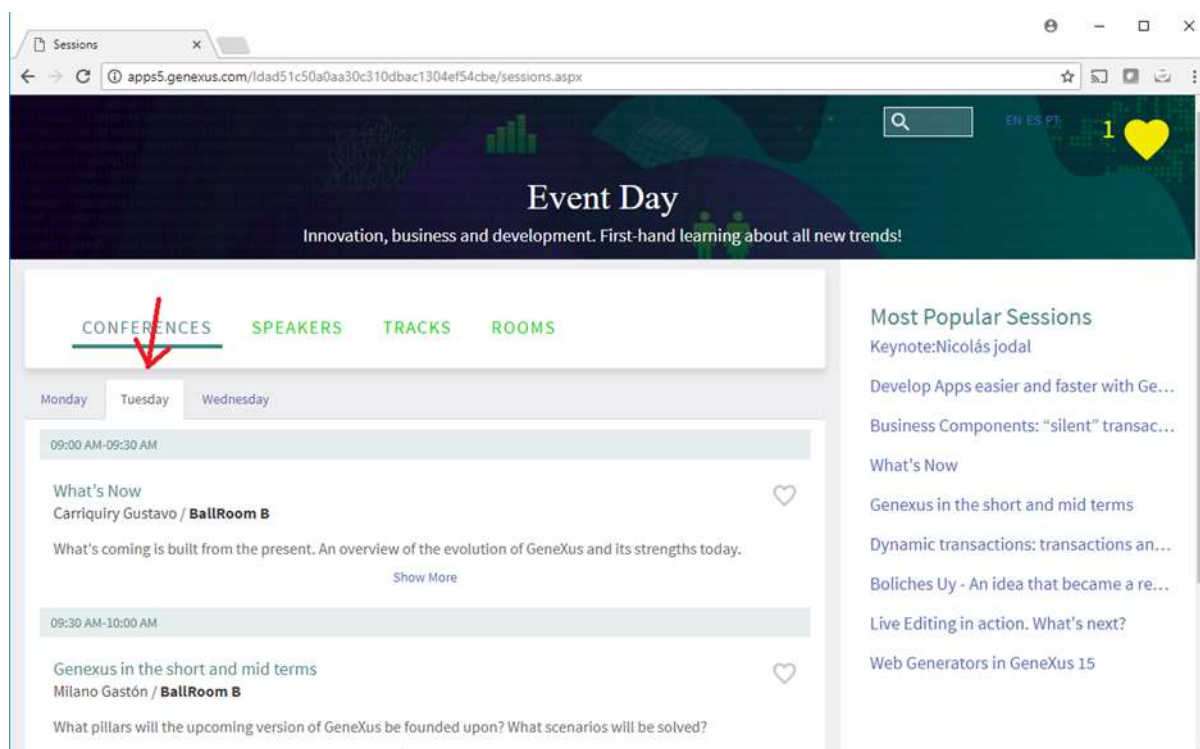
#### EndEvent

```

Event Tab1.TabChanged
do case
  case Tab1.ActivePage = 1
    Component1.Object = SessionsComponent.Create(&date1)
  case Tab1.ActivePage = 2
    Component2.Object = SessionsComponent.Create(&date2)
  case Tab1.ActivePage = 3
    Component3.Object = SessionsComponent.Create(&date3)
  endcase
endevent

```

Ejecute el web panel **Sessions**, y observe cómo se visualiza el control Tab, y cómo es posible ver las conferencias de cada día seleccionando cada una de las páginas:



## WEB - CALLOPTIONS TARGET

No vamos a hacerlo en el práctico, pero usted puede verlo en la KB.

El panel que muestra las conferencias más populares está hecho de esta forma –cuando estamos en una visualización small el panel de favoritos aparece como una barra flotante, mientras que cuando estamos en pantalla grande (desktop) el panel está fijo a la derecha.

Small:

Sessions

apps5.genexus.com/ldff3170080b7206d32d9cc51c71ca0d6e/sessions.aspx

iPad 768 x 1024 75% Online small

EN ES PT

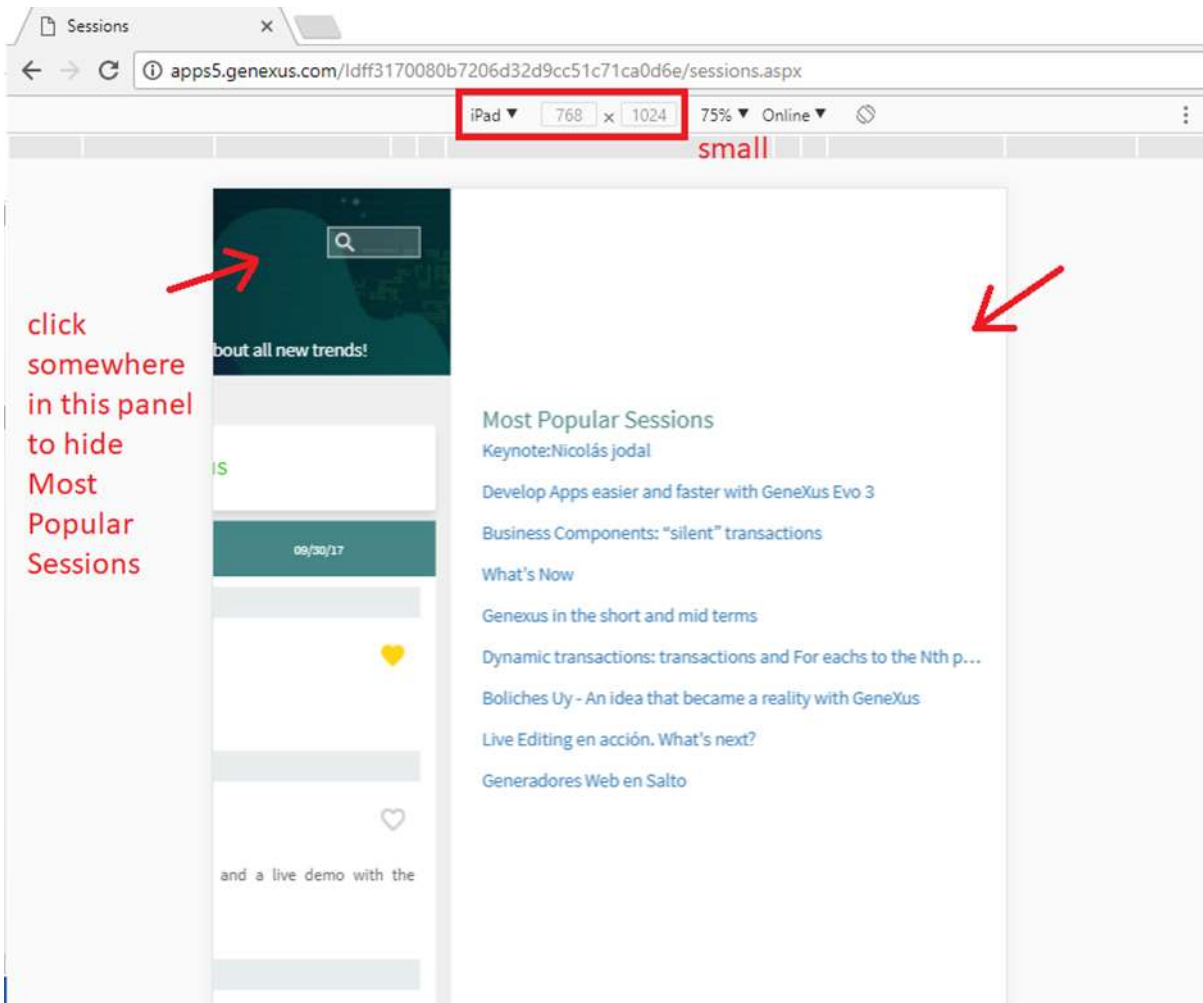
click to show Most Popular Sessions

Event Day

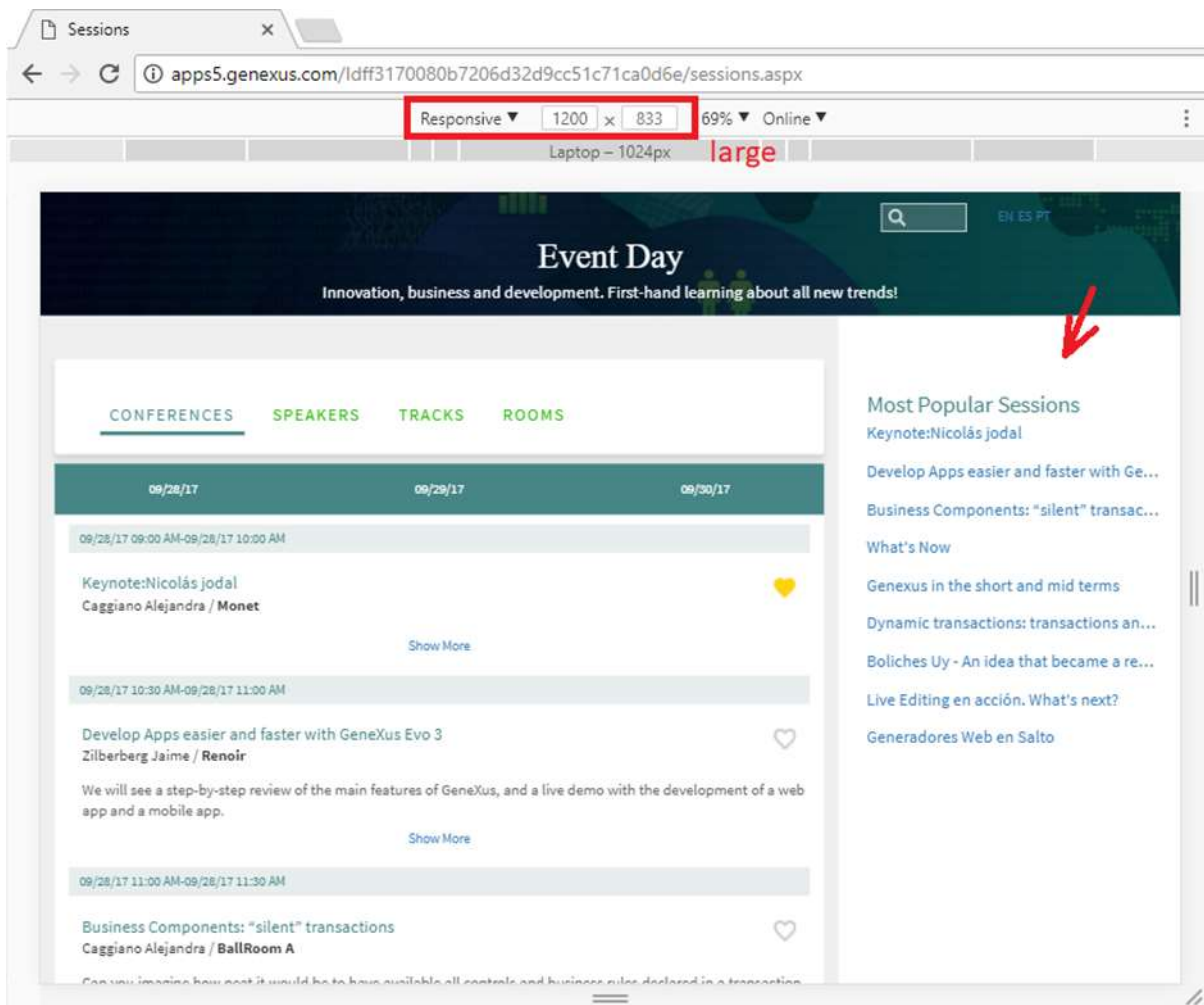
Innovation, business and development. First-hand learning about all new trends!

CONFERENCES SPEAKERS TRACKS ROOMS

09/28/17	09/29/17	09/30/17
09/28/17 09:00 AM-09/28/17 10:00 AM		
Keynote:Nicolás jodal Caggiano Alejandra / <b>Monet</b>		
<a href="#">Show More</a>		
09/28/17 10:30 AM-09/28/17 11:00 AM		
Develop Apps easier and faster with GeneXus Evo 3 Zilberberg Jaime / <b>Renoir</b>		
We will see a step-by-step review of the main features of GeneXus, and a live demo with the development of a web app and a mobile app.		
<a href="#">Show More</a>		
09/28/17 11:00 AM-09/28/17 11:30 AM		

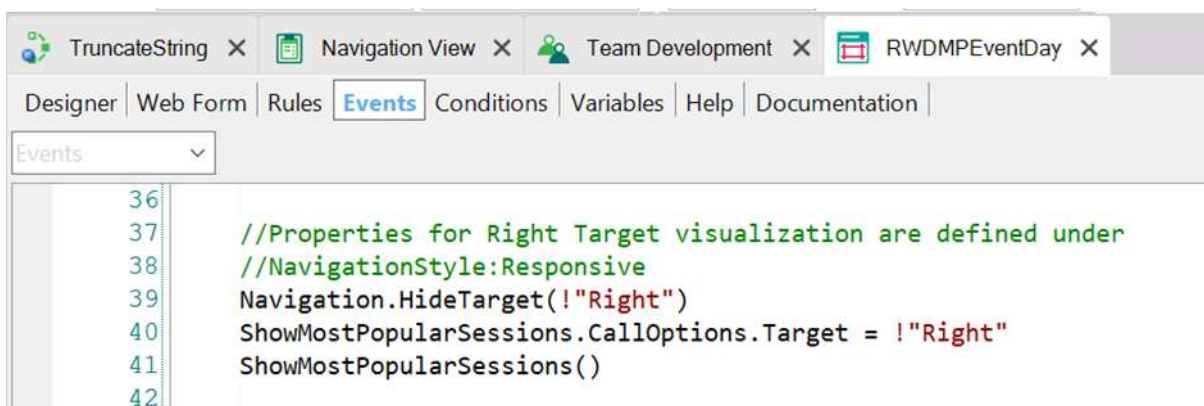


Large:



Esto se logra con la configuración de la sub clase **ResponsiveSlide** de la clase **NavigationStyle** en el Theme (recuerde que para los objetos del FrontEnd creamos otro theme, a partir del Carmine, CarmineFrontEnd).

Y en el evento Start de la Master Page del FrontEnd se hizo:

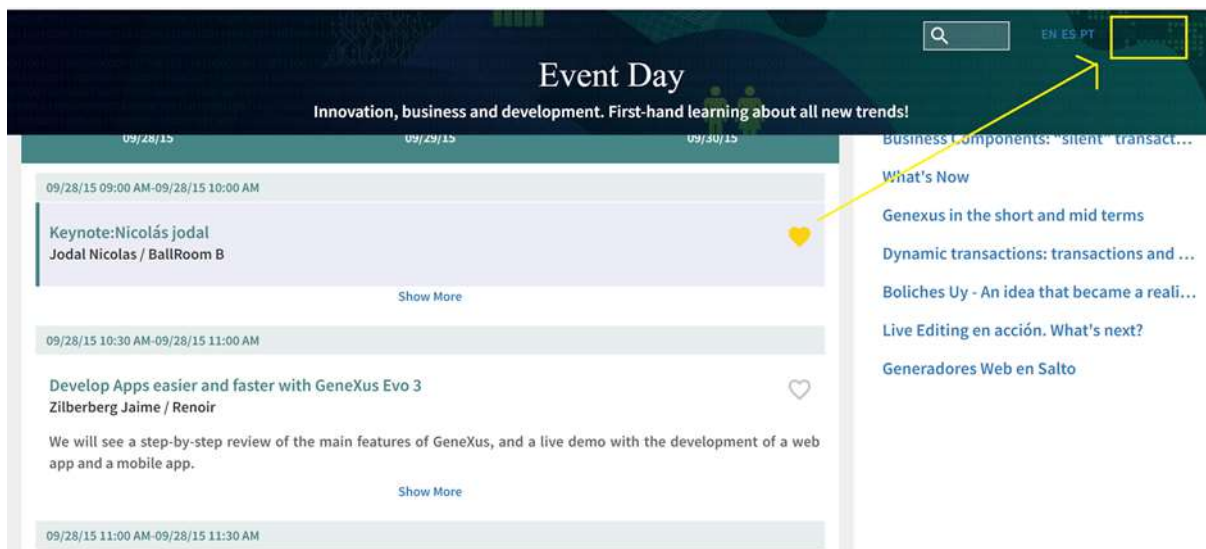


Observe que cuando el tamaño es Medium o mayor el modo de la clase ResponsiveSlide es Static, en cambio para Small y ExtraSmall es Slide. Y por eso, para esos tamaños hay que programar el click del control image Favorites (para esconder o mostrar).

