



Smart Devices & GAM

Práctico



Copyright © Artech Consultores S. R. L. 1988-2014.

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

Marcas Registradas

Artech y GeneXus son marcas registradas de Artech Consultores S.R.L. Todas las demás marcas mencionadas en este documento son propiedad de sus respectivos dueños.

1.	Primeros pasos	3
2.	Aplicación Web	4
3.	Aplicando el Pattern Work With for Smart Devices.....	6
4.	Entidades interrelacionadas	16
5.	Personalizando el Pattern “Work With for Smart Devices” “	20
	Personalizando el List	20
	Personalizando el Detail	22
	Themes.....	30
	Transitions	32
	Agregando Órdenes y Filtros.....	34
	Agregando Acciones	36
6.	Paneles para pedir datos al usuario	41
7.	Ejecución en el dispositivo [opcional].....	44
8.	Activación del GAM.....	46
9.	Uso de la API del GAM	51
10.	Links de interés	53

1. Primeros pasos

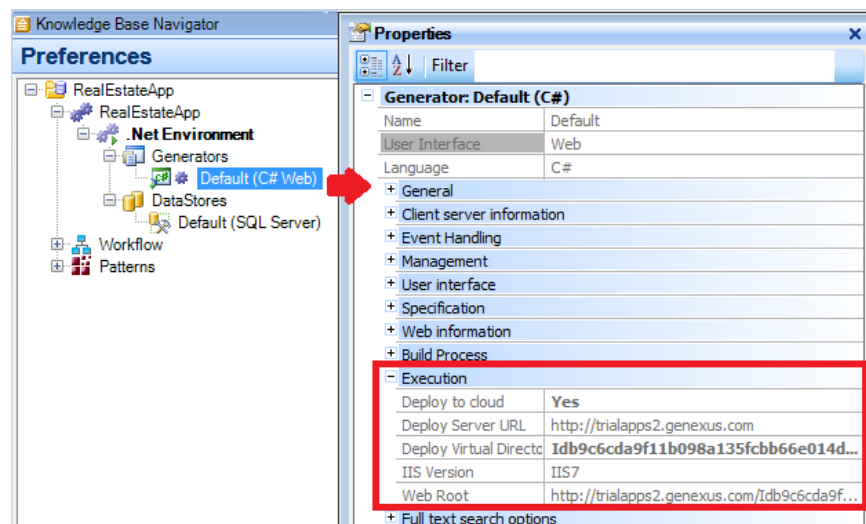
A lo largo de este práctico desarrollaremos una aplicación para una Inmobiliaria (en inglés: Real Estate), haciendo foco en la parte para Smart Devices. Prototiparemos por simplicidad en Android, dado que contamos con un emulador. Si tiene algún dispositivo (Android, BlackBerry o iOS) podrá probar la aplicación en él.

En nuestro sitio de capacitación encontrará videos que tratan todos estos temas:

- Básico (de X Ev2): <http://training.genexus.com/actualizacion-de-gx-ev1-a-ev-2/cursos-actualizacion-de-gx-ev-1-a-gx-ev-2-es?es>
- Avanzado de Smart Devices: <http://training.genexus.com/GeneXus-para-Smart-Devices/GeneXus-para-Smart-Devices?es>

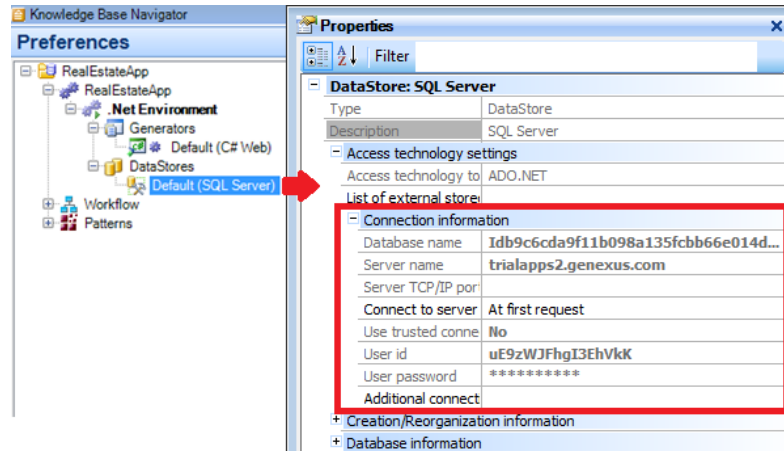
Si utiliza para este práctico la versión Trial de GeneXus X Ev2, **no** podrá elegir prototipar en otro environment distinto de .Net, ni hacerlo localmente (automáticamente se prototipará en la nube). Si cuenta con una versión full de GeneXus, podrá probar en otros ambientes así como prototipar local.

- 1) Cree una nueva KB (si usa la Trial se creará con Environment .Net, con generador Default C# Web).
- 2) Observe las propiedades del generador **Default(C# Web)**, en particular las del grupo Execution:



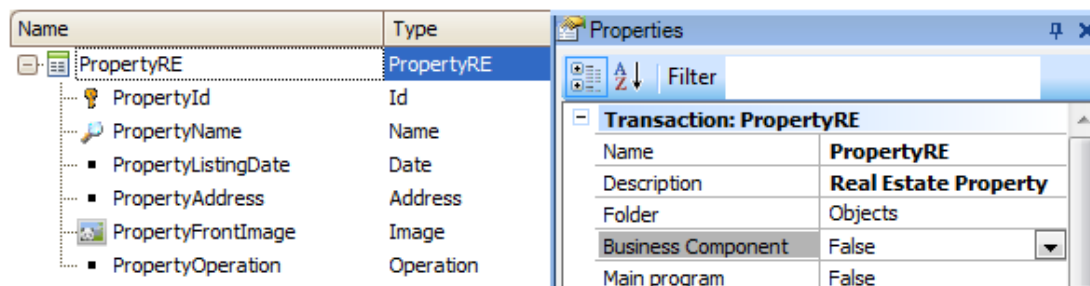
Se hará **Deploy to cloud**. Observe la URL del servidor. Si estuviera desarrollando con la versión full, la URL que vería sería: <http://apps3.genexusx.com> si prototipa en C# (.Net) o <http://apps2.genexusx.com> si lo hace en java o ruby

- 3) Observe las propiedades del Data Store Default.



2. Aplicación Web

- Cree la transacción PropertyRE (le pusimos este nombre en vez de “Property” para que no exista conflicto con el tipo de datos extendido Property; aprovechamos y también aclaramos en la Description que se trata de propiedades inmobiliarias: Real Estate Property).



Cree dominio **Id** Autonumber, **Name** = Character(30), **Operation** = Character(1), enumerado, que toma 2 valores: Rent o Sale.



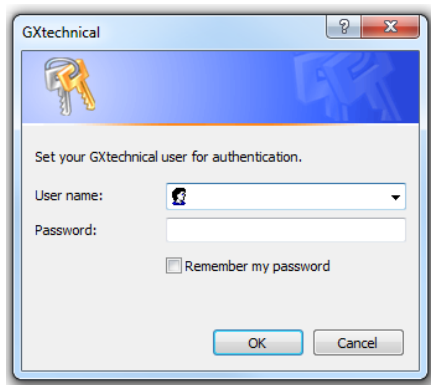
Observe los dominios predefinidos en GeneXus. Varios corresponden a dominios semánticos. ¿Qué significa esto?

Ingresa regla “Default” para PropertyListingDate, y regla “Error” si se deja la dirección vacía. Luego constatará que se disparan también en el Smart Device.

- Cree la transacción Neighborhood

Name	Type
Neighborhood	Neighborhood
NeighborhoodId	Id
NeighborhoodName	Name

- Haga F5. Se abrirá una ventana pidiéndole que se loguee con su usuario de GXTechnical. Si no tiene un usuario creado, créelo a través de gxtechnical.com/register

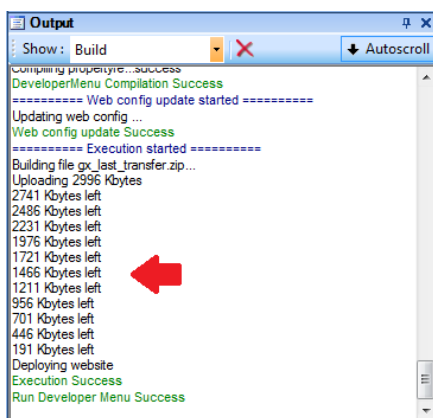


- Luego de logueado, se desplegará el reporte de análisis de impacto. **Ejecute la reorganización** para crear la base de datos y sus tablas (en la nube).

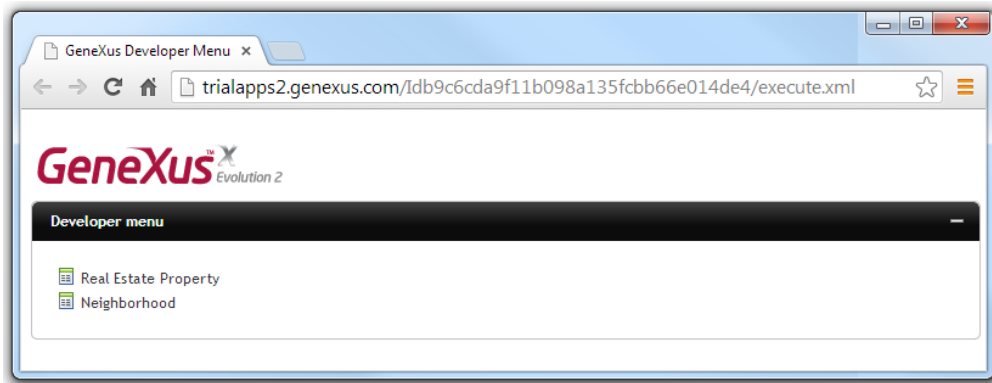


Observe que se crean automáticamente las listas de selección para los objetos web (Gx0010 y Gx0020)

Asegúrese de tener la ventana de output abierta. En un momento dado verá que va informando la cantidad de kbytes que se van subiendo a la nube.



- Al finalizar se abrirá el Developer Menu en el navegador, con los links para ingresar propiedades inmobiliarias y vecindarios. **Ingrese** algunos. Observe el look & feel (se está utilizando HTML 5).



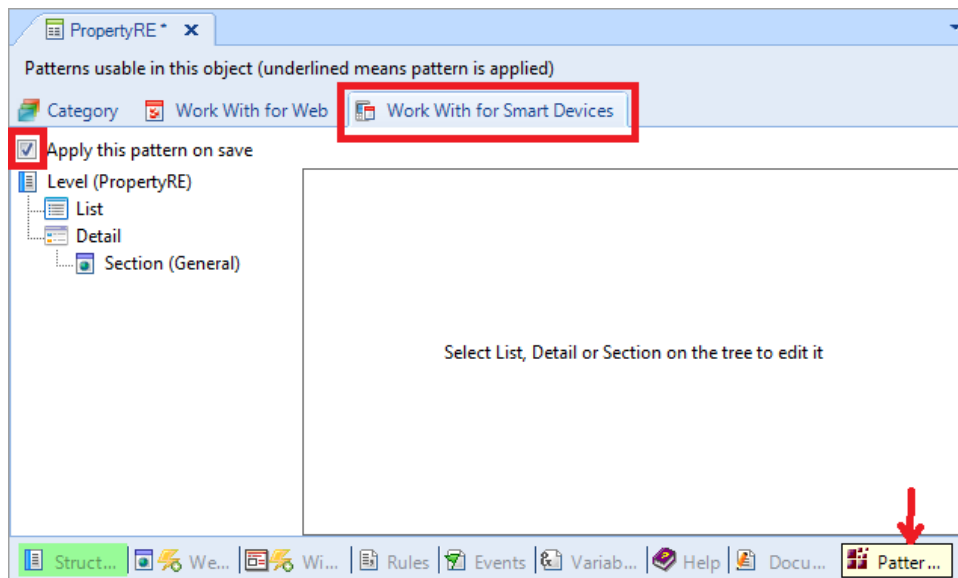
3. Aplicando el Pattern Work With for Smart Devices

- Aplique el pattern “Work With for Smart Devices” a la transacción PropertyRE.



Para ello:

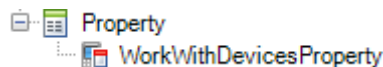
- Sección Patterns de la transacción. Elegir el patrón apropiado y “Apply this pattern on save”:



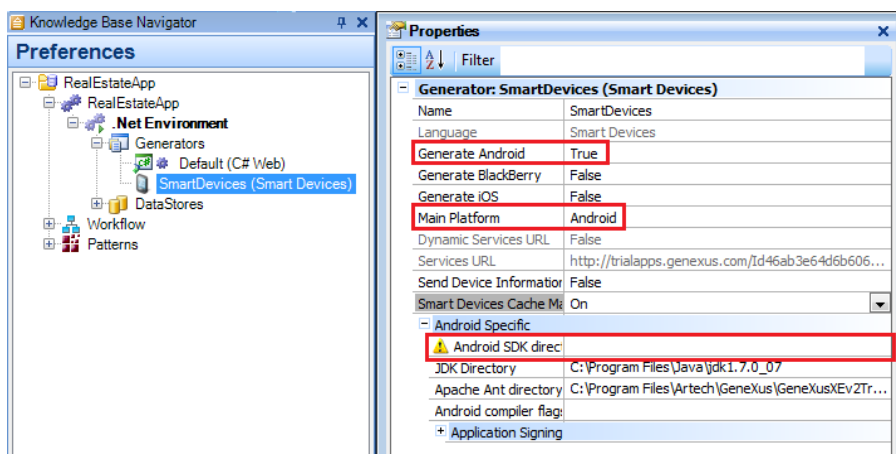
- Familiarícese con los nodos **List**, **Detail** y **Section(General)** de la instancia, viendo en cada caso la pantalla que aparece.
- Grabe y observe qué pasa con los siguientes elementos:
 - Propiedades de la transacción:

- **Business Component** = True
- **Expose as Web Service** = True
- **Web Services Protocol** = ReST Protocol

- Folder View / Property: aparece subnodo:



- Generador para Smart Devices como generador secundario



Si no aparece el path del directorio del SDK de Android, especifíquelo (suele ser: C:\Program Files\Android\android-sdk, o C:\Android\android-sdk)

- Aplique el pattern también a la transacción Neighborhood
- ¿Qué falta para poder probar la aplicación?



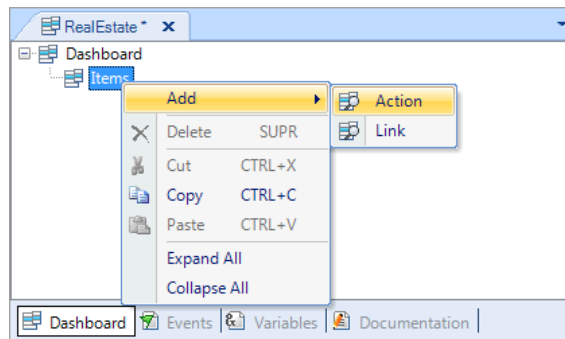
Si hace ahora **F5**, como no tiene ningún objeto main para la parte de Smart Devices, no tendrá qué ejecutar. Por tanto, antes de hacer F5...

Crear un **Dashboard** que ofrezca dos opciones al usuario: trabajar con el listado de propiedades inmobiliarias o con el listado de vecindarios.

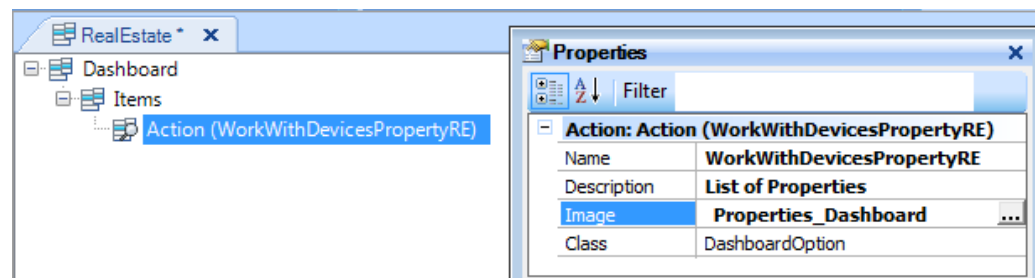


Para ello:

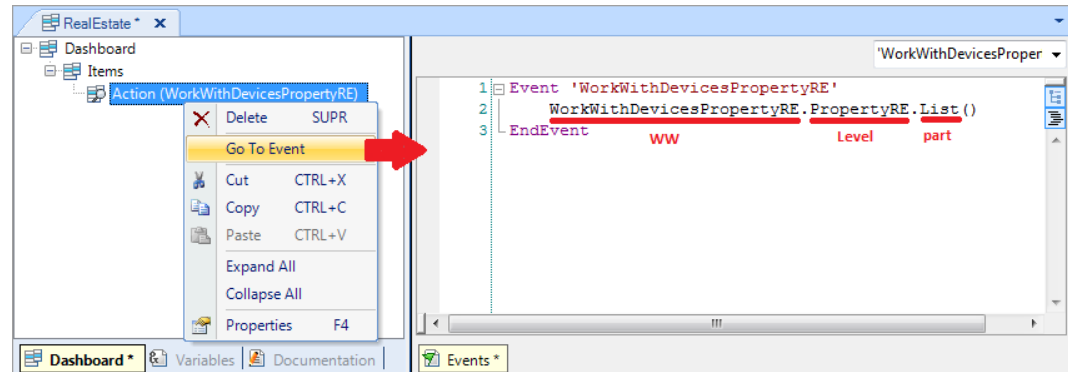
Agregar un ítem para invocar al “trabajar con” **Property**:



- Asócie una imagen a la opción (puede utilizar la del folder Images/Icons que los instructores le proveerán):



- Observe el evento asociado a esta opción:



Observe que aquí está invocando al Work With de Propiedades recién creado. ¿A qué parte del mismo? Al **List**.

