


## Control: Canvas



14-Controls.Canvas.And.Transformations.sp



Developing the mobile application

**Designing: Controls in layouts**  
Absolute positioning (Canvas) and Transformations

Cecilia Fernández | GeneXus Training

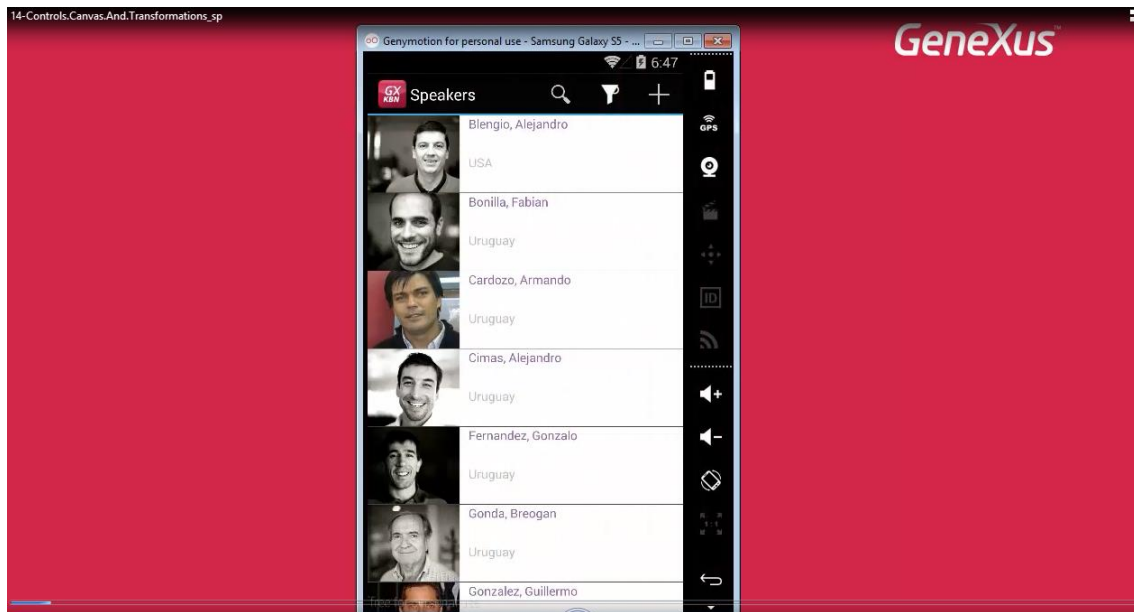
14-Controls.Canvas.And.Transformations.sp

Designing

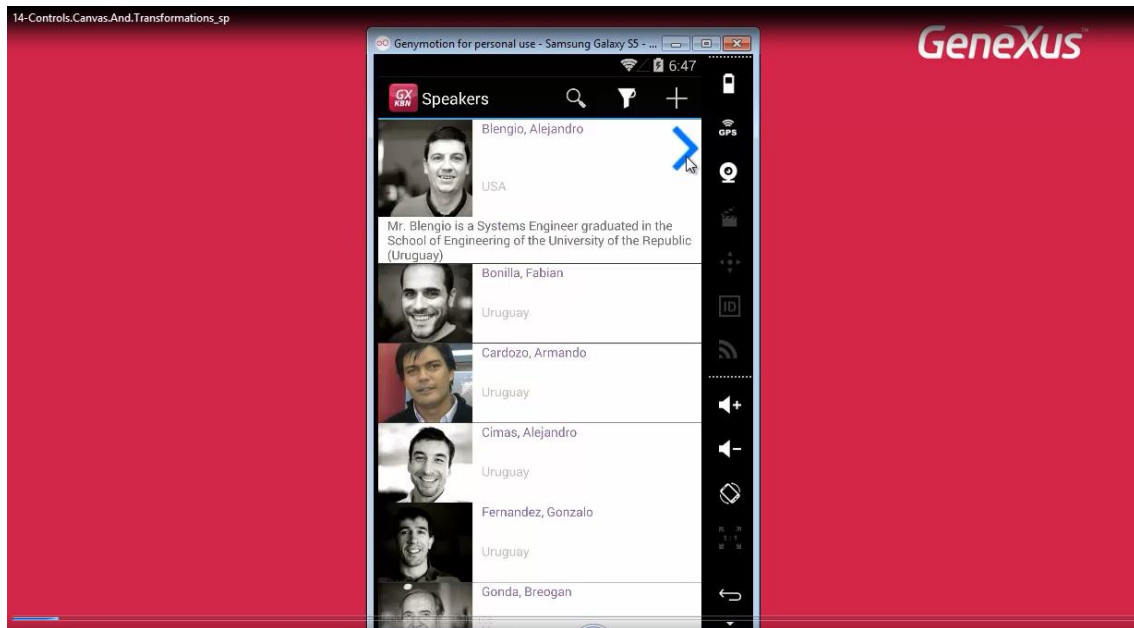
GeneXus

The diagram shows a red box labeled "Controls in layouts" on the left. To its right, a large red curly bracket groups three horizontal bars: two gray bars at the top and a purple bar at the bottom labeled "Canvas & Transformations". Three small gray squares are positioned between the two gray bars.

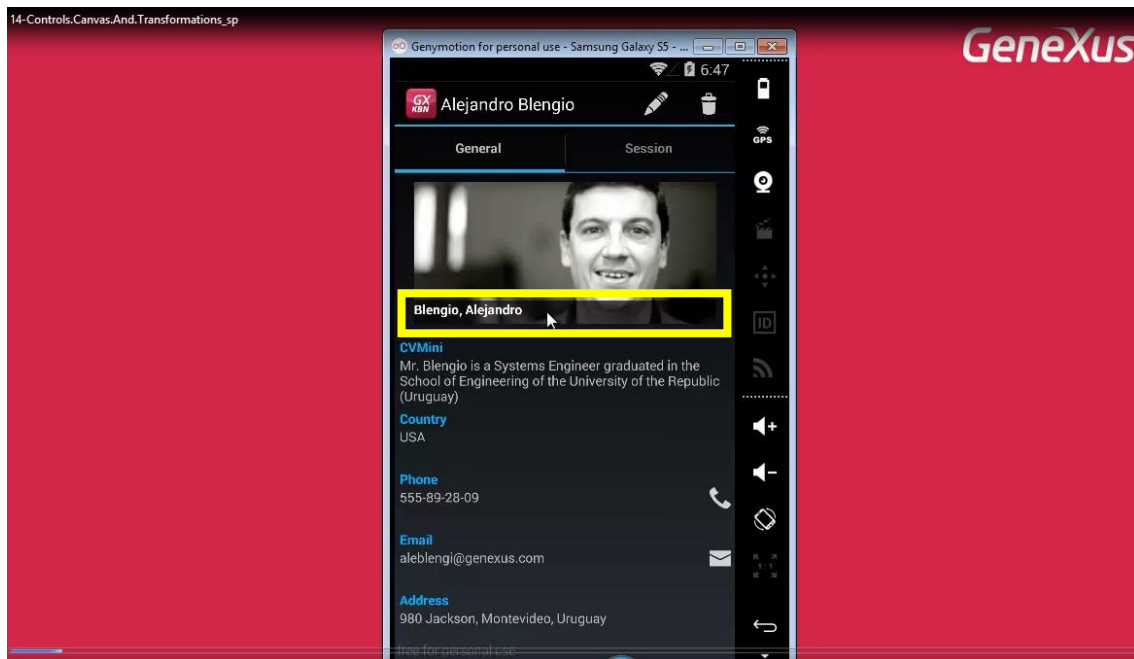
Estudiaremos la forma de lograr posicionamiento absoluto para los controles en la pantalla, permitiendo por ejemplo superponer controles. También veremos la posibilidad de desplazar, escalar, girar o modificar el tamaño de los controles.



Así, supongamos por ejemplo que al visualizar la información de un orador

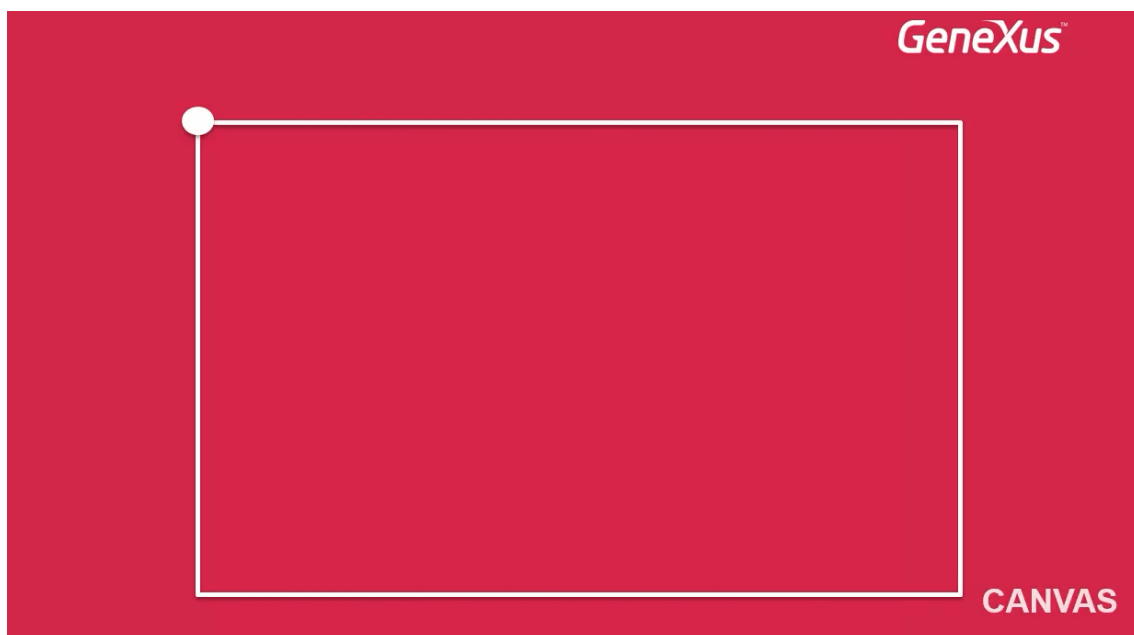


queremos que el nombre completo



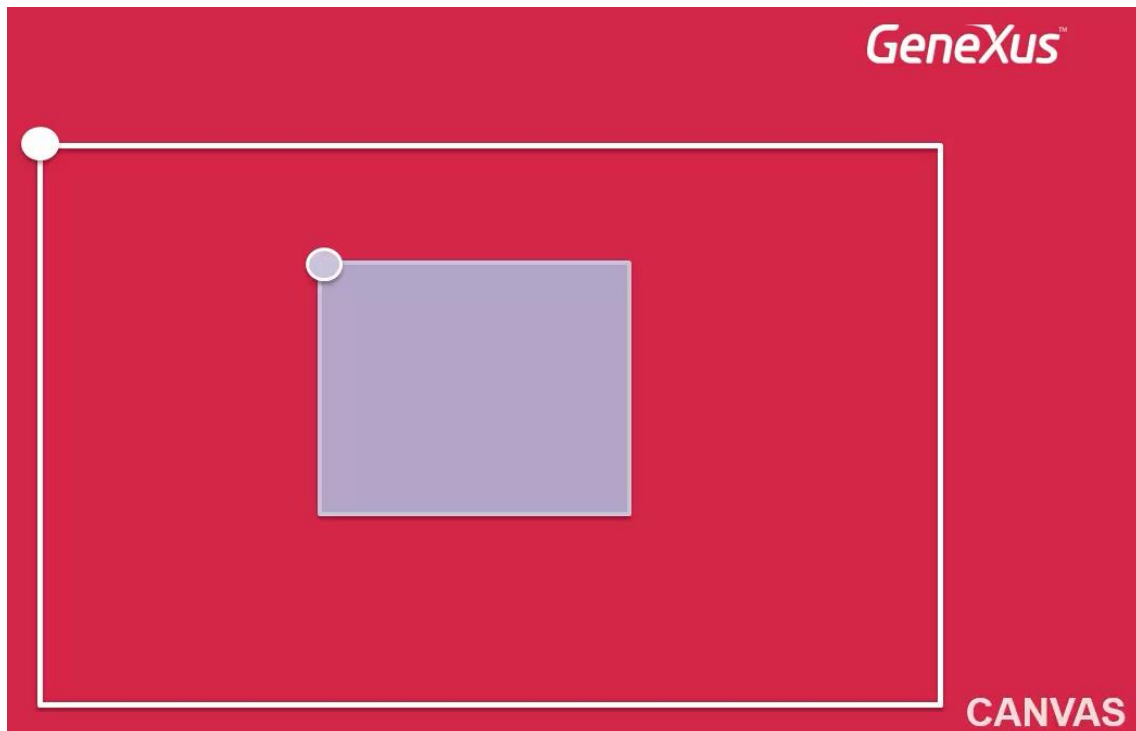
se visualice sobre la foto

Para ello, necesitaremos definir un contenedor de controles con posicionamiento absoluto

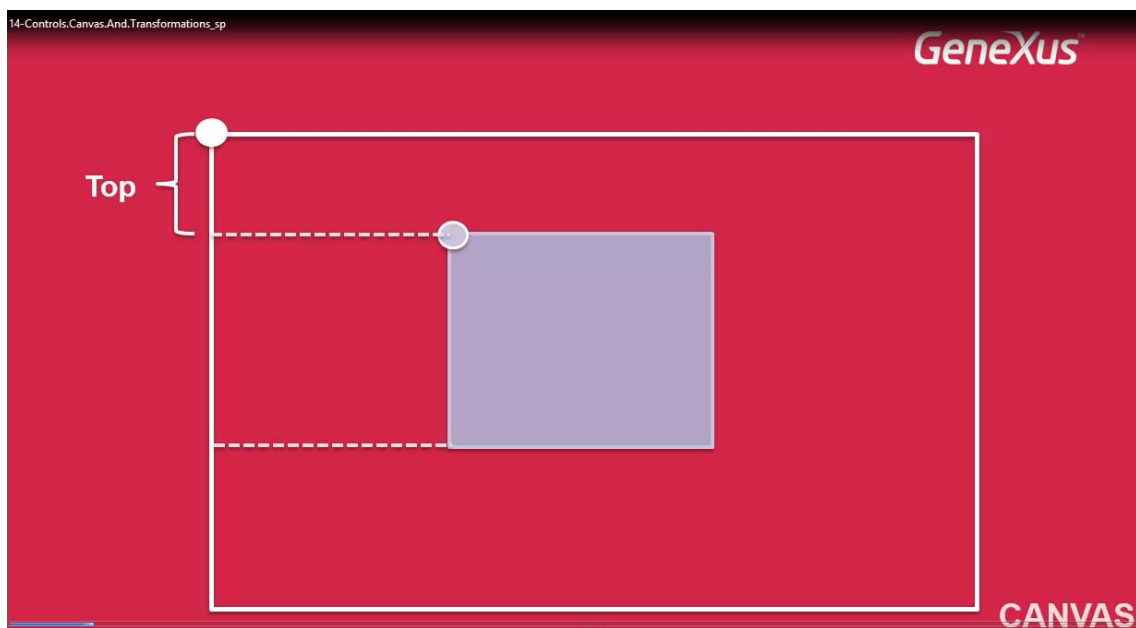


Este control es **en todo, como una tabla** pero recibe un nombre especial: CANVAS.

Teniendo un control CANVAS en el layout, todos los controles que insertemos dentro

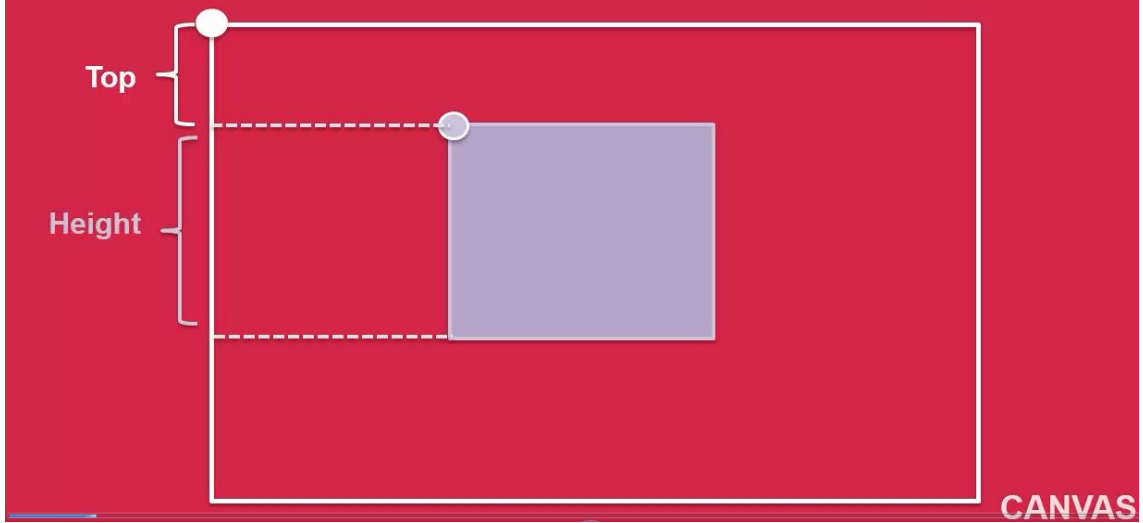


tendrán posicionamiento absoluto con respecto a los bordes del CANVAS. Así, si insertamos **este control**, tendremos que indicarle la distancia del borde superior: Top

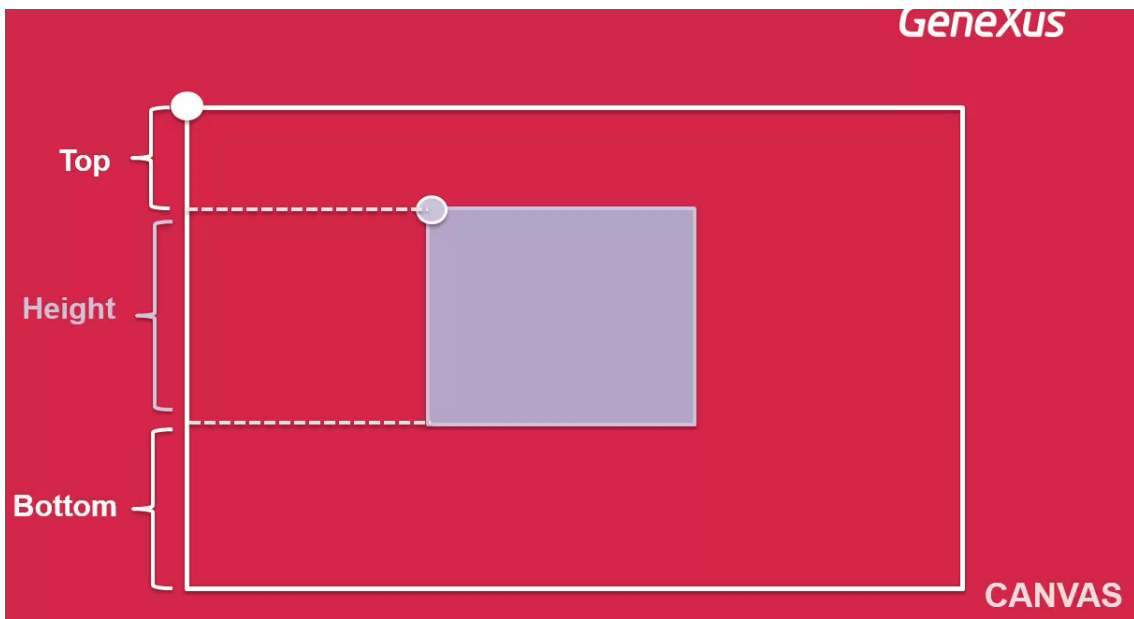


Alto del control: Height

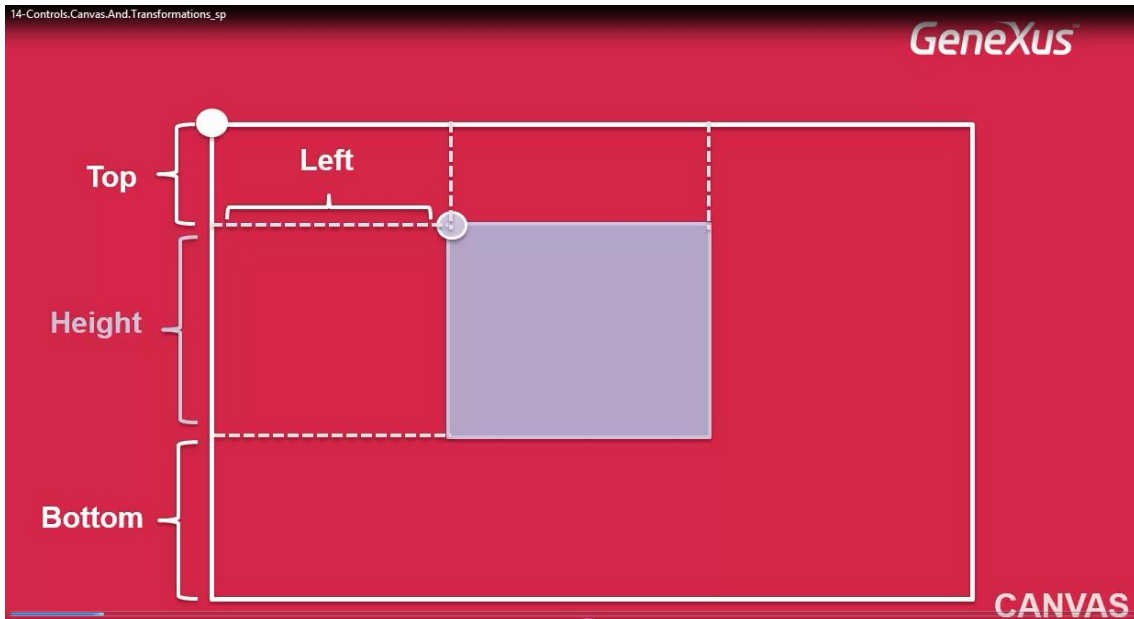




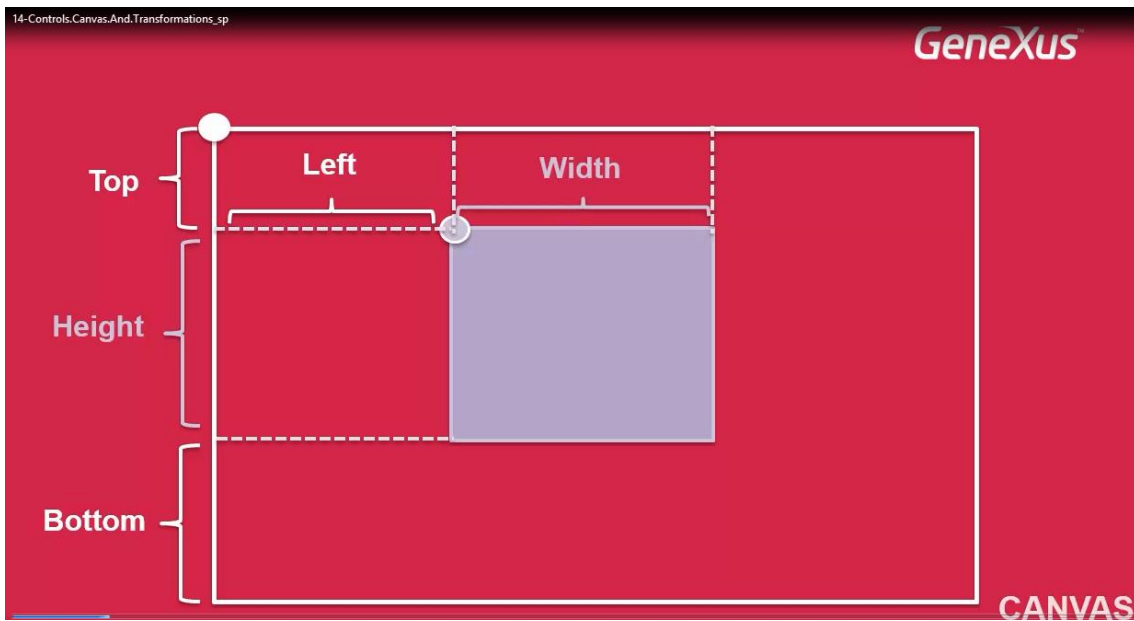
y la distancia del borde inferior: Bottom