## WEB - GLOBAL EVENTS

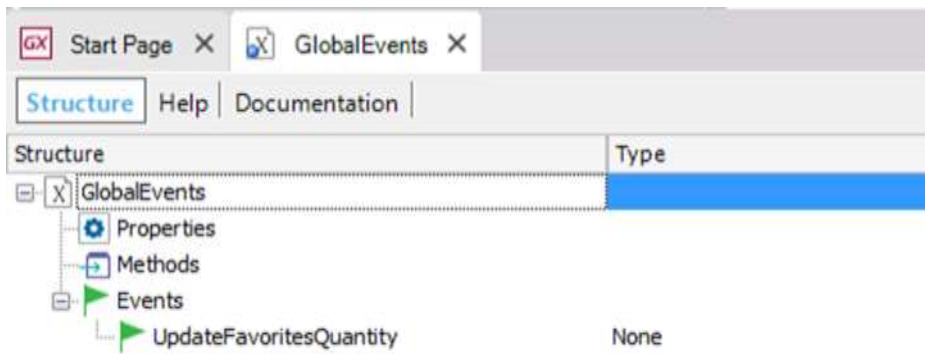
Global Events es una nueva funcionalidad muy potente, que permite que una acción en un Web Component (también valdrá para componentes SD, como veremos luego) dispare una acción en cualquier otro Web Component de nuestro formulario.

Supongamos que queremos tener en la Master Page del FrontEnd un Web Component donde mostremos la cantidad de conferencias que el usuario ha seleccionado como favoritas.



Lo ideal sería que el componente se actualizara a medida que el usuario marca o desmarca una conferencia como favorita, sin necesidad de refrescar todo el formulario. Esto es posible utilizando Global Events. Veamos cómo.

Abra el external object **Global Events** (observe que no se encuentra dentro del **módulo** GeneXus, dado que deberá poder ser modificado por el desarrollador) y agregue un evento *UpdateFavoritesQuantity*:



Y grabe el objeto. Ahora llamaremos al evento global recién creado desde el objeto donde el usuario selecciona o deselecciona una conferencia como favorita.

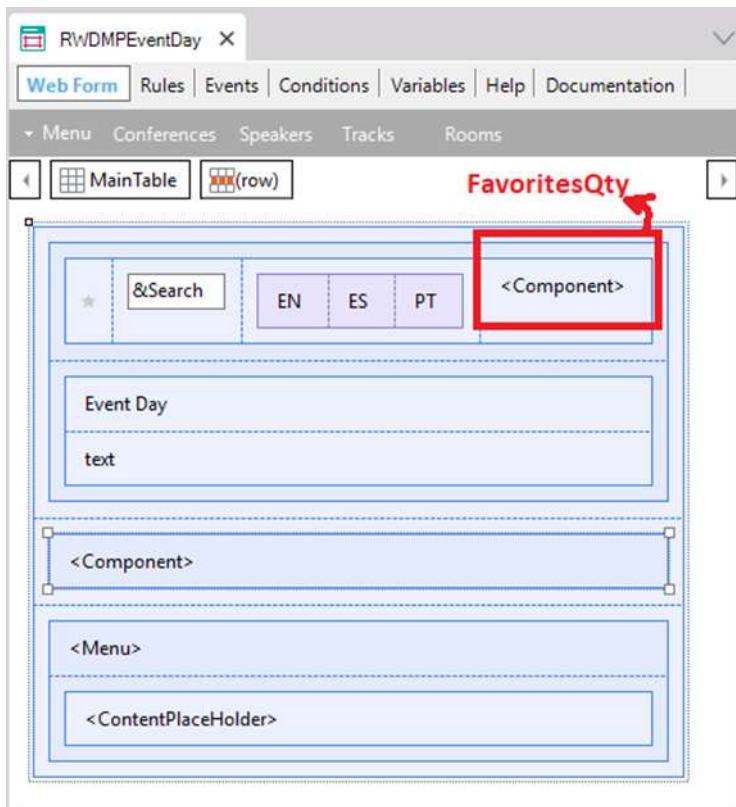
Abra el web component **SessionsComponent**, y en el evento `imgToggleFavorite.Click`, agregue la línea marcada en negrita:

```
Event imgToggleFavorite.Click
  &IsFavorite = ToggleFavorite(&SessionId)
  if &IsFavorite
    imgToggleFavorite.FromImage(BtnFavSelec)
  else
    imgToggleFavorite.FromImage(FavNo)
  endif
  GlobalEvents.UpdateFavoritesQuantity()
Endevent
```

De esta manera, cada vez que el usuario marque/desmarque una session como favorita se estará disparando el evento, que será *escuchado* por todo objeto que lo maneje (es decir, que lo tenga definido entre sus eventos).

Observemos que a la master page **RWDMPEventDay** le hemos agregado en el layout un control component de nombre `FavoritesQty`:





Nos hace falta solamente cargarlo con el web component **FavoriteCount** (abrirlo y observar que cuenta y muestra la cantidad de sessions favoritas), para lo que en el evento Refresh de la master page agregamos la línea en negrita:

```
Event Refresh
    WCRecentLinks.Object = RwdRecentLinks.Create(Form.Caption, ContentHolder.Pgmname)
    FavoritesQty.Object = FavoriteCount.Create()
EndEvent
```

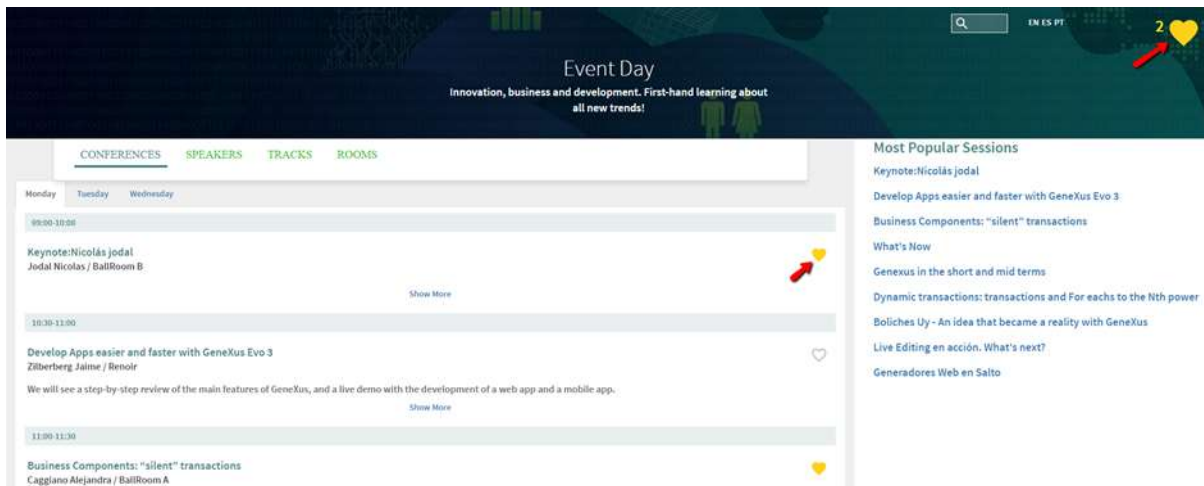
Lo que queremos es que cuando un panel o componente dispare el evento global UpdateFavoritesQuantity, este web component FavoriteCount actualice la cuenta.

Para ello alcanza con que programemos en su sección de eventos el evento global:

```
Event GlobalEvents.UpdateFavoritesQuantity()
    do 'GetCount'
EndEvent
```

Ejecute el web panel **Sessions**, y observe cómo se muestra en la parte superior derecha de la master page el componente, y cómo se actualiza al seleccionar o deseleccionar conferencias como favoritas:





## SD - APLICACIÓN ANDROID

Ahora nos dedicaremos a familiarizarnos y ampliar las funcionalidades de la app SD que ya está parcialmente desarrollada en esta KB.

Prenda la generación para Android. En las propiedades del Environment asegúrese de que la de nombre “Protocol Specification” tenga el valor Default que es “Unsecure (HTTP)”.

Observe el folder SD dentro del KB Explorer. Abra el objeto de nombre EventDay. Vea que el viejo objeto Dashboard ahora se llama Menu for Smart Devices. Realice un Run sobre él. Se compilará la app Android (llevará un rato) y se abrirá en el emulador (o en su dispositivo si lo ha conectado a la computadora).

Observar que hemos agregado a la KB un theme específico para Android: EventGXAndroid. Si va a SD Platforms / Any Android verá que se tiene configurado este theme.

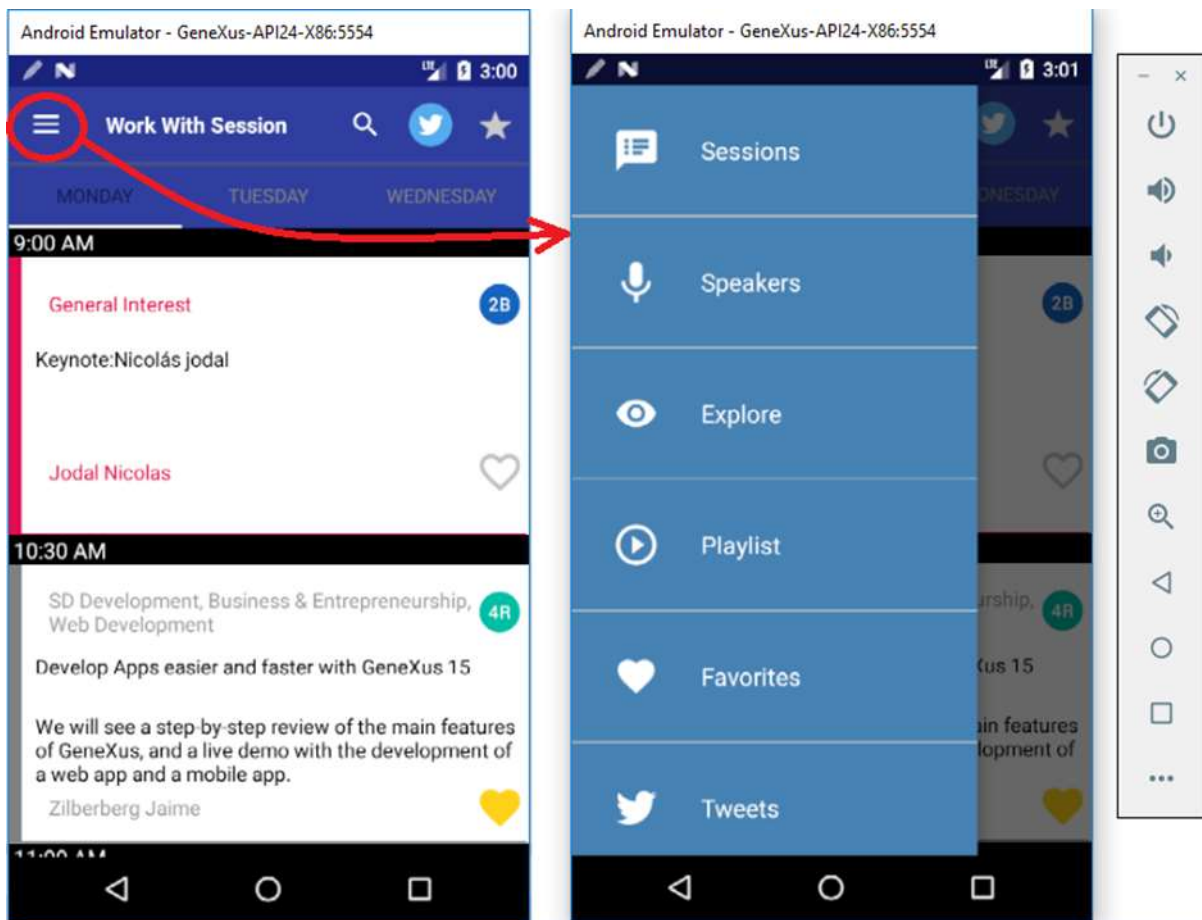
Cambie el tipo de navegación por **Slide**.

The screenshot shows a 'Properties' window with a 'Filter' field and a 'Document Outliner' on the right. The main content is a table with the following data:

Platform: Any Android	
Name	Any Android
OS	Android
Version	
Device Kind	All
Size	All
Theme	<b>EventGXAndroid</b>
Navigation Style	<b>Slide</b>
Default Layout Orien	Any
Bounds Name	
Minimum Shortest I	0
Maximum Shortest	0

**Nota:** A partir del upgrade 6 de GeneXus 15 se ha creado un nuevo theme default para aplicaciones móviles: el CarmineSD, en dos variantes: CarmineAndroid y CarmineiOS (no los verá en esta KB, puesto que el check box de CarmineSD está desmarcado), con variantes de rojo. Pero, como siempre, usted deberá utilizar el theme que responda a los colores y demás especificidades de la app que está desarrollando. En GeneXus 16 aparecerá con fuerza el concepto de Design System, relacionado con todo esto.

Se verá:



Navegue por la app. ¿Dónde se han configurado los colores de la application bar, de los íconos en ella, de las acciones, del background de la app?

Respuesta: en la clase Application del theme. Observe sus propiedades.

En lo que sigue probaremos algunas de las nuevas features realizando modificaciones sobre los objetos que implementan esta app Android.

Los Work with se encuentran bajo las transacciones, en el folder DataModel.

## SD - LIVE EDITING

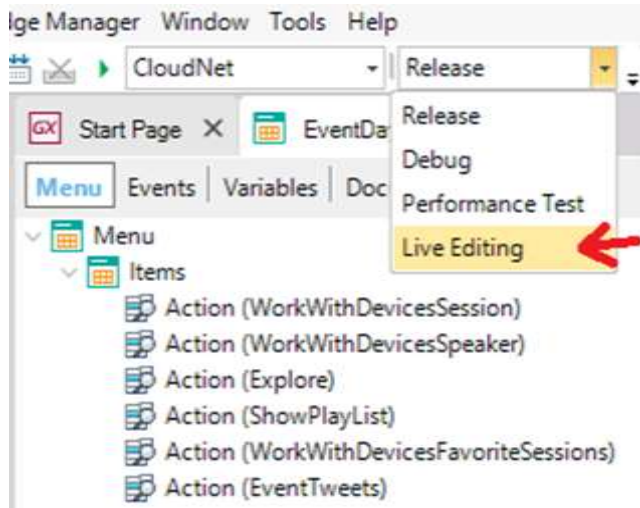
¿Qué tal si pudiéramos cambiar el diseño de nuestras pantallas y -SIN generar- ver los cambios directamente en nuestra aplicación SD?

Se implementó el “Live editing” y “Live Inspector” para lograr este objetivo y agilizar ampliamente el diseño de la aplicación.

Veamos cómo se activa y cómo funciona.

Seleccionar el valor **Live Editing** en el combo en la barra de GeneXus. De esta manera nuestra aplicación se mantendrá conectada con nuestro IDE.