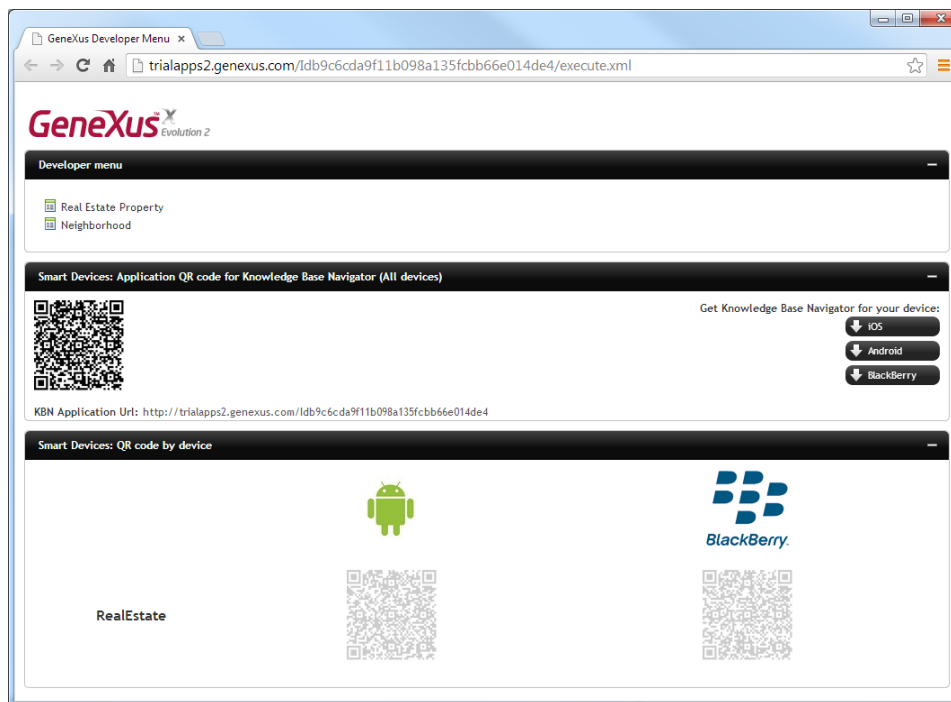
- De la misma manera, agregue un ítem para invocar al **WorkWithDevicesNeighborhood**.

- Grabe y observe que la propiedad **Main Program** está en "True". Ahora puede hacer **F5** para generar y ejecutar la aplicación en el emulador.

Notas:

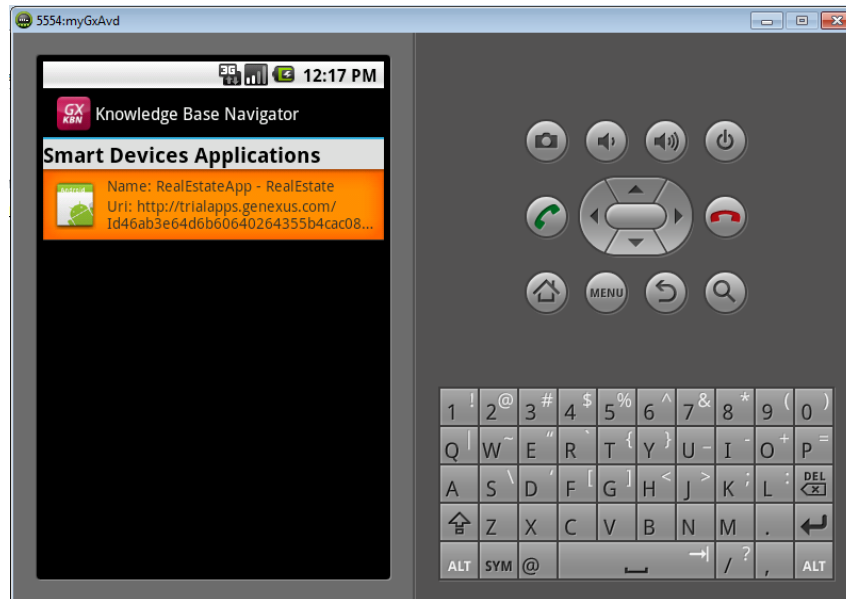
- Dado que no hemos especificado ningún "Startup Object", GeneXus abrirá tanto el Developer Menu web en el navegador predeterminado, como el emulador Android, para que podamos probar tanto la aplicación Web (.Net), como la de Smart Devices (Android), respectivamente.

- En el **Developer Menu**, además de los links para ejecutar los objetos de la aplicación Web, aparecerán **QR codes**: uno que encapsula la URL para poder ejecutar desde el KBN en el dispositivo que corresponda (a la derecha aparecen links para descargar el KBN de acuerdo a cada plataforma); los otros contendrán la aplicación para Smart Devices compilada, tanto para Android como para BlackBerry. No siempre se creará el archivo compilado. Dependerá de si se tiene "Startup object" configurado y de la modalidad de ejecución que se empleó (si F5, Run o Run with this only sobre un objeto main, etc.) En nuestro caso aún **no tenemos archivo compilado**, por lo que aparecen deshabilitados.

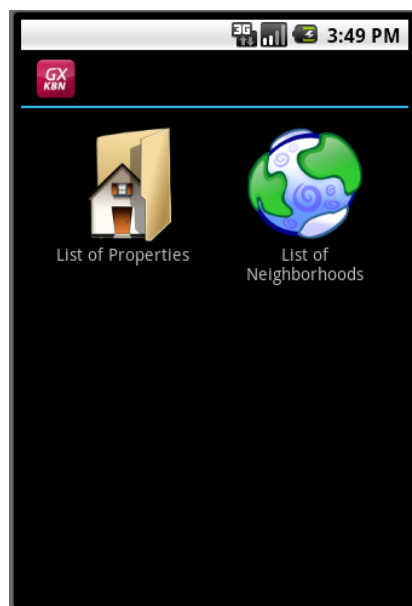


Vea más sobre esto en <http://wiki.gxtechnical.com/commwiki/servlet/hwikibypageid?18260>

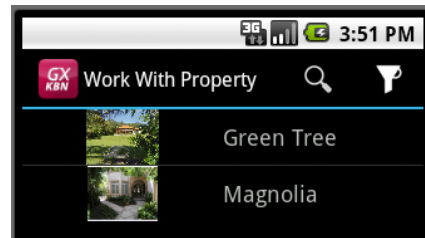
En nuestro caso el emulador está abriendo el KBN (Knowledge Base Navigator) que muestra la **URL** para ejecutar el **dashboard** de la aplicación (único objeto main que tenemos hasta el momento). Recuerde todo lo estudiado sobre la arquitectura. Aquí no estará ejecutando un compilado, sino un intérprete (el KBN)



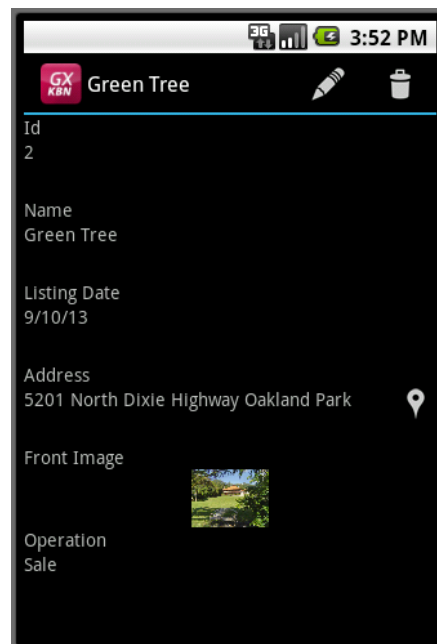
- Al hacer *tap* sobre la URL, se muestra el layout correspondiente al Dashboard:



- Haga tap sobre “**List of Properties**”
 - Observe que aparecen las mismas propiedades que ingresó antes cuando ejecutó el Developer Menu web (al final de la sección 2).
 - Observe los atributos que se están mostrando de cada propiedad. Vaya a GeneXus a buscar en el pattern el Layout de este **List**, para ver la correspondencia.



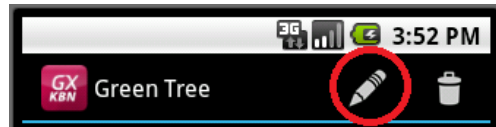
- Haga *tap* sobre una cualquiera de las propiedades listadas, y vea cómo aparece el detalle de la misma (💡 ¿este es el Layout correspondiente al modo **View** o al **Edit**?):



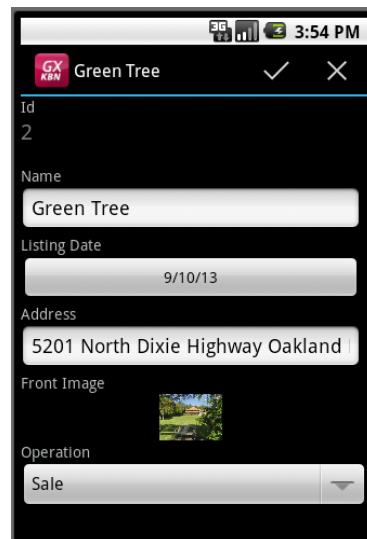
- Observe qué sucede cuando hace *tap* sobre el ícono que aparece en el campo de dirección (💡 recuerde aquello de los **dominios semánticos**: ¿con quién se está interactuando para lograr esa funcionalidad? Recuerde uno de los pilares: la integración con funcionalidades nativas del dispositivo)
- Vuelva con el botón de back:



- **Edite** la propiedad inmobiliaria que está visualizando ( para ello observe el ícono que aparece arriba a la derecha)



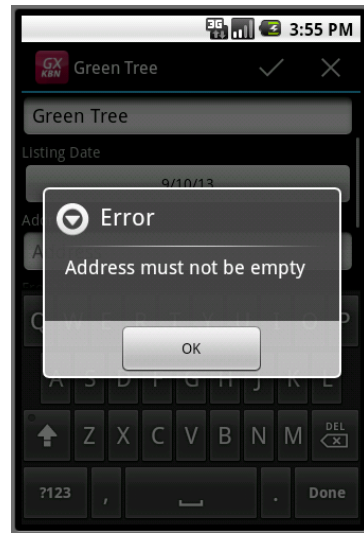
Al hacerlo ¿qué Layout se le estará desplegando para esta **sección general** del detalle? ¿El del modo **View** o **Edit**?



- **Elimine** el contenido del campo de **dirección** y confirme.



¿Le permitió grabar los cambios? Debería obtener una pantalla como la que sigue:



- Cancele.
- Vuelva a intentar modificar la Propiedad Inmobiliaria, pero esta vez sin involucrar regla error. Por ejemplo: modifique la fecha (observe el Date Picker).

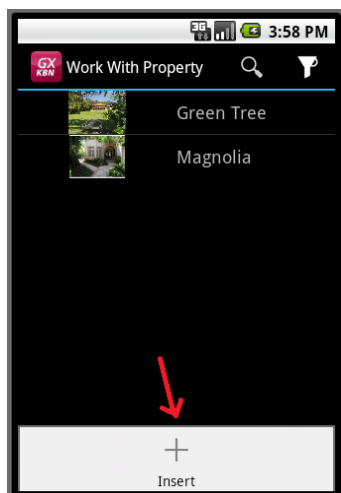


Modifique la Operación (de Rent por Sale, o viceversa) y grabe.

- Ahora vuelva a la "List of Properties" (con botón back mostrado antes), y visualice el MENU.

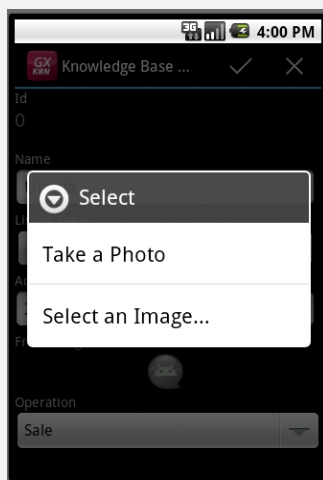


- Inserte una nueva propiedad.

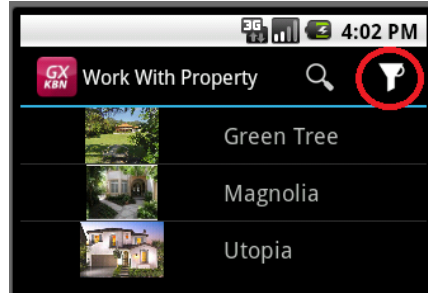


Observe que:

- El Id es autonumber, así que no tiene que ingresar valor.
- Cuando edita el campo Name, automáticamente se despliega el teclado.
- la fecha aparece con el valor que especificó en la regla Default de la transacción, pero, como siempre con esta regla, puede modificarlo.
- Al hacer *tap* sobre el campo de la imagen del frente de la propiedad, se le ofrece la posibilidad de tomar una foto con la cámara del dispositivo (en el emulador se podrá tomar una imagen ficticia) o seleccionar una de la galería de imágenes.



- Grabe y observe cómo aparece la nueva propiedad en la lista.
- **Filtre** las propiedades por Operation (Sale o Rent)



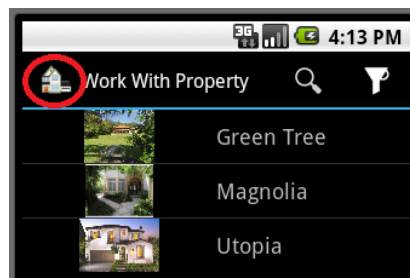
- Back, y ahora haga un **Search** por nombre. Observe que el filtro se realiza buscando indistintamente por nombre o dirección.
 - Trate de ubicar en el pattern en GeneXus el lugar donde se encuentran especificados estos **filtros** y **searchs**.
- Ahora en lugar de interpretar la aplicación a través del KBN, ejecute en el emulador el **compilado**.



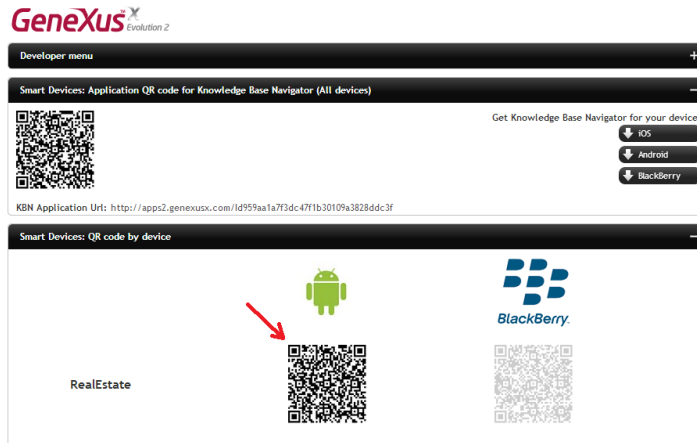
Tiene varias alternativas.

- Hacer botón derecho sobre la pestaña del Dashboard y elegir la opción “Run With this Only”. El efecto será el mismo que en el caso anterior. **Elegir esta opción**, así el F5 nos sigue generando la aplicación web y la de Smart Devices interpretada.
- Especificar el **Dashboard** como **Startup Object**. (botón derecho sobre la pestaña del mismo / Set As Startup Object o Preferences/.Net Environment, propiedad “Startup Object”). En el próximo F5 compilará sólo la aplicación que depende de este objeto, es decir, en nuestro caso, sólo la aplicación para Smart Devices cuyo menú es el dashboard. No abrirá el Developer Menu web, sino sólo el Emulador con el dashboard compilado ejecutándose.
- Existen más alternativas. Le sugerimos buscarlas en nuestro wiki en otro momento.

Nota: Si entre las propiedades del Dashboard indicó un ícono en la propiedad “Android Application Icon”, será el que saldrá en el emulador a la izquierda (en lugar del KBN)

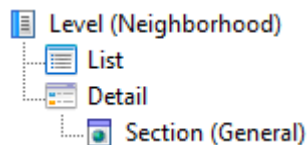


- Recuerde que al compilar, aparecerá en el Developer Menu el QR Code correspondiente. Puede obtenerlo mediante View / Show QR Codes:



4. Entidades interrelacionadas

- Observe la instancia del Pattern aplicado a Neighborhood:



- Relacione las entidades PropertyRE y Neighborhood (cada propiedad pertenece a un vecindario).