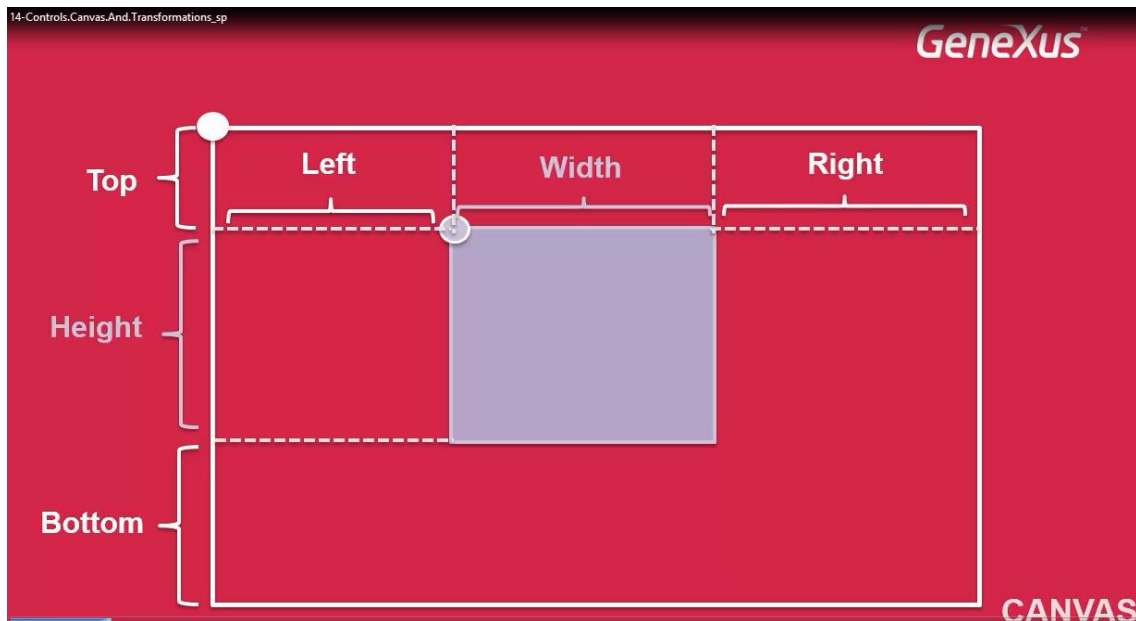
Y por otro lado, la distancia de la izquierda: Left



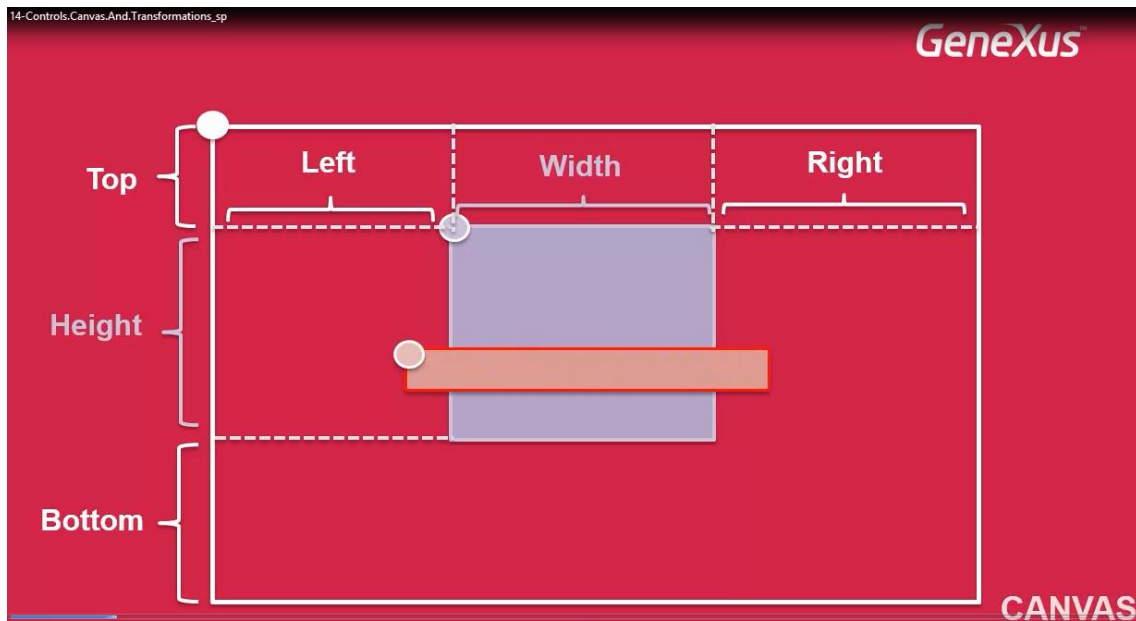
el ancho del control: Width



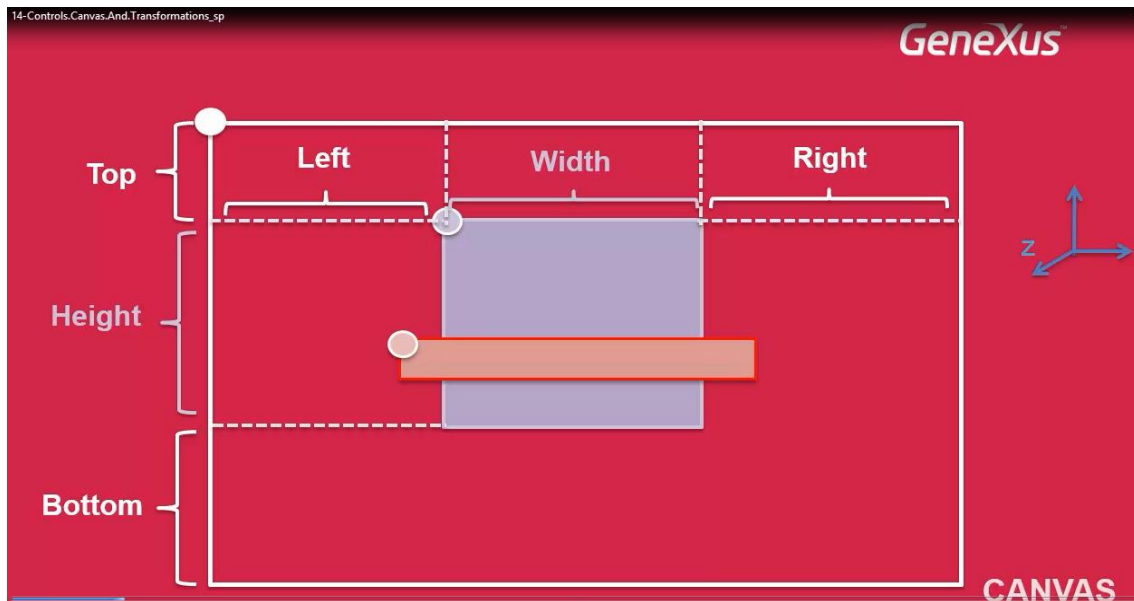
y la distancia del lado derecho: Right



Asignando estas medidas convenientemente es que podemos superponer los controles



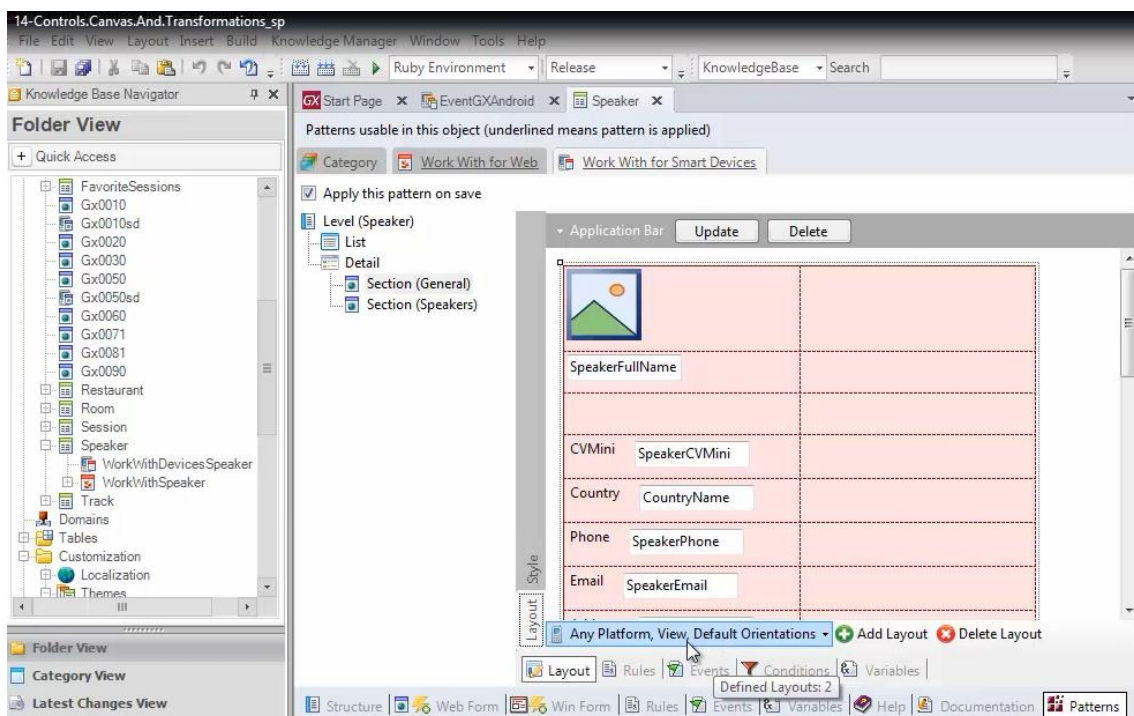
Necesitaremos por tanto, asignar también, un valor para el eje Z



Que determinará la capa de cada control para saber cuáles estarán encima de cuáles.

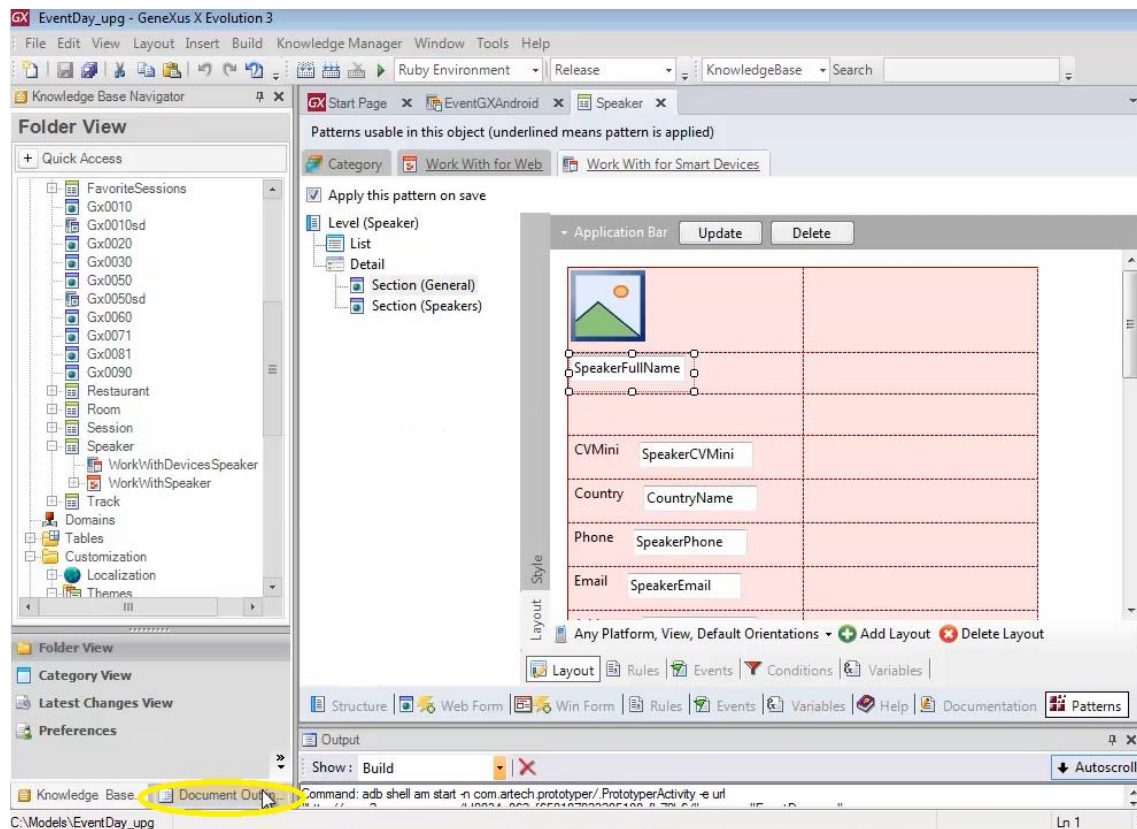
Asume valores discretos: cero es la capa de más abajo, uno es la siguiente, y así, sucesivamente.

Entonces, si vamos a GeneXus

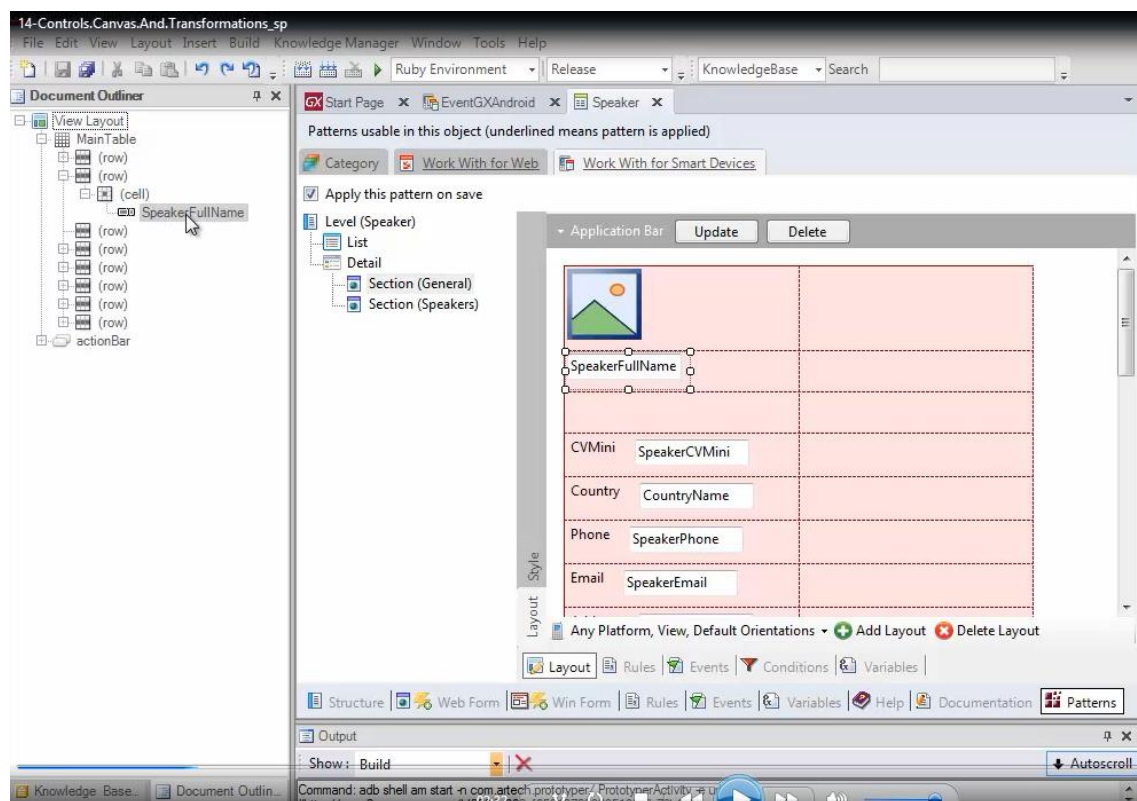


estamos en la Section General, en la pantalla en modo View... y tenemos la imagen del orador y el nombre completo en 2 filas diferentes de la tabla.

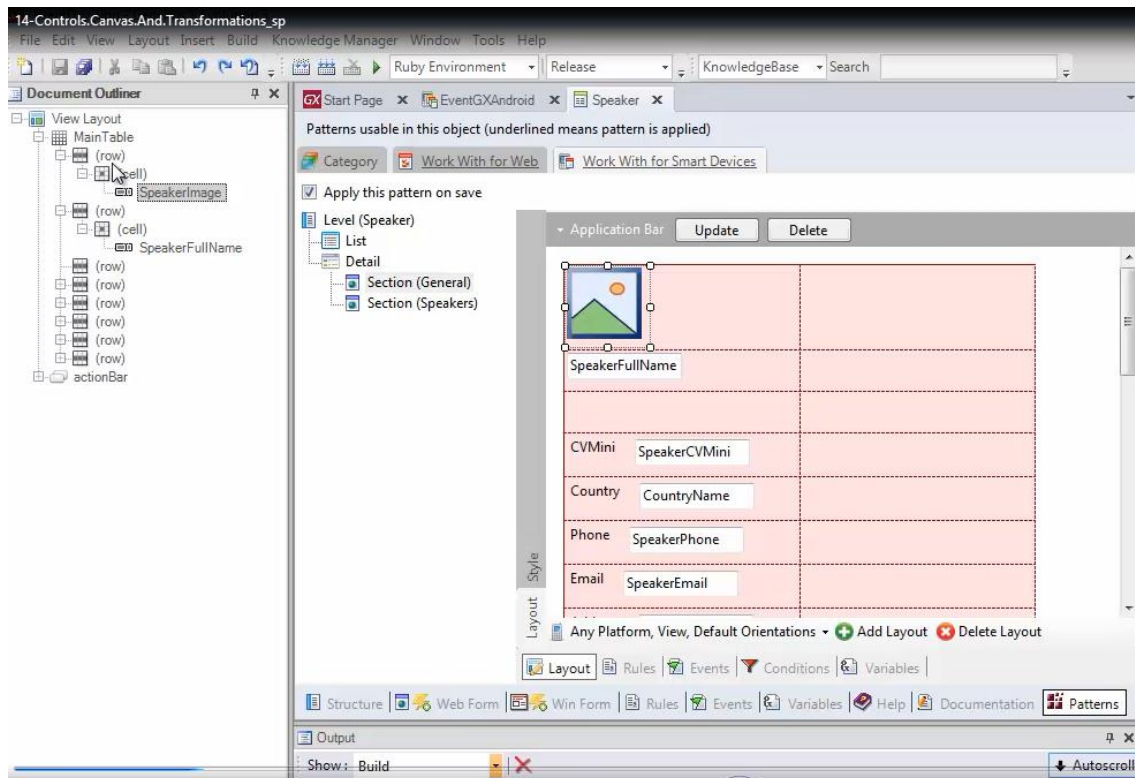
Podemos editar el Document Outliner, que nos va a ser muy útil en el caso de Canvas



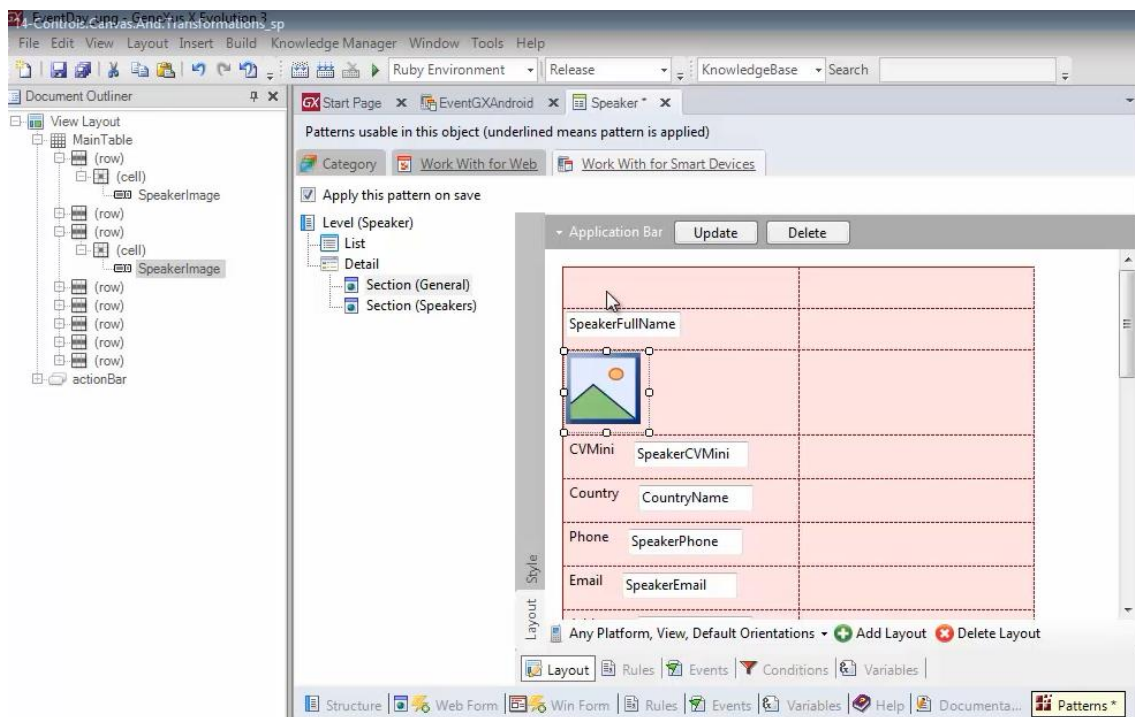
Y ver entonces que SpeakerFullName está en una de las celdas, de una de las filas:



Y la imagen está en la primera:

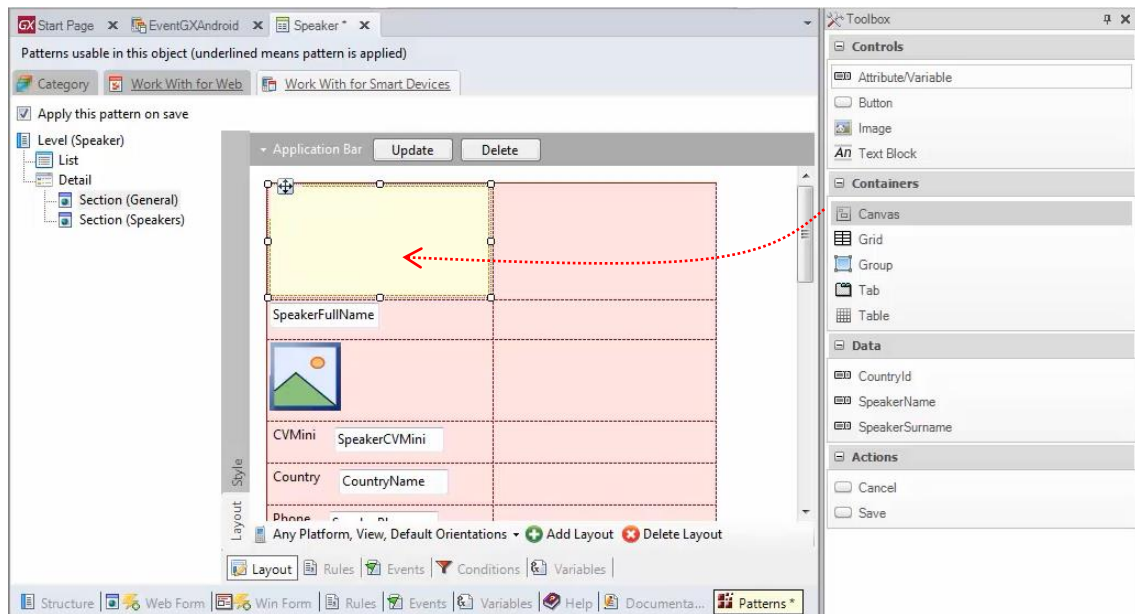


Vamos a mover la imagen provisoriamente para abajo para hacer espacio

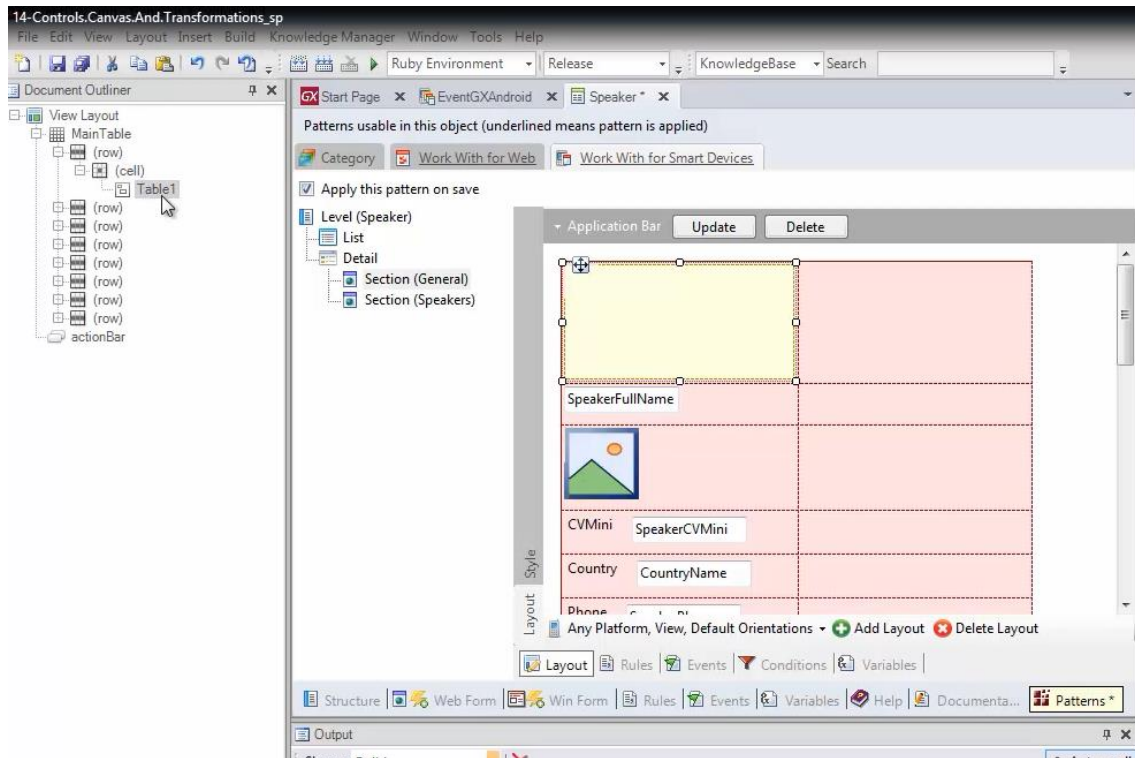


y en esta primera fila vamos a insertar un control : canvas



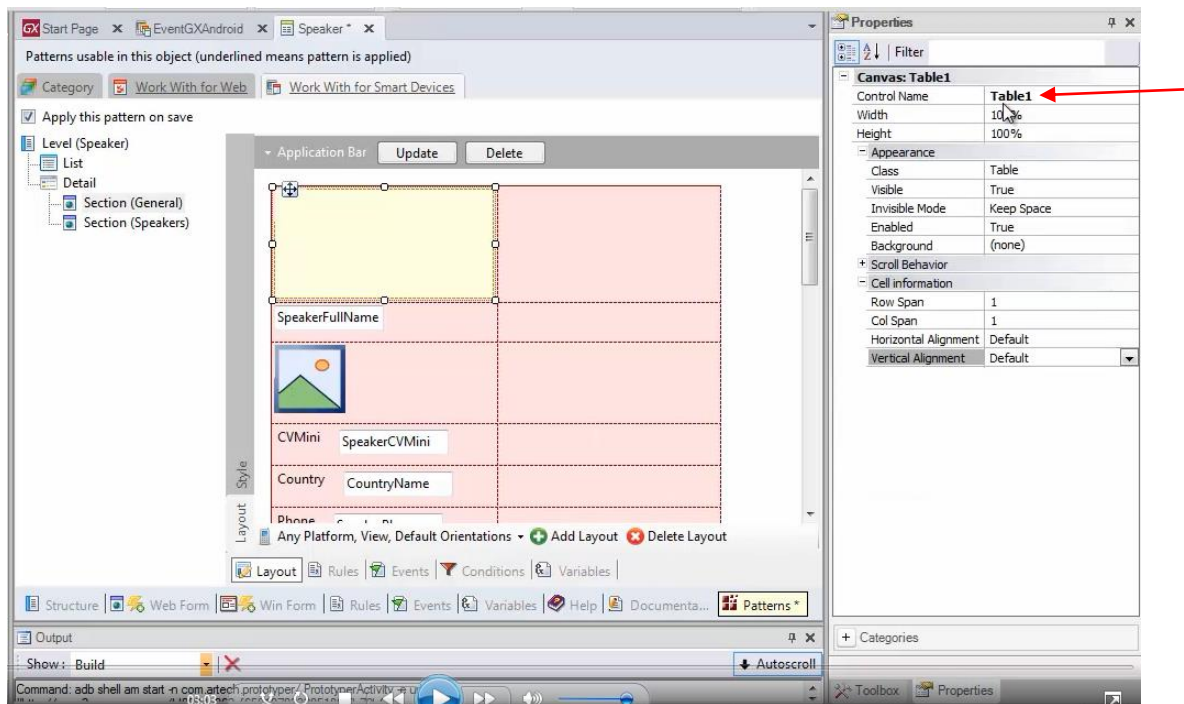


Si observamos ahora la fila en el Document Outliner

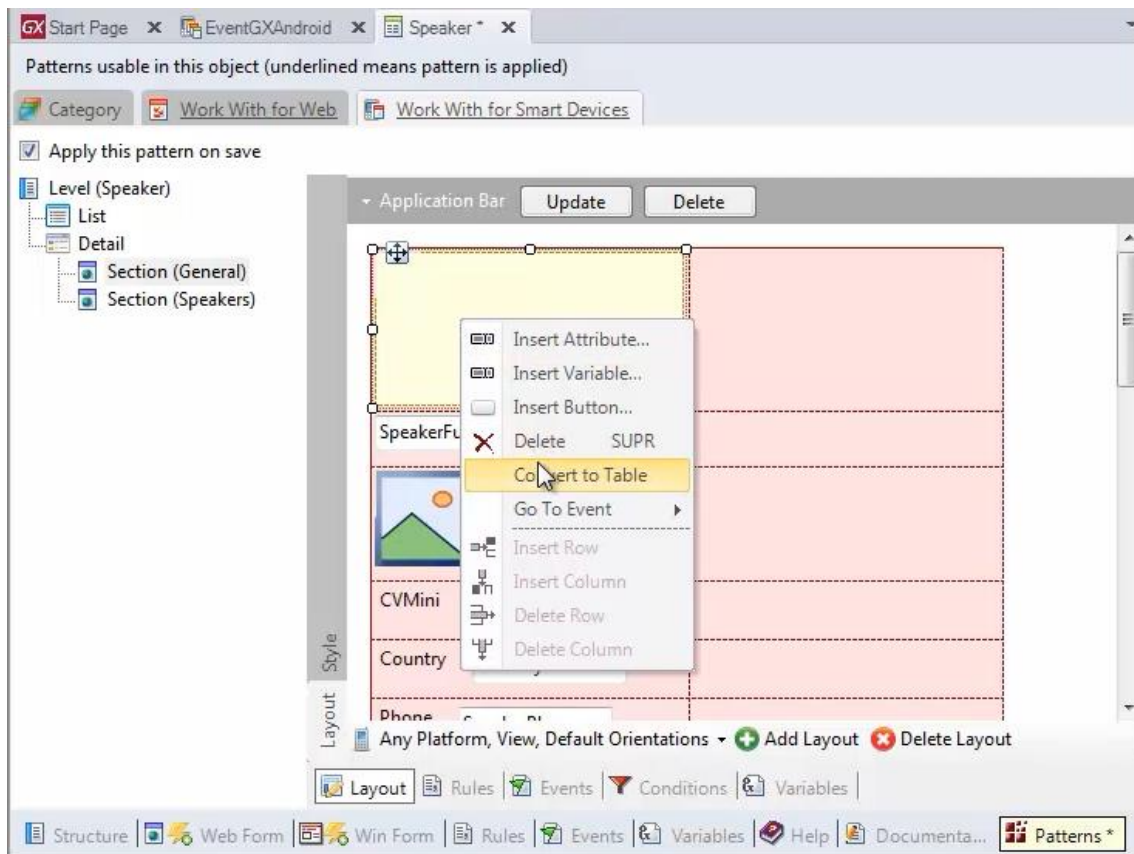


vemos que aparece con el nombre **Table1**

y si vamos a editar las propiedades, también vemos que tiene el nombre: **Table1**

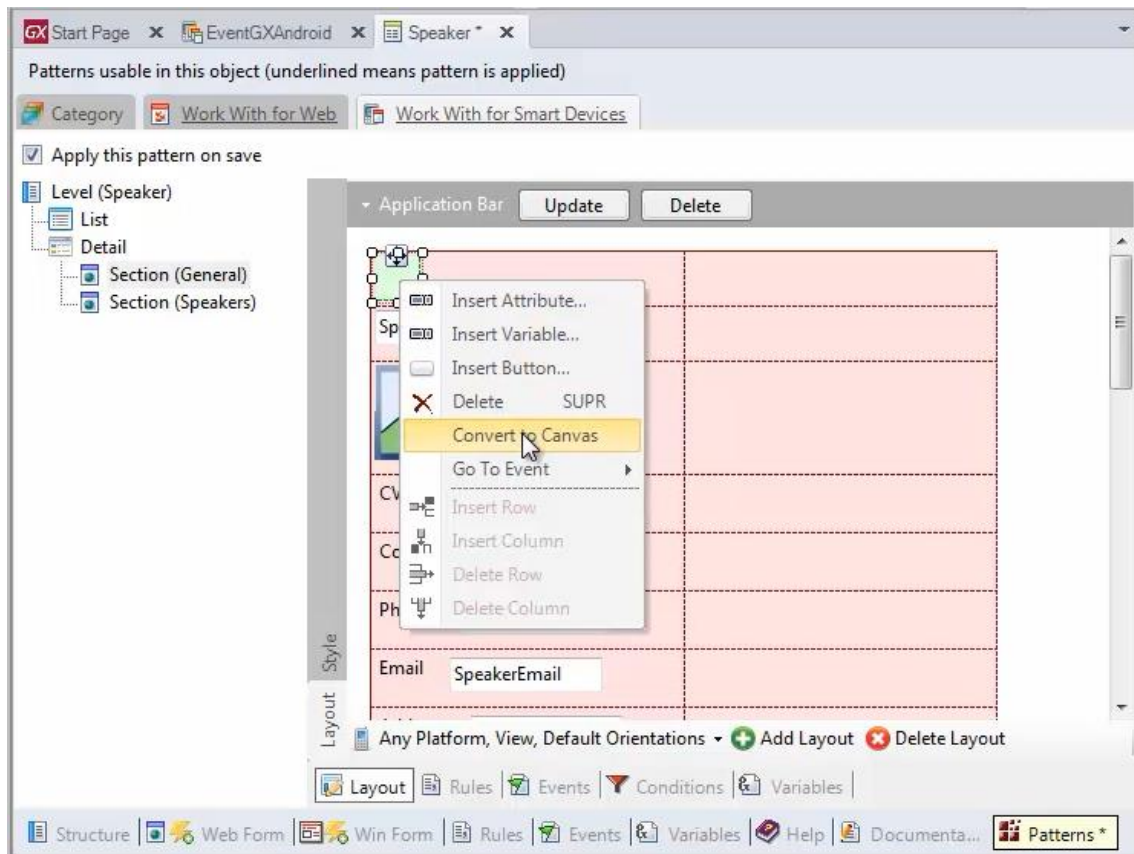


De hecho, en todo momento podemos convertir

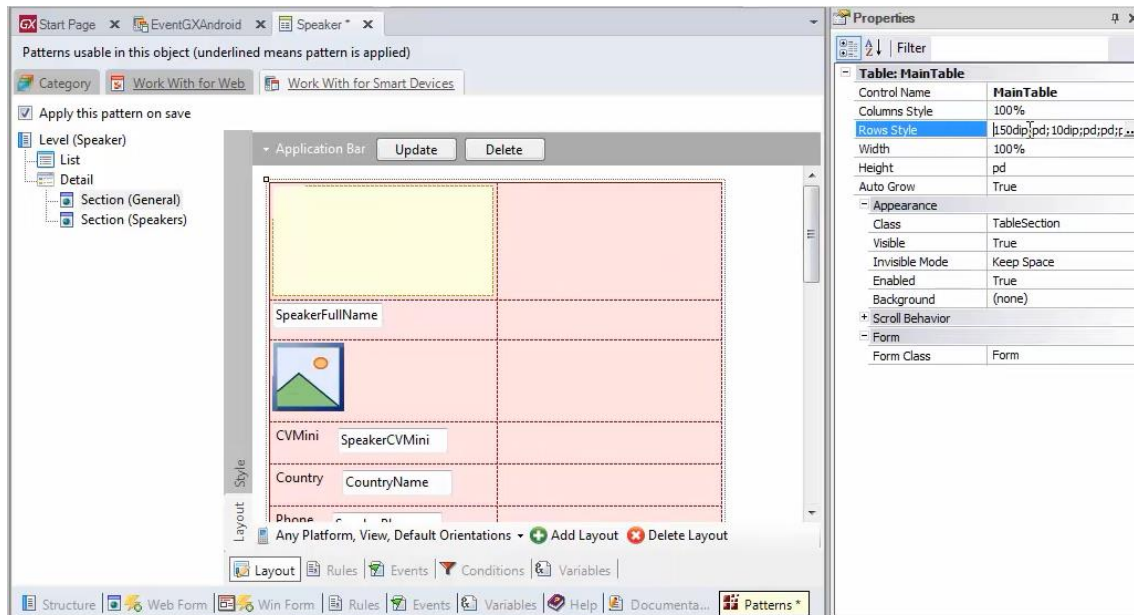


Un control CANVAS a una TABLA común y corriente y viceversa:

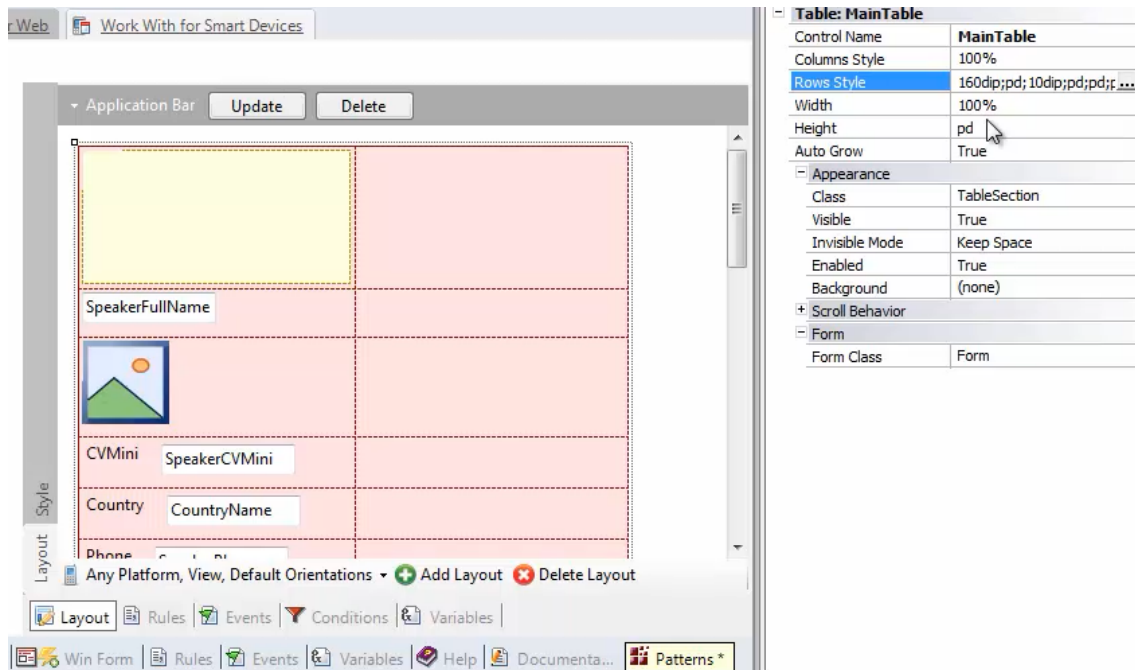




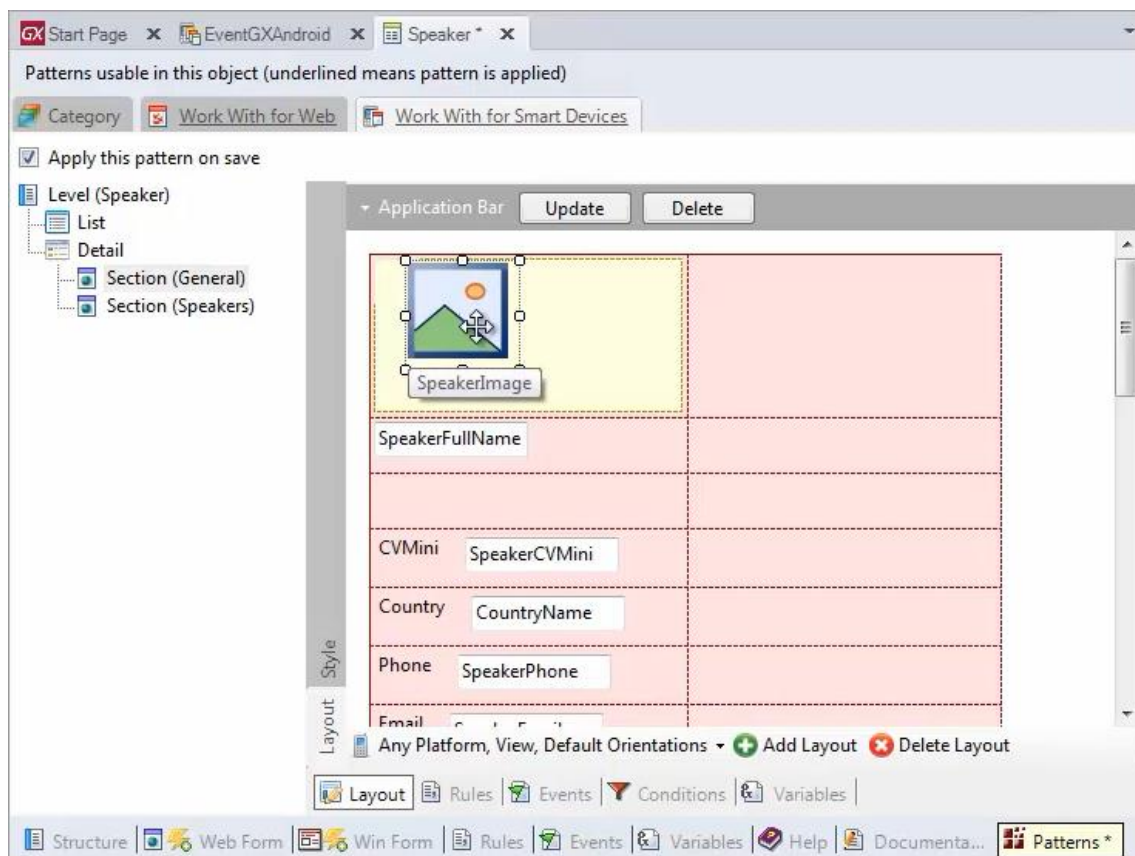
Vamos a ver el tamaño que ocupaba la 1era fila: 150 dips



Vamos a agregarle 10 dips para que nos quede de 160:

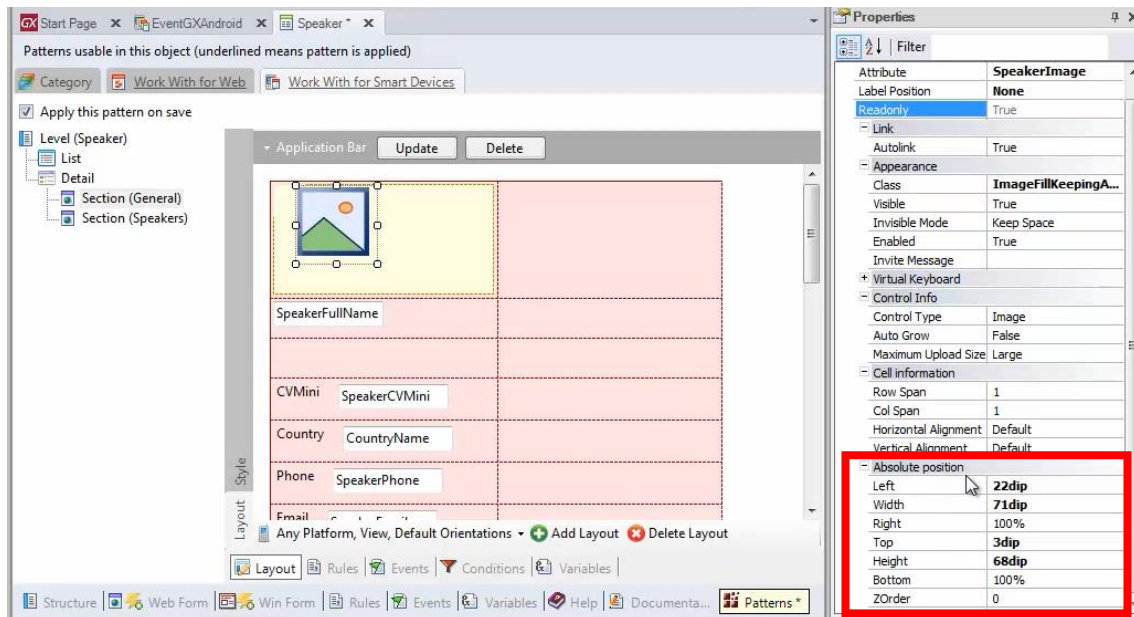


Y ahora sí, vamos a insertar la imagen



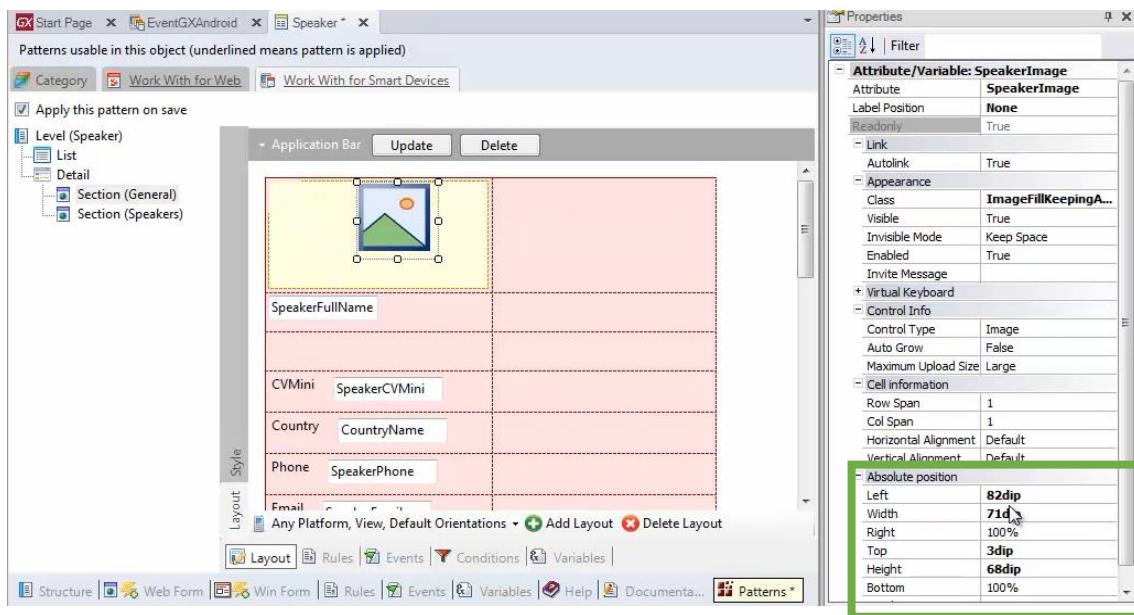
dentro del contenedor CANVAS.