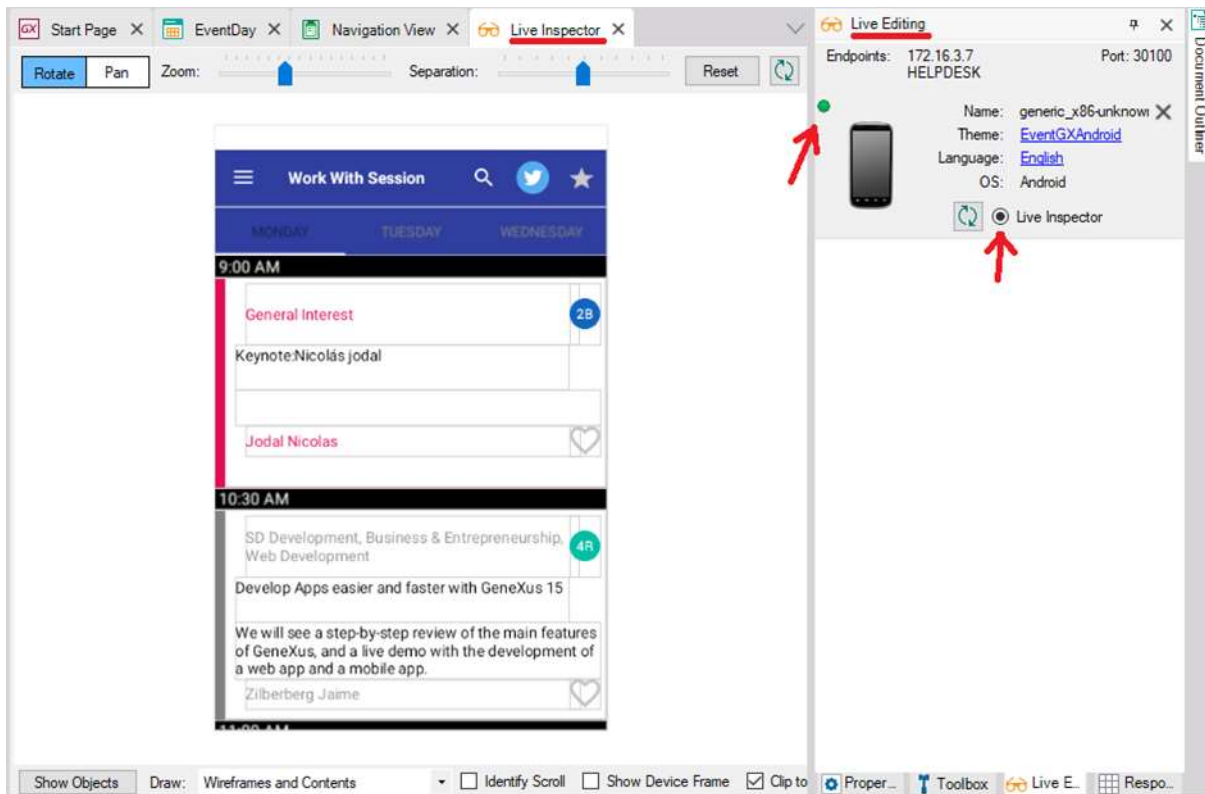
Para activarlo, hay que cambiar el valor de este combo, configurando el valor “Live Editing”:



Hecho esto, ya podemos ejecutar nuestra aplicación. Dado que ahora vamos a estar desarrollando y probando la app móvil, le convendrá configurar al Menu for Smart Devices **EventDay** como Startup object, así el F5 hace un Run de él.

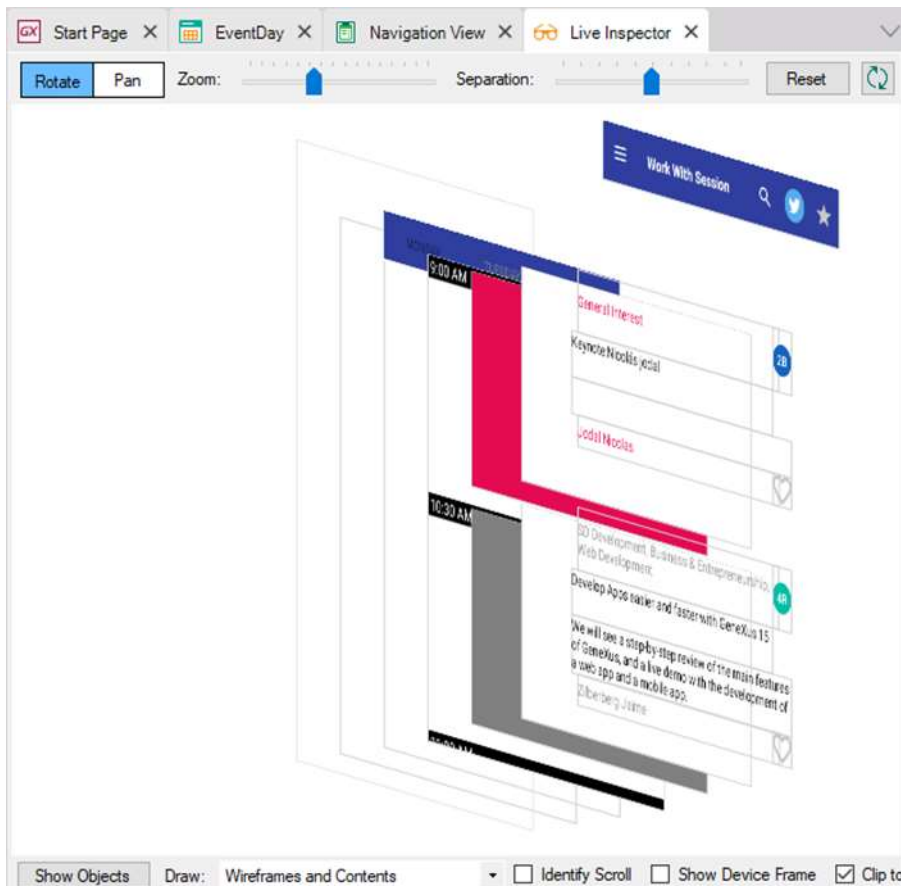
Se abrirá la app en el **Emulador Android**. Además, se abrirán dos ventanas:

- “Live Editing” mostrando el nombre del dispositivo que está ejecutando actualmente la app –en este caso la instancia del emulador– y con el que se está conectado (vea el punto verde que lo indica).
- “Live inspector” que permite analizar en runtime la configuración de propiedades de cada ítem en un objeto con UI y modificarlas para ver el resultado in-line, mejorando enormemente la productividad del desarrollador comparando con GeneXus XEv3 (donde el desarrollador, para poder ver sus cambios, necesariamente debía re-compilar la aplicación cada vez).

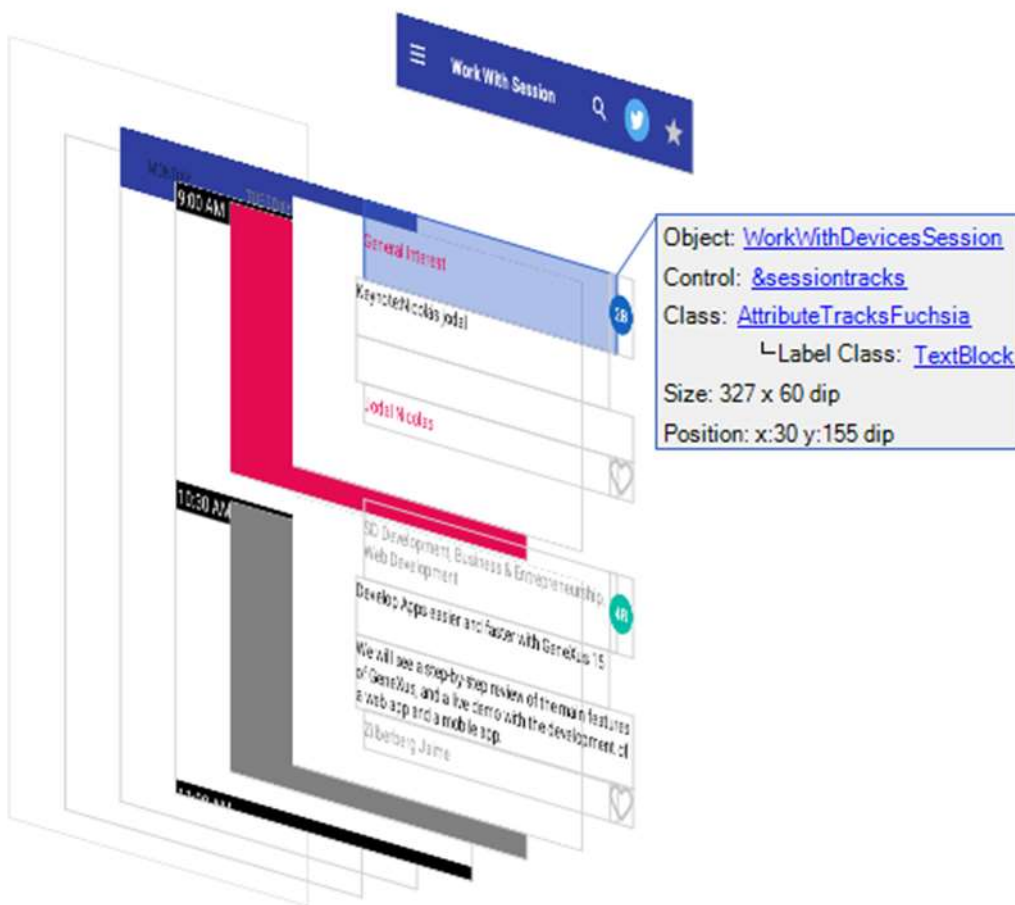


En el inspector se muestra la pantalla que está ejecutándose en ese momento dentro de la aplicación (emulador) y podemos visualizar cómo se reflejan los cambios realizados en tiempo real.

Pruebe hacer clic, dejar presionado y mover el mouse. Verá que se le mostrarán las distintas capas que conforman la pantalla del objeto que se está ejecutando:

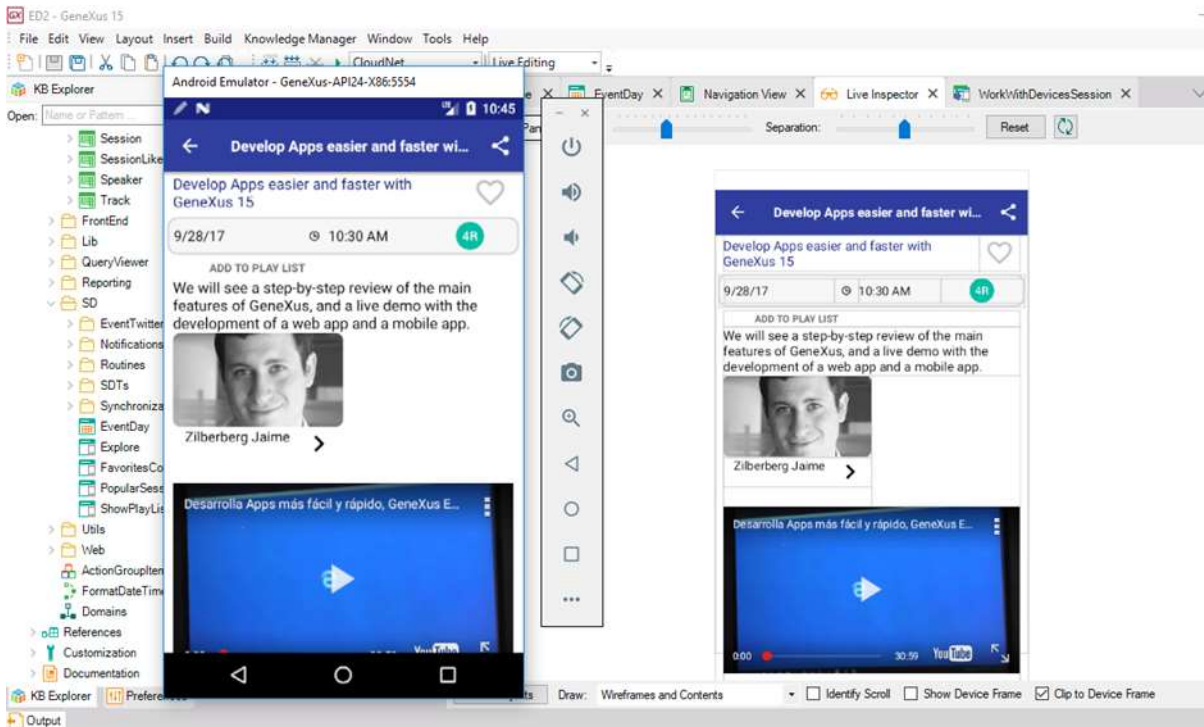


Observar que lo que se haga en la app en el emulador se reflejará en esta pantalla automáticamente. Observe también que pasando el mouse sobre los distintos elementos, se le informa de qué control se trata. Y si hace clic, se le muestra la info ampliada (objeto, control, clase y subclases del theme, tamaño, posición):



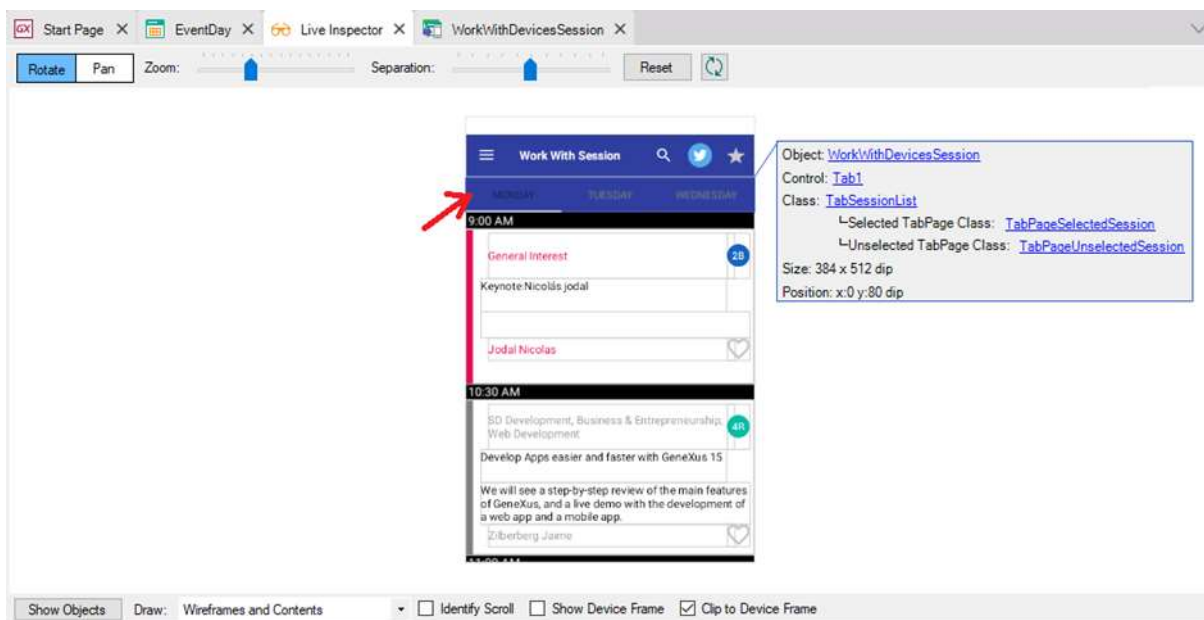
Con el botón Reset puede volver a la visualización plana.

Ahora vaya a la app en el emulador, y cambie de pantalla. Por ejemplo, haga tap sobre la segunda conferencia. Verá que automáticamente la ventana de Live Inspector le muestra esta pantalla también.



Ahora veamos su utilidad.