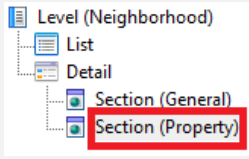


Para ello agregue en la transacción PropertyRE los atributos NeighborhoodId y NeighborhoodName (permitiendo nulos en NeighborhoodId, de lo contrario le fallará la reorganización) y grabe:

Name	Type	Description	Nullable	Formula
PropertyRE	PropertyRE	Real Estate Property		
PropertyId	Id	Property Id	No	
PropertyName	Name	Property Name	No	
PropertyListingDate	Date	Property Listing Date	No	
PropertyAddress	Address	Property Address	No	
PropertyFrontImage	Image	Property Front Image	No	
PropertyOperation	Operation	Property Operation	No	
NeighborhoodId	Id	Neighborhood Id	Yes	
NeighborhoodName	Name	Neighborhood Name		

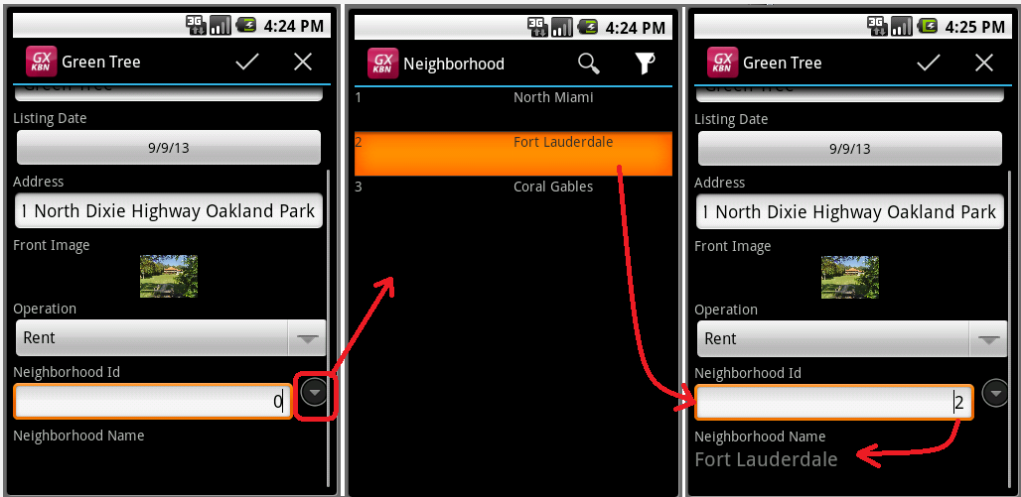
- Vuelva a observar la instancia del pattern Neighborhood ( si lo tenía abierto de antes, ciérrelo y ábralo nuevamente, para refrescarlo. Entre las propiedades de la sección, puede cambiarle Caption y Name, por “Property”; de esta manera, abreviamos el nombre y en el tab en ejecución seará ésa la caption, como verá unos pasos más adelante).

 ¿Por qué aparece ese nuevo nodo bajo Detail? ¿Qué significa?

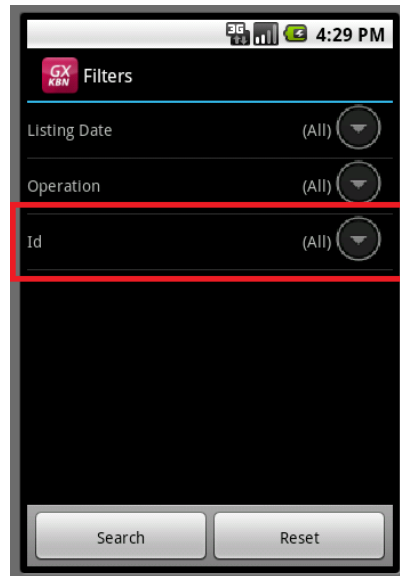


- Para corroborar, presione **F5, reorganice** y constátelo.
Pruebe en el “List of Properties”:
 - a. **Editar** una propiedad para ingresarle el vecindario.

 Observe que se le ofrece elegir el vecindario entre los existentes. ¿Qué Layout se está desplegando para ello? Se trata de una **lista de selección** creada automáticamente. Búsquela en GeneXus.

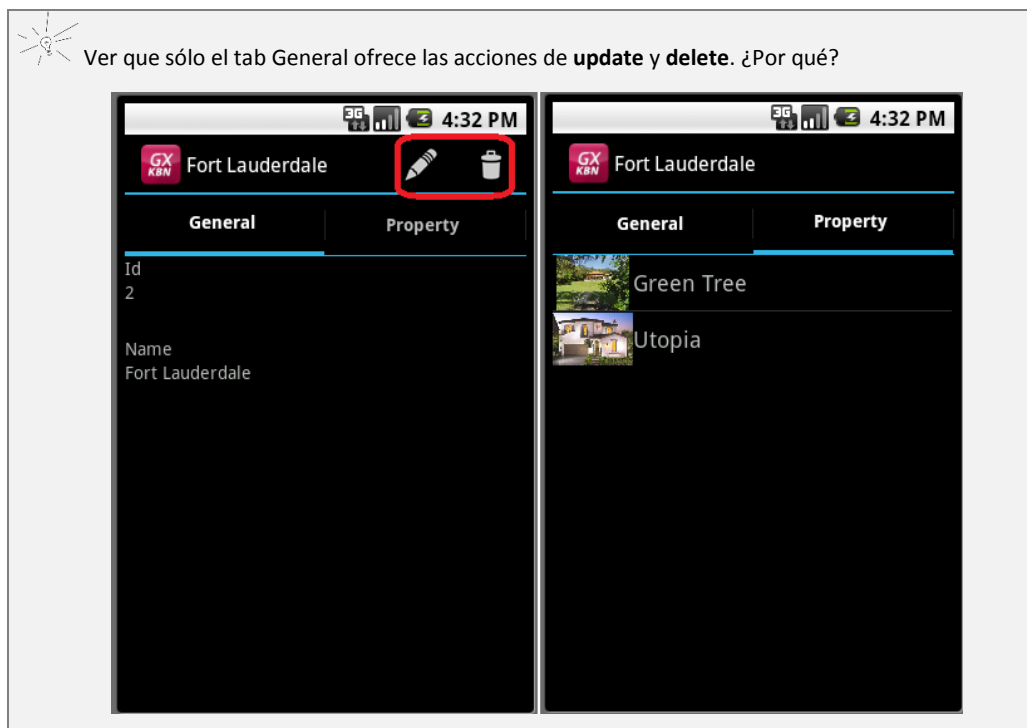


- b. Agréguele de igual modo el vecindario a las demás propiedades que tenga ingresadas.
 - c. Vuelva al “List of Properties” y observe que se agregó un filtro por vecindario. Véalo en GeneXus.



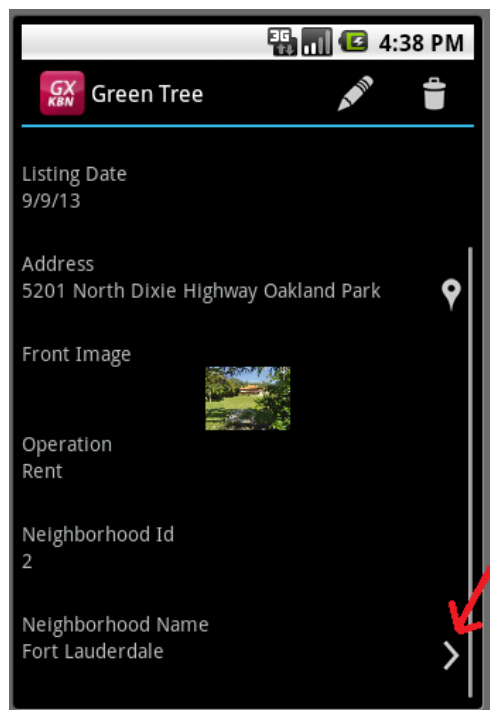
Verificar:

- d. A través del "List of Neighborhoods" ingresar a uno de los vecindarios y observar que aparece el tab **General** y el tab **Property**, con el primero activo.

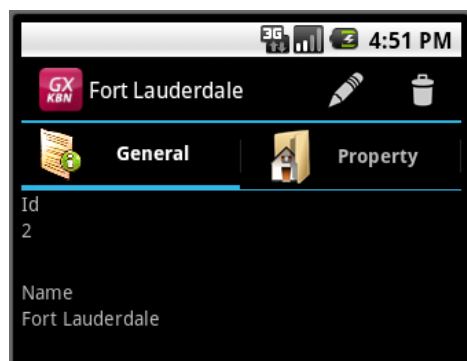


- o Hacer *tap* sobre una de las propiedades listadas en el tab Property. (¿Qué Layout se abrió?)

- En la propiedad, deténgase en el Neighborhood. Haga *tap* sobre él y observe la circularidad:



- ¿Dónde se configura que las secciones del detalle de un vecindario aparezcan como tabs?
- ( vaya al Layout del nodo **Detail** y observe el control "<All Sections Content>").
- ¿Cómo consigue que los tabs tengan imágenes? (si está ejecutando el KBN, puede que tenga que dar "back" y volver a entrar, para ver las imágenes):



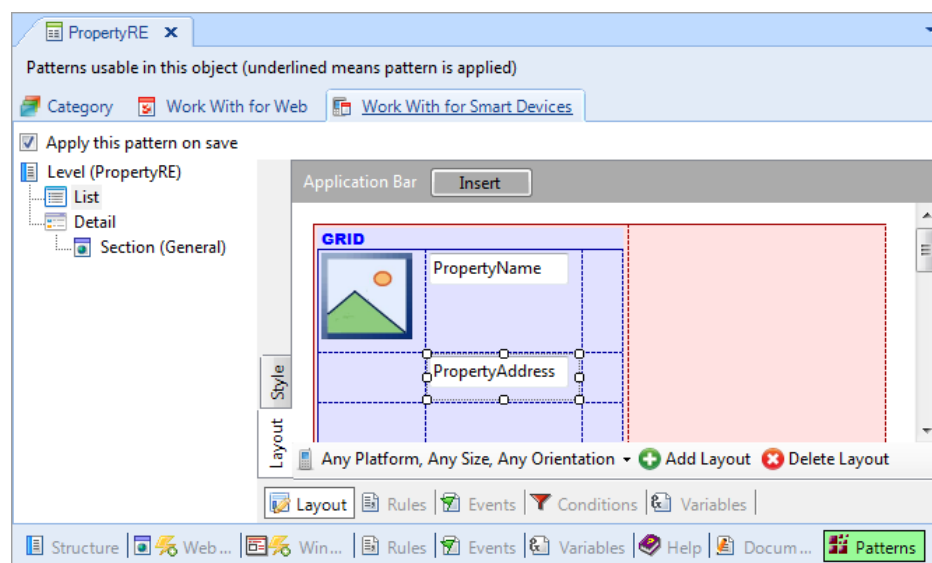
Una vez hecho este recorrido, ya hemos reconocido a un nivel básico dónde está cada cosa y cómo funciona. Estamos en condiciones de pasar a la siguiente etapa: personalizar.

5. Personalizando el Pattern “Work With for Smart Devices”

Personalizando el List

- Editar el Layout del List del **WorkWithDevicesProperty** y agregar el atributo PropertyAddress.

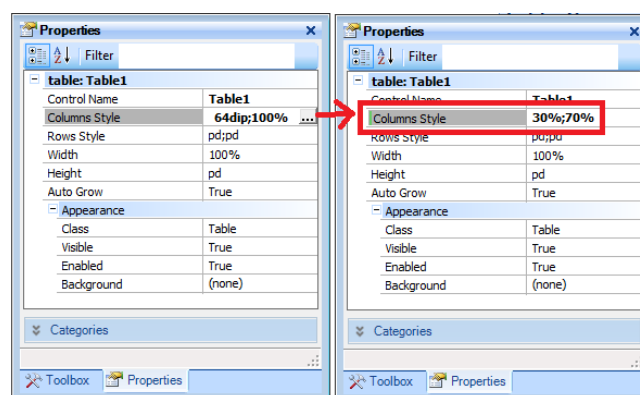
El Layout deberá quedar como se muestra (para quitarle el Label ¿qué propiedad del control atributo debe modificar?)



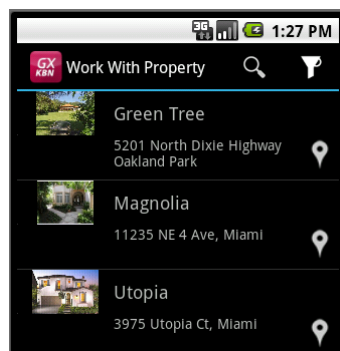
- Observe qué tamaño ocupará la primera columna y qué tamaño la segunda. Haga que la primera (la de la foto) ocupe el 30% del ancho de la pantalla.



Para ello:



- Ejecutar la aplicación presionando **F5** y verificar los cambios realizados:

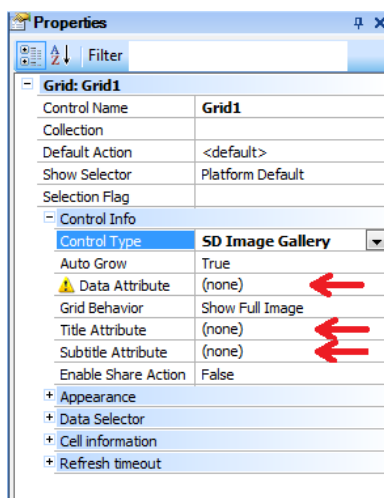


- Ahora deseamos que las propiedades se muestren, no como una lista, sino como una galería de imágenes, identificándose cada propiedad por su foto.



Para ello:

En el Grid del List, configurar la propiedad **Control Type = SD Image Gallery**, ingresando como **Data Attribute** el que contiene la foto (PropertyFrontImage) y si lo desea, en la propiedad **Title Attribute** puede especificarle el atributo PropertyName y en **Subtitle Attribute**, NeighborhoodName.



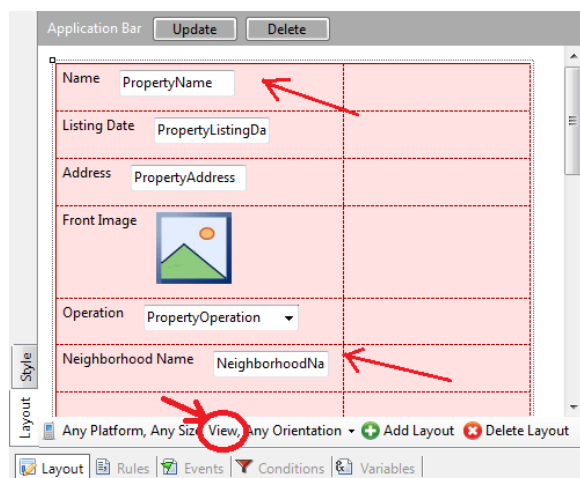
- **Ejecutar** la aplicación (F5) y observar el cambio en la visualización de la lista de propiedades inmobiliarias. Desplácese por las propiedades, y elija ver el detalle de alguna.



- **Deshacer** lo hecho, volviendo al listado default del grid (por líneas).

Personalizando el Detail

- Se desea que cuando se está **visualizando** una propiedad inmobiliaria, no se muestren los ids (PropertyId, NeighborhoodId), y que éstos sólo se muestren cuando se está en **Edit**. A su vez, en **Edit** no se desea mostrar el NeighborhoodName. **Implementelo**.



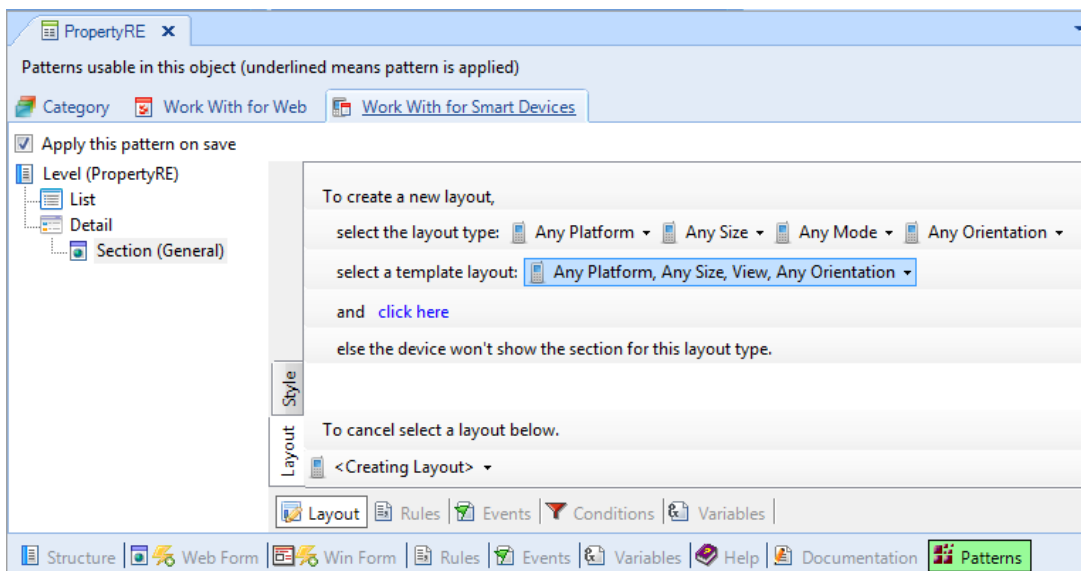
- **Ejecutar** la aplicación (F5) y verificar los cambios realizados.

- Además del layout genérico para la pantalla de visualización del detalle de una propiedad inmobiliaria, queremos tener uno **específico** para la aplicación ejecutada en dispositivos con **Android** (es decir, que en Blackberry o iOS se use la genérica y en Android la específica). En este layout para Android se desea que la imagen de la propiedad sea lo primero que se muestra.