Si observamos con atención todas las propiedades de la imagen, se le están sumando estas bajo el grupo **Absolute Position**:



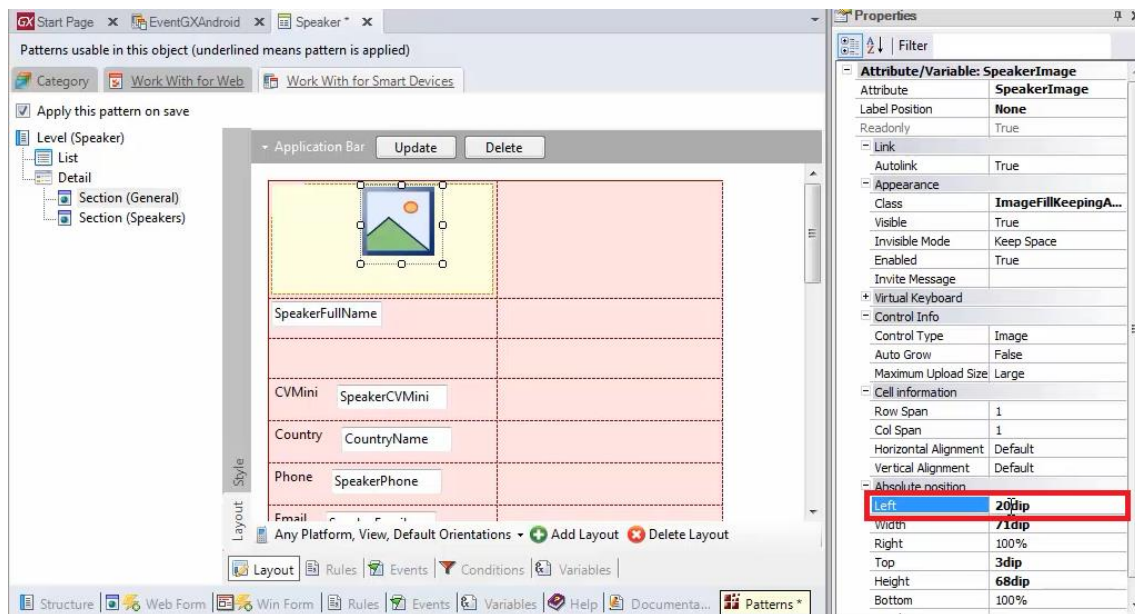
Estas propiedades son ni más ni menos que las que veíamos recién para poder especificar la posición absoluta del control en relación al contenedor.

Así, si nosotros movemos la imagen

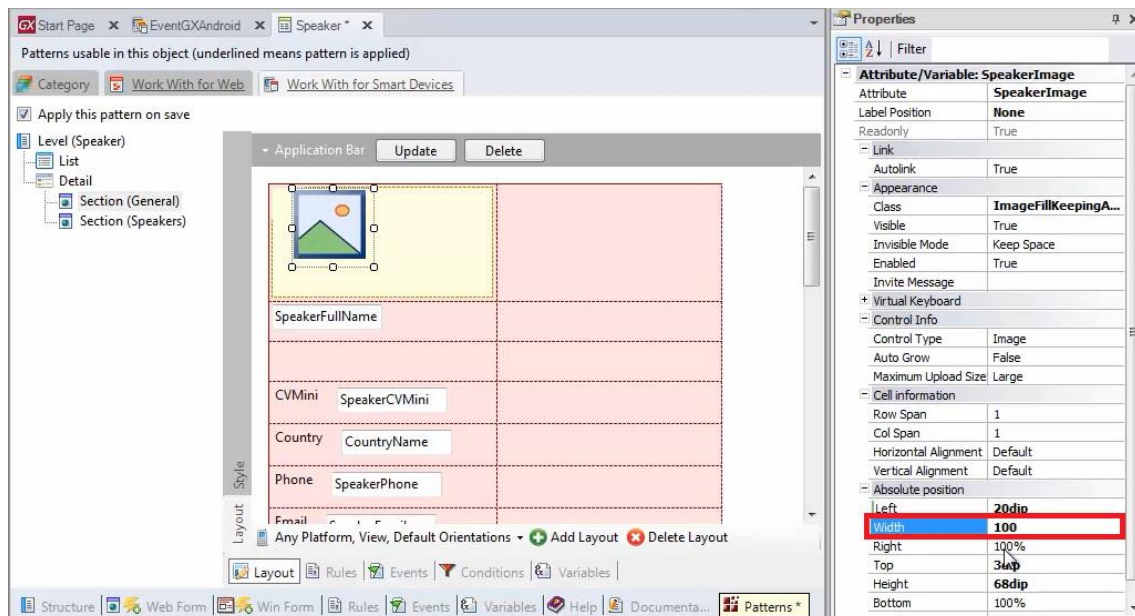


observemos que se modificaron estos valores.

Vamos a especificar que queremos que la imagen del orador esté a 20 dips de la izquierda

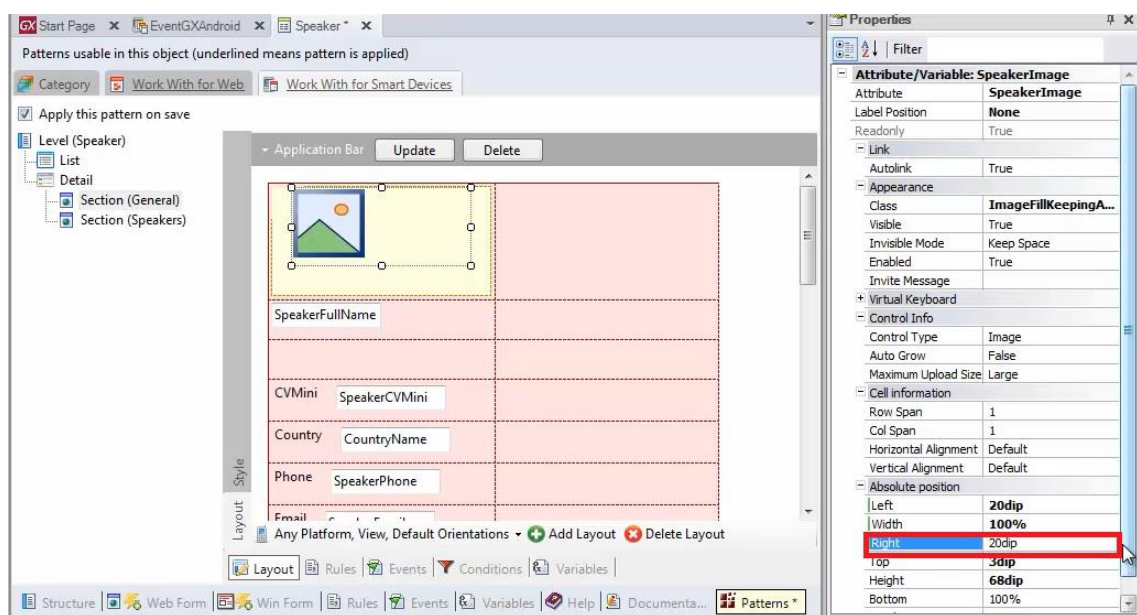


que su ancho corresponda al 100% del ancho del contenedor

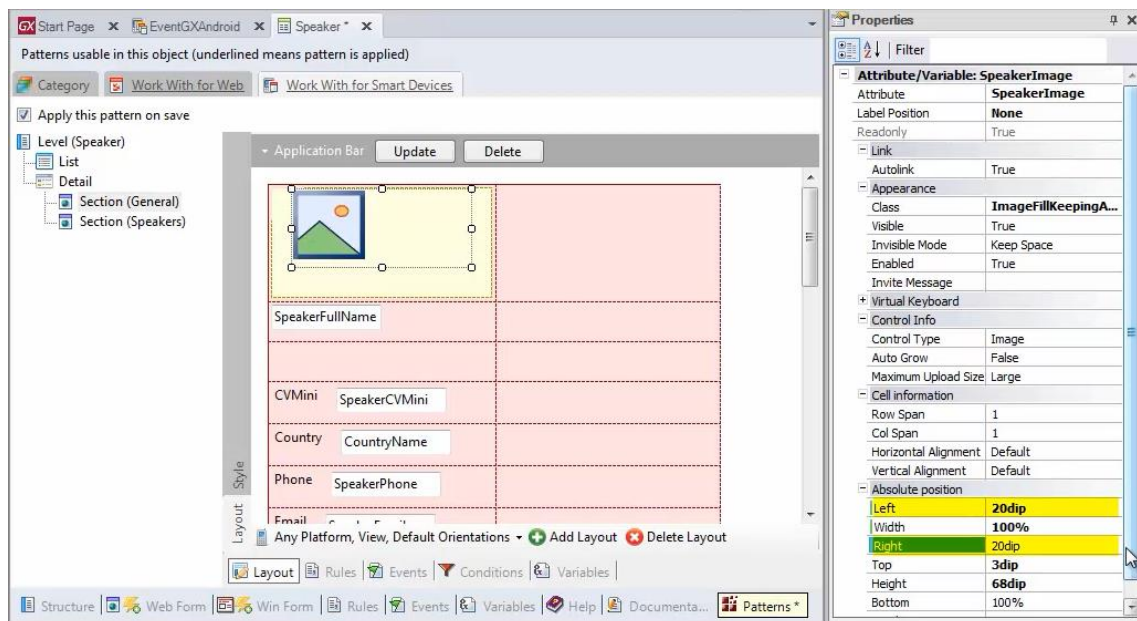


pero que esté además a 20 dips del margen derecho

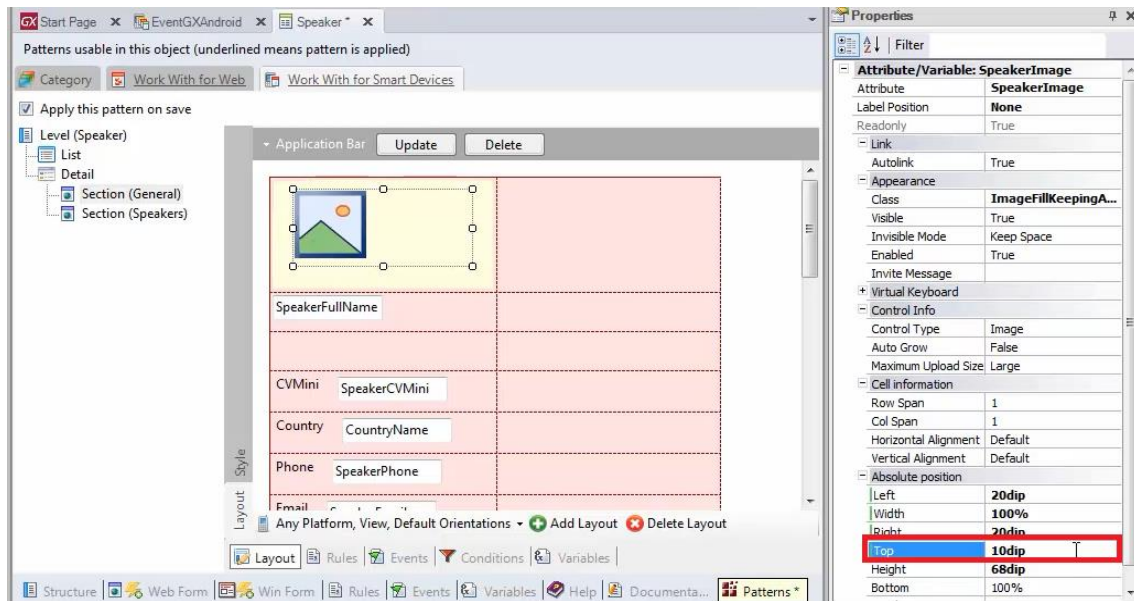




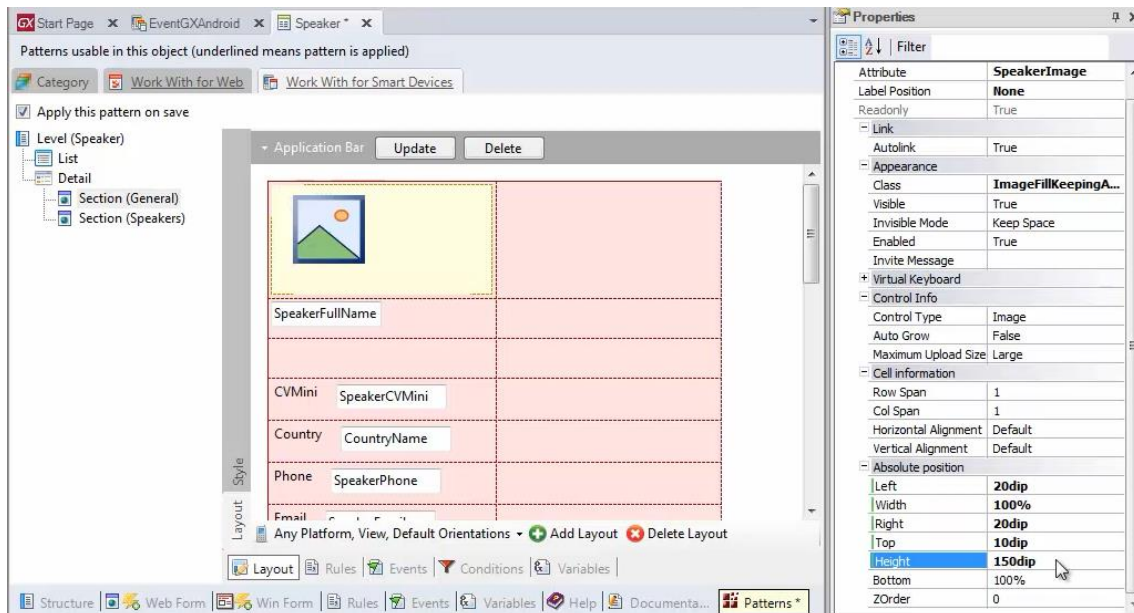
Recordemos que el porcentaje siempre se aplica sobre lo que resta de quitar los valores fijos. En este caso, el Width será del 100% del ancho total del Canvas, menos 40 dips (los 20 de la izquierda y los 20 de la derecha).



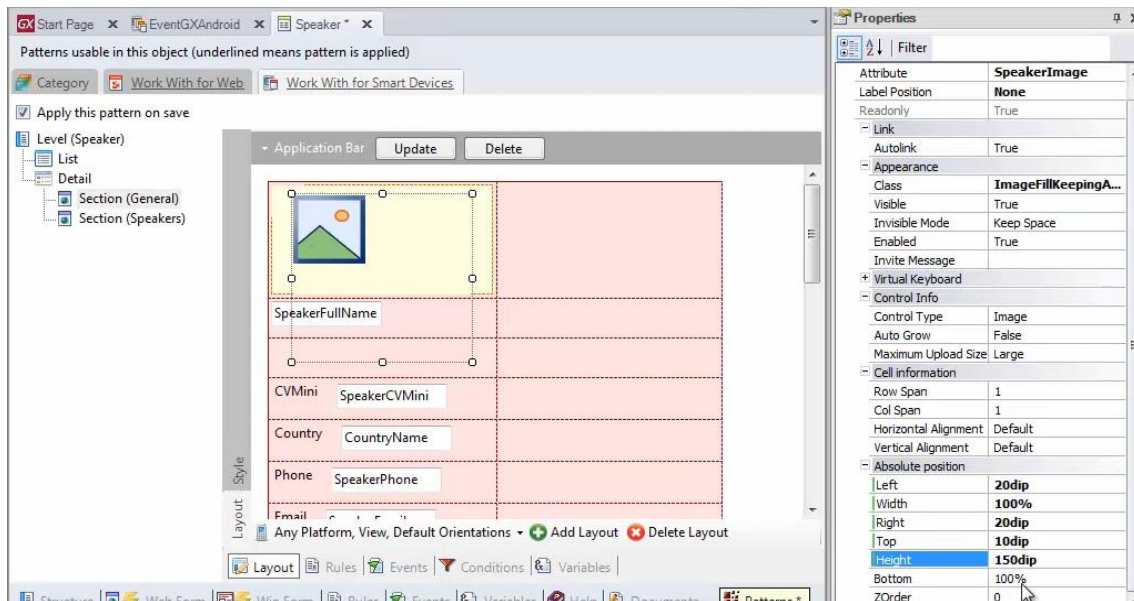
Por otro lado vamos a querer que esté a 10 dips del borde superior



que el alto sea de 150 dips

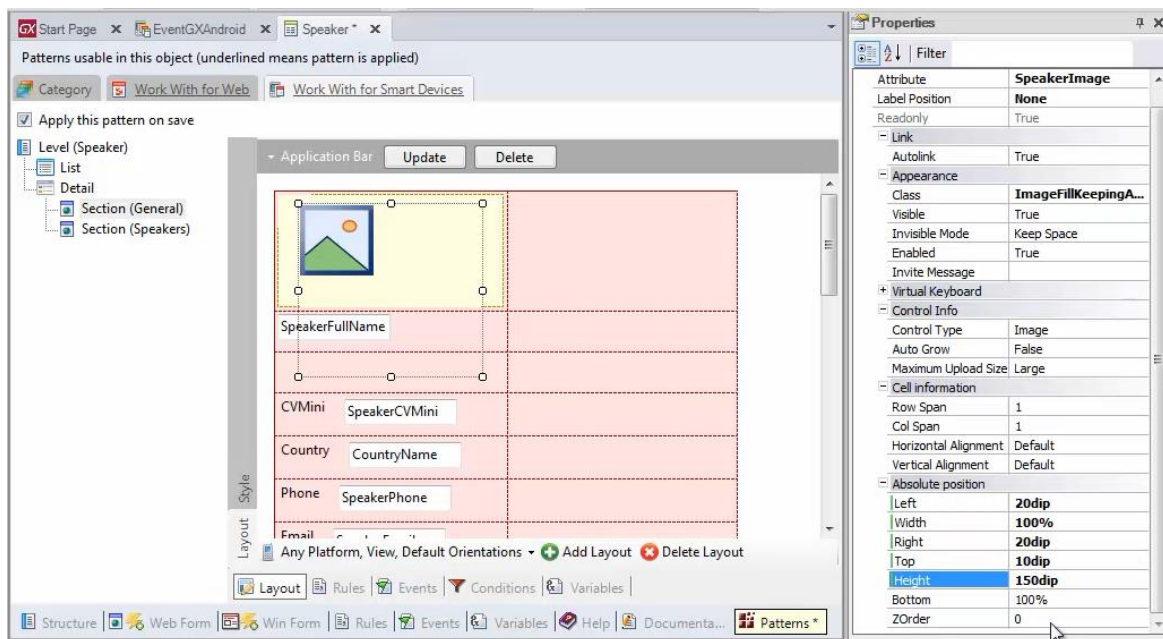


Y la distancia del borde inferior sea el 100% restante



Como la celda era de 160 dips y el Height y Top sumados dan 160, entonces la distancia del borde inferior será cero.

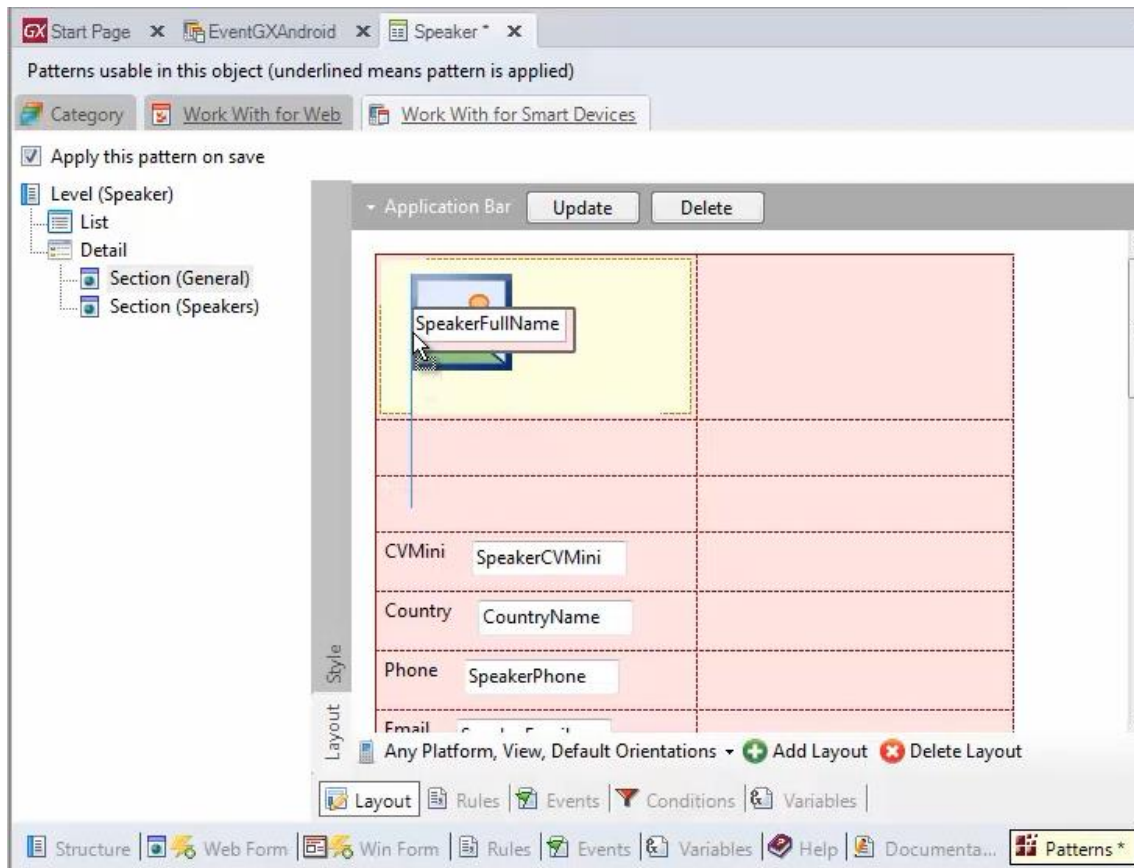
Observemos que orden en relación al eje Z, está en cero.



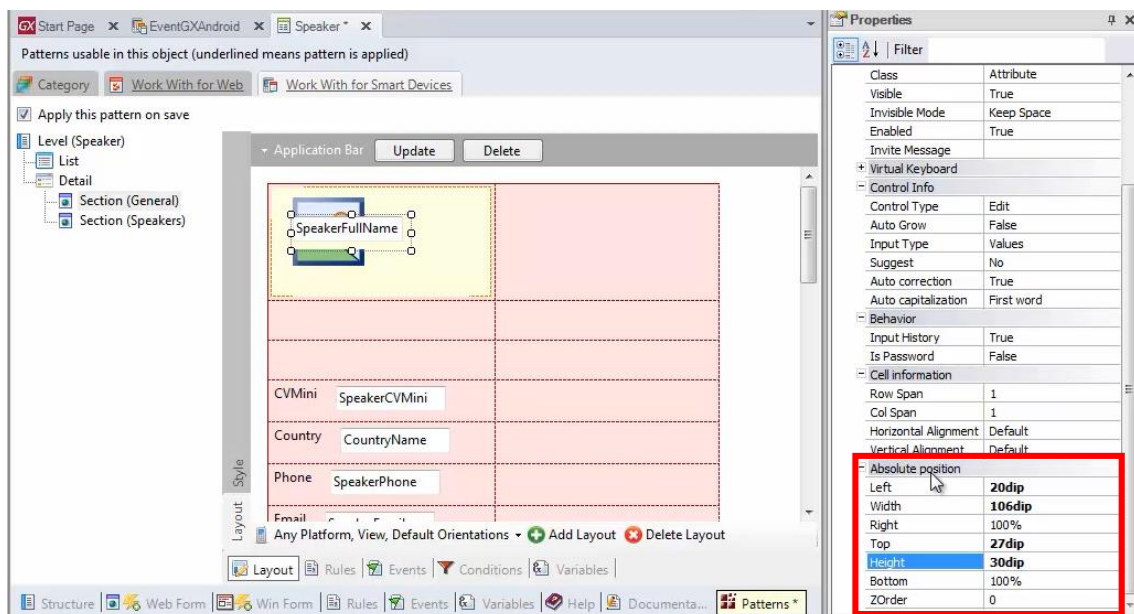
Es decir, este control será el que aparecerá más abajo.