Vuelva a visualizar el List de Sessions. (Si perdió la conexión presione en el emulador donde se le indica Re-Try o haga Ctrl+F5 –Run without building–). Dentro del Live Inspector haga clic en el tab posicionado debajo de la application bar:

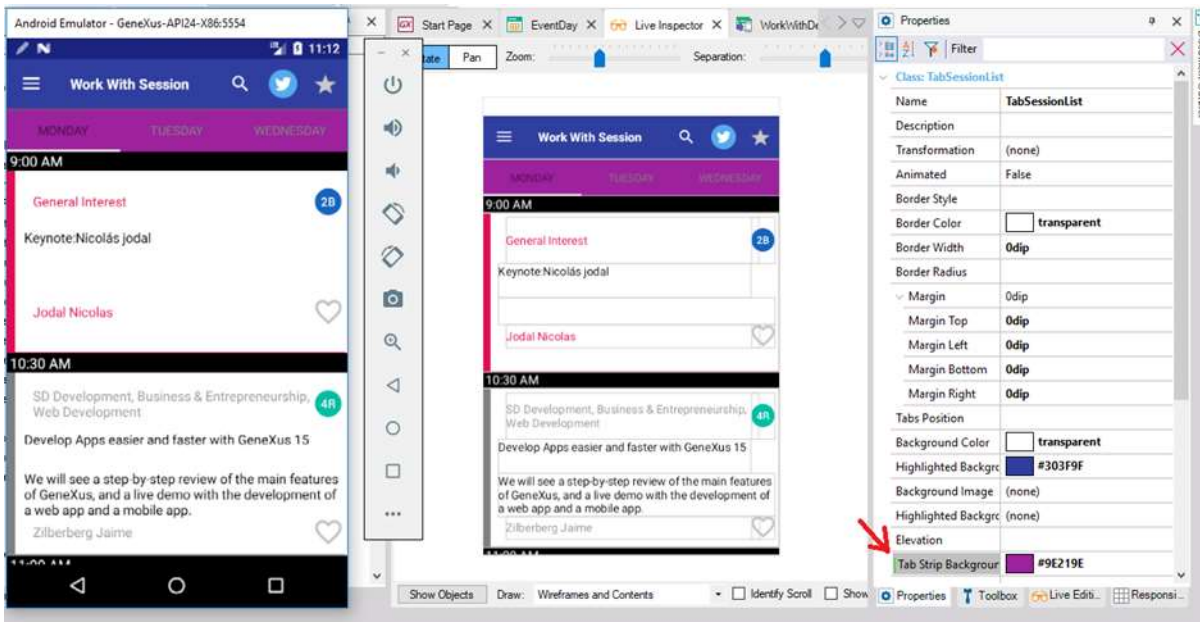


Luego haga clic en el link donde aparece el nombre de la clase asociada al control, TabSessionList. Observe la ventana de propiedades:



Se cargarán automáticamente las propiedades de la clase seleccionada.

Modifique la propiedad **Tab Strip Background Color = #9E219E** y visualizará los cambios tanto en el emulador como en el Live Inspector:

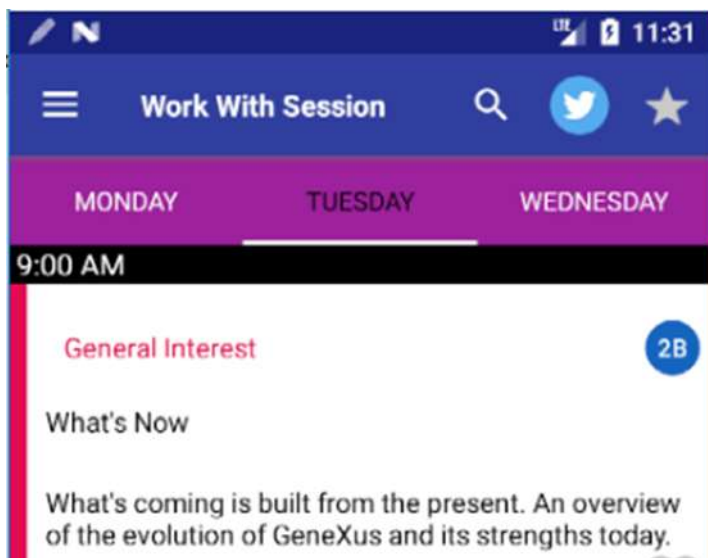


El color del font del tab cuando está seleccionado y cuando no lo está hace que casi no se vean los textos. Pruebe cambiar los colores por unos más contrastantes.

**Solución:** vuelva a hacer clic en el Live Inspector sobre el tab y ahora haga clic en la subclase **TabPageSelectedSession** y luego en la **TabPageUnselectedSession**.



Por ejemplo, dejarlo en negro cuando está seleccionado el tab y en blanco cuando no lo está:



De esa forma, se pueden seguir modificando propiedades tanto del control como de la clase asociada al control hasta lograr el diseño deseado, sin tener que compilar para probar.

Antes de continuar, abra el objeto PopularSessions y *descomente*, si no lo hizo antes, el código del evento Load (siempre y cuando haya definido en su KB la transacción dinámica FavoritesStatistics que se le pedía mucho antes en este práctico). Ese panel es invocado desde la Action Bar del List de Sessions, al hacer tap sobre la imagen de la estrella.

## AGREGANDO DISEÑO CON LIVE EDITING

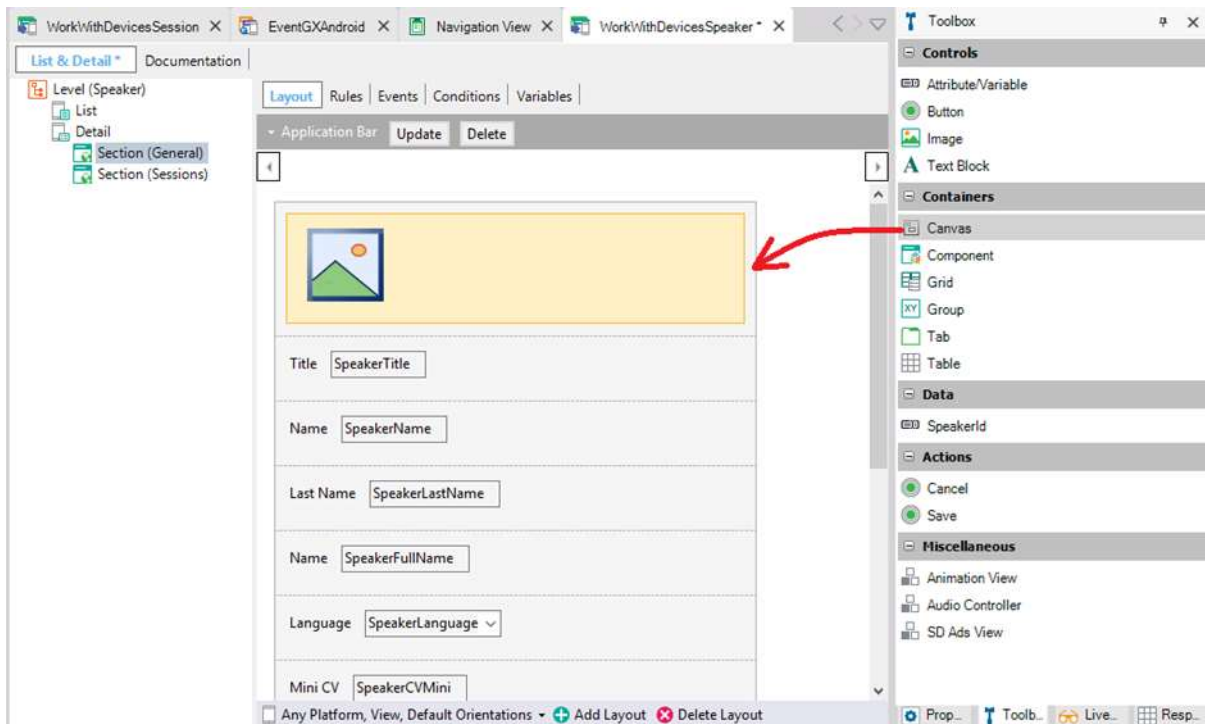
Si hacemos tap sobre una charla del List de Sessions se despliega su información. Al hacer tap en el nombre del orador, se despliegan los datos del orador sin ningún tipo de diseño.

Vamos a modificar ese layout:



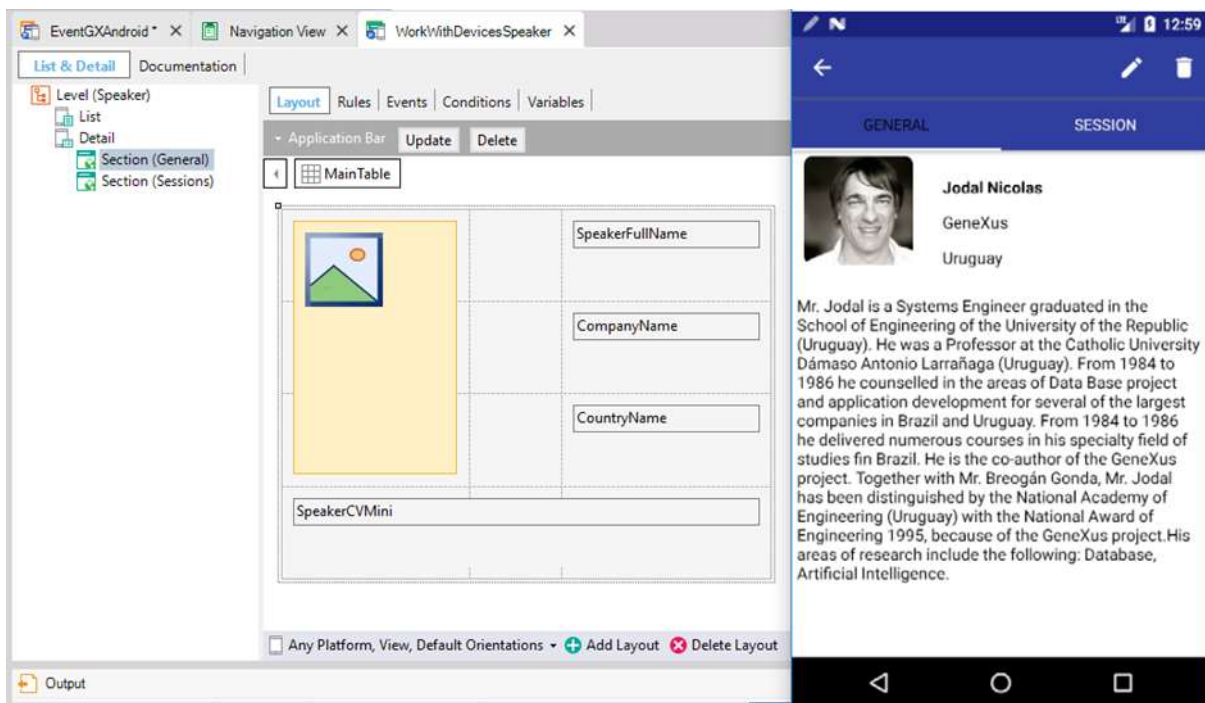
En el IDE, abra el objeto `WorkWithDevicesSpeaker`, Section (General), layout View.

Comenzamos colocando un `Canvas` al principio de la `Tabla`. Y al atributo `SpeakerImage` (que está más abajo) lo arrastramos dentro del `Canvas`. Cámbiele su clase por `ImageFillKeeping`.



Ir observando en el emulador cómo van viéndose los cambios a medida que se hacen.

Intente que el layout le quede más o menos así (aunque la idea es que pruebe Live editing. Puede intentar otro diseño de layout que le guste más):



Para que el resultado sea más o menos el que ve en pantalla: a la tabla canvas configúrele el Row span en 3 y el Height en 100%.

A la imagen la clase ImageSpeakerMain, Auto Grow False y Absolute position:

- Top: 5dip
- Left: 50%
- Bottom: 100%
- Right: 50%
- Width: 100dip
- Height: 100dip
- Z-Order: 0

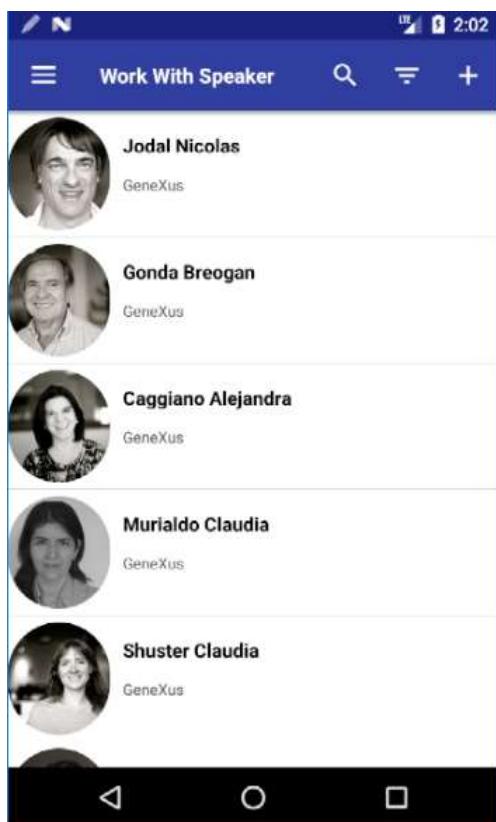
Para dejarle un margen izquierdo a los atributos, cuenta con la clase AttributeAttributeLeftMargin5.

**Nota:** si hubiera tenido que agregar un atributo que no se encontraba ya cargado, Live editing no lo iba a ayudar e iba a tener que hacer un build. ¿Por qué?

## SD - CONTROL HORIZONTAL GRID, SMART GRID, FLEX GRID

El control Horizontal Grid nos permite visualizar los elementos de una lista de forma horizontal en vez del clásico grid con filas.

Si ejecutamos el Menu for Smart Devices **EventDay** en el emulador, y vamos a **SPEAKERS**, veremos la lista de oradores de la forma habitual:

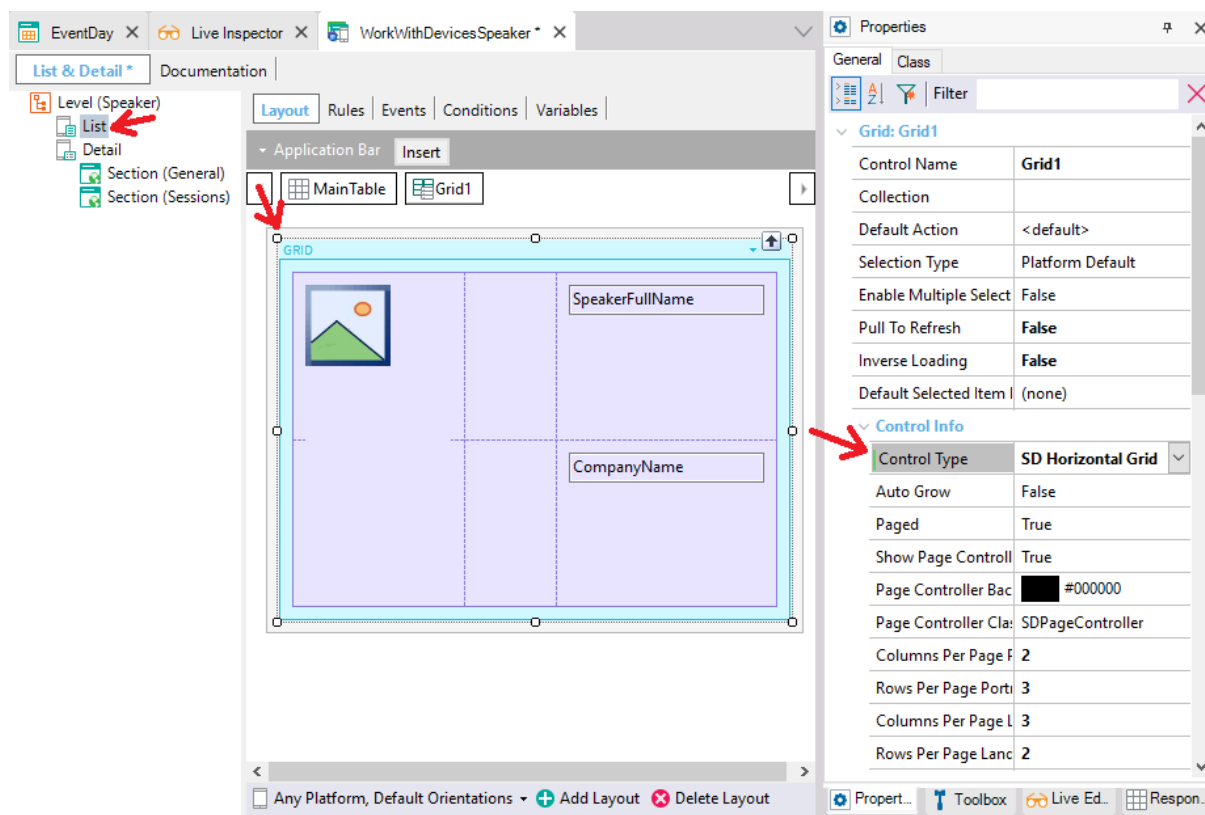


Pero queremos verlos de esta otra forma:



Para ello:

Abra el objeto **WorkWithDevicesSpeaker**, y en el nodo **List**, seleccione el grid. Configure la propiedad **“Control Type”** con el valor **“SD Horizontal Grid”**, así como las propiedades mostradas en la siguiente imagen, que permiten controlar la cantidad de filas y columnas por página:

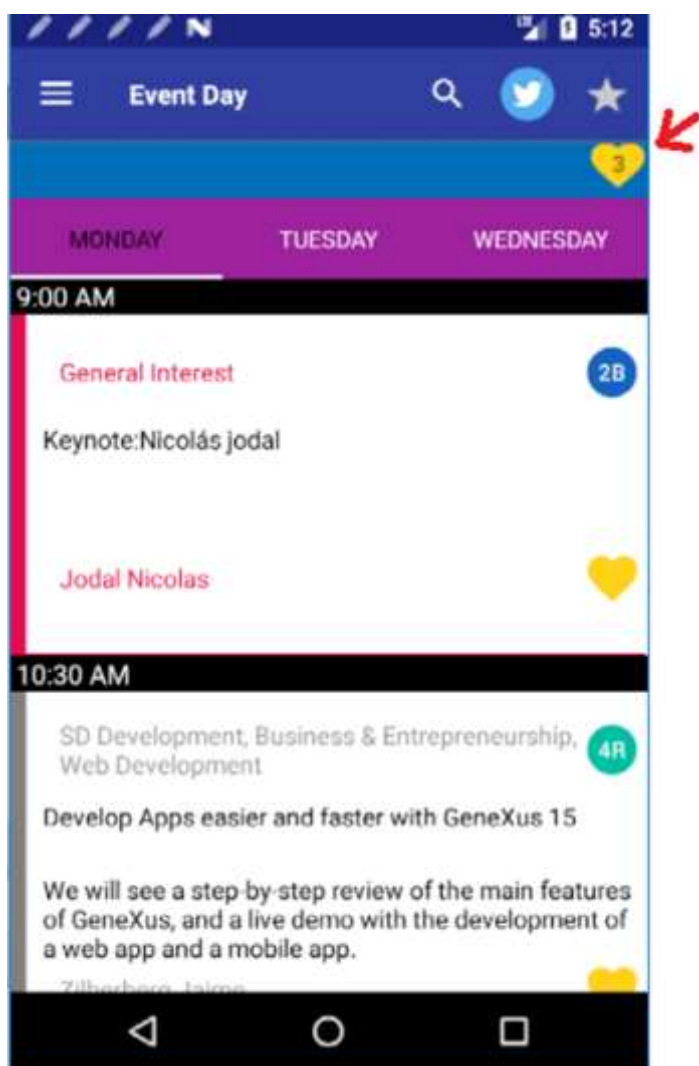


Pruébalo en ejecución. ¿Tuvo que compilar?

Vea que al desplazarse hacia la derecha, los oradores se mueven en bloque. Pruebe modificar el control type ahora por SD Smart Grid, Y pruebe variar las propiedades que se le presentará, para ver las diferencias entre ellas y con el Horizontal Grid. También puede probar el SD Flex Grid (son de GeneXus 16).

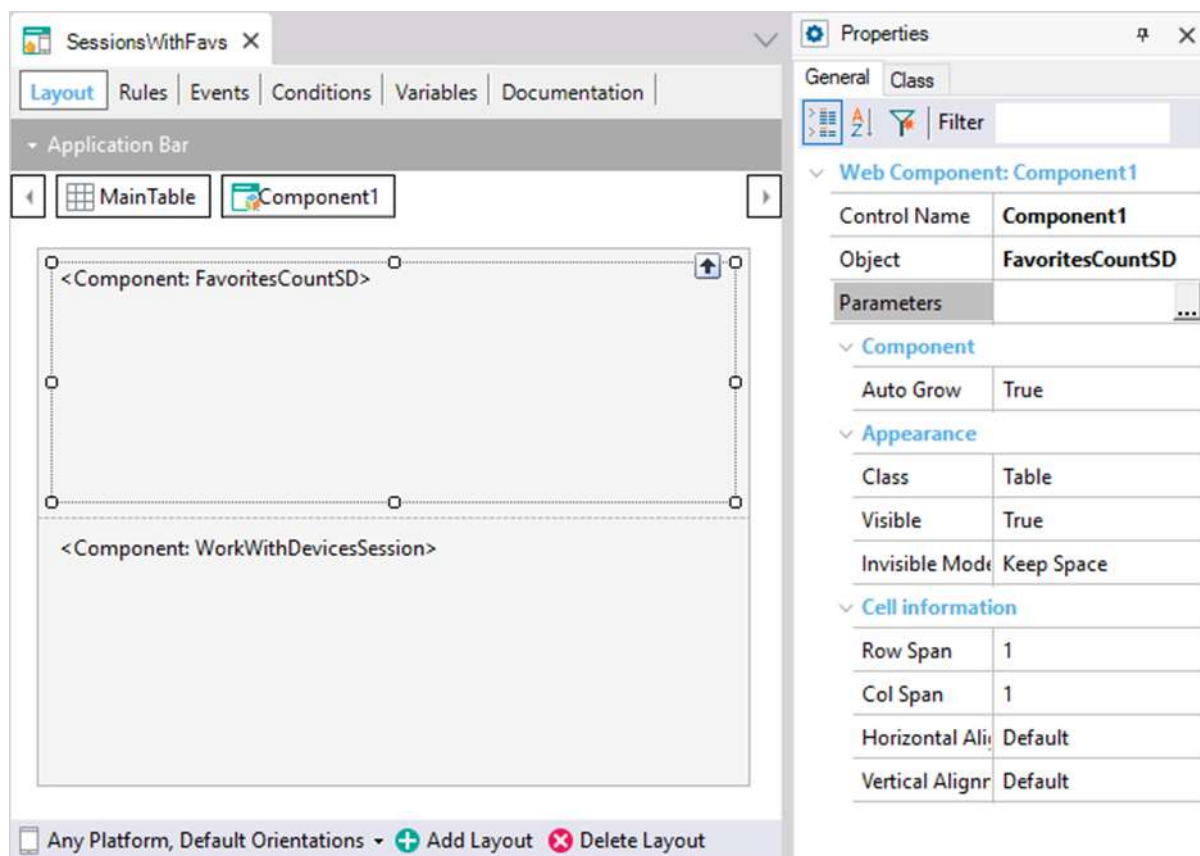
## SD COMPONENTS Y GLOBAL EVENTS

Deseamos modificar la forma en que se muestra el List de Sessions, de manera tal que cuando se *favoritea* una conferencia en cualquiera de los tres días, se observe arriba el número total de conferencias marcadas como favoritas por ese dispositivo (recuerde que habíamos implementado esto mismo para el frontend web):



Una solución para no tener que modificar el layout del `WorkWithDevicesSession` ni el del panel `FavoritesCountSD` (que cuenta y muestra los favoritos) es crear un panel nuevo y utilizar componentes:



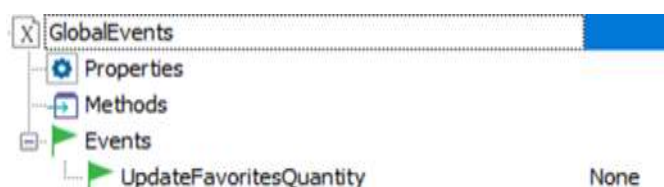


Recomendamos que la propiedad Rows Style de MainTable sea 35dip; 100%.

Con esta implementación conseguimos que se disparen los eventos ClientStart, Start, Refresh y Load de este panel, e inmediatamente esos mismos eventos del componente 1 y luego los del componente 2. Con eso quedará cargada la página como vimos antes.

Nota: Desde el u12 de GeneXus 15 puede invocarse dinámicamente al objeto a ser cargado en el componente.

Pero ahora queremos que si el usuario hace tap sobre la imagen de favoritos del List de Sessions esto dispare el evento global que ya habíamos definido para Web:



Y que el panel for Smart Devices “escuche” este evento, para poder calcular nuevamente la cantidad de favoritos y mostrarla.

Para lo primero, en el evento &Fav\_Image.Tap del List de WorkWithDevicesSession habrá que agregar la línea en negrita:

```

Event &Fav_Image.Tap
  Composite
    InOutFavoriteSession( SessionId )
    GlobalEvents.UpdateFavoritesQuantity()
    Refresh
  endcomposite
Endevent

```

Y en el panel FavoritesCountSD agregar el evento:

```

Event GlobalEvents.UpdateFavoritesQuantity()
  Do 'GetCount'
endevent

```

Modifique en el Menú EventDay la primera acción para que en lugar de invocar al List de sessions, invoque a este nuevo panel:

```

Event 'WorkWithDevicesSession'
  SessionsWithFavs()
  //WorkWithDevicesSession.Session.List()
EndEvent

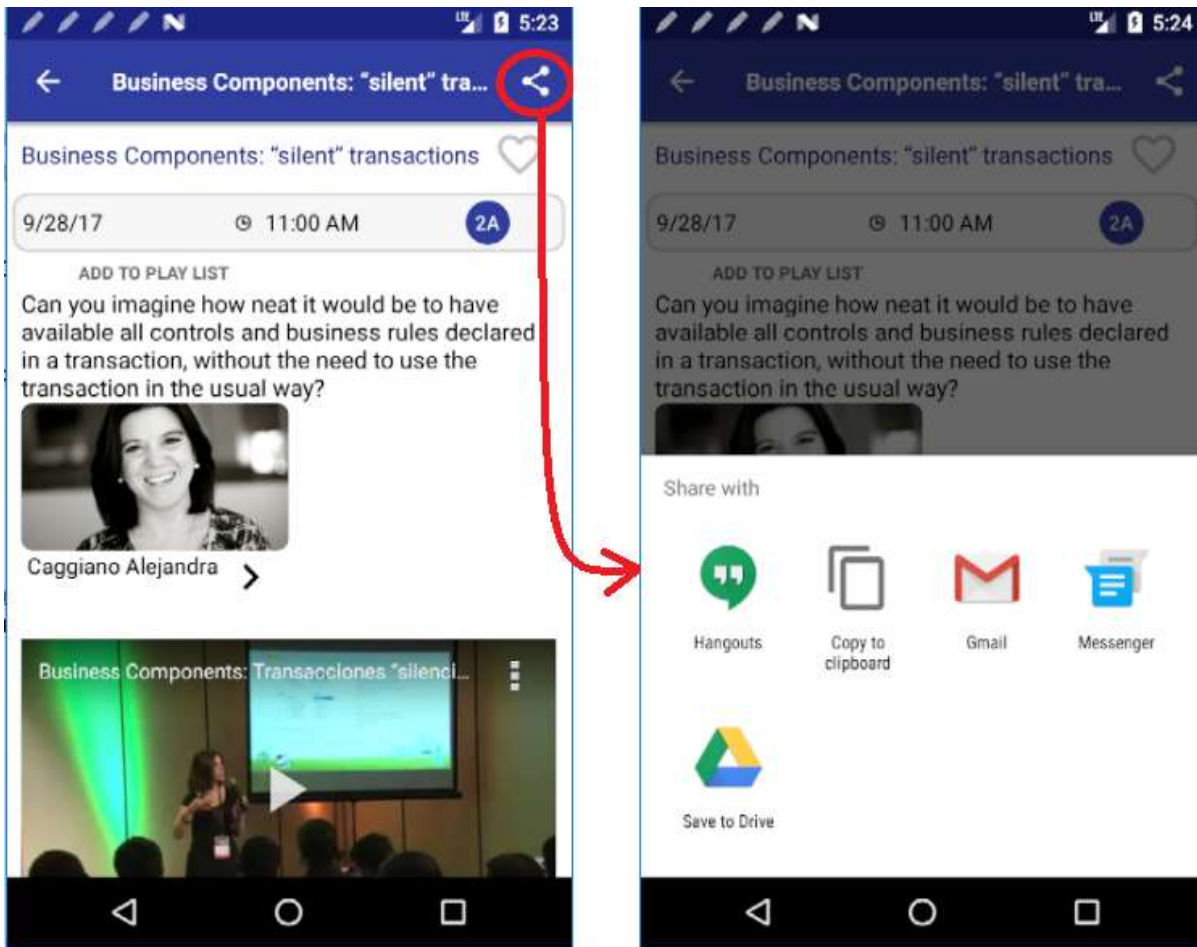
```

Ejecute y pruebe.

Observe cómo la application bar se compone de la unión de ambas application bars (las de ambos componentes. En nuestro caso, el primero está vacío, por lo que sólo se ven las acciones del Workwith).

## SD – COMPARTIR INFO MEDIANTE ALGUNA DE LAS APPS INSTALADAS EN EL DISPOSITIVO (SHARE EO)

Si observa la información detallada de una conferencia, verá que se ha agregado a la action bar un botón Share:



Si lo observa en GeneXus, verá que se ha programado el evento asociado a ese botón:

Event 'Share'

```
GeneXus.Social.Share.ShareText(SessionName, !"http://www.genexus.com",
    SessionInitialDate.ToString())
```

Endevent

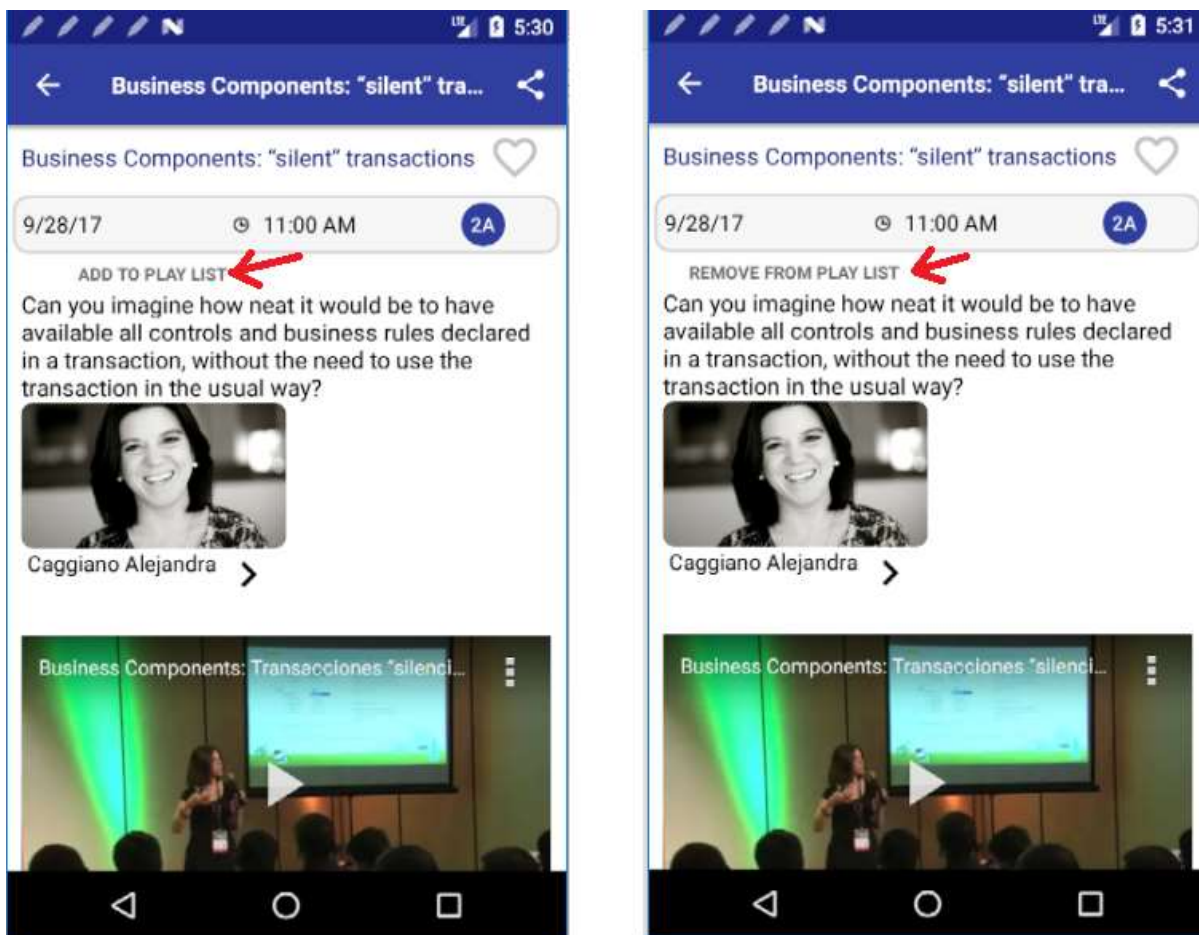
## SD - AUDIO CONTROLLER Y COLAS DE REPRODUCCIÓN

En la versión GeneXus 15 se agregaron nuevos métodos, eventos y SDTs dentro de la api **Audio** que nos ayudan a manejar las colas de reproducción. También se agregó un **control Audio Controller** para brindar la posibilidad de personalizar el reproductor.