Para ello:

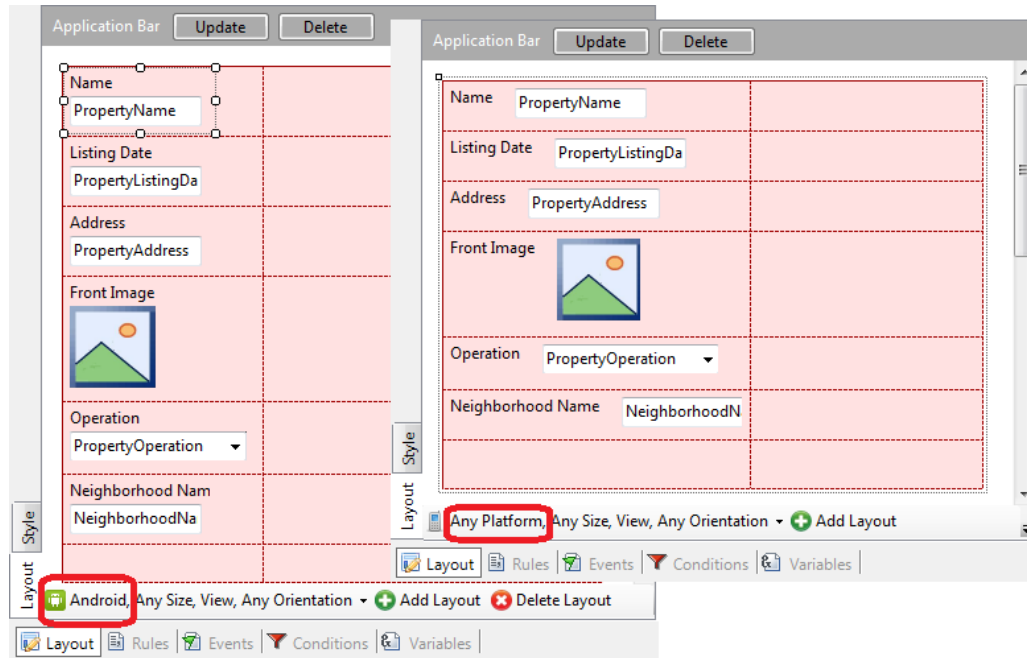
Section(General) / View. Agregue un Layout, para plataforma Android. Observe que puede inicializar el nuevo Layout con uno igual al que estaba trabajando antes (el por default: Any Platform, Any Size, View, Any Orientation) o el de Edit, o inicializarlo vacío. Elija la opción que se presenta en la siguiente imagen y haga clic donde se indica “click here”.



Asegúrese de estar editando el nuevo layout. Observe las diferencias en la ubicación de las etiquetas.

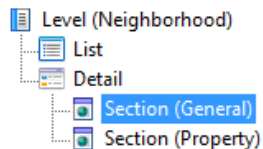


¿Por qué? ¿Dónde está configurada esta propiedad? (posicione sobre cualquier atributo del Layout y observe el valor de la propiedad **Label Position**. ¿Qué puede inferir acerca del default de la plataforma “Android” para “Any Size”?



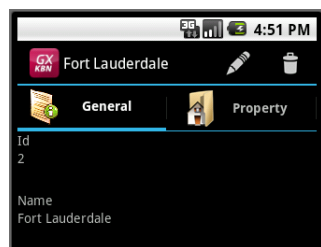
Mueva la imagen de modo que quede primera y observe el cambio en ejecución (💡 dado que estamos generando para **Android**, si estuviéramos generando para iPad o Blackberry, ¿qué **Layout** se mostraría?):

- Edite el pattern "Work With for Smart Devices" de **Neighborhood**. Observe las diferencias entre el Layout de **Section(General)** y el de **Section(Property)**.

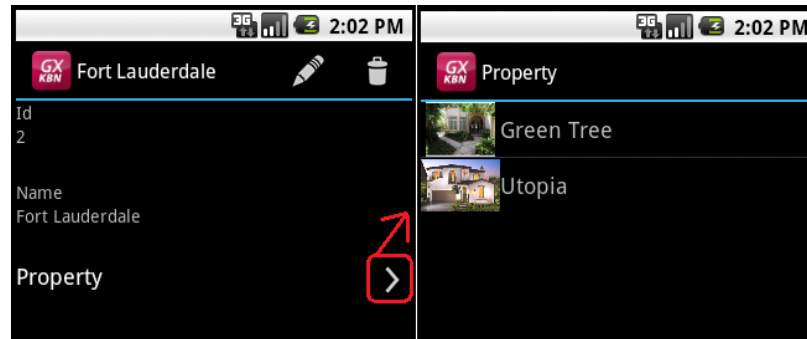


Note que mientras el Layout de Section(General) será plano, el de Section(Property) contendrá un grid para mostrar las propiedades.

- Ya habíamos visto que por defecto en Android las secciones se mostraban en ejecución como tabs:

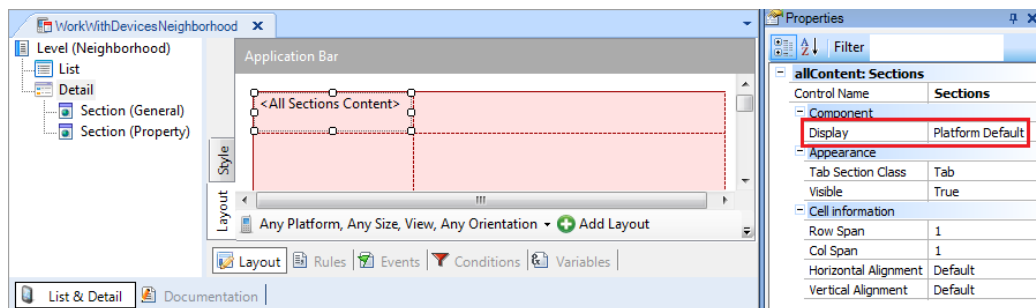


Nos piden que no se muestren así, sino que las propiedades inmobiliarias del vecindario se ofrezcan como un link a una pantalla independiente. **Implementélo.**



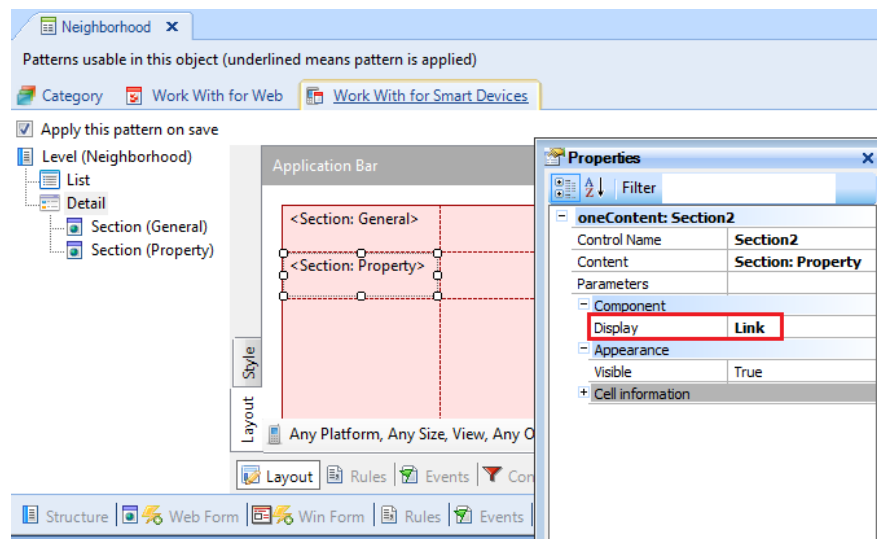
Para ello:

Observe el nodo Detail y las propiedades del control que aparece por defecto "<All Sections Content>":

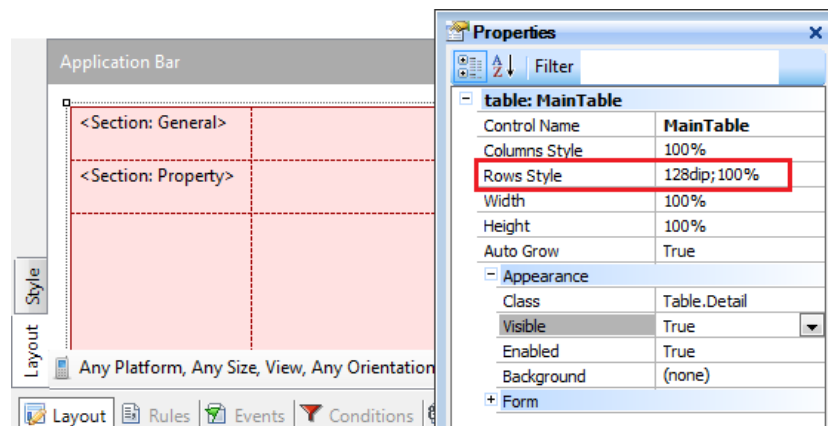


¿Qué significa la propiedad Display? Observe que los valores que ofrece son —además de Platform Default—, **Tab**, **Inline** y **Link**.

Nota: Recuerde que para que pueda manipular los contenedores de secciones individualmente, deberá eliminar <All Sections Content> del Layout, tras lo cual aparecerá en la toolbox junto con un contenedor por cada sección.

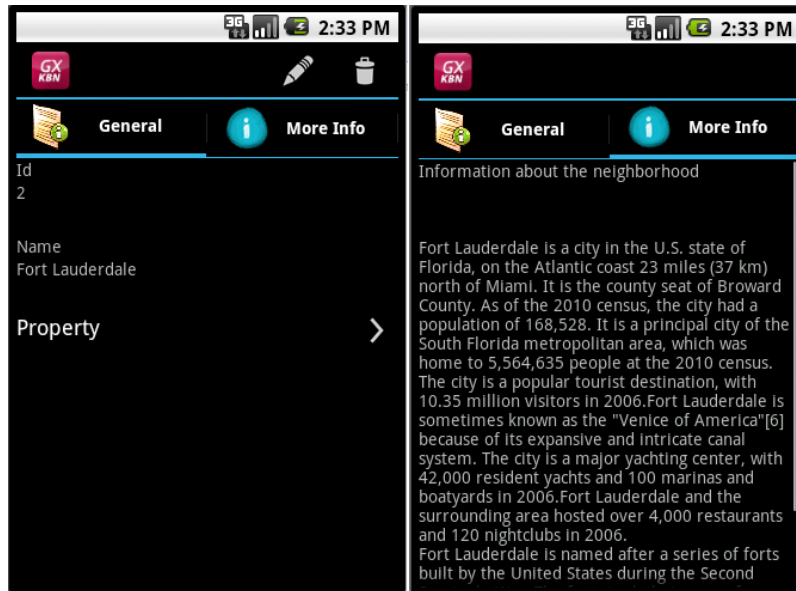


Observe en la solapa Style que no está mostrando la Section Property ¿Por qué? Observe la propiedad **Rows Style** de la tabla. Modifique el tamaño de las filas para que entren ambas secciones:



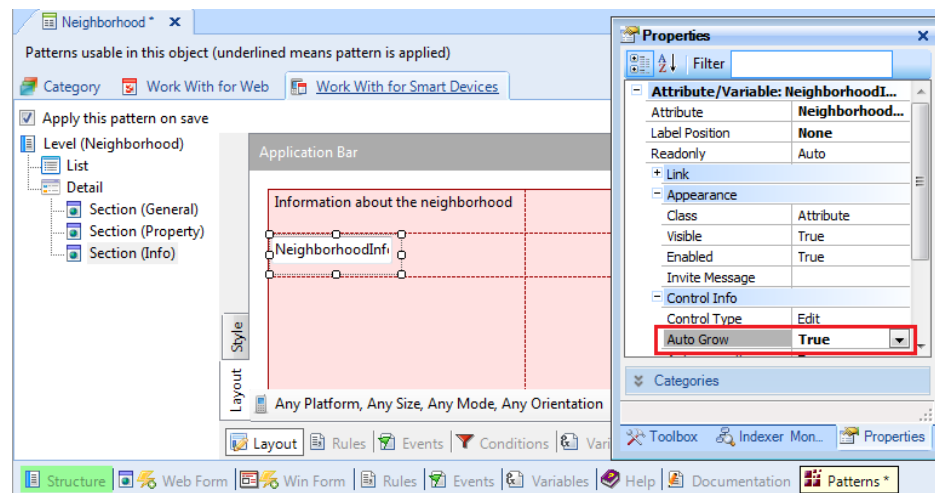
[OPCIONAL]

- Se necesita agregar a la estructura de la transacción Neighborhood un **atributo NeighborhoodInfo** de tipo de datos Varchar(1024) para almacenar información útil del vecindario. Observe que automáticamente el atributo se agrega al Layout de **Section(General)**. Sin embargo, en modo visualización (view) no se desea que aparezca allí, sino en una solapa (tab) aparte, "More Info":

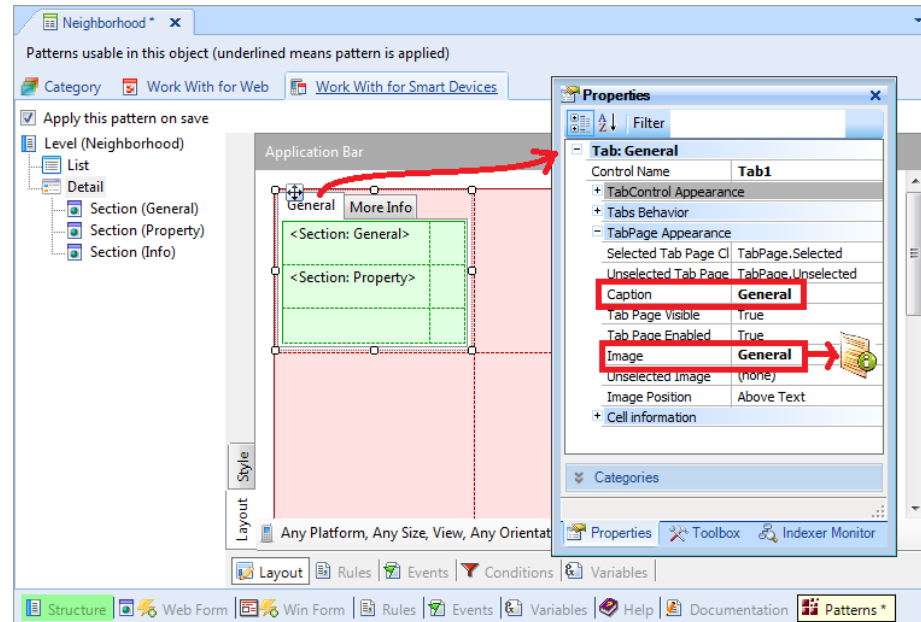


Para ello:

- Posiciónese sobre el nodo **Detail** y con botón derecho haga **Add → Section**. Observe cómo se agregará una nueva sección (Section Code será el nombre, que no puede contener espacios). Su Layout se crea vacío.
- Inserte allí un text block y el atributo **NeighborhoodInfo** recién creado, cambiándole la propiedad **AutoGrow** a True:

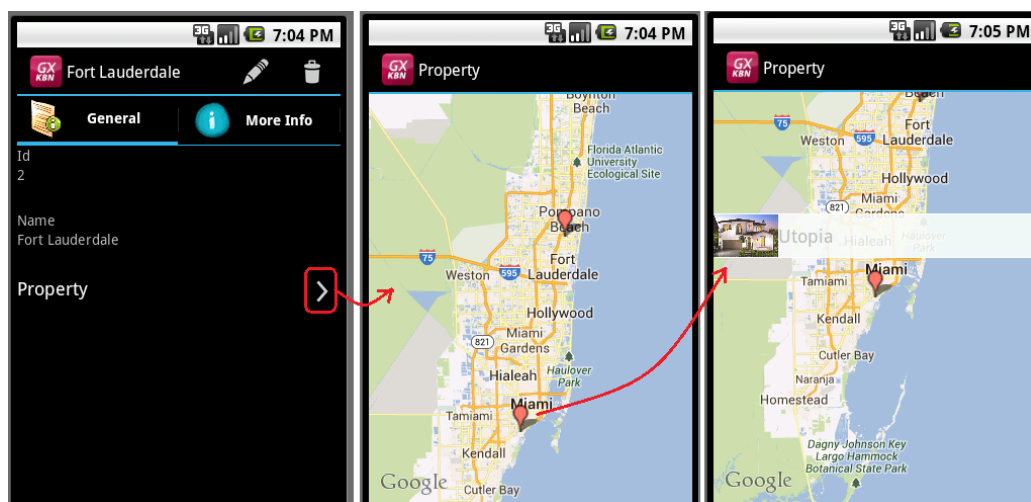


- Agregue a la sección la regla **Parm** para que se filtre por el vecindario.
- Observe el nodo Detail. No fue agregada la nueva sección automáticamente. ¿Por qué? Observe que el control "Section: Info" se encuentra en la Toolbox para poder ser utilizado.
- Agregue un control Tab, configurando los **Captions** de los ítems de la siguiente manera y arrastre a cada ítem el control <Section> correspondiente. Asóciele una imagen a cada ítem del tab control:



Nota: Recuerde editar las propiedades de la tabla interna al tab General, para que entre toda la información en la pantalla. Por ejemplo: Rows Style = 128dip; 100%

- Pruébalo en ejecución (recuerde ingresar información de por lo menos un vecindario para probar lo efectuado).
- Si ahora queremos que las propiedades del vecindario se muestren, no como lista de ítems, sino como puntos en un mapa, ¿cómo lo implementa?