Ahora queremos también mover el atributo SpeakerFullName para adentro del control CANVAS



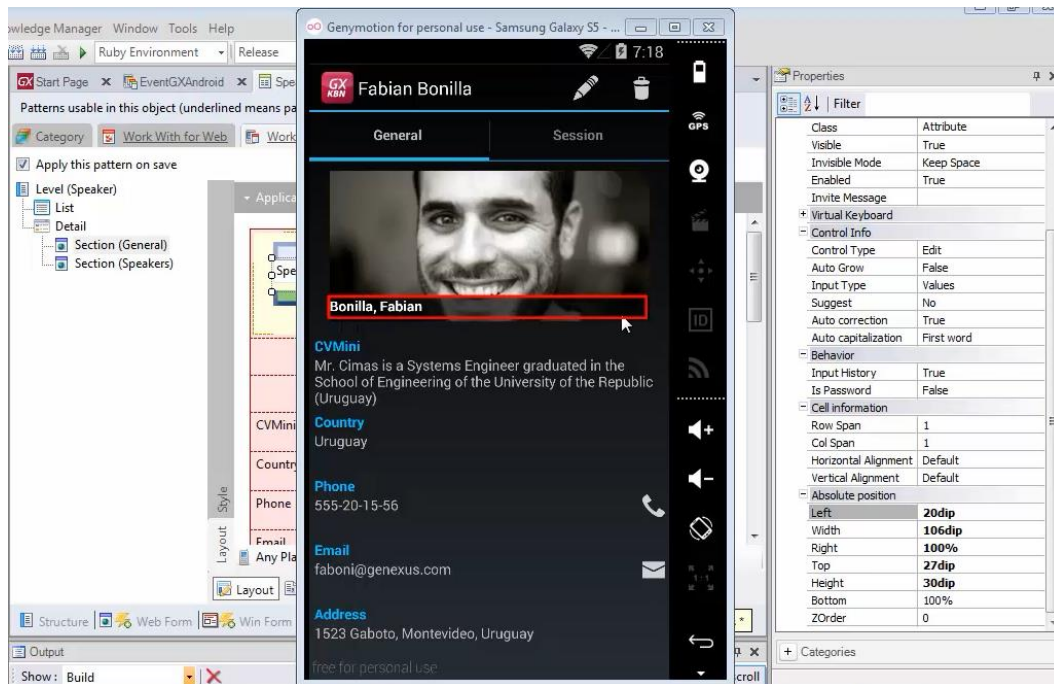


Observemos que otra vez se han agregado estas propiedades del grupo: Posición Absoluta

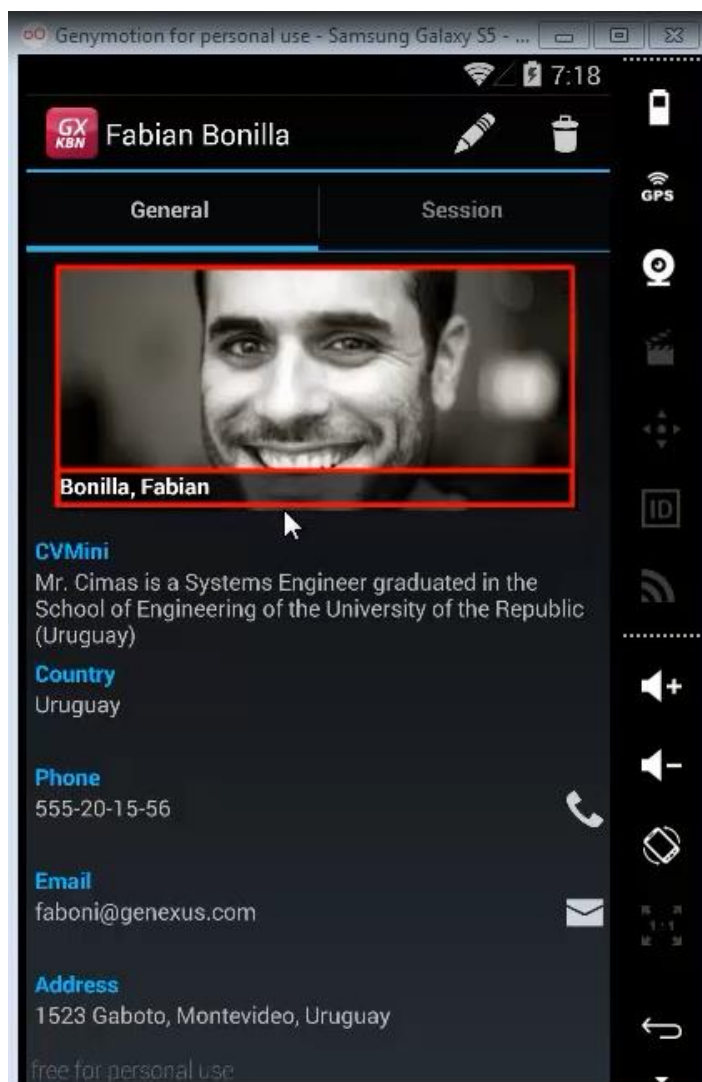


Ahora vamos a configurar estas propiedades, teniendo que queremos que este control

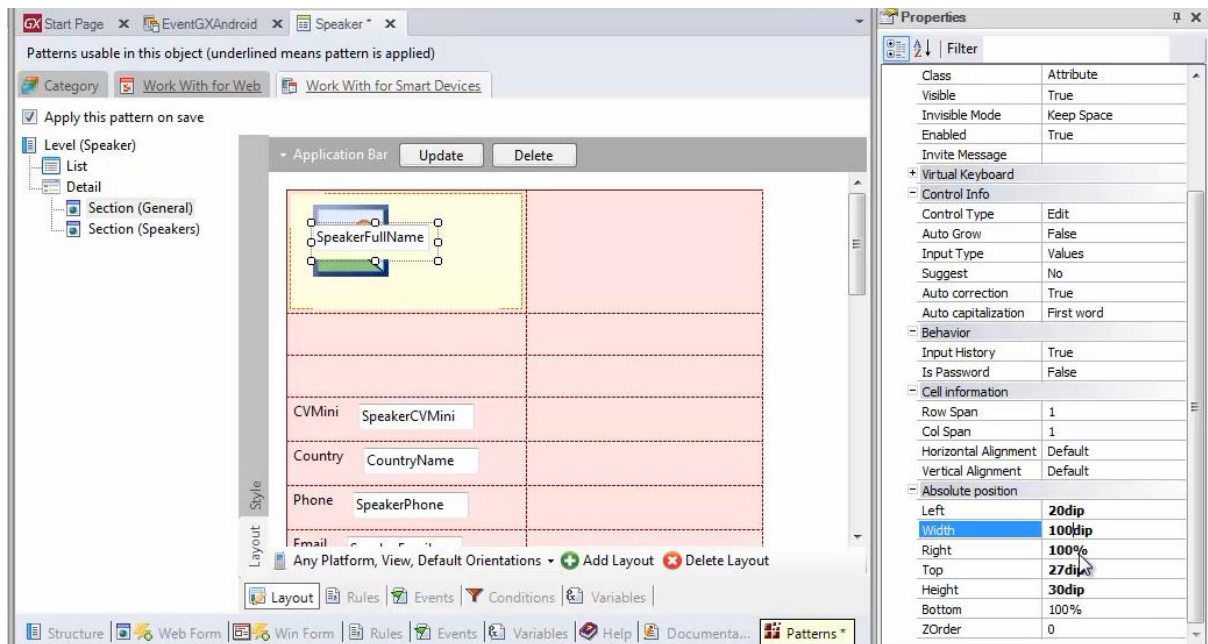




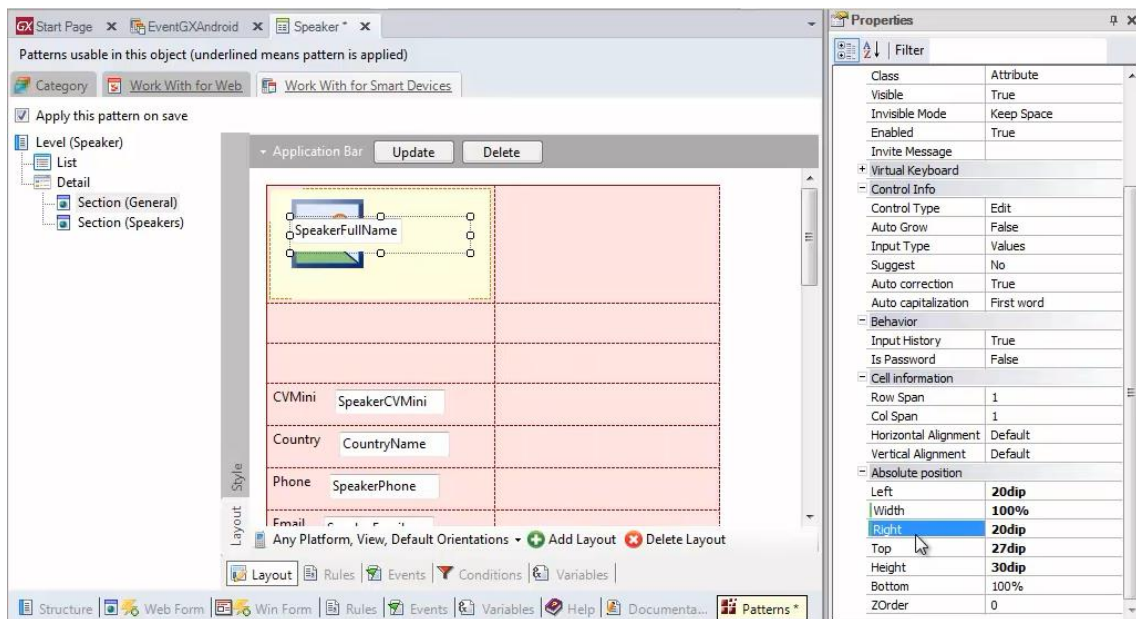
se superponga absolutamente a la imagen del orador, de acuerdo al margen inferior



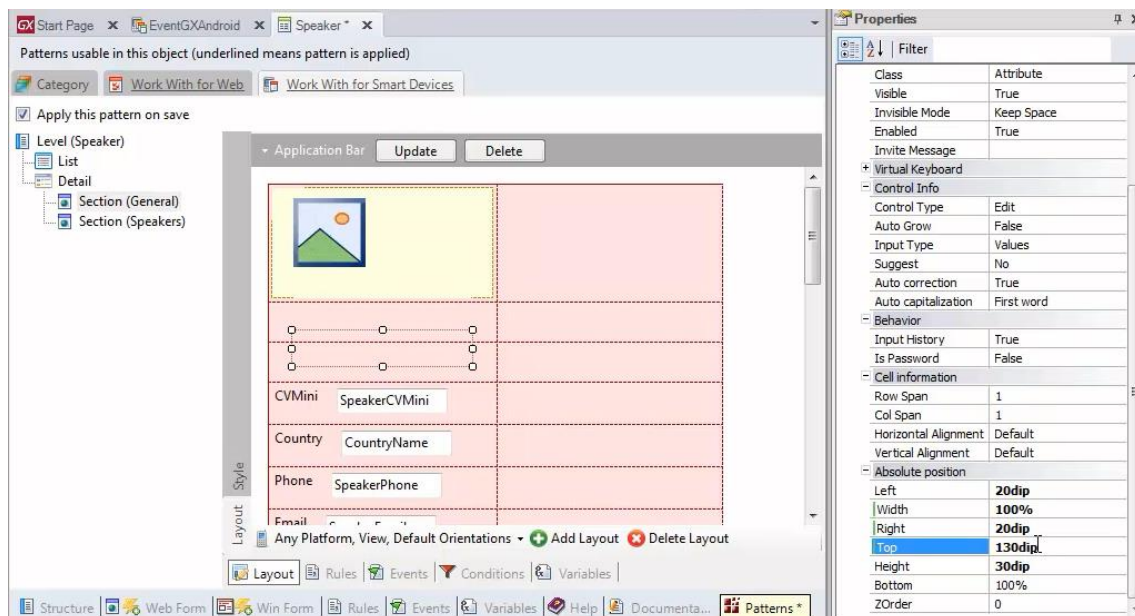
Por lo que el Width será también del 100%



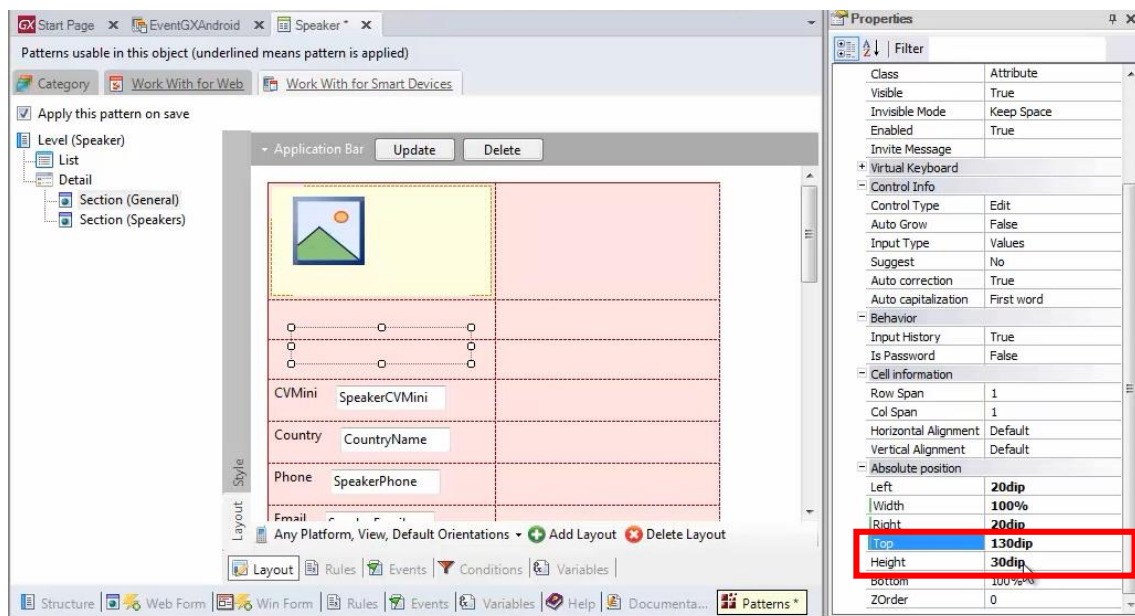
el borde derecho será de 20 dips



El margen de arriba será de 130 dips

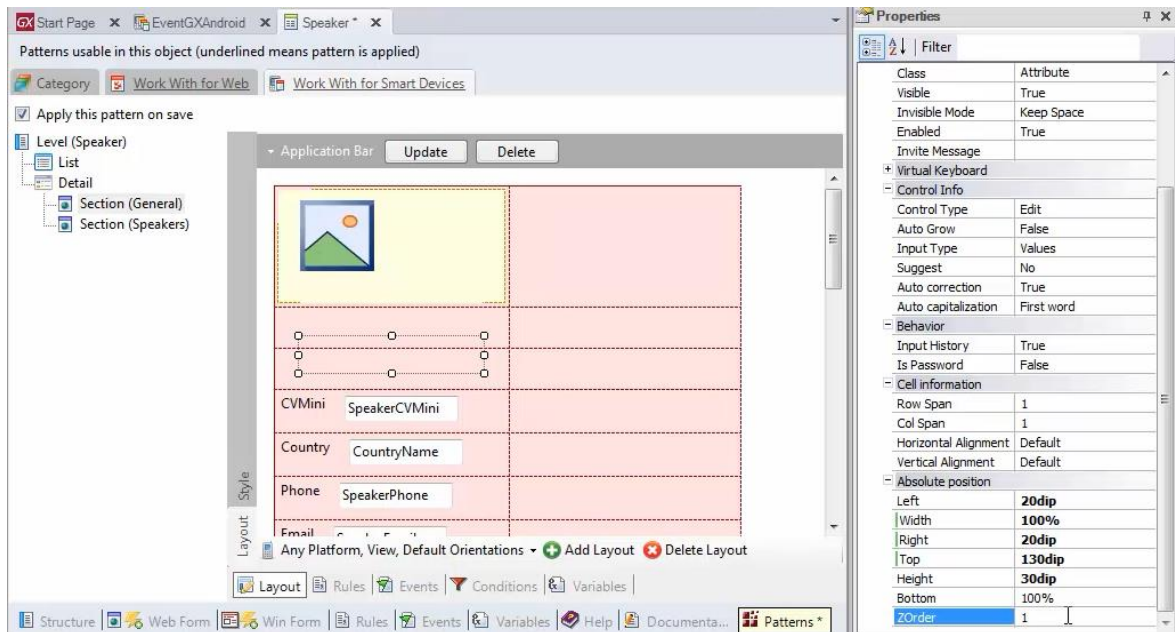


dado que el alto del control será de 30 dips



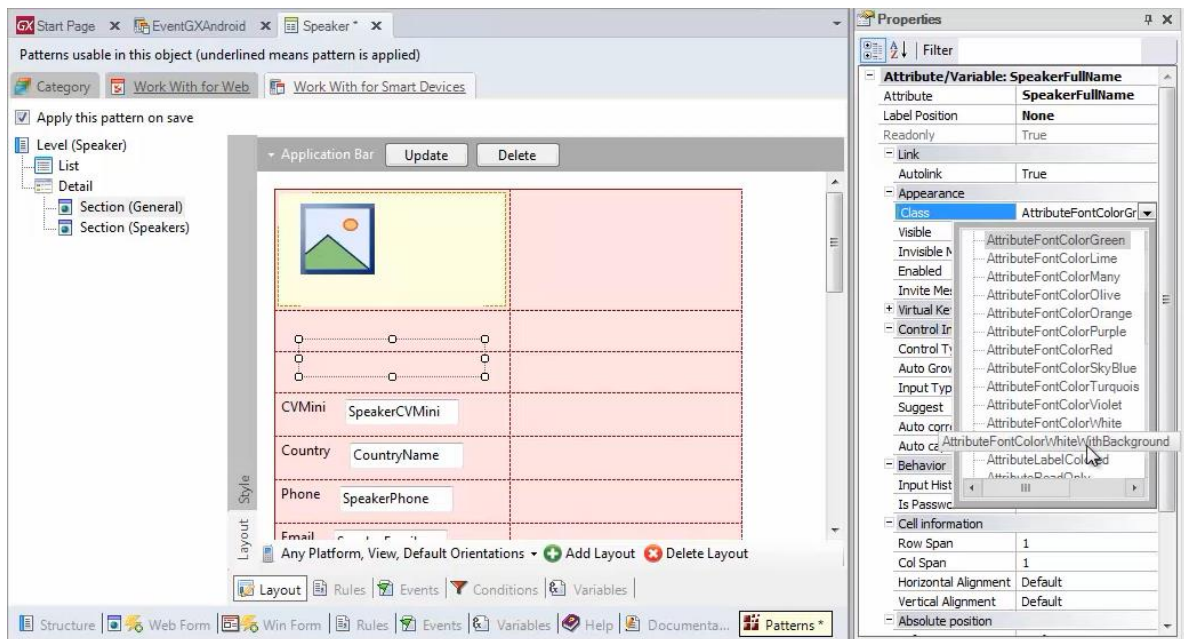
lo que juntos suman 160, por tanto, el 100% restante será cero... y el orden relativo al eje Z, será: 1





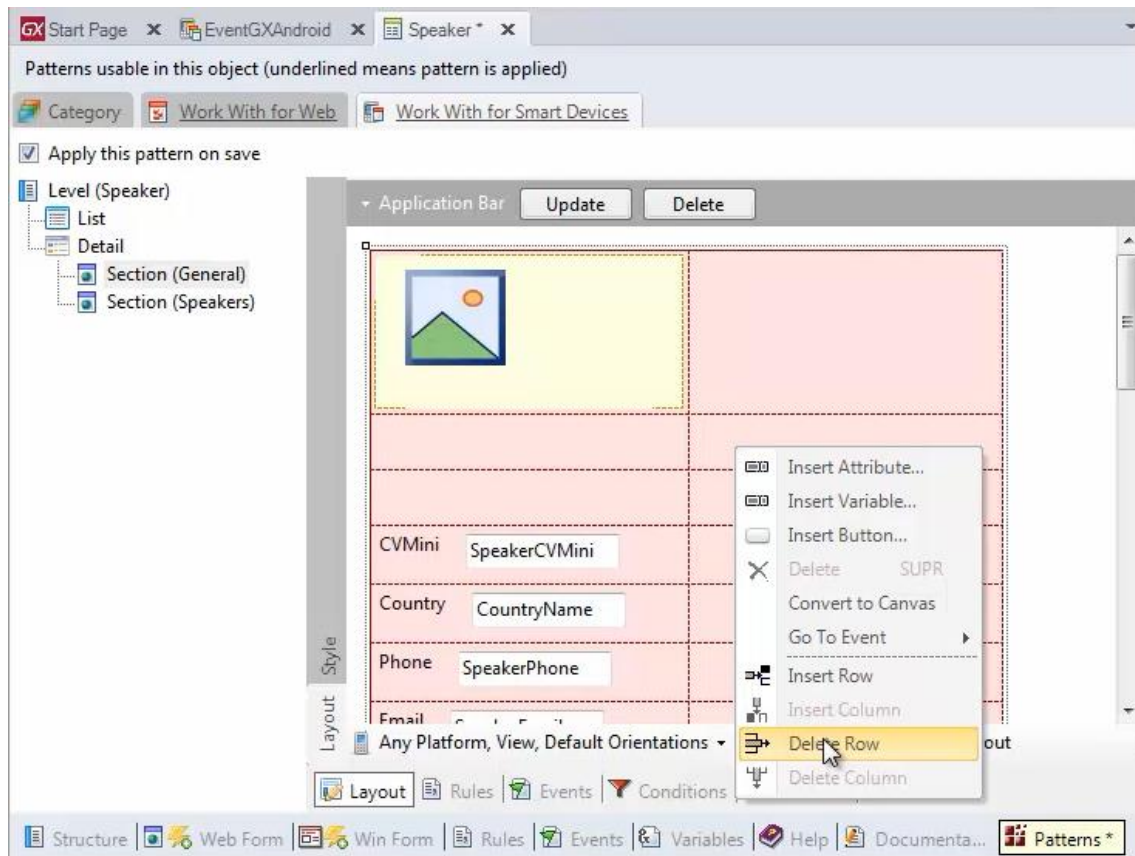
porque deseamos que este control aparezca encima de la foto.