En nuestro ejemplo, los nuevos SDTs que utilizaremos son: el SDT **MediaItem**, que contiene la información para cada audio y el SDT **MediaQueue**, que mantiene una lista de **MediaItems**, es decir, la propia cola de reproducción.

Cada conferencia tiene ingresada en el atributo **SessionAudio** la url de un archivo mp3 con el audio de esa conferencia. Observe que en la pantalla de **Detail** de la conferencia aparecerá un texto que,

dependiendo de si la conferencia está o no ingresada en su playlist, le permite agregarla o eliminarla.



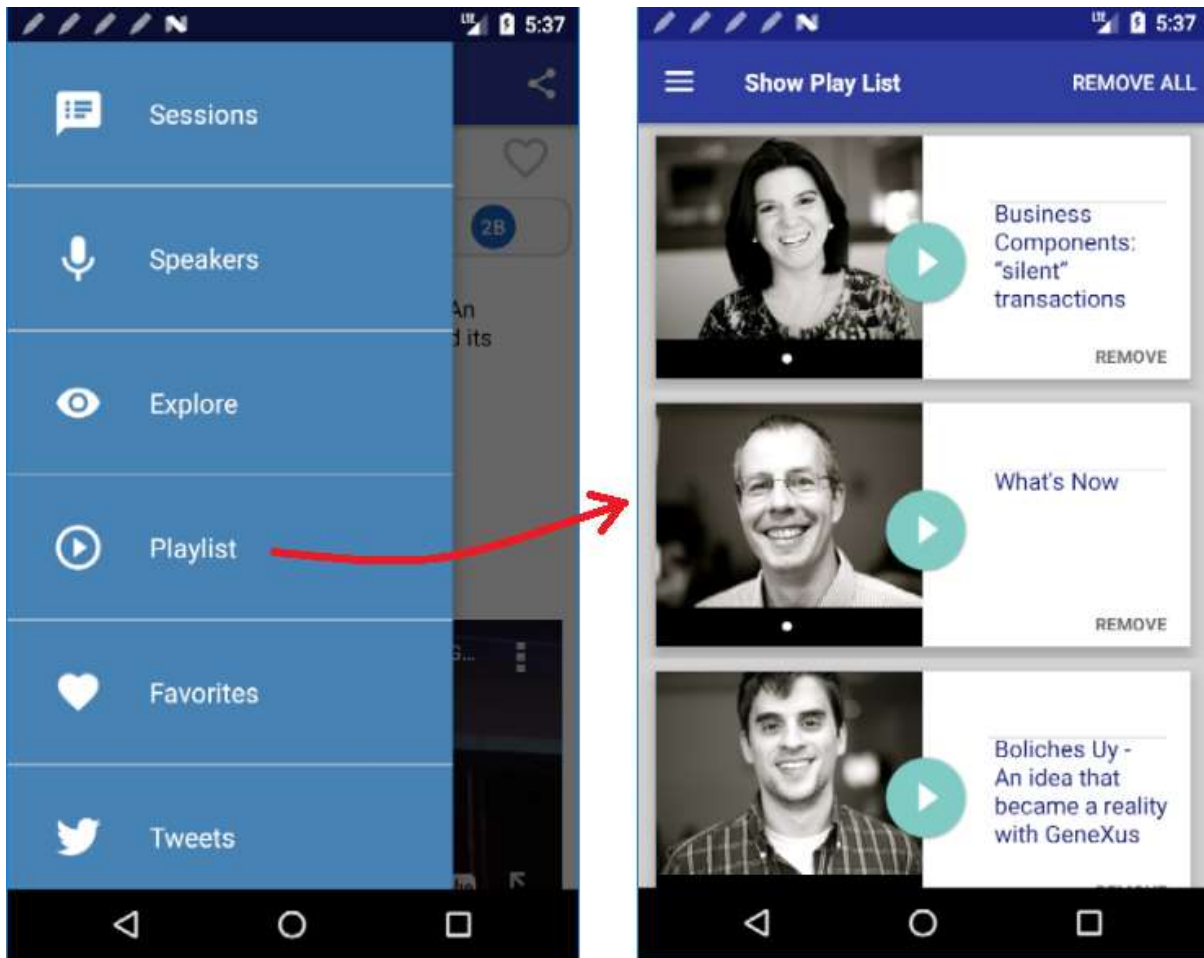
**Nota al margen:** observe que se debería cargar también un video en la pantalla. Si no se carga es que le faltó agregar al objeto main (en nuestro caso el menú EventDay) la clave de Android Google Services API Key.

La Playlist es por dispositivo. Observe la transacción que almacena esa información:

Name	Type
Playlist	Playlist
PlaylistId	VarChar(128)
SessionId	Id
PlaylistPlay	Boolean
SessionName	Name
SessionAudio	Url, GeneXus

El atributo PlaylistId corresponderá al ClientInformation.Id.

Agregue algunas conferencias a la playlist del dispositivo marcándolas desde el Detail de cada una. Luego acceda a la opción PlayList del menú:

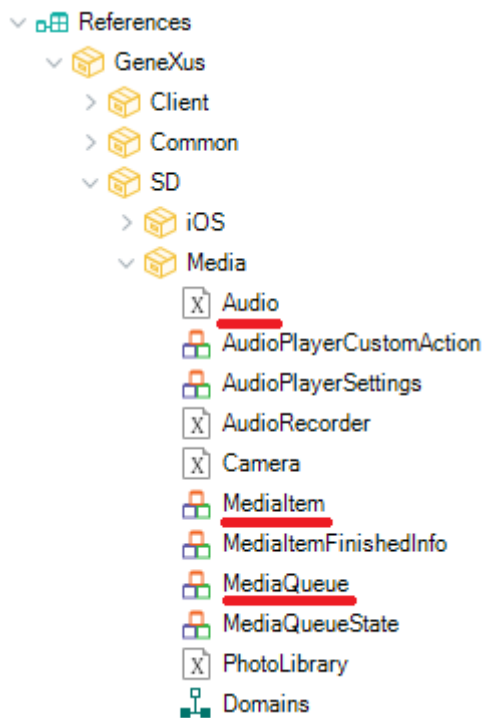


Presione el botón Play de alguna de ellas y verá cómo comienza a reproducirse el audio. Esta implementación corresponde a la versión anterior de GeneXus, cuando la api Audio no contaba con las funcionalidades específicas para soportar colas de reproducción.

Abra el panel ShowPlayList donde se ha implementado lo anterior a través de un grid de tabla base PlayList que carga todas las sessions que corresponden a PlaylistId = ClientInformation.Id.

**Nota:** si el Build arroja un error indicando que se ha excedido en la cantidad de métodos, modifique la propiedad Multidex Build del objeto main (en nuestro caso el menú EventDay) pasándola a True.

Observe cómo se implementó la reproducción y el stop utilizando la api Audio. Ábrala. Y note que ahora se encuentran los nuevos métodos para colas, junto con SDTs:



El MediaQueue será una colección de Medialtem. Esos dos SDTs son los que le permitirán cargar los datos de la cola de reproducción.

Querremos implementar lo mismo que hacíamos a mano en este panel, ahora a través de una cola de audios.

Para ello:

Cree un panel **AutomaticPlayList** y en su layout arrastre el nuevo control **AudioController** desde la toolbox.

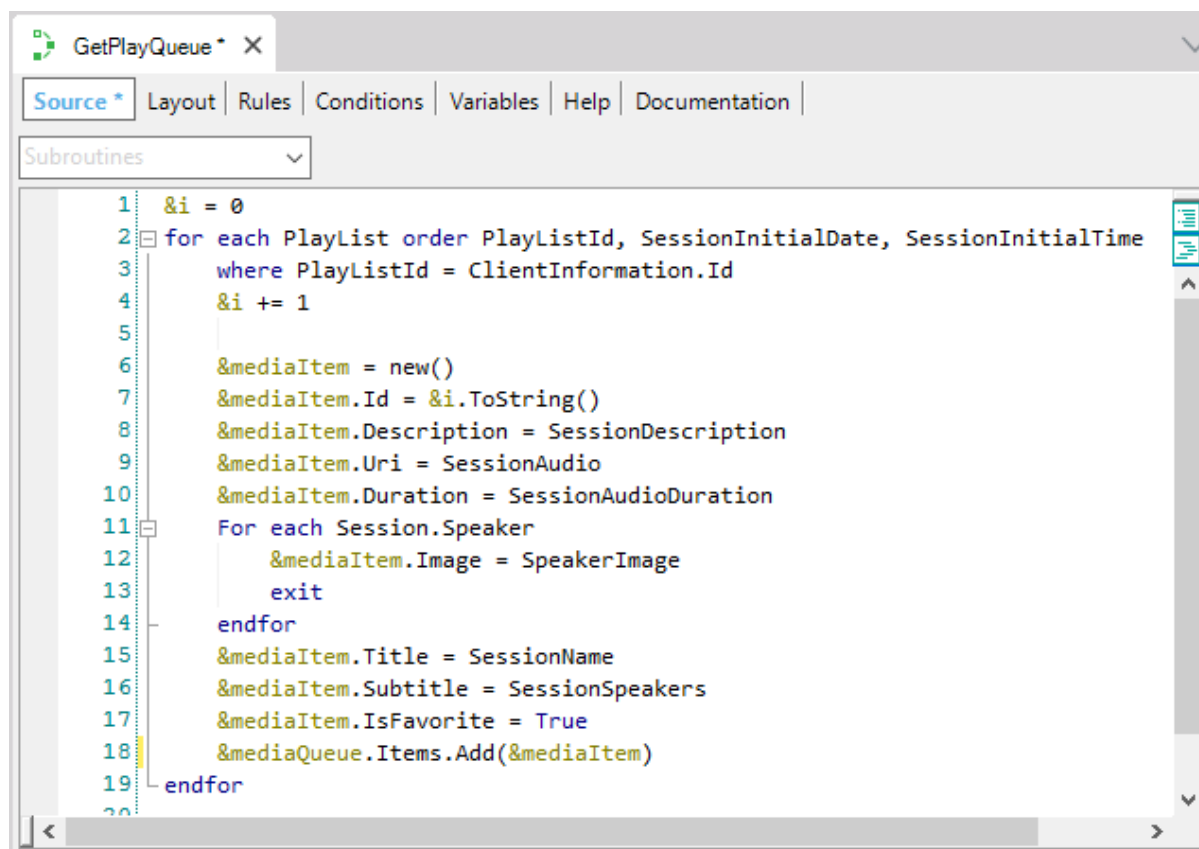
Y luego haga:

```

Event ClientStart
  Composite
    &mediaQueue = GetPlayQueue()
    Audio.SetQueue(&mediaQueue)
  endcomposite
Endevent

```

El procedimiento **GetPlayQueue** ya se le ofrece implementado en la KB. Observe que lo que hace es recorrer la tabla **PlayList** y cargar un item en la cola por session de ese **ClientInformation.Id**:



```
1 &i = 0
2 for each PlayList order PlayListId, SessionInitialDate, SessionInitialTime
3   where PlayListId = ClientInformation.Id
4   &i += 1
5
6   &mediaItem = new()
7   &mediaItem.Id = &i.ToString()
8   &mediaItem.Description = SessionDescription
9   &mediaItem.Uri = SessionAudio
10  &mediaItem.Duration = SessionAudioDuration
11  For each Session.Speaker
12    &mediaItem.Image = SpeakerImage
13    exit
14  endfor
15  &mediaItem.Title = SessionName
16  &mediaItem.Subtitle = SessionSpeakers
17  &mediaItem.IsFavorite = True
18  &mediaQueue.Items.Add(&mediaItem)
19 endfor
20
```

Puede sustituir la invocación a ShowPlayList en el Menú EventDay por una invocación a este panel AutomaticPlayList, o también podría agregar un componente en ShowPlayList al final del Layout:



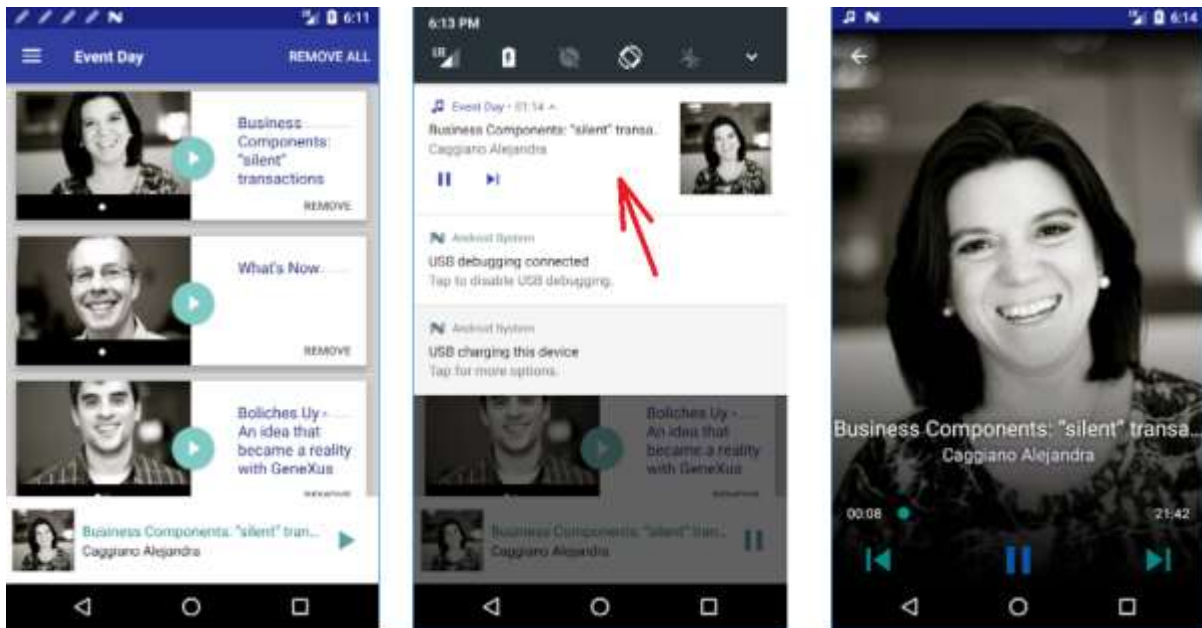
The screenshot shows the GeneXus IDE interface. On the left, the 'Layout' editor displays a 'MainTable' component. The table contains a 'Remove' button, a 'SessionName' text field, and a landscape image. Below the table, a component labeled '<Component: AutomaticPlaylist>' is visible, with a red arrow pointing to it. The 'Properties' panel on the right shows the configuration for 'Table: MainTable'.

Table: MainTable	
Control Name	MainTable
Columns Style	100%
Rows Style	100%;5dip;pd;20dip
Width	100%
Height	100%
Auto Grow	True
Appearance	
Class	TableDetail
Background	(none)
Visible	True
Invisible Mode	Keep Space
Enabled	True
Scroll Behavior	
Scroll Factor	1
Zoom Factor	0
Scroll Attachm	Parent
Form	
Form Class	Form

Y que se cargue con el objeto AutomaticPlaylist.

Ejecute y pruebe. Presione play y verá cómo se le muestra la cola en modo “mini”, e incluso puede manejarla desde otra app pues se le muestra entre las notificaciones. Y si maximiza la cola de reproducción se le abrirá a pantalla completa el audio que se está reproduciendo y se le ofrecen los botones para ir al anterior o siguiente.