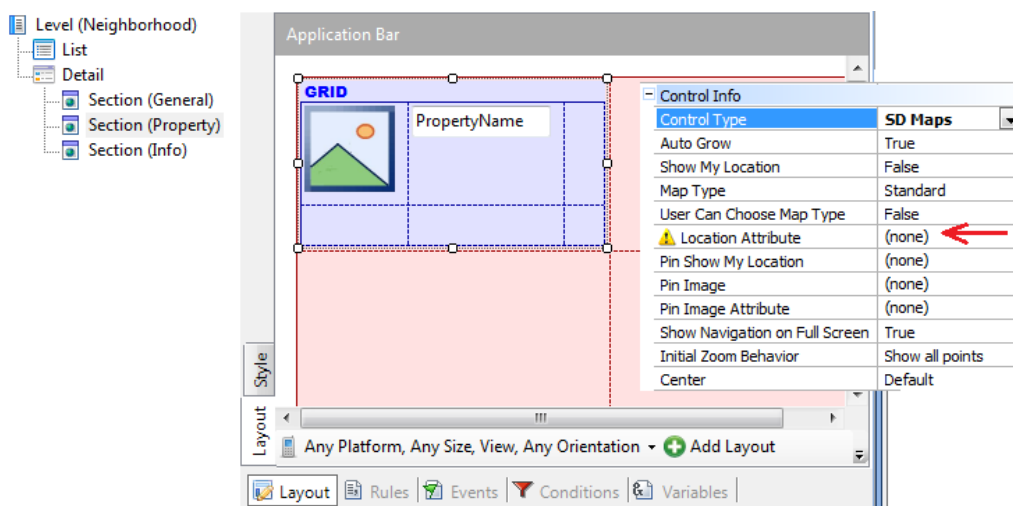




Para ello:

Dado que en el Layout de **Section (Property)** se encuentra un grid, pueden utilizarse para la visualización los mismos tipos de control grid vistos para el List. Utilizaremos el SD Maps, que requiere que exista un atributo de tipo geolocation. Entonces:

- Agregue un atributo **PropertyGeolocation** a la estructura de la transacción Property, para almacenar la longitud y latitud de la propiedad inmobiliaria. Obsérvese que asumirá como tipo de datos el dominio semántico Geolocation.
- Luego vaya al **Section(Property)** del Work With para **Neighborhood** y en el grid modifique la propiedad **Control Type**, a **SD Maps**, especificando el atributo PropertyGeolocation como el "Location attribute".



- Haga **F5** (tendrá que **reorganizar**). Asigne geolocations a las propiedades que tiene ingresadas y pruebe lo realizado.



Para ello:

Puede utilizar google maps para obtener latitud y longitud de las direcciones que utilizó, haciendo botón derecho sobre el punto en el mapa y luego "¿Qué hay aquí?". Aquí le pasamos algunas:

Address: 11235 NE 4 Ave, Miami

Geolocation: 25.879081,-80.190101

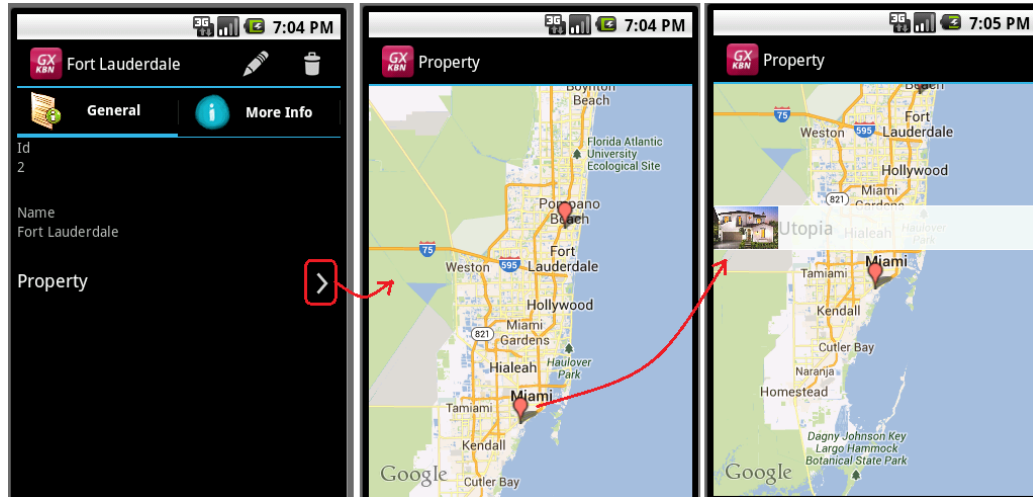
Address: 5201 Norht Dixie Highway Oakland Park

Geolocation: 26.1922,-80.134461

Address: 3975 Utopia Ct, Miami

Geolocation: 25.71711,-80.255472

Nota: Cuando se estudien las apis, más adelante, podrá resolverlo codificando la obtención del dato geolocation a partir del address, utilizando el método GetLocation de la api GeoLocationAPI. Así no será necesario que el usuario lo ingrese.

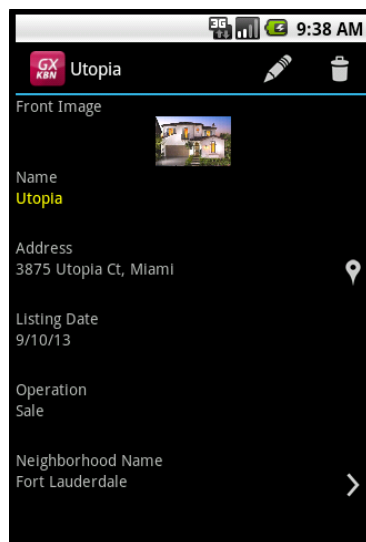


Observe que haciendo *tap* sobre una de las ubicaciones, se muestran los datos del grid, y si se hace *tap* sobre esta información, se muestra todo el detalle.

Themes

Los Layouts con los que hemos trabajado hasta el momento, no permitían configurar aspectos de diseño de los controles. Para poder hacerlo, tenemos los themes.

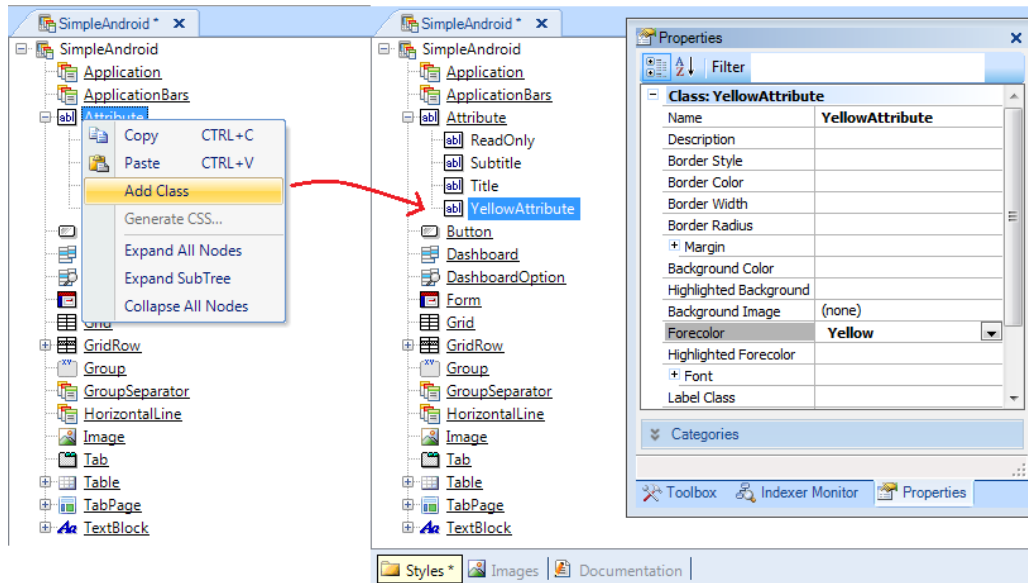
Por ejemplo, queremos poder configurar que algunos atributos puedan desplegarse con forecolor amarillo, en Android. En particular, el atributo `PropertyName` al visualizar una propiedad inmobiliaria. **Implementelo** y pruébelo en ejecución.



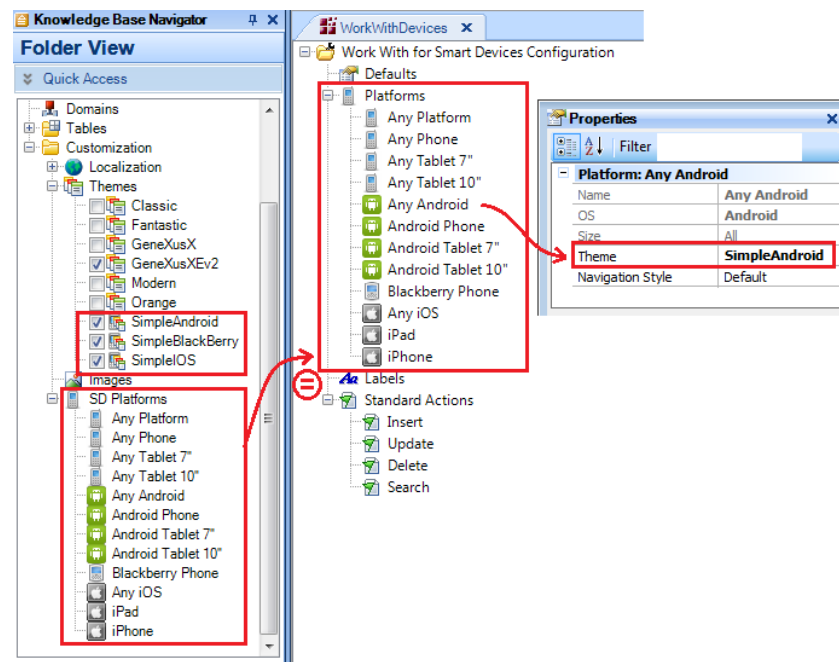
Para ello:

- Modifique el theme predeterminado para Android: "SimpleAndroid"

- Cree una subclase del nodo Attribute, con el nombre que desee, y modifíquela la propiedad Forecolor, como mostramos (luego grabe):



Observe que en el **Folder View** tenemos bajo el nodo Customization, propiedades a configurar para cada plataforma de Smart Devices posible:

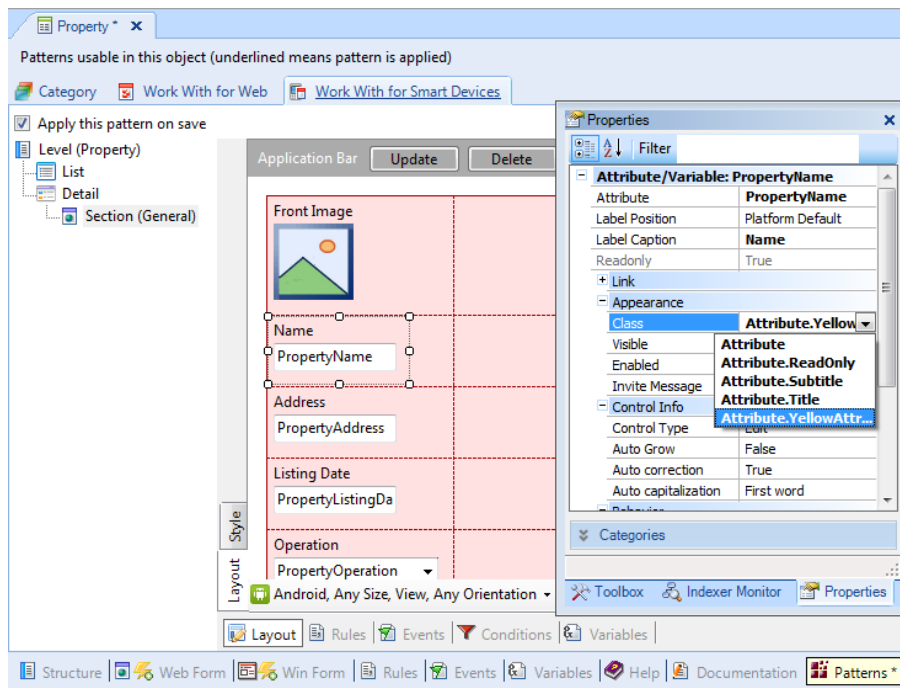




Las mismas propiedades se encuentran bajo **Preferences/Patterns**, en una instancia que configura las propiedades de los Work With for Smart Devices. Para cada nodo bajo Platforms, tendrá la posibilidad de configurar un theme diferente. Por ejemplo, si quisiera un diseño diferente cuando se trata de tabletas de 7", de 10" o de teléfonos.

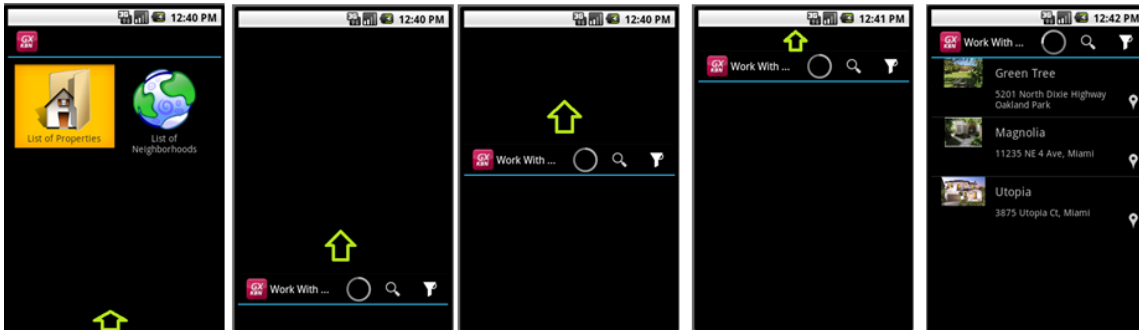
Si tiene un theme configurado para "Any Platform" y uno para "Any Android" y usted está generando para Android, ¿cuál tomará la aplicación?

- Ahora configure que al visualizar una propiedad inmobiliaria (**Detail**), el nombre de la misma salga amarillo. Vaya a **Section(General)** y elija para el atributo PropertyName la clase YellowAttribute (asegúrese de estar en el Layout para Android y no para "Any Platform").



Transitions

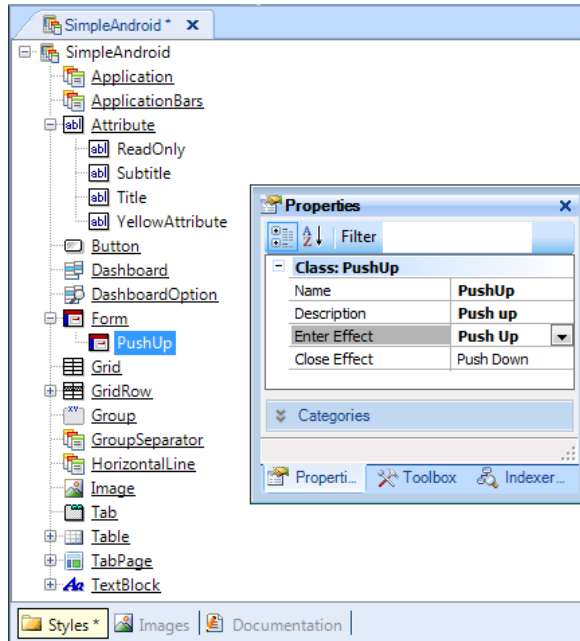
Si desea que el listado de propiedades se abra siempre desde abajo, ¿cómo lo consigue?



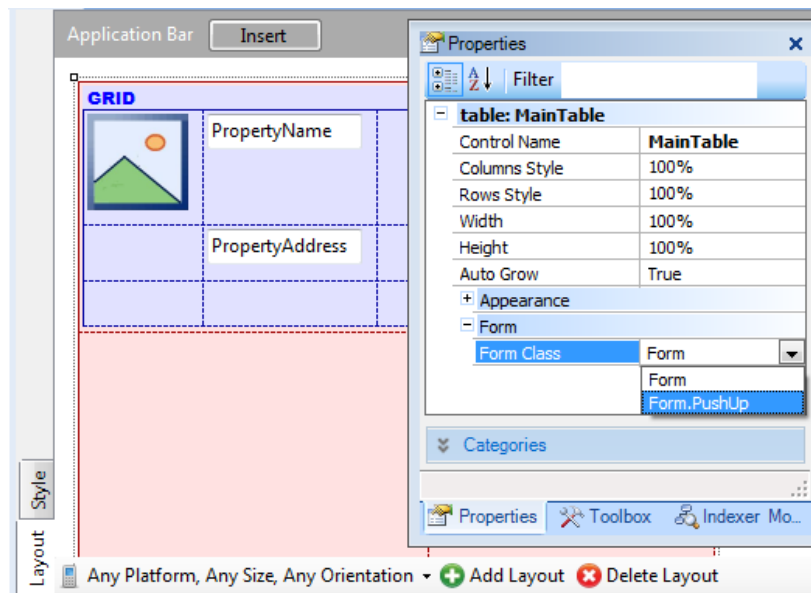


Para ello:

Crear una subclase de la Form (aquí le llamamos PushUp, pero puede llamarse como usted quiera). Configure el Enter Effect.