También vamos a aprovechar a cambiarle la clase a este atributo por:  
AttributeFontColorBlueWithBackground

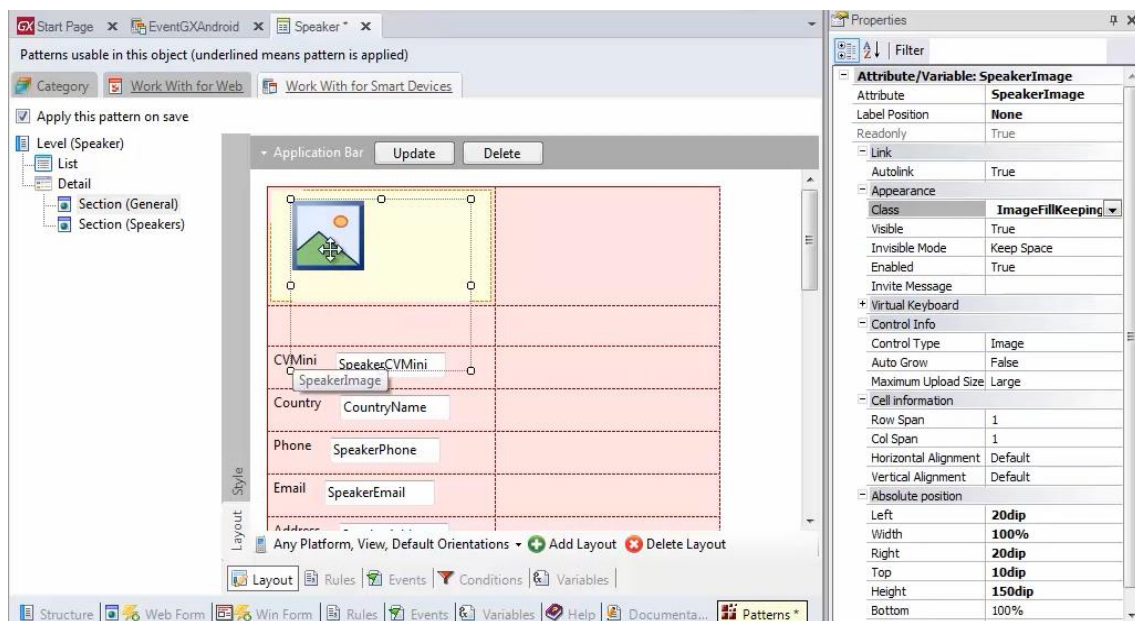


para que el fondo tenga cierta transparencia y se vea un poco de la imagen del orador de fondo.

Vamos a eliminar esta segunda fila porque ya no la necesitamos más



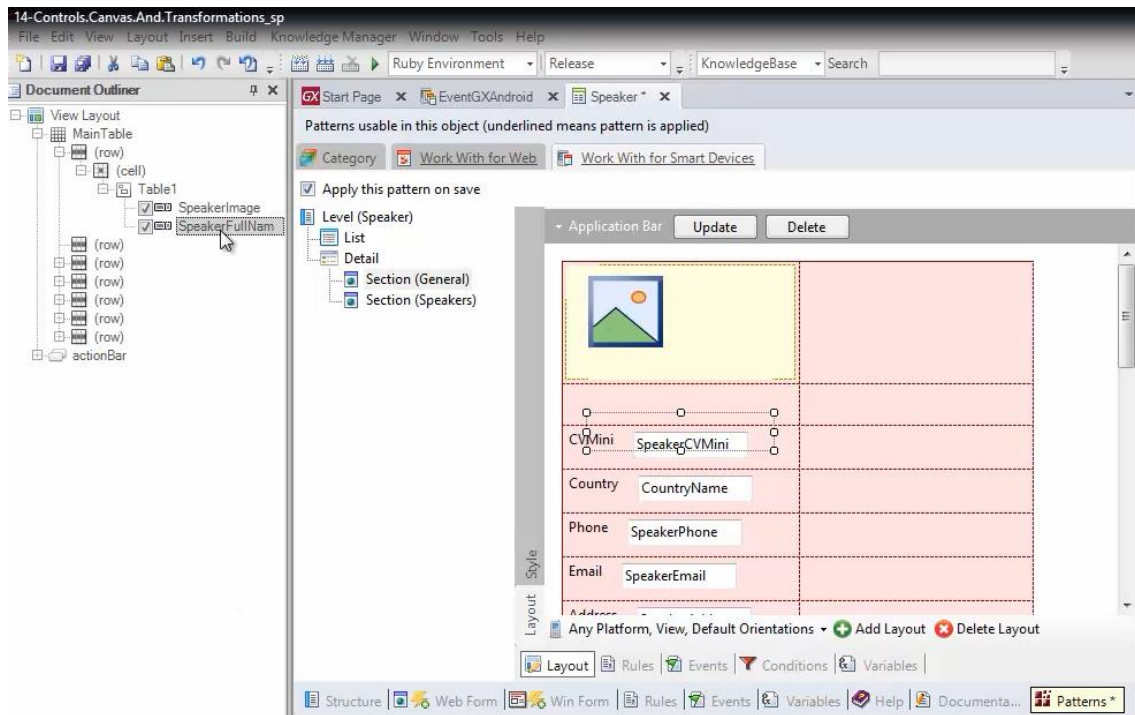
y reparemos ahora en lo siguiente: si queremos ahora cambiar alguna de las propiedades de la imagen, estamos viendo que se superpone la imagen en el layout:



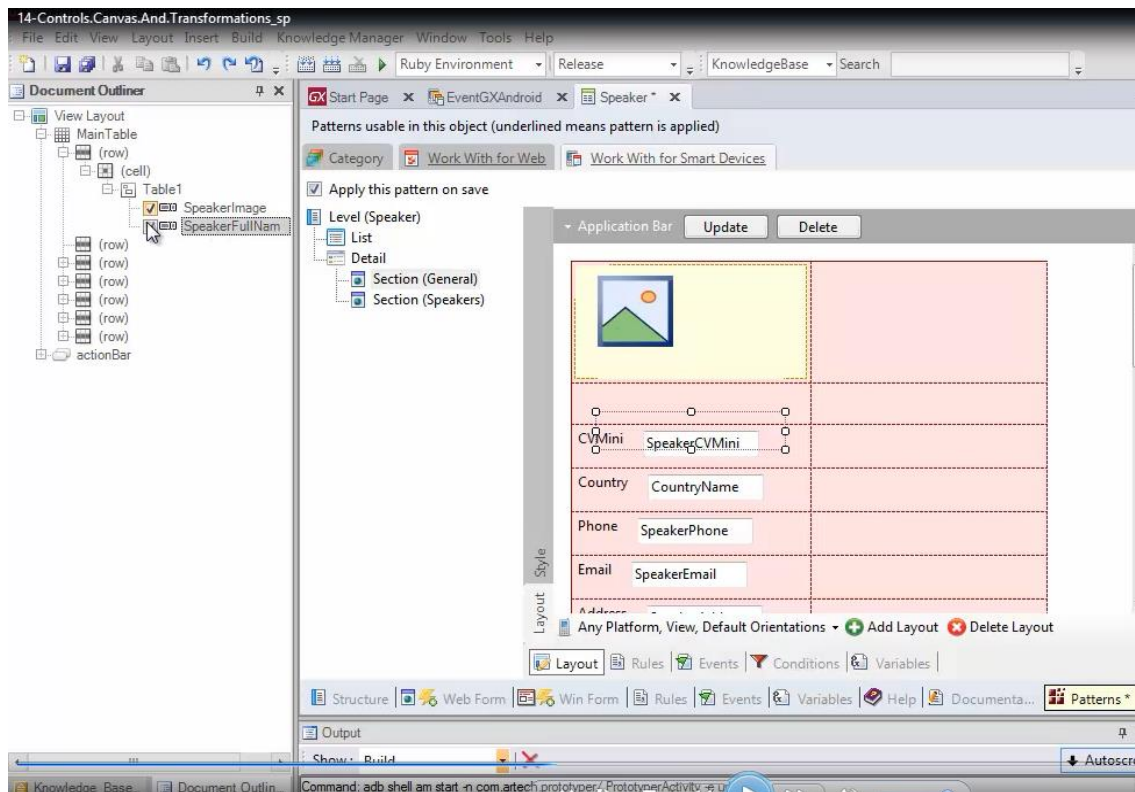
pero aún así la podemos ver y podemos trabajar con sus propiedades..

Pero si ahora queremos acceder a las propiedades del atributo ya no lo tenemos disponibles. ¿Cómo hacemos entonces para editarlo?

Aquí es donde se va a tornar indispensable usar el **Document Outliner** y vemos que aquí sí obtenemos la forma de posicionarnos sobre el control:

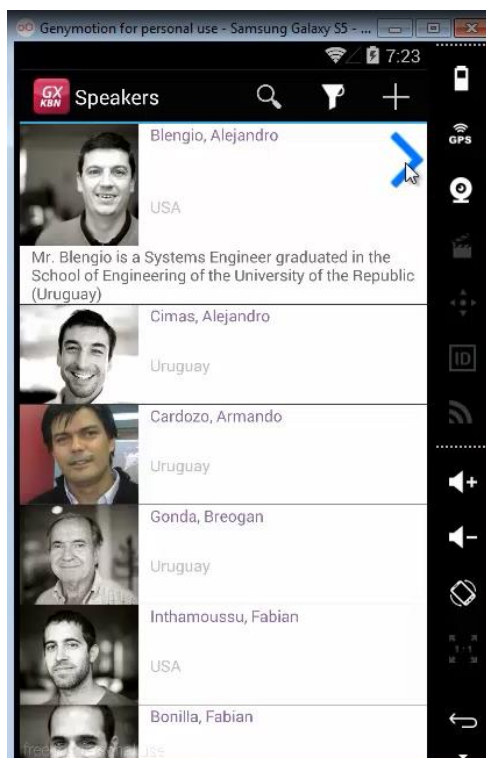
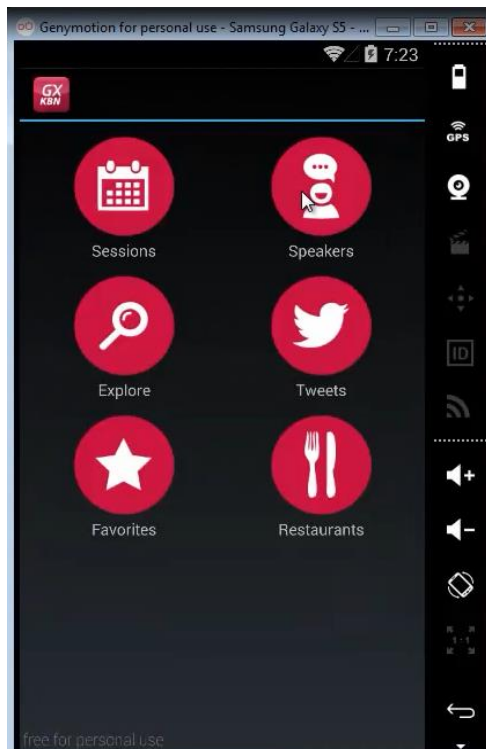


Observemos además, que dentro del control CANVAS, todos los controles que se encuentran allí tienen estos chck boxes

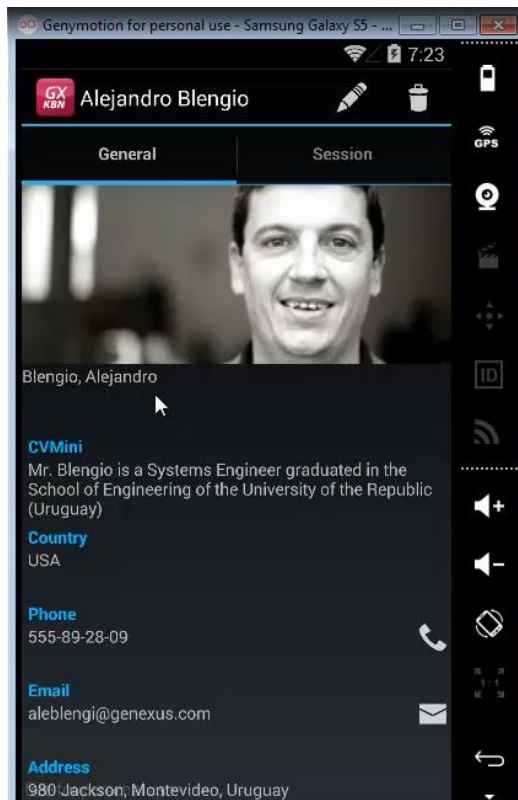


para permitir ocultar los controles y hacerlos visibles justamente para poder trabajar con las diferentes capas.

Antes de dar F5, si ejecutamos la aplicación tal como la tenemos

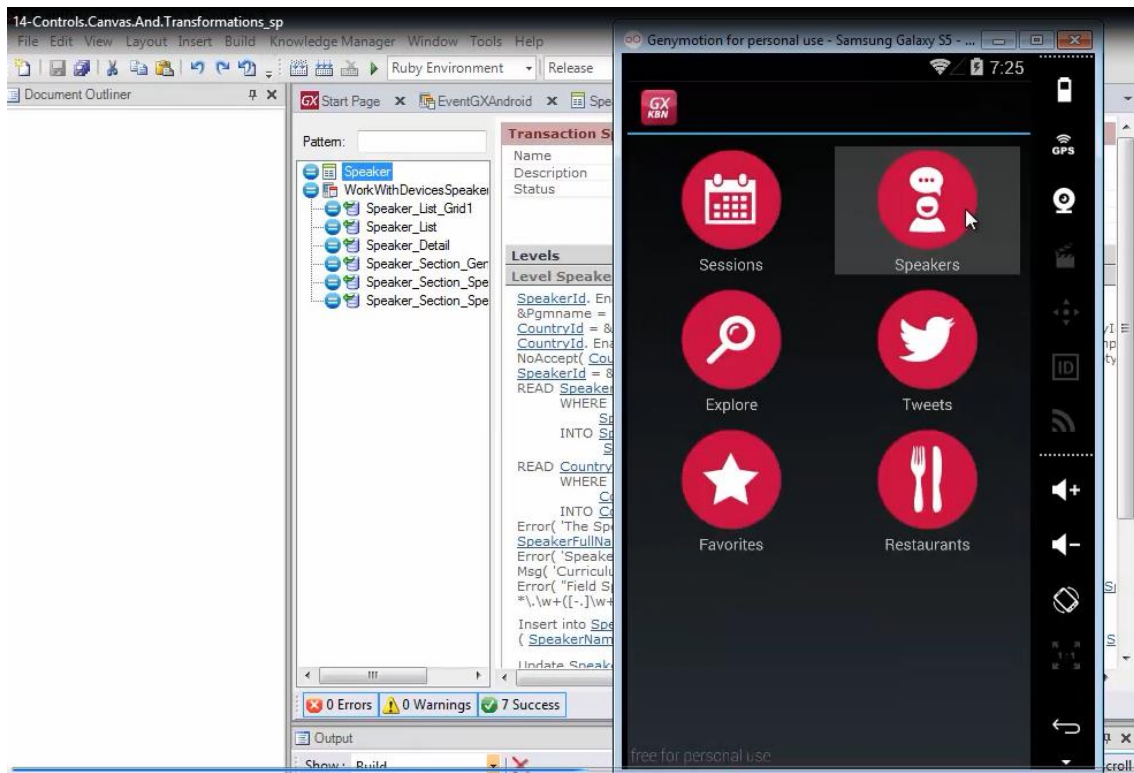




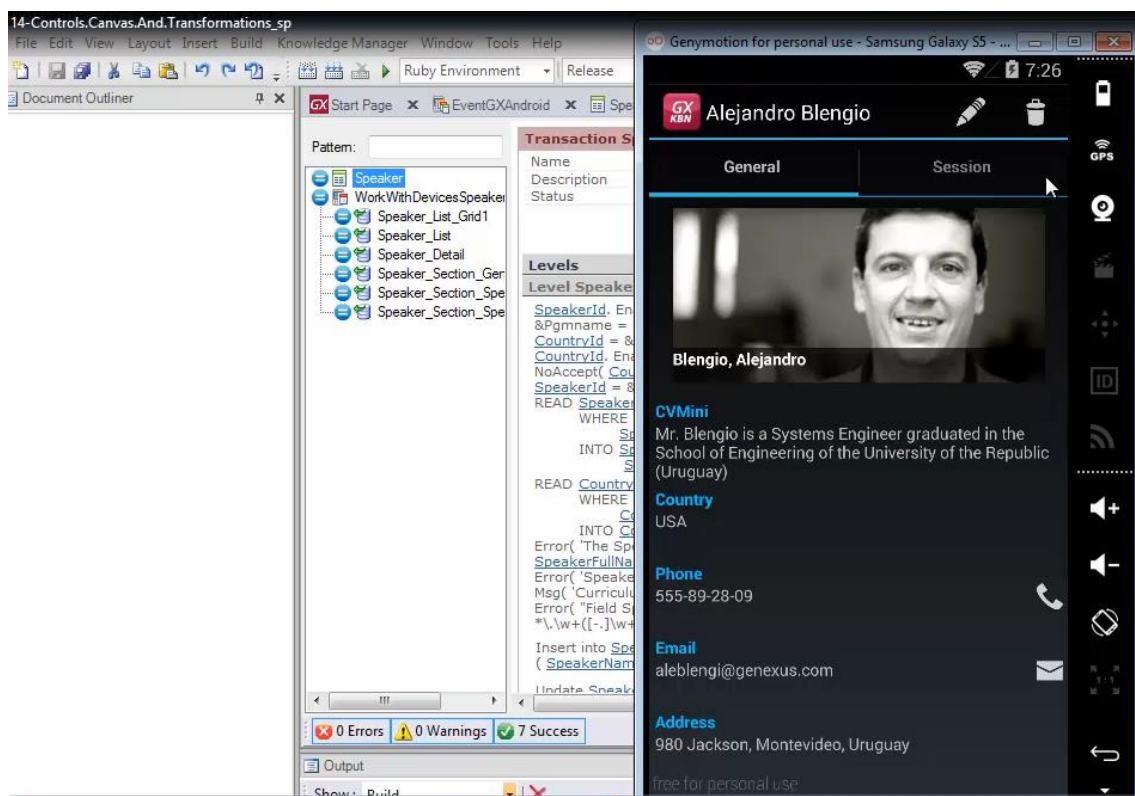
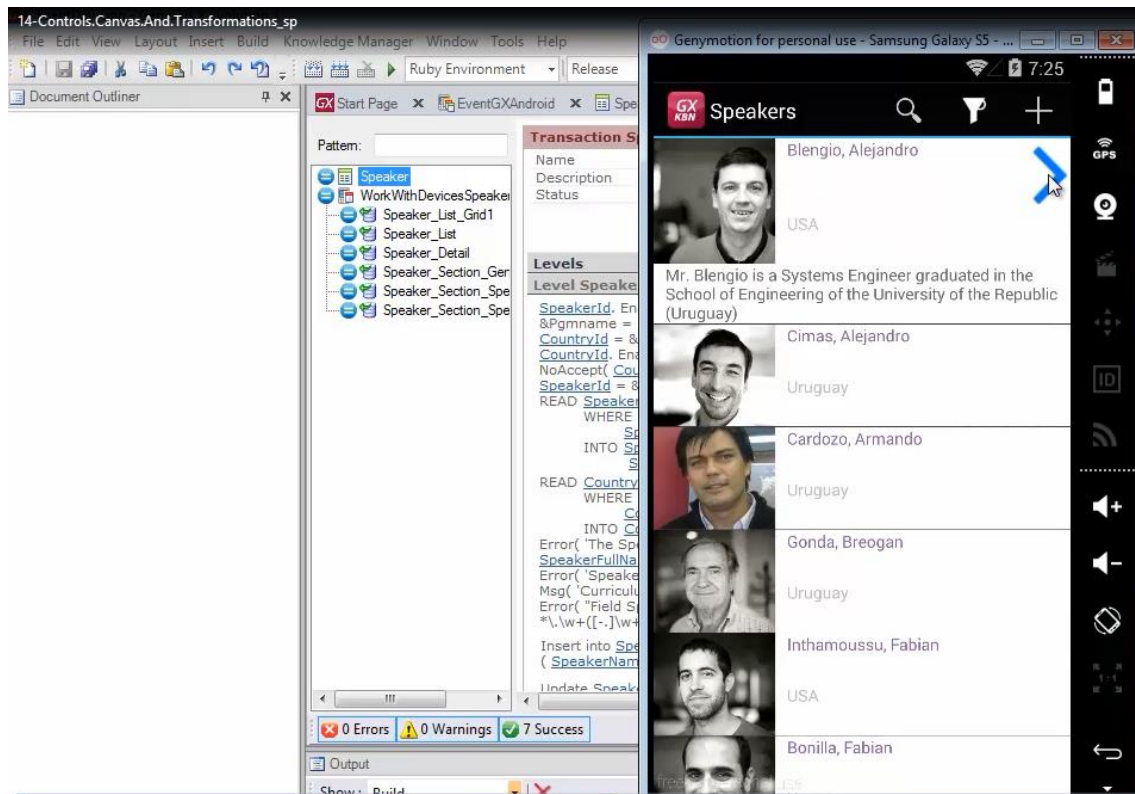


Vemos que los controles están saliendo cada uno en una celda diferente (de una fila diferente).