¿Dónde se configuran los colores, tamaños, etc., de título, subtítulo, imagen, botones para el modo “mini” y para el modo “full”?

Abra el theme (EventGXAndroid) y observe la propiedad AudioController:

Properties	
Class: AudioController	
Mini Player	
Title Label Class	MiniTitleLabel
Subtitle Label Class	MiniSubtitleLabel
Image Class	MiniImage
Play/Pause Button Class	MiniPlayPauseButton
Full Screen Player	
Title Label Class	FullTitleLabel
Subtitle Label Class	FullSubtitleLabel
Image Class	FullImage
Play/Pause Button Class	FullPlayPauseButton
Other Buttons Class	FullButton

Luego observe las propiedades de esas subclases.

## SD – PULL TO REFRESH

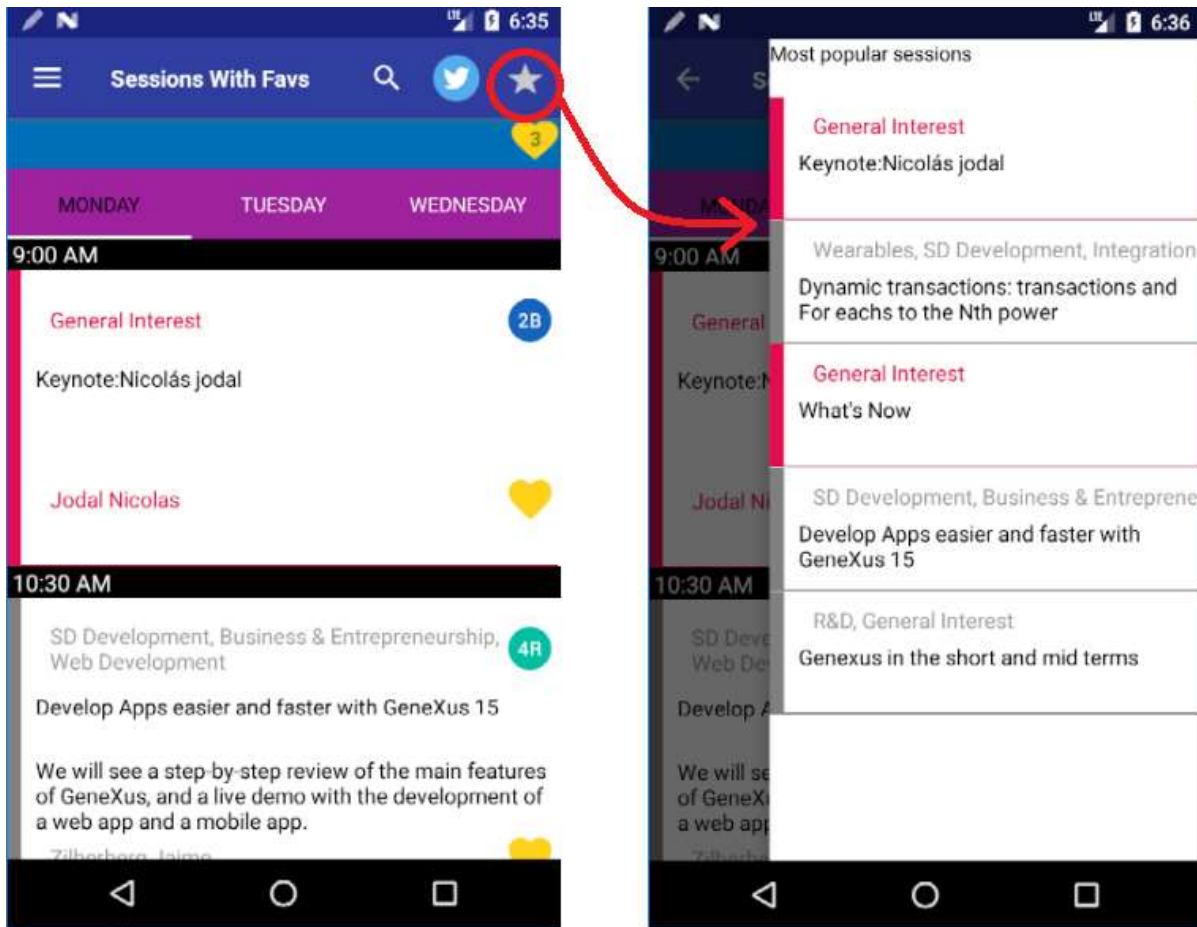
Si elige ver los tweets en el menú, encontrará la funcionalidad de Pull to refresh en acción.



Abra el panel EventTweets y posicionándose en el grid. Vea la propiedad Pull To Refresh en True. Observe también la propiedad Inverse Loading.

## SD – DESPLEGAR PANTALLA A LA DERECHA: CALLOPTIONS Y NAVIGATION EO

Si ahora deseamos que al presionar la imagen con la estrella en la application bar del List de Sessions, en lugar de abrirse el panel con las sessions más populares en una pantalla independiente, se abra a la derecha, ¿cómo lo realiza?



Respuesta:

Entre los eventos del WorkWithDevicesSession nodo List, agregar el ClientStart que se ejecutará una sola vez, cuando el objeto se abra, y luego, mientras se mantenga abierto no volverá a ejecutarse. Allí invocamos al panel PopularSessions en la ventana de la derecha, que al principio permanecerá oculta.

```
Event ClientStart
  Composite
    PopularSessions.CallOptions.Target = !"Right"
    PopularSessions()
  endcomposite
Endevent
```








Y luego modificamos el evento 'Fav' asociado al botón con la estrella, para que al presionarlo se muestre esa pantalla.

```
Event 'Fav'
  Navigation.ShowTarget(!"right")
Endevent
```

## REPORTING (OPCIONAL)

En esta parte del práctico vamos a hacer un recorrido por los reportes que ya están implementados en la KB. Consulte al instructor sobre si realizar este recorrido.

Todos los queries implementados tienen como objetivo formar parte de un dashboard de controles con indicadores de análisis de los “likes” que los diferentes usuarios van dando a las conferencias del evento.

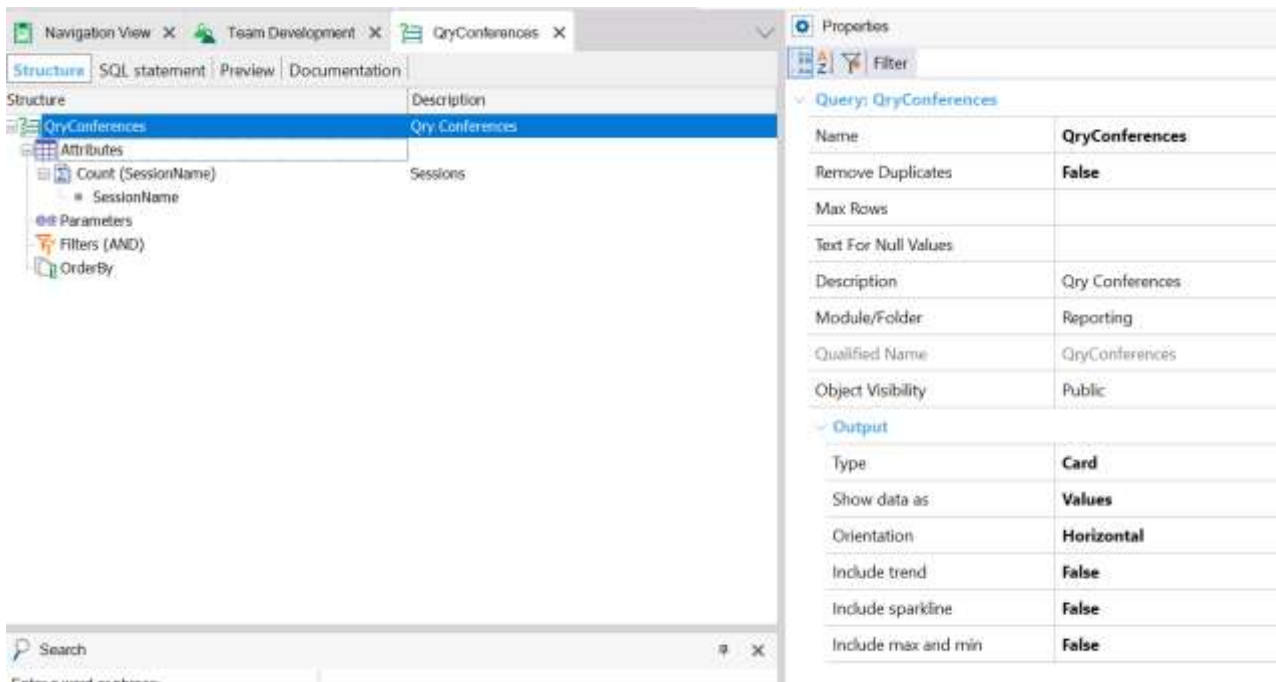
- ▼  Reporting
  -  QryConferences
  -  QryLikes
  -  QryLikesChart
  -  QryLikesPivot
  -  QryLikesTimeline
  -  Reports

### CARD: QRY CONFERENCES

La consulta ‘Conferencias’ muestra en una tarjeta la cantidad de conferencias definidas en el evento.

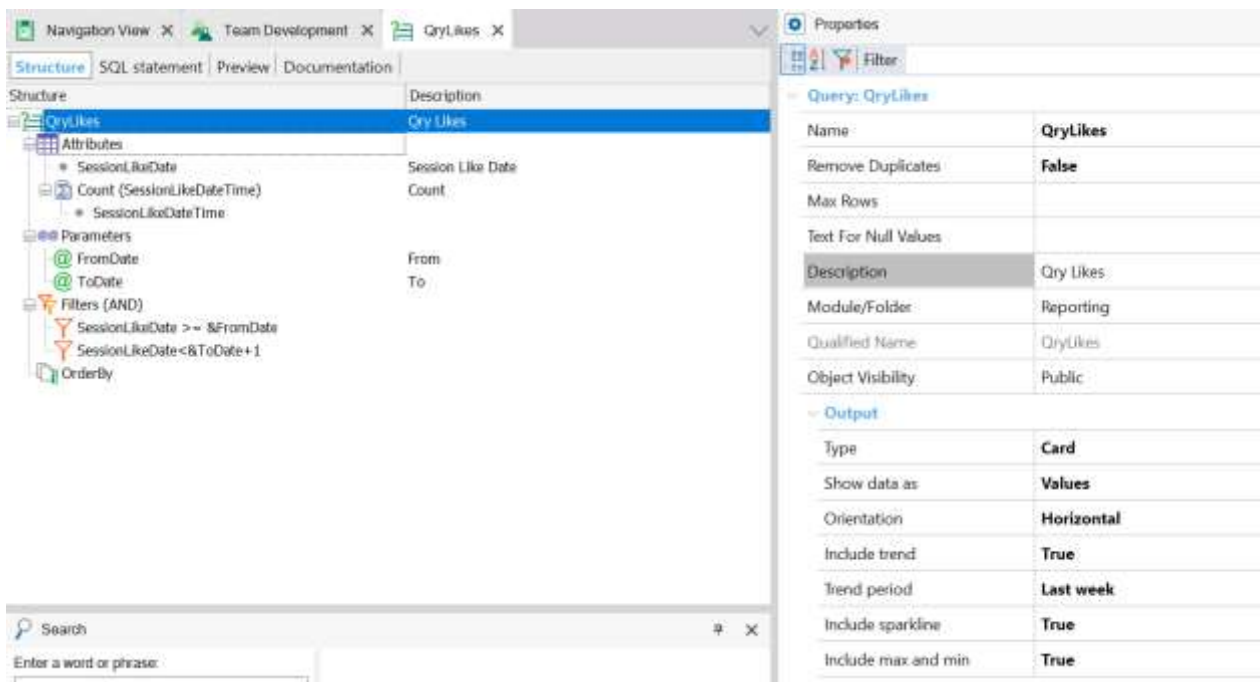
Este es un dato que –en nuestro caso– no varía con el tiempo por lo cual no se incluye la tendencia (Include Trend=False) ni la información de variabilidad (sparkline, max and min).

Es la consulta más básica que sirve para mostrar información sumariada (podría ser total de likes, etc).

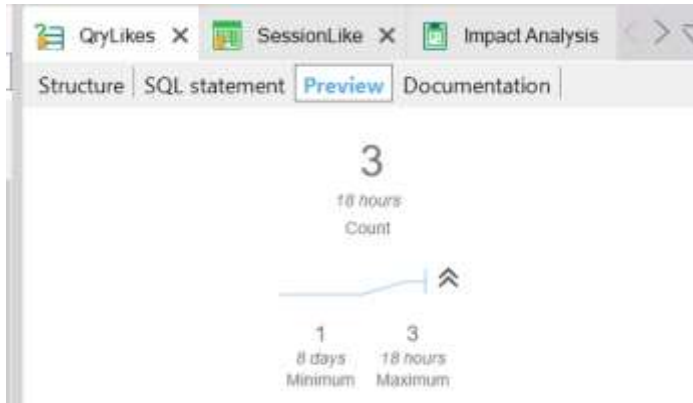


## CARD: QRYLIKES

Muestra una tarjeta con la cantidad de likes en total que ha habido para todas las conferencias, y como es un dato que puede ir variando con el tiempo muestra la tendencia de la última semana, incluyendo información de variabilidad (sparkline, max, min).



Si vamos al tab Preview podemos ver cómo se va a ver la tarjeta en nuestra aplicación. La visualización puede cambiarse vía los Themes, aunque en este caso estamos usando la visualización por defecto



The screenshot shows the 'Class: QueryViewer' configuration panel. The panel is expanded to show the 'Card specific' section, which includes fields for 'Container class', 'Element', 'Value class', 'Time class', and 'Title class'. The 'Trend' section includes fields for 'Trend image class', 'Upward trend image', 'Downward trend image', and 'Sideward trend image', all set to '(none)'. The 'Max and min' section includes fields for 'Value class', 'Time class', and 'Title class'.