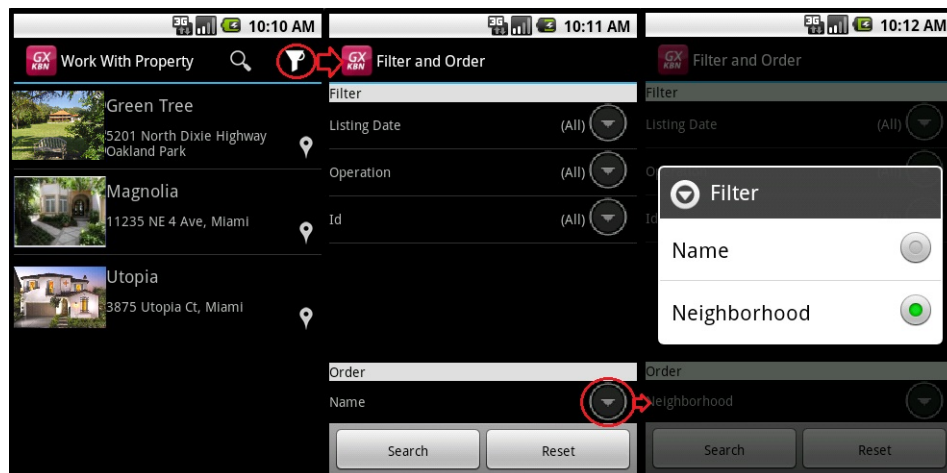


Luego vaya al layout del List del Work with de propiedades inmobiliarias, y configure como Form Class, la recién creada.



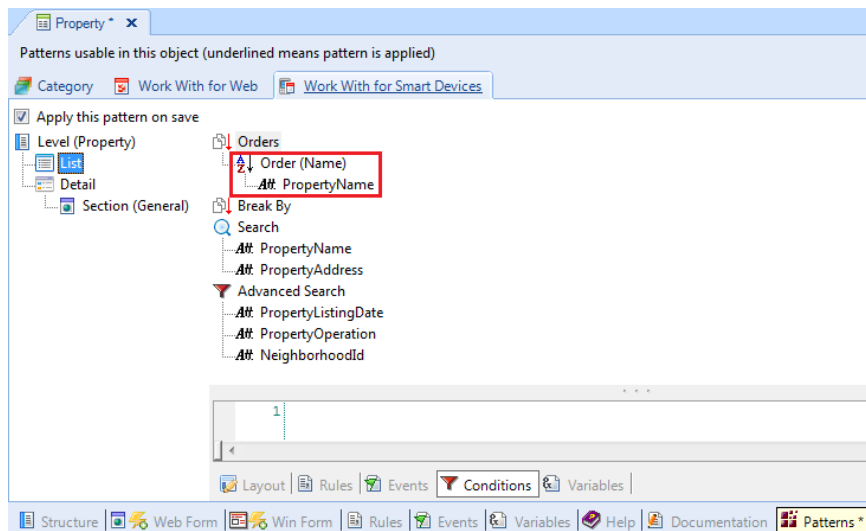
Agregando Órdenes y Filtros

- El usuario quiere poder elegir en el “List of Properties”, a través del ícono filter, si desea ver la información ordenada por nombre de propiedad o por vecindario. Implementélo.

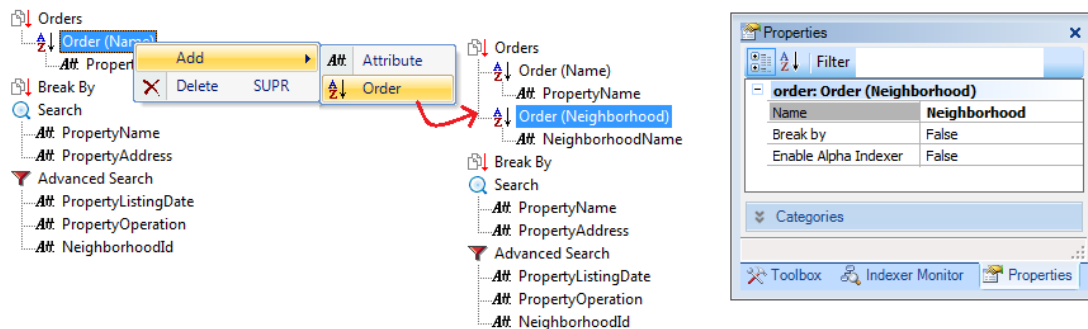


Para ello:

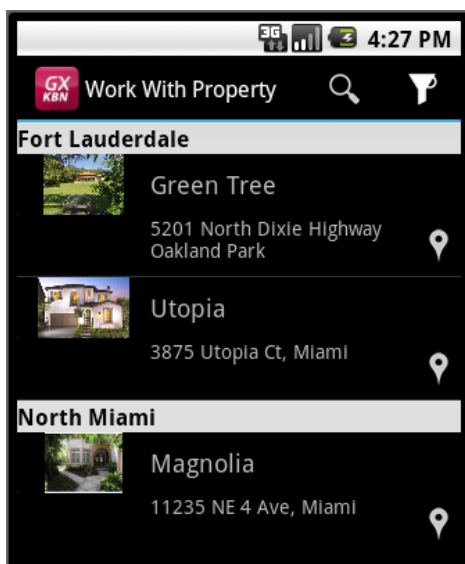
- Seleccionar la pestaña **Conditions** dentro del List:



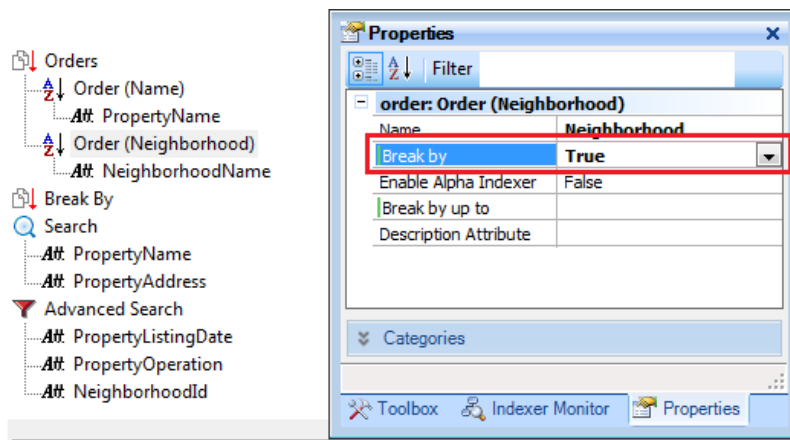
- Agregar un Orden por el atributo **NeighborhoodName**:



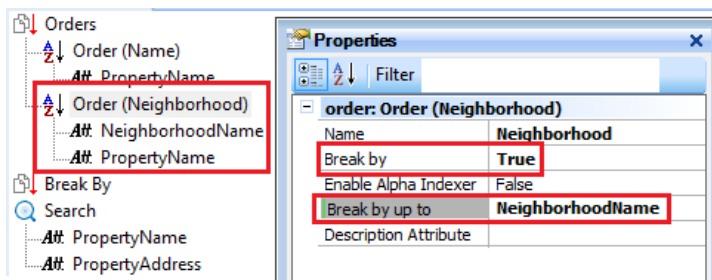
- Ahora, al elegir mostrar las propiedades ordenadas por vecindario, se quiere que aparezcan no sólo ordenadas, sino **agrupadas** por vecindario.



Para ello, edite:



- ¿Y si ahora se necesita que dentro de cada grupo (vecindario) se ordenen por nombre de propiedad?

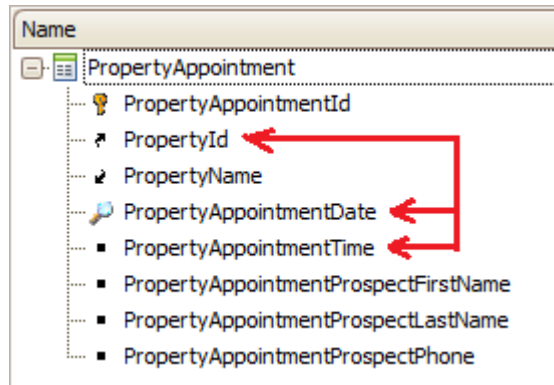


Agregando Acciones

- Se necesita agendar visitas de posibles clientes (prospects) a cada propiedad.

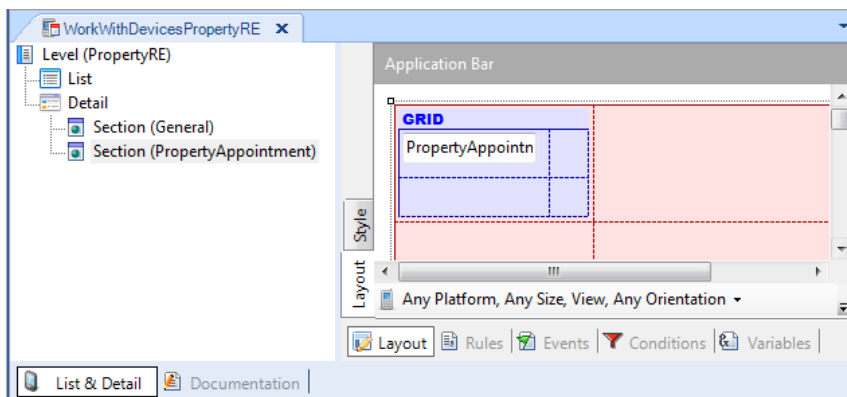
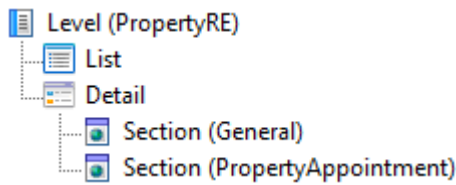


Para ello, crear la transacción PropertyAppointment con la estructura que se muestra:

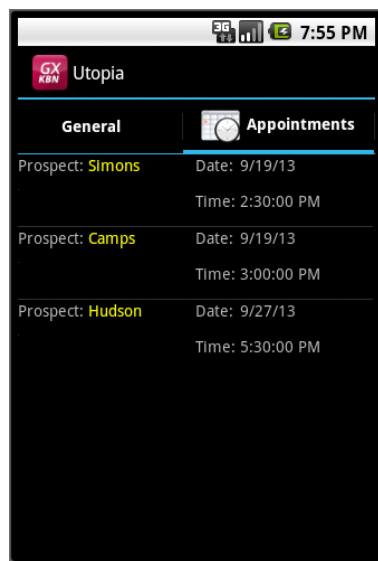


Observación: {PropertyId, PropertyAppointmentDate, PropertyAppointmentTime} son una clave candidata, por lo que podría definir un índice único para representarla. A los efectos de este práctico no es necesario.

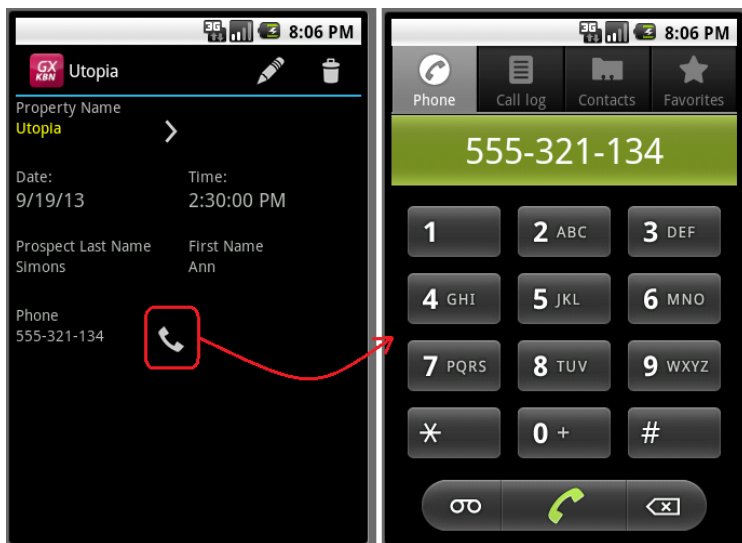
- Aplicarle el "Work With for Smart Devices" a la transacción PropertyAppointment y verificar que se añada al **WorkWithDevicesPropertyRE** la **Section(PropertyAppointment)**



- Si quiere agregar o quitar atributos del Layout de la sección PropertyAppointment, hágalo. Puede acortar el Caption de la sección para que no sea tan largo (por ejemplo: **Appointments**) y asociarle una imagen a ambas secciones. Para comprobar el comportamiento, **haga F5 (reorganice)**. Ingrese algunas visitas a través del Developer Menu web y pruebe en el emulador que esas visitas aparecen listadas para la propiedad correspondiente.



Observe que sin programar nada, al hacer *tap* sobre una de las visitas desplegadas, se visualiza su información (utilizando el Detail del WorkWith de **PropertyAppointment** correspondiente). Esta será la acción por defecto ("Default Action"). En la pantalla que sigue se ve este Layout personalizado.



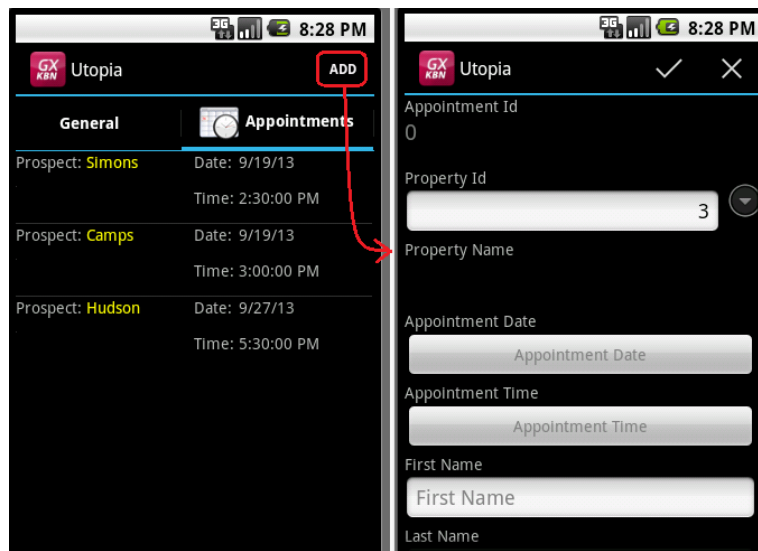
Observe también que el atributo elegido como descriptor es el que se muestra en este Layout arriba del todo (en nuestro caso, la fecha de la appointment). Puede cambiarlo por el apellido del prospect (vea el evento Start del nodo Detail).

Observe lo que sucede automáticamente con el teléfono (está utilizando la funcionalidad nativa del dispositivo)



Observe en el nodo **List** la Application Bar: tiene un botón **Insert**, vaya a ver la programación del evento asociado. ¿A quién se está invocando? Ahora vaya al nodo **Section(PropertyAppointment)**. Observe que la Application Bar está vacía y que tampoco dentro del Layout existen botones. Aquí no se está ofreciendo la posibilidad de insertar una nueva visita (appointment): sólo se listan las visitas preexistentes de esa propiedad.

- Desde la sección que muestra las visitas agendadas para una propiedad, se quiere ofrecer la posibilidad de insertar una nueva. **Implementelo** y pruébelo.

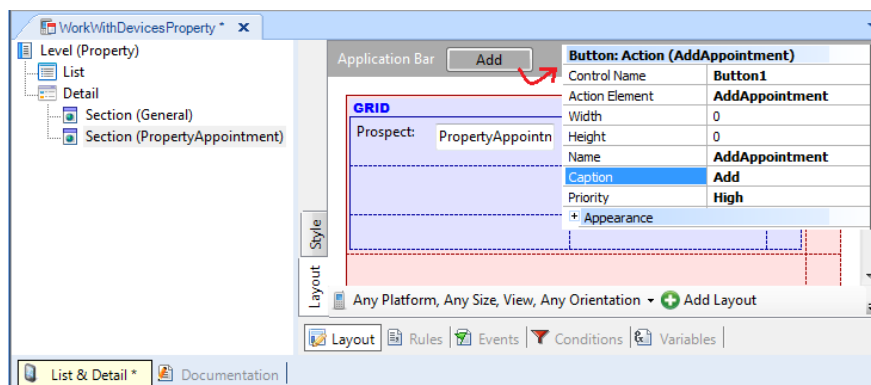


Para ello, en la **Section(PropertyAppointment)** del WorkWithPropertyDevices:

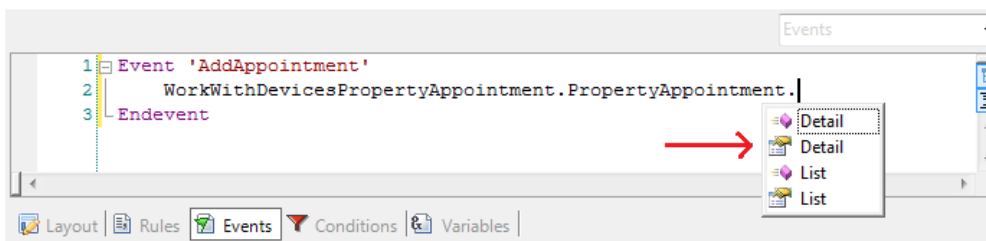
- Botón derecho sobre la Application Bar/ Insert Button
- Ingrese el nombre de la acción que se asociará automáticamente al botón (e.g. AddAppointment).



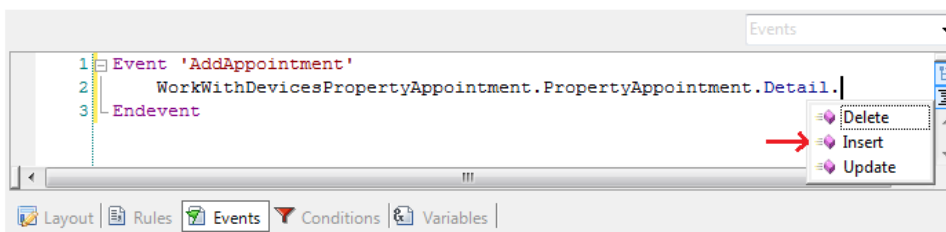
Observe que puede cambiar la prioridad (al cambiarlo a High, el botón aparecerá siempre visible) y el caption para acortar el nombre a **Add**.