Ahora sí, hagamos F5

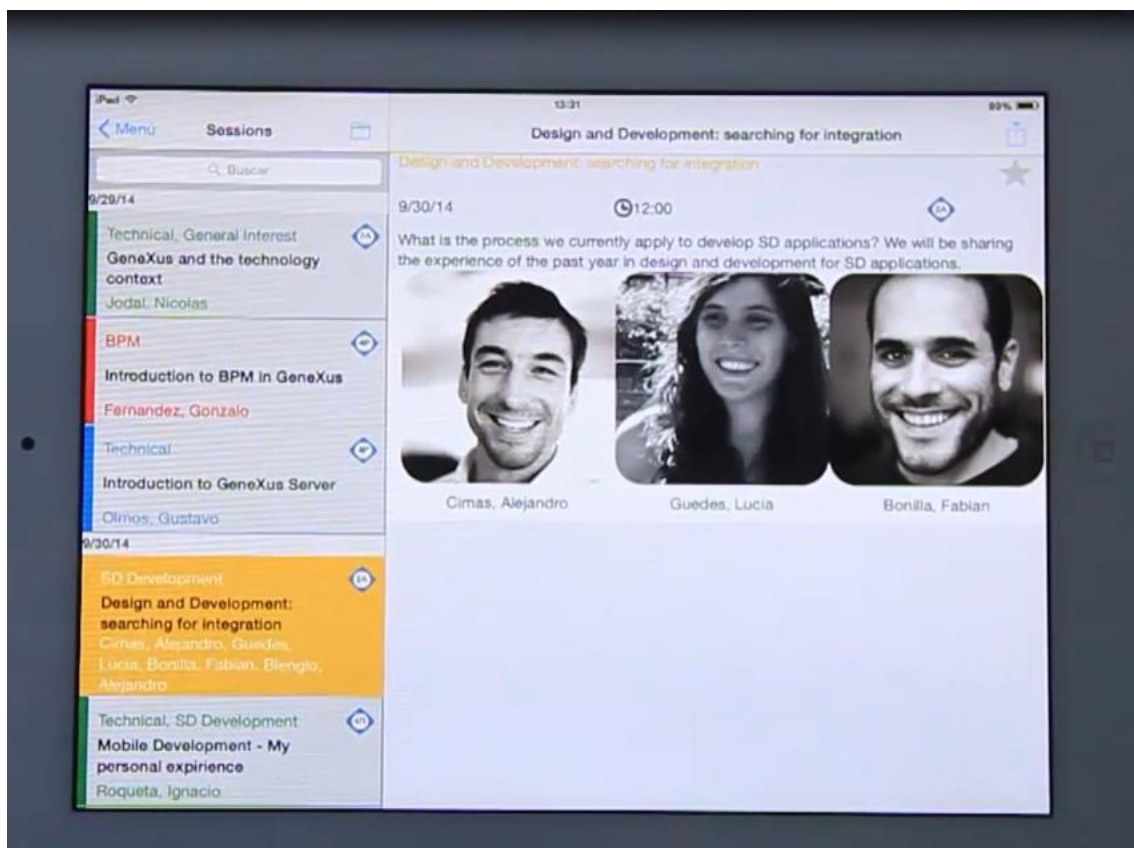
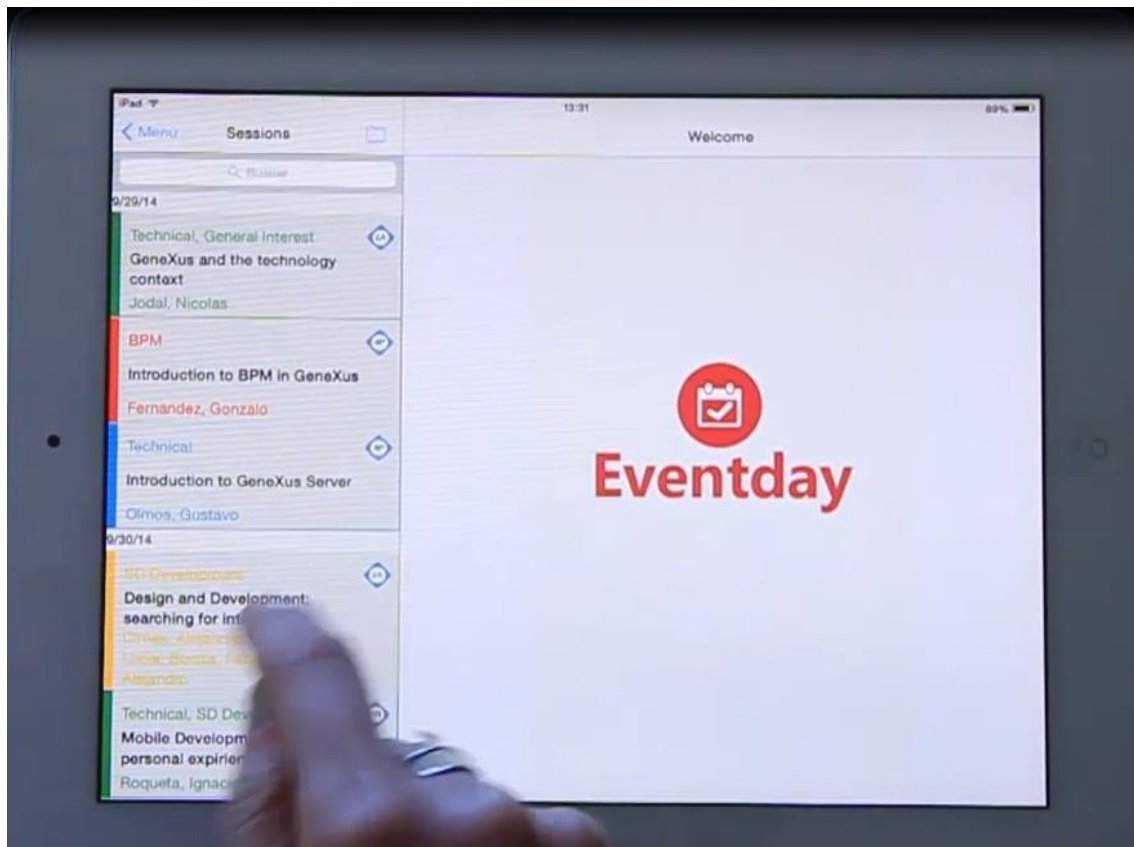




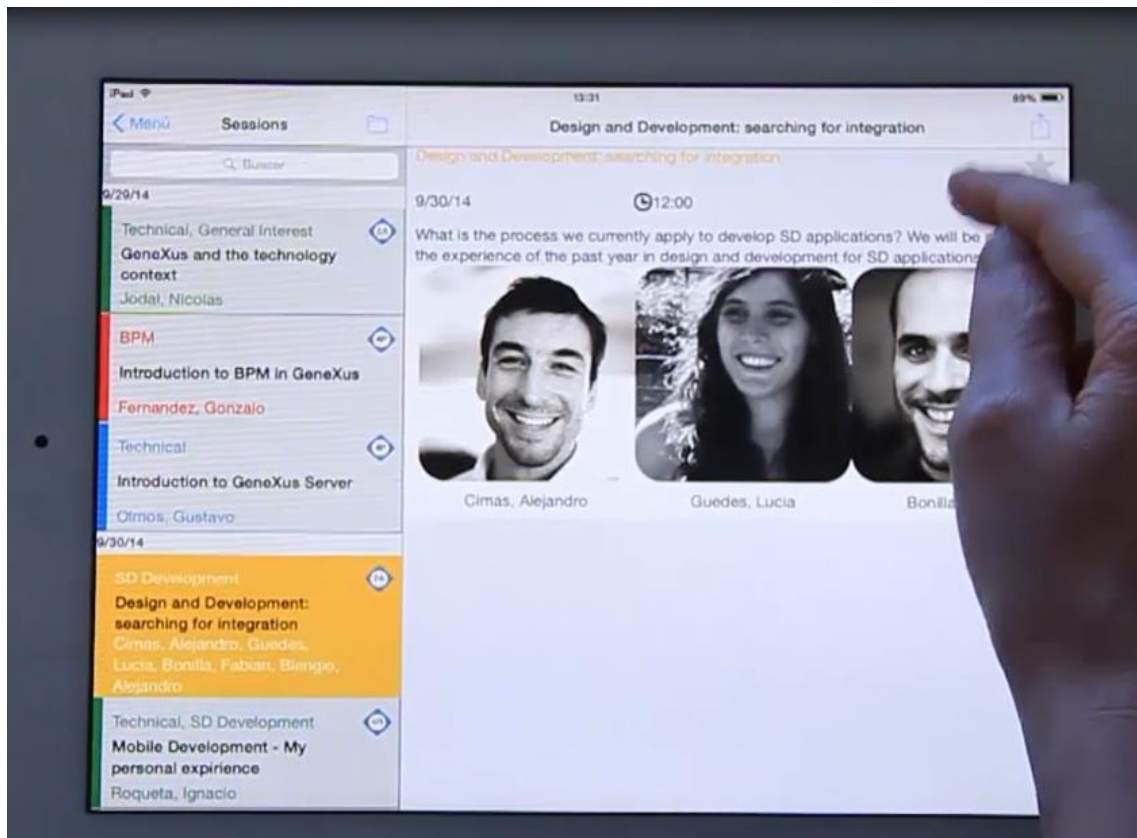


y vemos cómo se superponen.

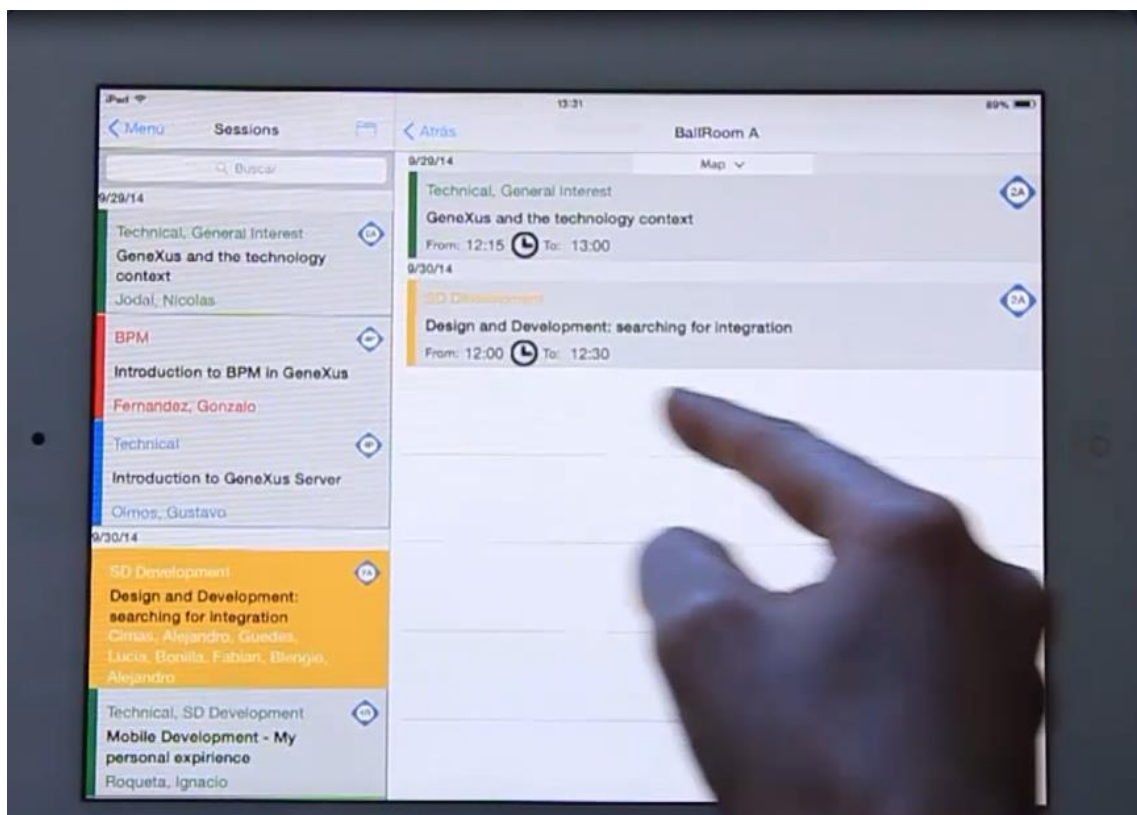
Veamos otro ejemplo:



Estamos visualizando la información de una conferencia y queremos ver la información de la sala

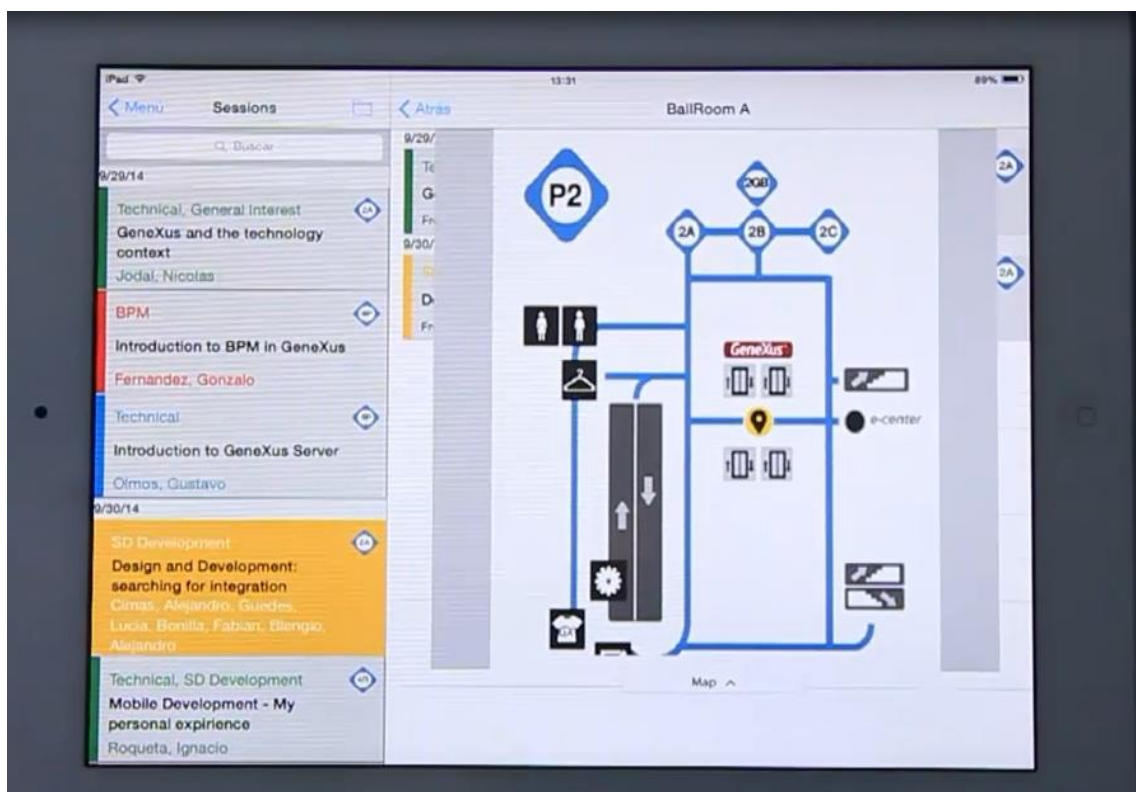
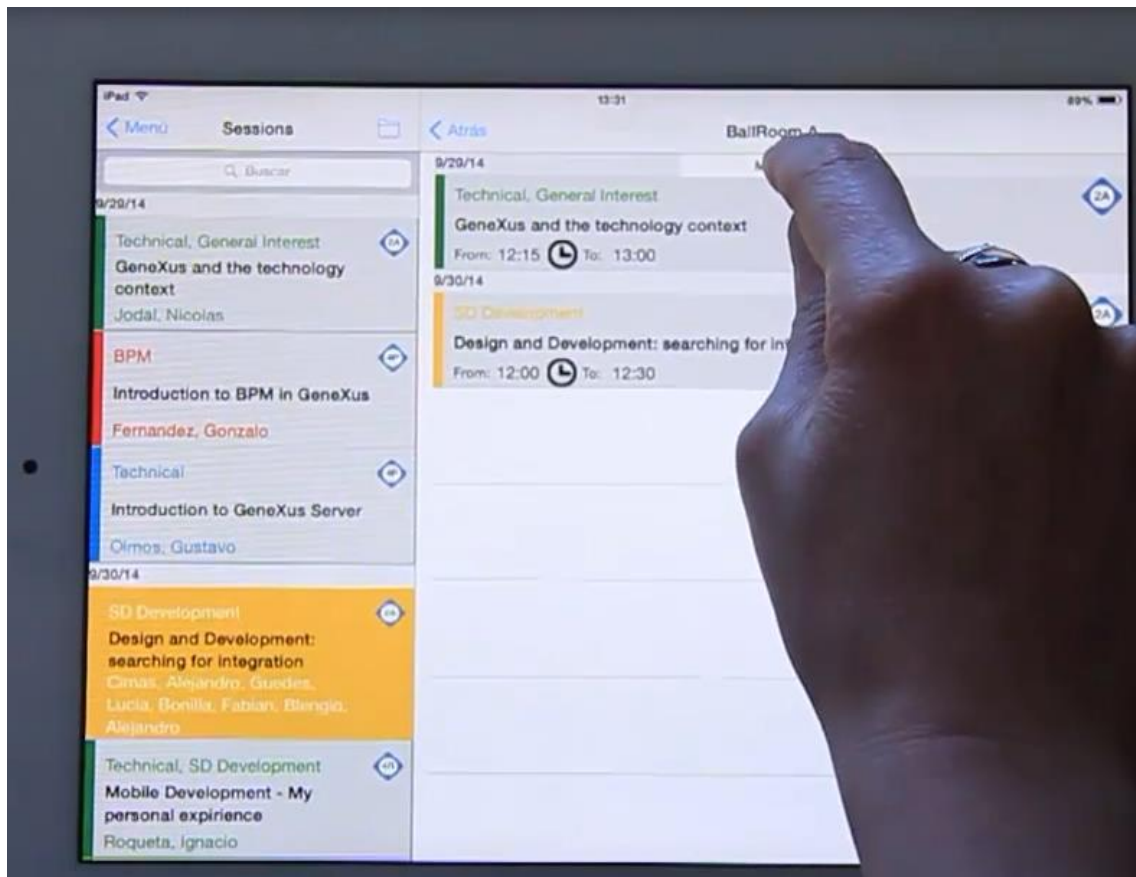


Se está mostrando en un grid las conferencias de esa sala

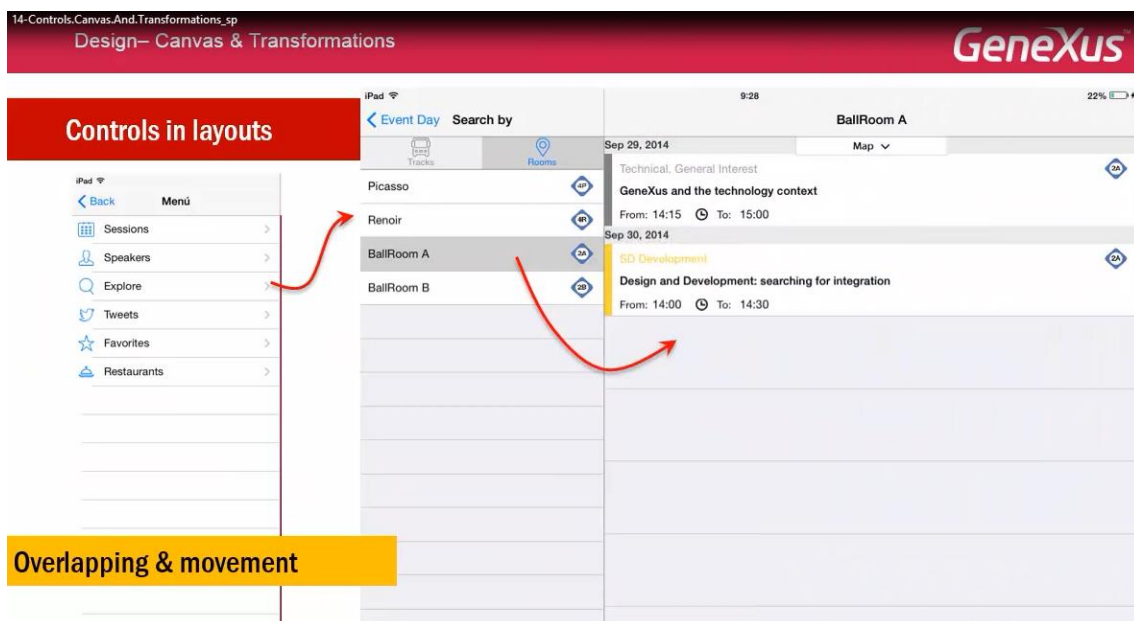
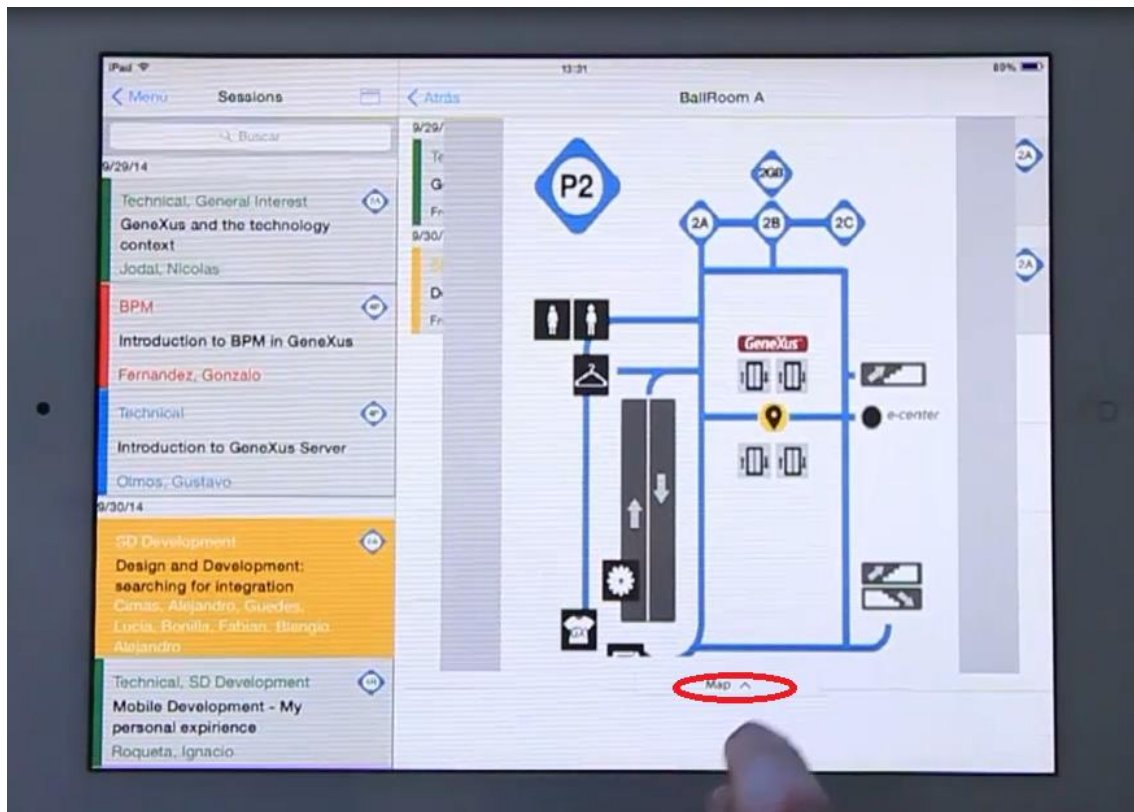


pero además





se permite desplegar expandiendo la ubicación de la sala en un plano del edificio... y colapsar ese mapa

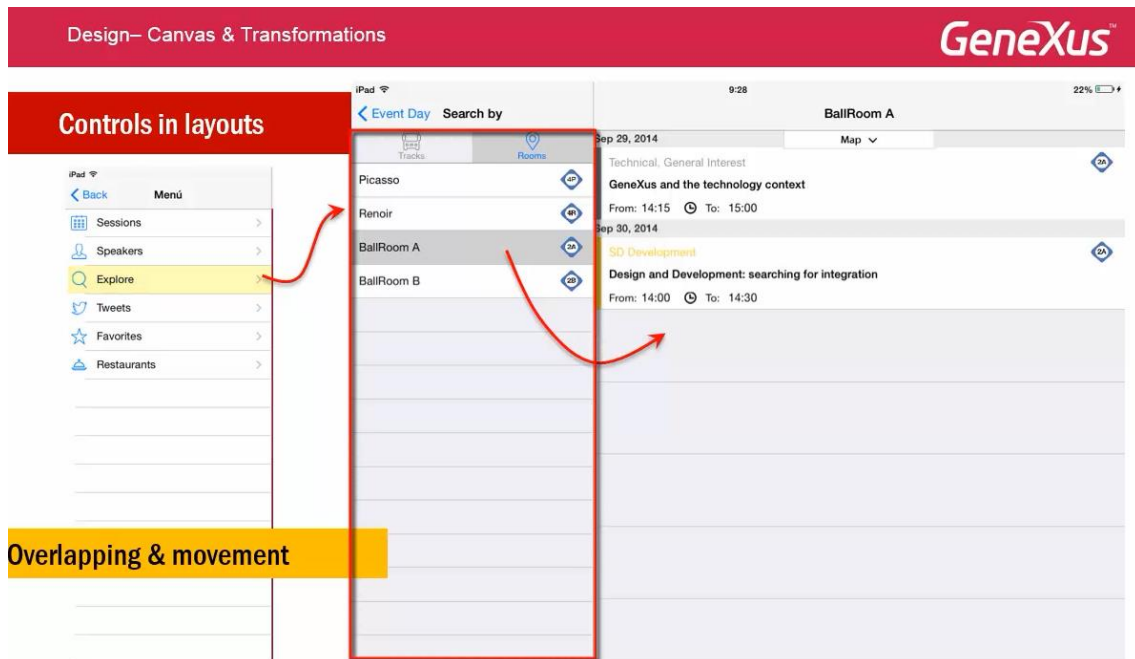


Este es un ejemplo de solapamiento de controles **más** desplazamiento con movimiento.