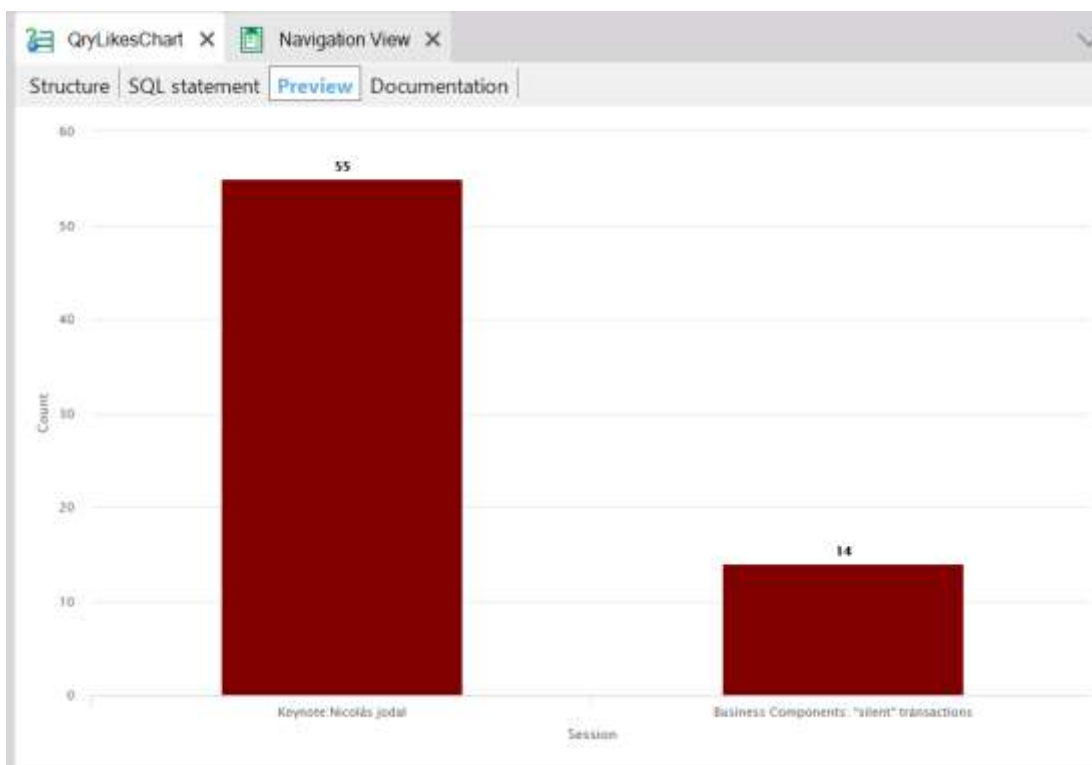
Class: QueryViewer	
Card specific	
Container class	
Element	
Value class	
Time class	
Title class	
Trend	
Trend image class	
Upward trend image	(none)
Downward trend image	(none)
Sideward trend image	(none)
Max and min	
Value class	
Time class	
Title class	

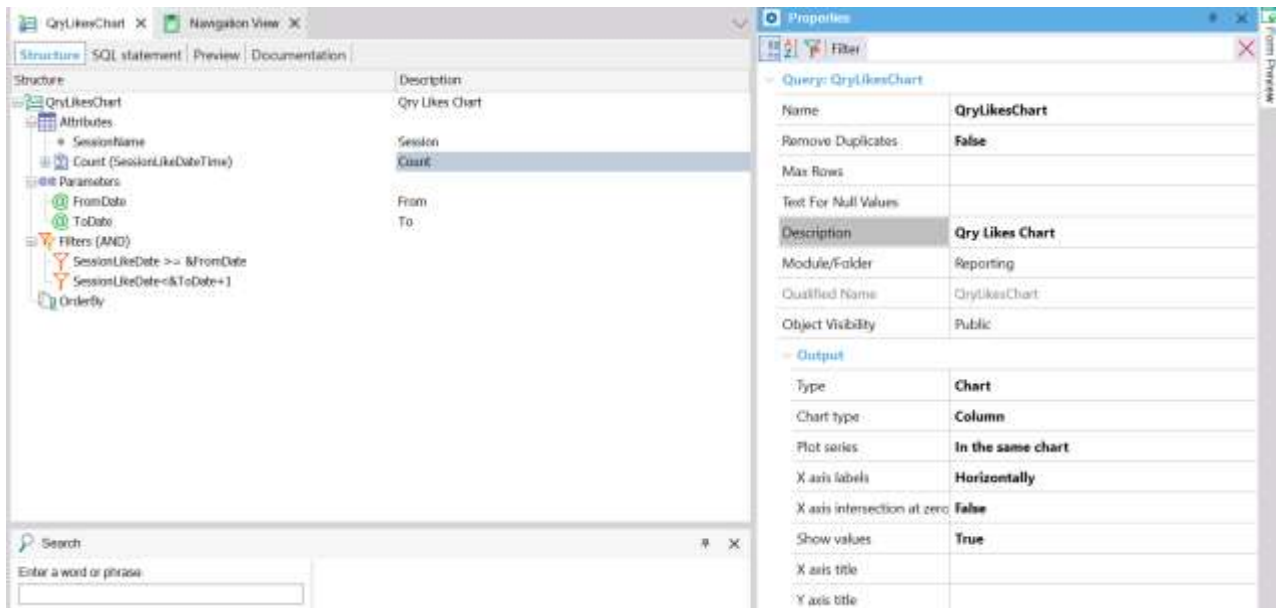
En la pestaña SQL Statement se puede ver la consulta que se va a ejecutar para obtener los datos. En caso de tener temas de performance, es importante analizar la consulta para asegurarse que esté haciendo un acceso razonable a la base.

```
Navigation View x Team Development x QryLikes x Carmine x
Structure SQL statement Preview Documentation
SELECT F1,
       Count(iF2) AS F2
FROM   (SELECT SessionLikeDateTime AS F1,
              SessionLikeDateTime AS iF2
        FROM   SessionLike
        WHERE  (SessionLikeDateTime >= &FromDate)
              AND (SessionLikeDateTime < (&ToDate + 1)))
GROUP BY F1
```

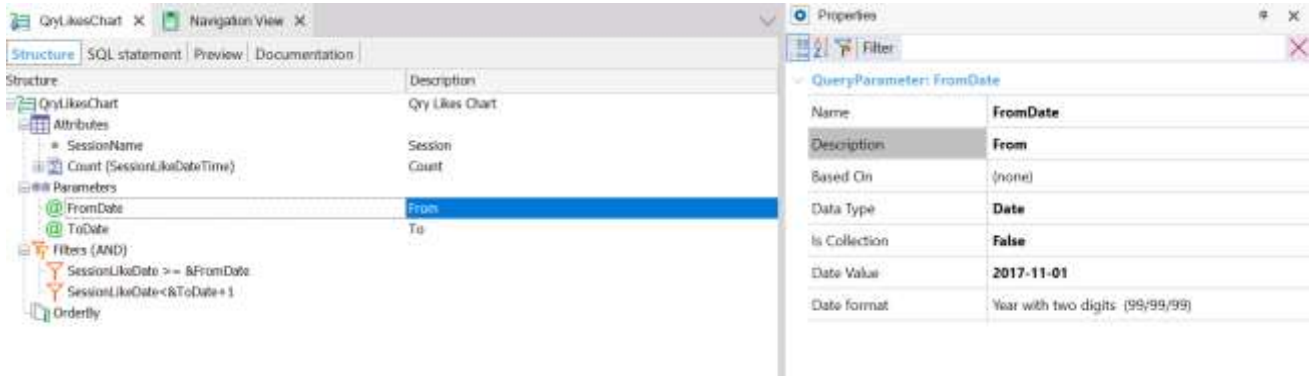
## CHART: QRYLIKESCHART

Muestra una tabla con los likes por conferencia en un periodo de tiempo:





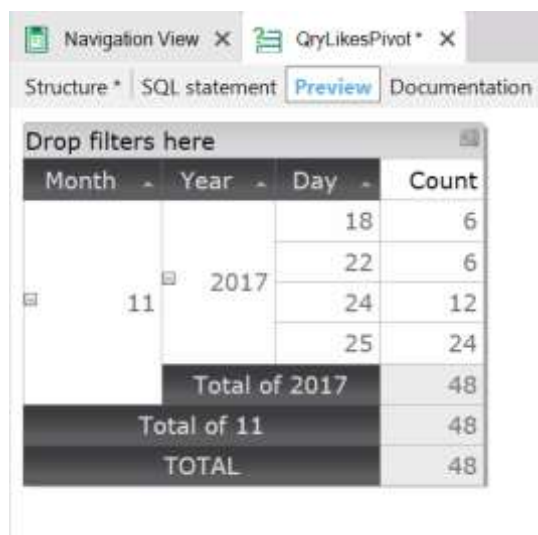
En la vista previa, los valores de los parámetros se pueden especificar al definir los mismos, por ejemplo, al realizar el 'preview' vamos a ver likes que sucedieron después del '2017-11-01'





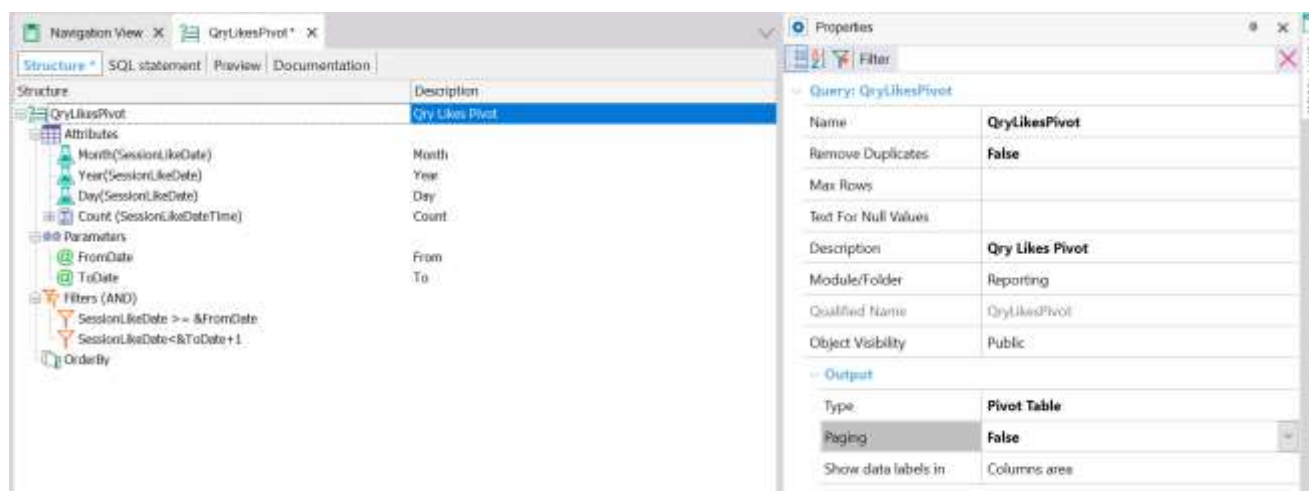
## PIVOT: QRYLIKESPIVOT

Es una consulta que muestra una tabla dinámica de likes por periodo de tiempo.



Drop filters here

Month	Year	Day	Count
11	2017	18	6
		22	6
		24	12
		25	24
Total of 2017			48
Total of 11			48
TOTAL			48



Structure

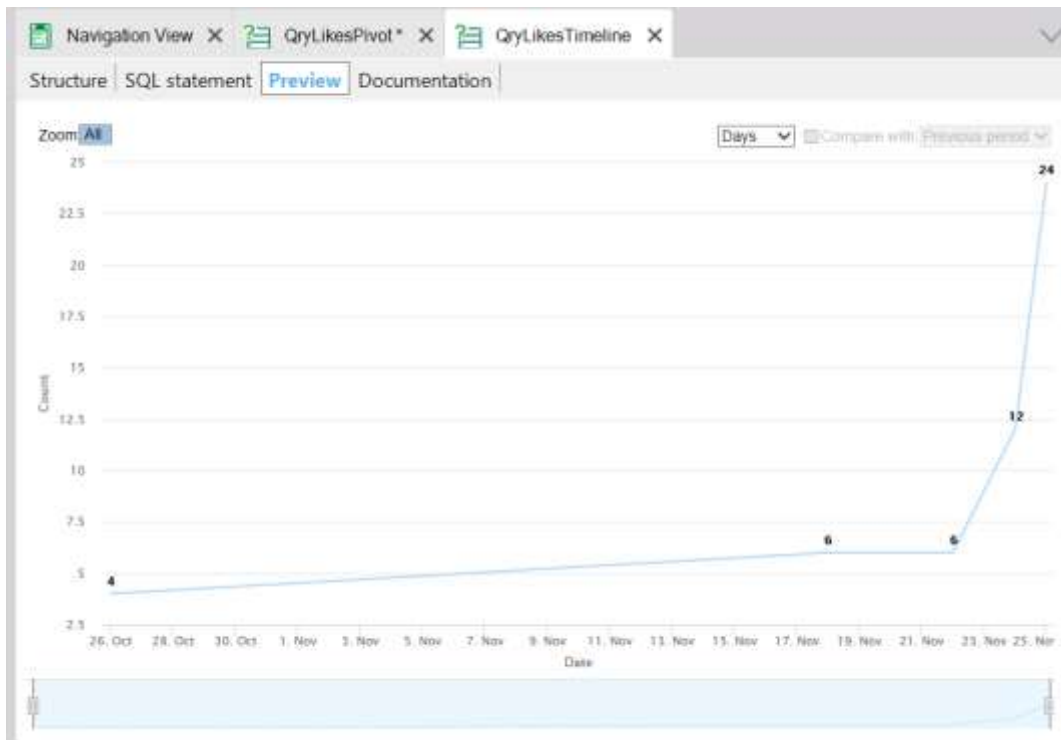
Structure	Description
QryLikesPivot	Qry Likes Pivot
Attributes	
Month(SessionLikeDate)	Month
Year(SessionLikeDate)	Year
Day(SessionLikeDate)	Day
Count (SessionLikeDateTime)	Count
Parameters	
FromDate	From
ToDate	To
Filters (AND)	
SessionLikeDate >= &FromDate	
SessionLikeDate < &ToDate + 1	
OrderBy	

Properties

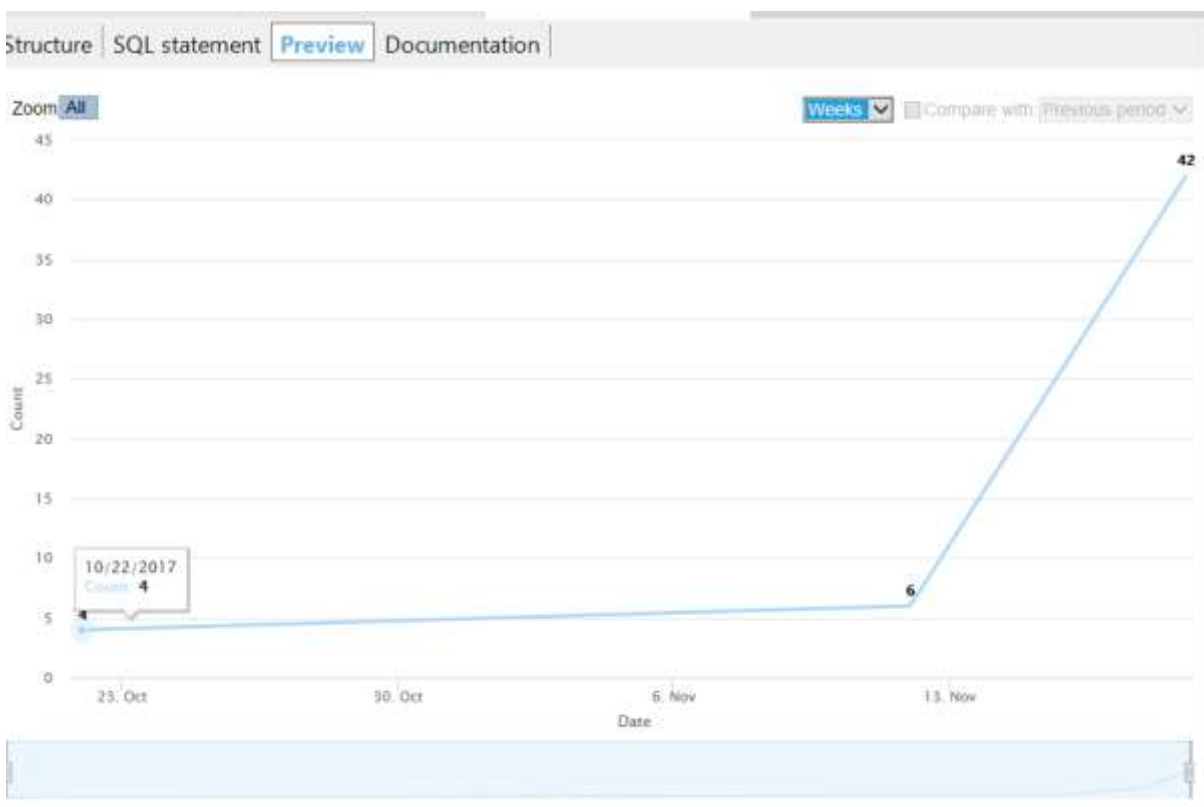
Name	QryLikesPivot
Remove Duplicates	False
Max Rows	
Text For Null Values	
Description	Qry Likes Pivot
Module/Folder	Reporting
Qualified Name	QryLikesPivot
Object Visibility	Public
Output	
Type	Pivot Table
Paging	False
Show data labels in	Columns area

## TIMELINE: QRYLIKESTIMELINE

Muestra la evolución de cantidad de likes a lo largo del tiempo. El timeline tiene funcionalidades pre-definidas para comparar una tendencia con otro periodo de tiempo (i.e evento 2017 vs evento 2016, etc.), agrupar:



Poder visualizar la información en varios niveles: días, semanas, meses:



Y también se puede achicar/agrandar el periodo haciendo clic en la barra inferior.

La funcionalidad del preview es la misma que la que tendrán los controles en la aplicación web.