- ¿A quién debe invocar la acción? ¿En qué modo? Para ello deberá ir a la sección Events:



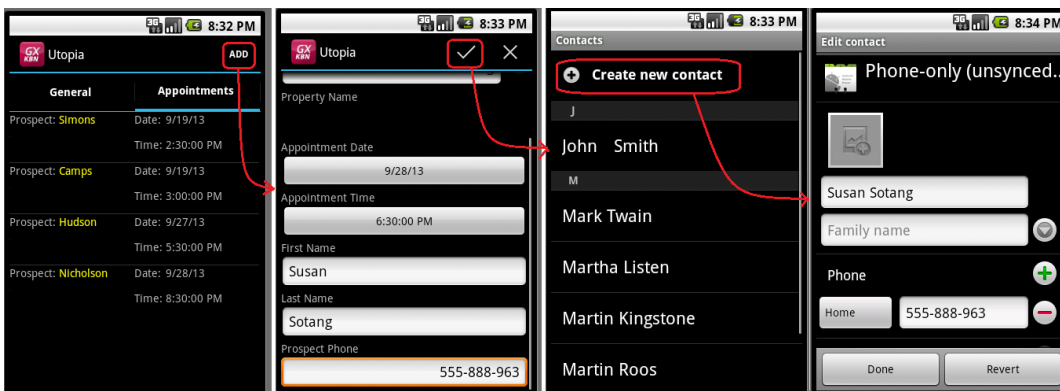
Necesita invocar al Detail del Work With de PropertyAppointment, en modo Insert.



Necesitamos enviarle como parámetro PropertyId. Para ello debemos cargar una variable Business Component de tipo PropertyAppointment y pasarla por parámetro (la variable será inout, volverá cargada con los valores ingresados por el usuario en la pantalla del detalle):

```
Event 'AddAppointment'
    &appointment.PropertyId = PropertyId
    WorkWithDevicesPropertyAppointment.PropertyAppointment.Detail.Insert (&appointment)
Endevent
```

- Ahora queremos interactuar con algunas funcionalidades del dispositivo. Por ejemplo, queremos que al **agregar** una nueva visita, se ingresen los datos del prospect en la **libreta de direcciones**.





Para ello:

Puede arrastrar al evento el objeto **AddressBook** del folder **SmartDevicesApi** o tipearlo. Utilizaremos su método **AddContact**. Recuerde que los parámetros de las apis son posicionales y obligatorios.



¿Por qué GeneXus le advierte que debe escribir comando composite si olvida especificarlo?

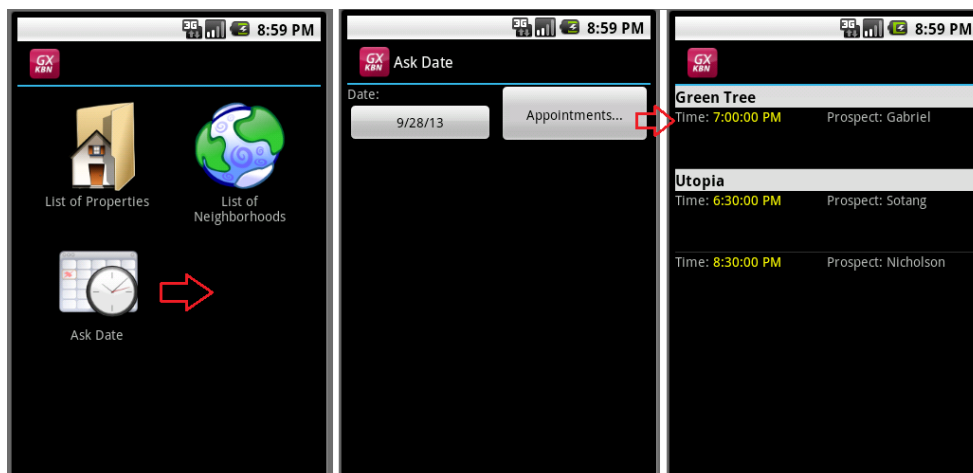
```
Event 'AddAppointment'
  Composite
    &appointment.PropertyId = PropertyId
    WorkWithDevicesPropertyAppointment.PropertyAppointment.Detail.Insert(&appointment)
    AddressBook.AddContact( &appointment.PropertyAppointmentProspectFirstName,
                          &appointment.PropertyAppointmentProspectLastName, '',
                          &appointment.PropertyAppointmentProspectPhone, '', '', '')
  Endcomposite
Endevent
```

Haga **F5** e intente ingresar una nueva visita. Luego de salvar la visita, se le abrirá la aplicación de contactos.

Nota: Si cuenta con un dispositivo (el emulador no contiene Calendar), podría también agendar la visita en el calendario del dispositivo.

6. Paneles para pedir datos al usuario

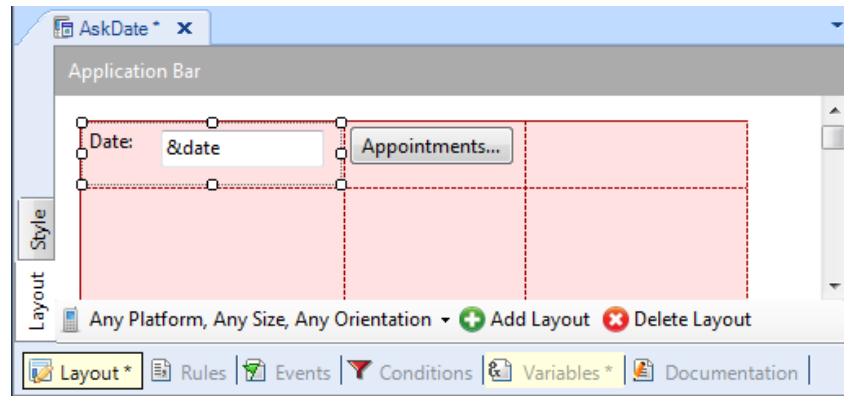
Queremos poder listar solamente las visitas agendadas para un día determinado. Por lo que necesitamos que el usuario ingrese la fecha deseada y entonces proceder a mostrar las visitas.



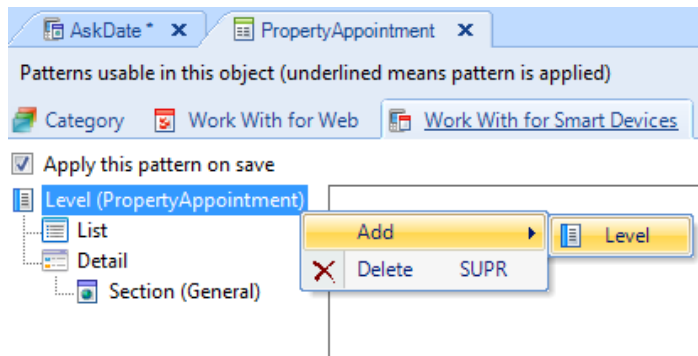


Para ello:

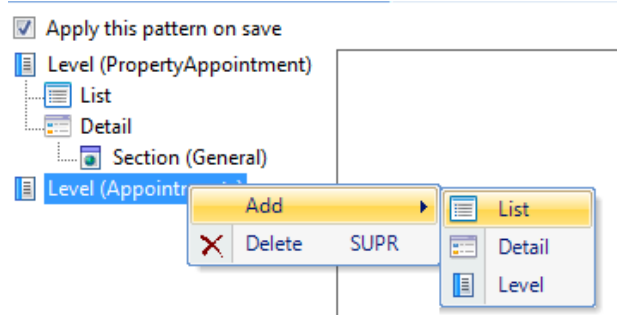
- Cree un objeto del tipo “Panel for Smart Devices” de nombre **AskDate**.
- Agregue una variable **&date** y luego insértela en el Layout (asegúrese que la propiedad Readonly quede en False, de manera tal que el usuario pueda ingresar valor para la misma)
- A su lado haga botón derecho e ingrese un nuevo botón (vea cómo esta vez estará dentro del Layout).

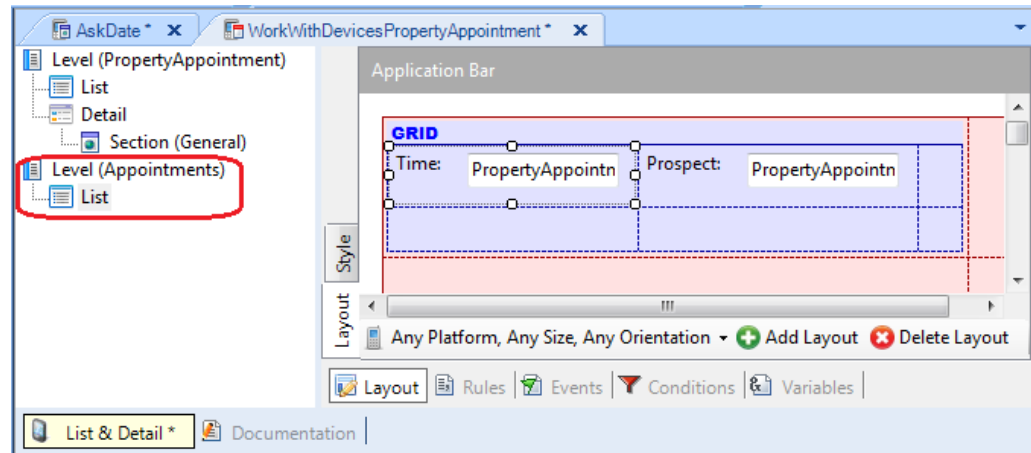


¿A quién invocará la acción asociada al botón? ¿Al List del Work With for Smart Devices de la transacción PropertyAppointment? Si invocamos a este panel, vamos a tener que modificarlo, para que ahora reciba un parámetro: la fecha. Si esto no nos sirve, entonces una posibilidad es crear un **Level paralelo** en el Work With de PropertyAppointment. Hagámoslo:



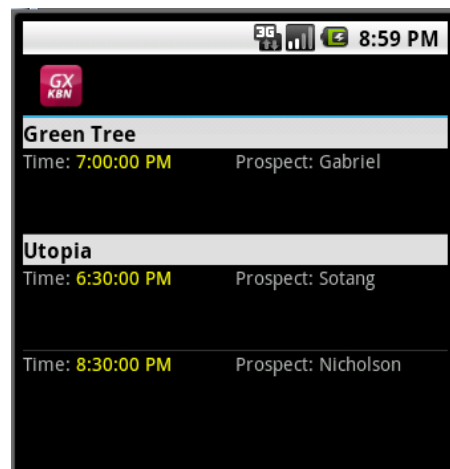
- Creamos ese nuevo Level con un nodo List, que es en el que diseñaremos el Layout para este caso.



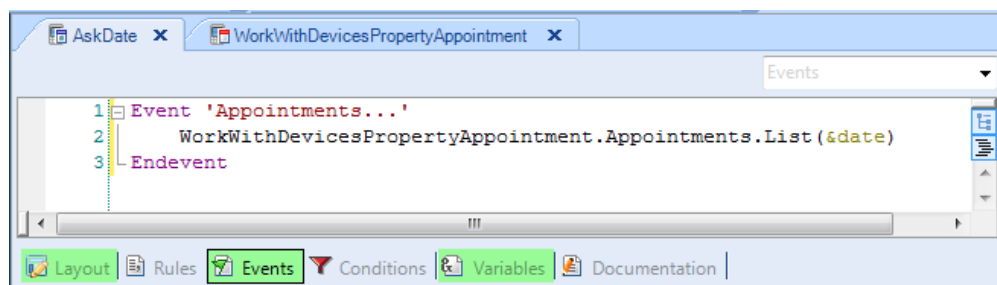


Y especificamos parm(in: PropertyAppointmentDate);

Nota: además, hemos agregado un orden por PropertyName, PropertyAppointmentTime, con un break by PropertyName, así nos salen agrupadas las visitas por propiedad; y hemos personalizado el Layout, para que nos salga la información así:



- ¿Qué nos resta? Ir al panel que permite ingresar una fecha, y en el evento asociado a la acción invocar al Level que acabamos de crear, pasándole el parámetro &date (la variable):



- Agregue al Dashboard este panel. Así, podremos ver en ejecución lo deseado.

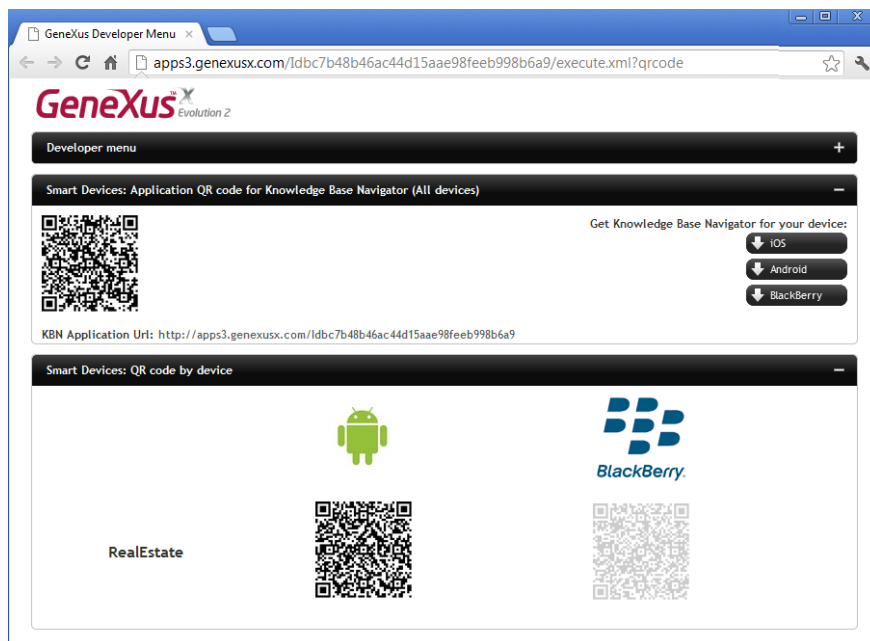
7. Ejecución en el dispositivo [opcional]

- Si cuenta con un dispositivo **Android**, especifique al dashboard como “Startup Object” y haga F5, o directamente haga botón derecho sobre la pestaña del dashboard y “Run” (si no desea especificar startup object).

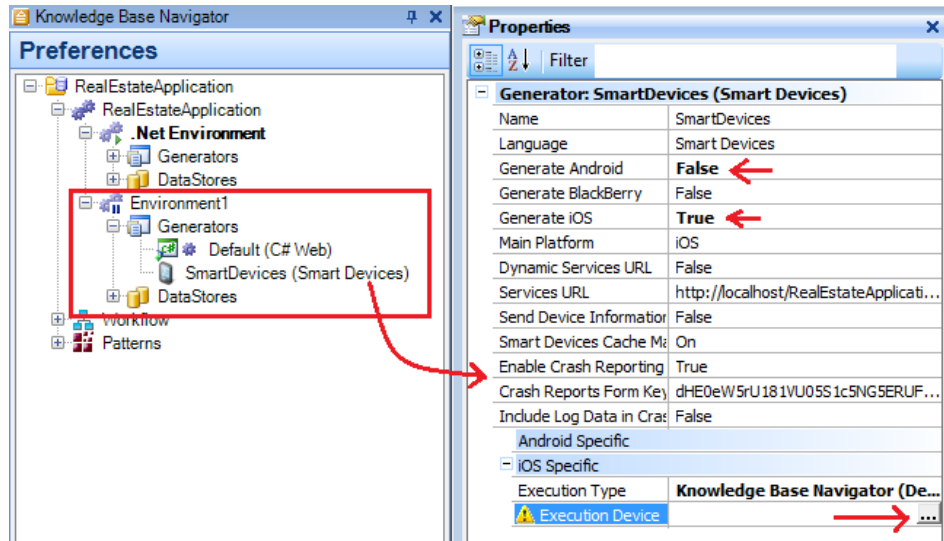


Recuerde que habrá un único startup object por environment (por tanto, dependiendo de si es un objeto web o SD, la parte de la aplicación que podrá probar, si lo configura).

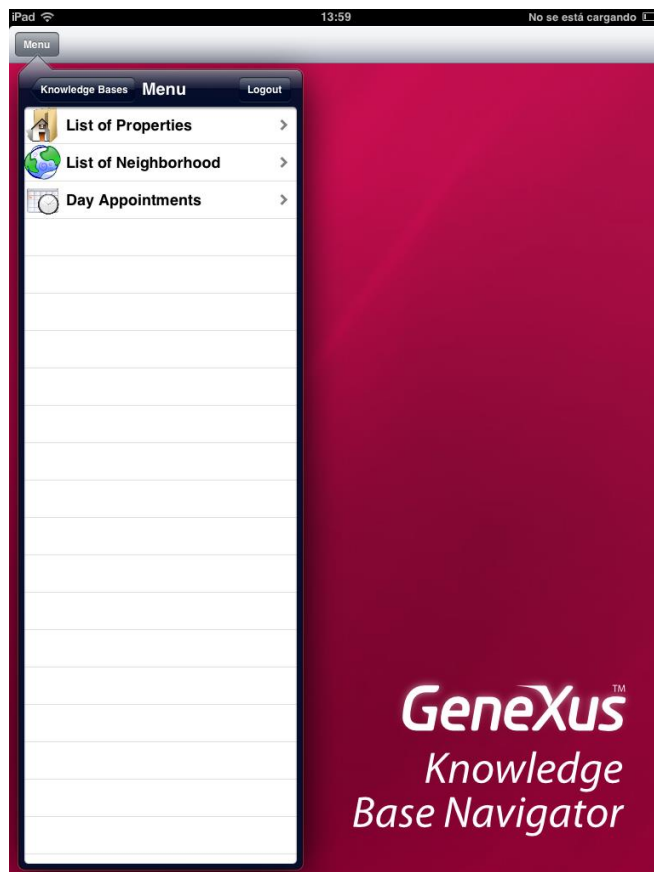
- Tiene dos alternativas rápidas:
 - conectar el dispositivo a la PC/notebook (instalándose previamente el driver necesario y habilitando USB debugging en el dispositivo), o
 - instalarse la aplicación en el dispositivo, bajándosela de la nube –o copiando el compilado (archivo apk) en la SD card–. Puede ver todas las posibilidades aquí: <http://wiki.gxtechnical.com/commwiki/servlet/hwikibypageid?14910>.
- Si seguimos la opción **b**: haga View/**Show QR Codes**. Ejecute la aplicación de su dispositivo que le permite scannear y realice la lectura del Código QR que aparece (que contiene codificada la URL del compilado donde fue subida la aplicación Smart Device en la nube). Automáticamente se descargará el compilado en su dispositivo y podrá instalarlo y ejecutar.



- Si cuenta con un dispositivo **iOS**, descárguese el KBN correspondiente (lo obtiene de iTunes, de la url que le muestra el Developer Menu) y ejecútelo. Allí loguéese con su usuario de GXtechnical. De esa manera, quedará su dispositivo almacenado en la información de su usuario.
- Luego vaya a GeneXus, y, si tiene la versión full creése un nuevo Environment, modificando las propiedades tal como se muestra en la siguiente imagen –si tiene la versión Trial, no podrá crear un nuevo ambiente, así que modifique el existente–:

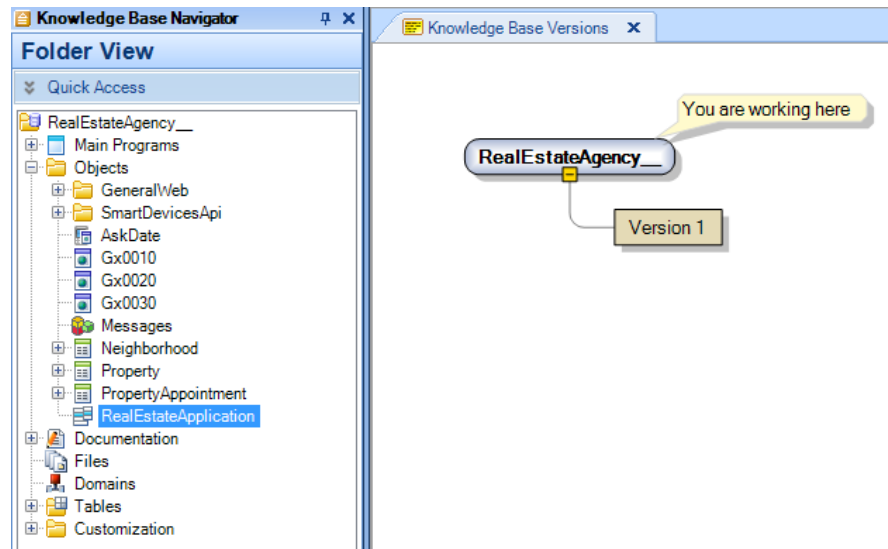


En la propiedad "Execution Device", aparecerá el listado de dispositivos asociados a su usuario. Elija el que registró en el paso anterior. Active el nuevo environment (botón derecho/Set As Target Environment) y haga un F5.



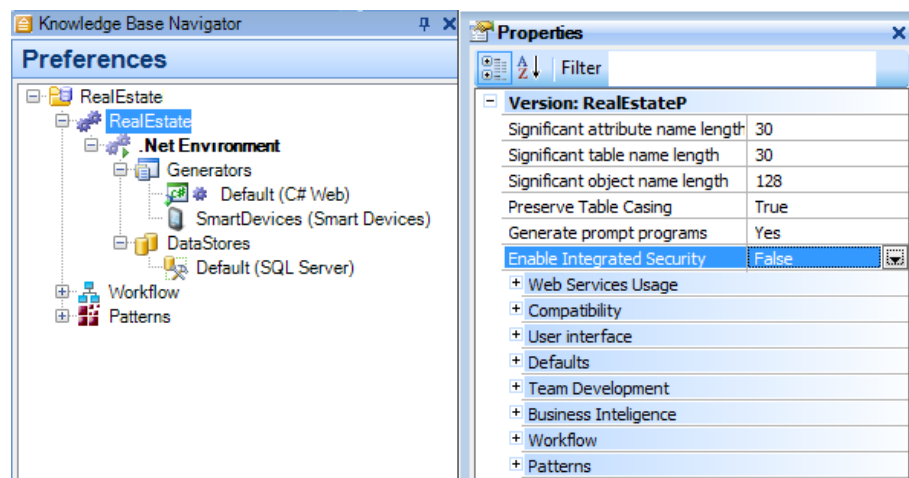
8. Activación del GAM

- Antes de proseguir, congele la versión actual de desarrollo, para tener un respaldo del estado actual de la KB.
- Esto se logra desde la opción View/Versions, Freeze sobre la versión.

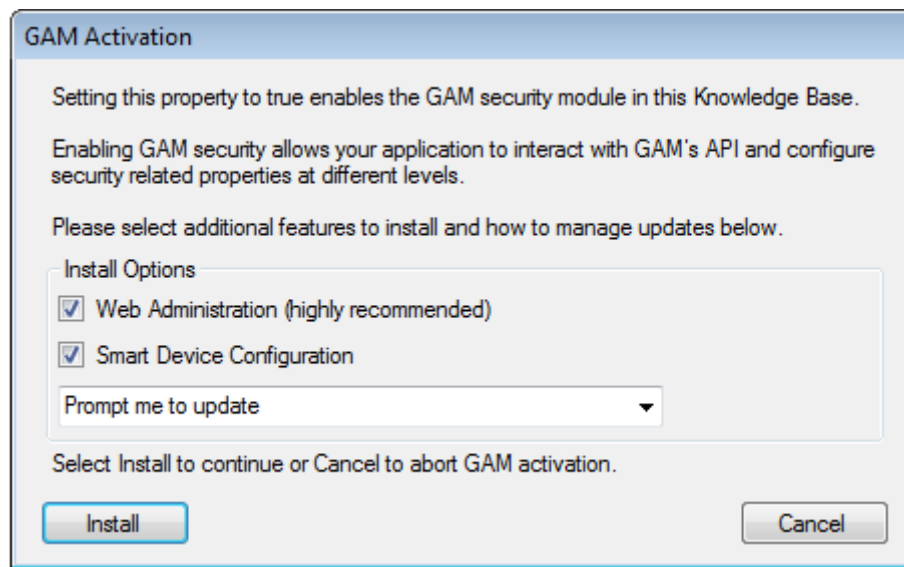


- Incorporar el GeneXus Access Manager (GAM) para obtener un control de acceso automático.

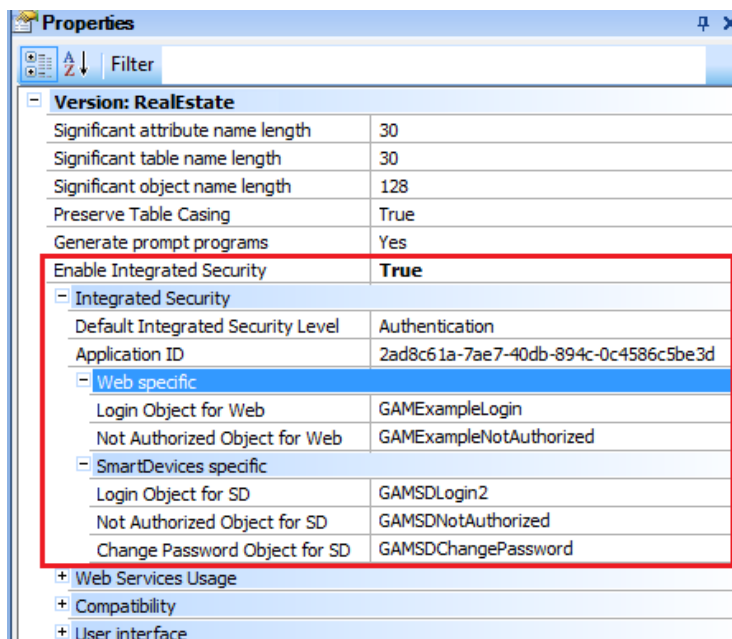
Para ello:



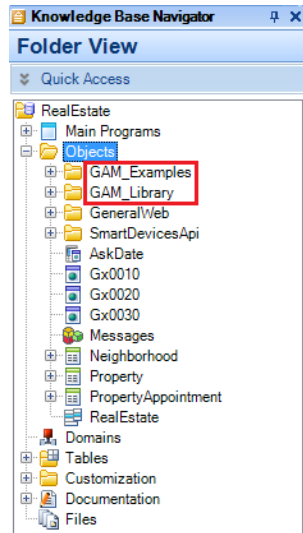
Configure la propiedad **Enable Integrated Security** en True, presione Install:



Observe las propiedades que se abren:

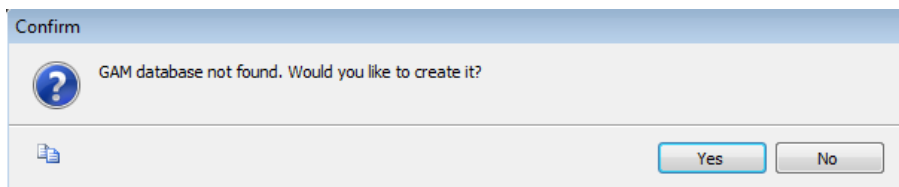


- Observe todo lo que se importa en la KB tras el cambio anterior (en el Folder View encontrará dos nuevas carpetas, GAM_Examples y GAM_Library).



¿Qué objetos son los que se configuran automáticamente en las propiedades “Login Object for Web” y “Login Object for SD” en las propiedades mostradas más arriba?

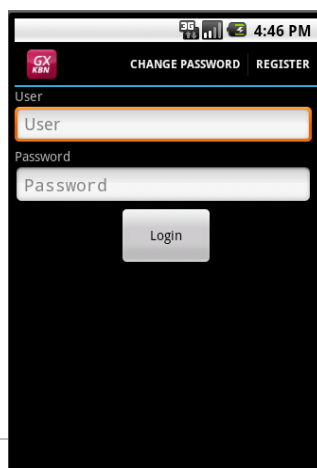
- Haga **Rebuild All** y luego **F5**. Confirme que desea crear la base de datos del GAM:



Nota: se requiere el MySQL driver para .Net para poder crear la base de datos del GAM contra MySQL: 'libmysql.dll'. Si no está prototipando con .Net (en ese caso ya la tiene), asegúrese de instalarla en C:\Windows\SysWOW64 si su CPU es de 64 bits y en C:\Windows\System si es de 32 bits, desde el siguiente link: <http://sourceforge.net/projects/mysql/drivers>. Vea más en nuestro wiki: <http://wiki.gxtechnical.com/commwiki/servlet/hwikibypageid?2041>

Asegúrese antes de volver a hacer el Rebuild all que le haya quedado en True la propiedad Enable Integrated Security

- Al ejecutar la aplicación en el Smart Device, ingrese los siguientes datos para la conexión:



usuario: **admin**
password: **admin123**

- Vaya al Developer Menu, e intente ejecutar una transacción. No podrá, sin pasar antes por el Login. Pruébelo.

- Desde el Developer Menu ejecute el Web panel **GAM Home** y observe el Backend para administrar la seguridad:

Update	Roles	Password	Delete	Authentication	Name	First Name	Last Name
				local	admin	Administrator	User

- Definir un nuevo usuario desde la opción Users
- Configurar:

- User Name = agency01
- Password = agency01

User

GUID	DemoCurso2013
Namespace	
* User Name	agency01
* Email	agency01@gmail.com
First Name	Real Estate
Last Name	MyHouse
* Password	
* Password confirmation	
External Id.	
Birthday	/ / 28
Gender	Not Specified
/ / 12:00 AM	
Don't want to receive additional information	☐
Cannot change password	☐
Must change password	☐
Password never expires	☐
User is blocked	☐
Security Policy	(None)
☐	

* Information required.

Confirm **Cancel**

9. Uso de la API del GAM

Los agentes inmobiliarios pueden ingresar y visualizar las propiedades que poseen en venta o alquiler accediendo a la aplicación para Smart Devices, desde sus dispositivos.

Pero como la Transacción PropertyRE es única para todos los agentes inmobiliarios, se debe identificar al agente que ingresa una Propiedad en el sistema.

Además, en la Lista de Propiedades que se muestra en la aplicación para Smart Devices, se requiere que los agentes inmobiliarios sólo puedan visualizar los registros ingresados por ellos y no la información ingresada por otro agente inmobiliario.

Para lograr esto se debe:

- Almacenar en la Transacción PropertyRE, la identificación del agente inmobiliario que ingresa la Propiedad.
- En el WWSD Property, mostrar solamente las Propiedades del agente inmobiliario logueado.



Para ello siga los siguientes pasos:

Cambios en la TRN PropertyRE

- Agregue un nuevo Atributo llamado UserId en la TRN PropertyRE (para identificar el agente inmobiliario que se loguea a través del GAM).

El atributo se define de tipo GAMUserIdentification, que es un dominio que se importa junto con los objetos del GAM.

- En las reglas de la Transacción agregar:

```
Equal (UserId, GetUser());
```

Donde Get User será un procedimiento que se creará a continuación. De esta forma se captura automáticamente la información del usuario logueado y se guarda en la Transacción al ingresar una propiedad (recuerde que la regla equal, cuando no se está en modo insert filtra, por lo que las propiedades inmobiliarias ya ingresadas no se mostrarán al usuario logueado).

Nuevo procedimiento GetUser

- Definir el procedimiento **GetUser** con la siguiente lógica:

```
&UserId = GAMUser.GetName();
```

Esto permite obtener el usuario logueado en la aplicación.

GAMUser es una variable estática que instancia la información del usuario logueado, pudiéndose utilizar métodos asociados como el GetName, GetEmail, etc. para obtener la información relacionada con el usuario.

En este caso se utiliza el método GetName para obtener el nombre del usuario logueado.

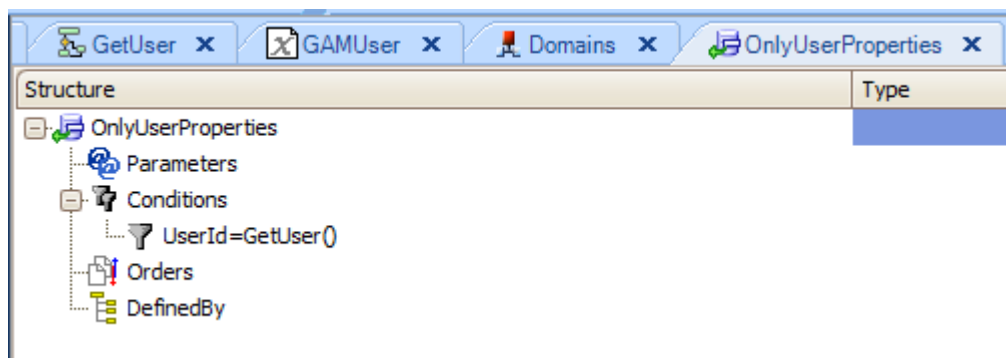
- En las Reglas del procedimiento GetUser agregar:

```
Parm(out:&UserId);
```

De esta forma al ingresar una nueva Propiedad a la aplicación, se obtendrá la información del agente inmobiliario logueado y se salvará la información en la tabla al confirmar el registro.

Cambios en WWSO Property

- Crear data selector como el siguiente:



Notar que el procedimiento **GetUser** es el definido en el paso anterior.

- Editar el WorkWithDevicesProperty y en la propiedad Data Selector del Grid del List, configurar el Data Selector recién creado, para filtrar por las propiedades del Usuario logueado. (La otra posibilidad era escribir la condición en la sección de Conditions del List)

Ejecutar la aplicación

- Ejecutar la aplicación Web, ingresar a la TRN PropertyRE, ingresar con el usuario agency01/agency01
- Verificar que ya existan registros ingresados (para esos registros se tendrá el atributo UserId vacío)
- Agregar una nueva propiedad, verificar que en atributo UserId aparece el usuario agency01.
- Ejecutar la aplicación para Smart Devices, ingresar con el usuario agency01/agency01
- Ingresar al List de Propiedades, verificar que sólo se ven las Propiedades ingresadas por el usuario agency01

10. Links de interés

Además de los videos en nuestro sitio de capacitación (<http://training.genexus.com/cursos/todos-los-cursos?es>), encontrará documentación detallada en nuestro wiki:

Smart Devices Generator – table of contents

<http://wiki.gxtechnical.com/commwiki/servlet/hwikibypageid?20516>