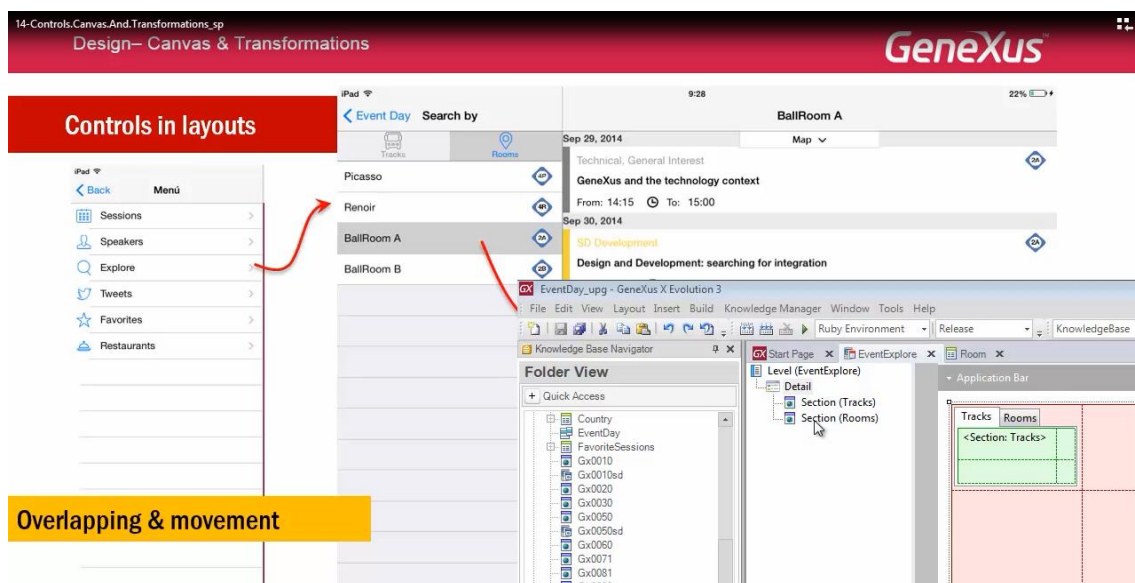
Necesitamos que en el Detail de WorkWith de Rooms, uno de los controles superpuestos se desplace en la pantalla de forma animada, para lograr una interfaz de usuario dinámica.

Investiguémoslo.

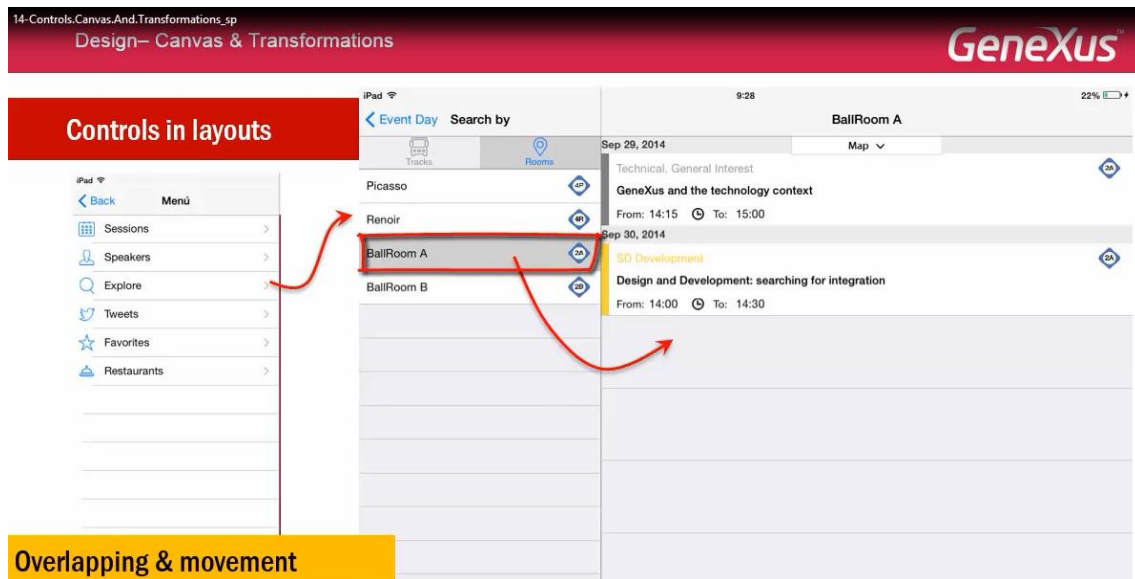
Al detalle de una sala, podemos acceder también desde el menú invocando al objeto "Event Explore"



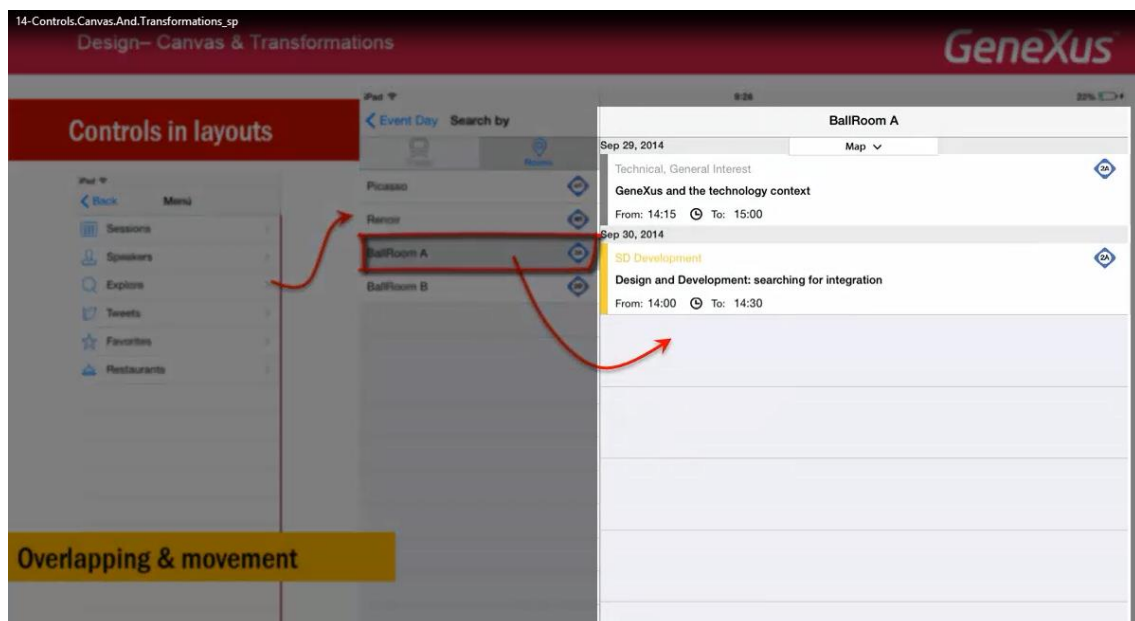
un work with for Smart devices, creado en forma manual del que solo se utiliza su nodo Detail



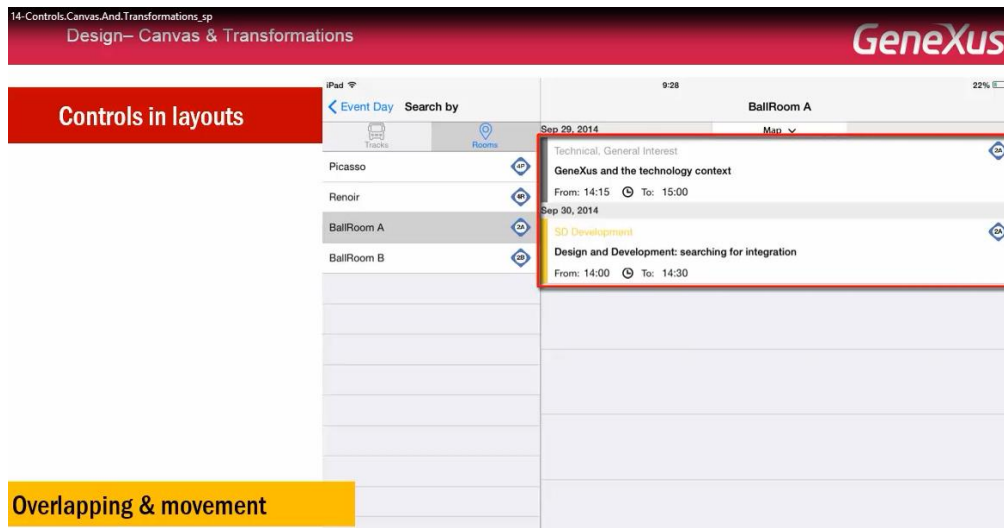
con sus 2 secciones: Tracks y Rooms, que muestran las listas de trucks y salas respectivamente.  
Si desde la lista de salas, se elige una, en el ejemplo BallRoom A:



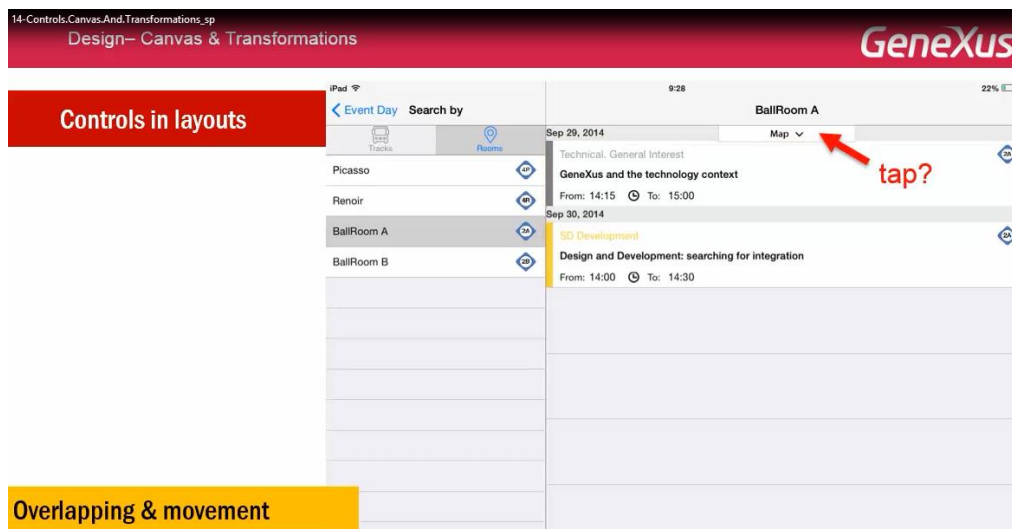
se invoca al Detail del Work With de Rooms para ver los datos de esa sala elegida:



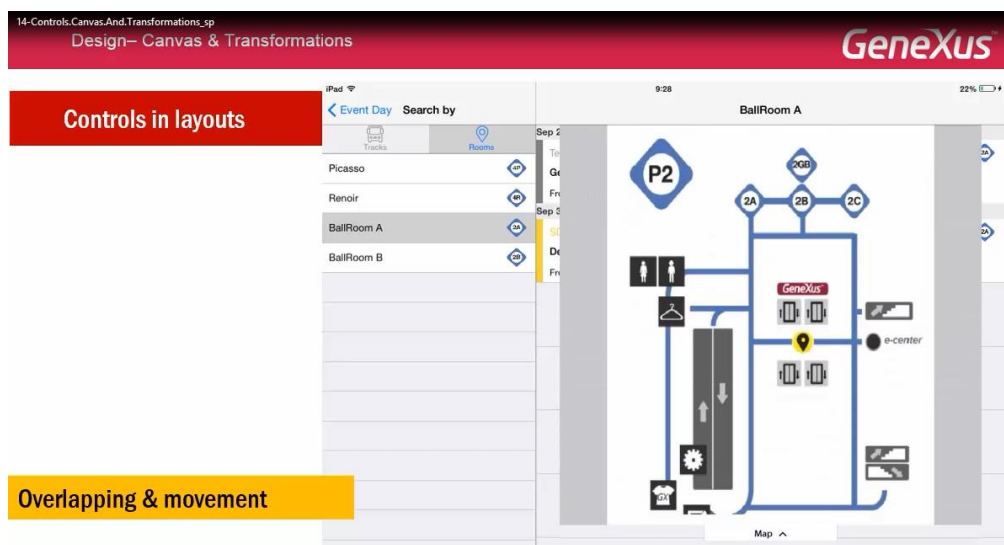
Si tenemos esta aplicación instalada en un ipad, veremos que se nos despliega la lista de conferencias agendadas en esa sala



Tenemos arriba un botón que muestra una imagen para expandir el mapa



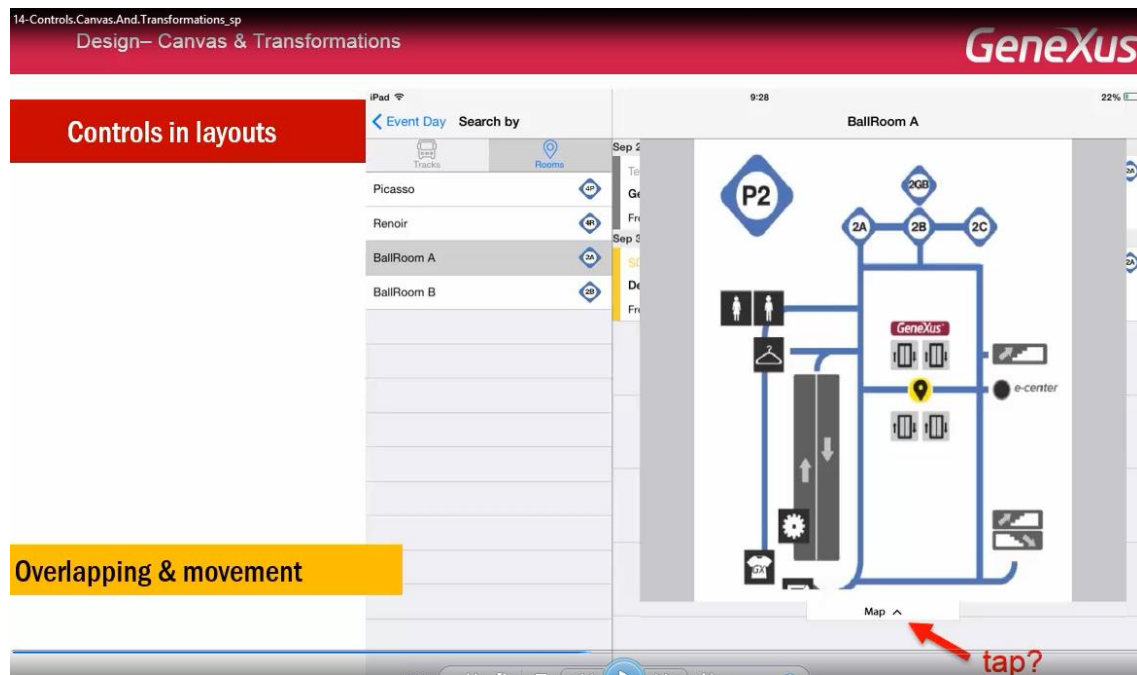
Haciendo TAP



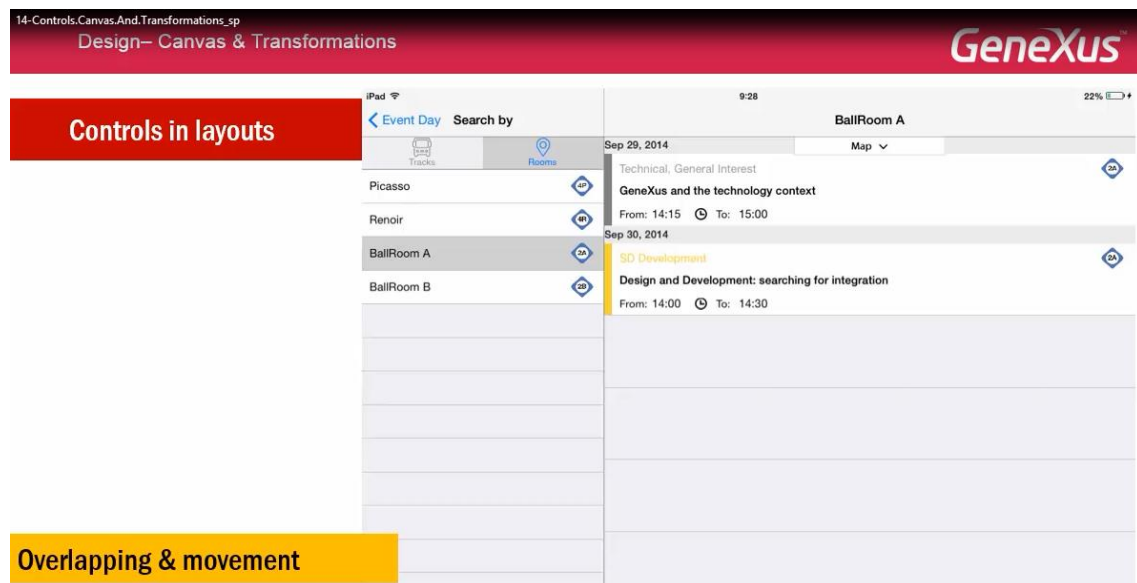


Muestra el mapa solapado al grid de abajo.

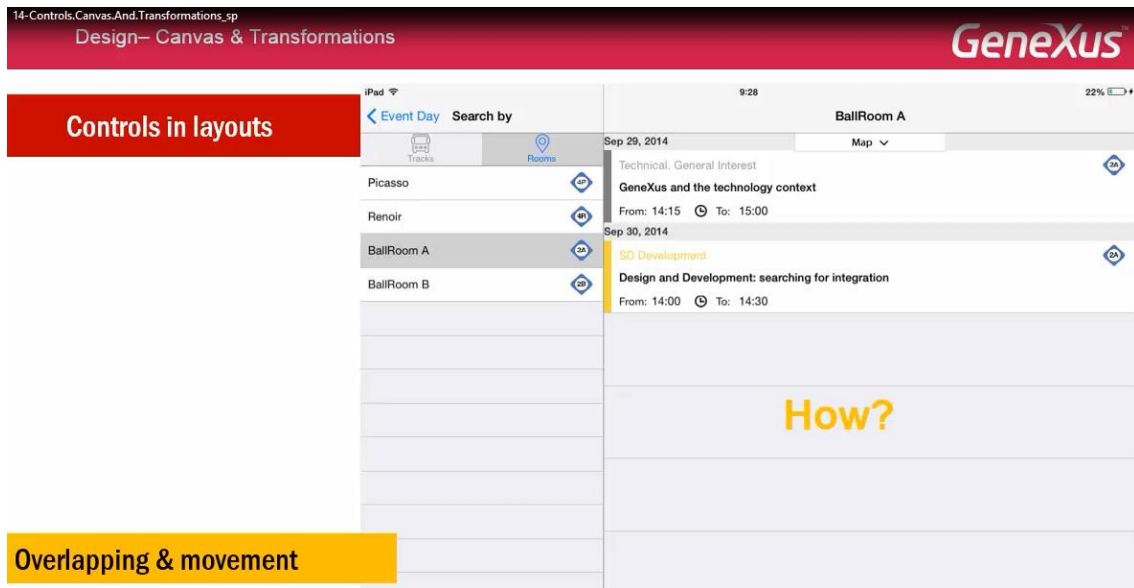
Si ahora volvemos a hacer TAP en el botón



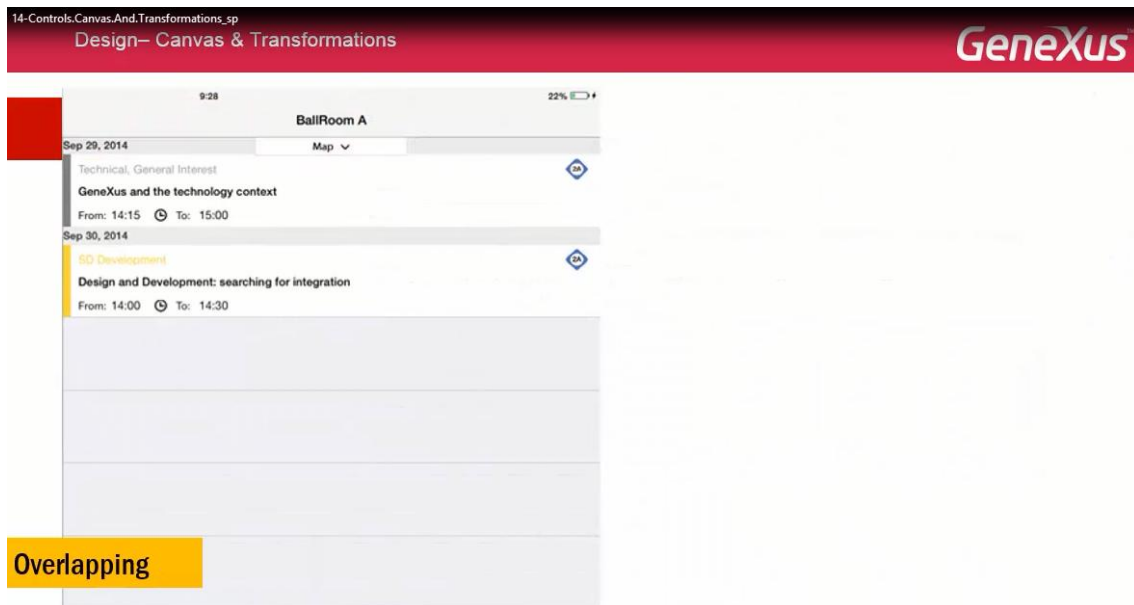
podemos ver cómo desaparece el mapa, volviendo a la imagen inicial



¿Cómo implementamos esto?

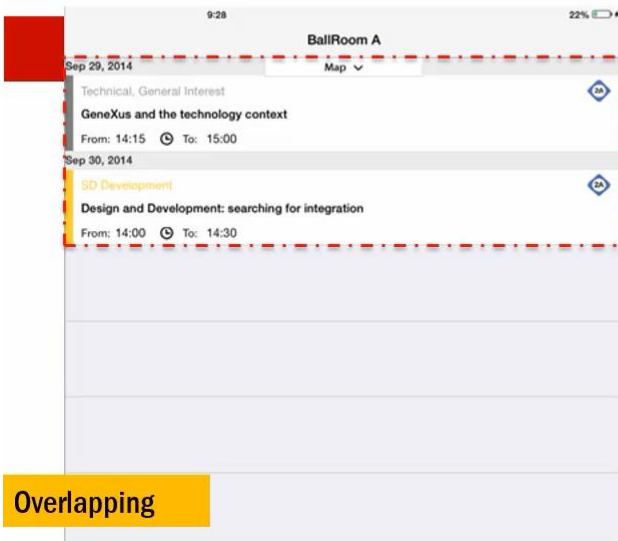


La parte de solapamiento ya la sabemos

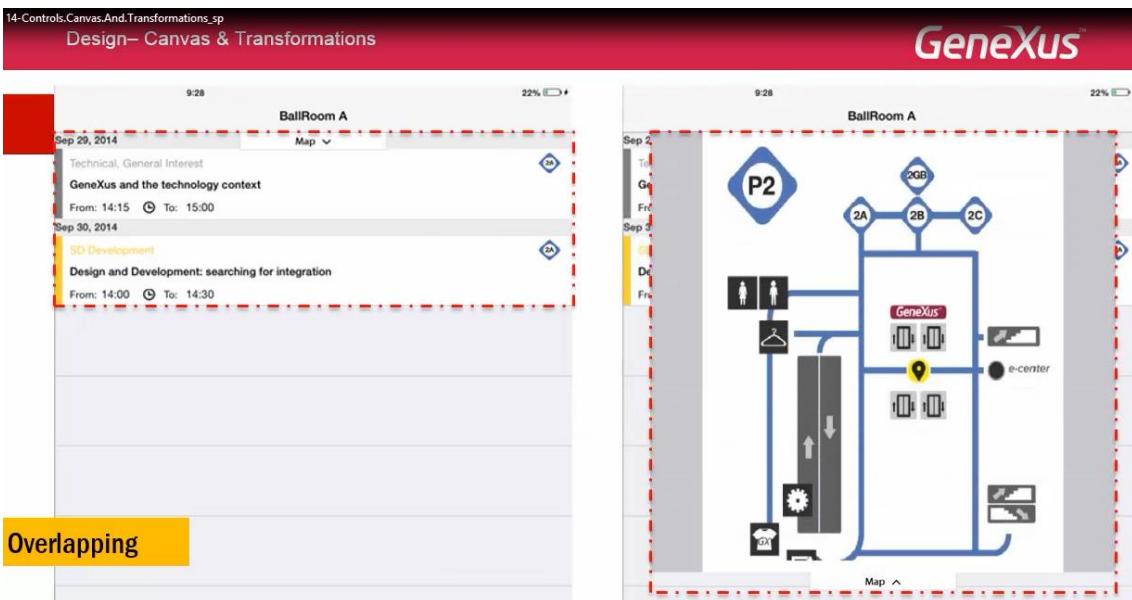


Tenemos en este Layout, en una misma área de pantalla, la superposición de 2 controles:

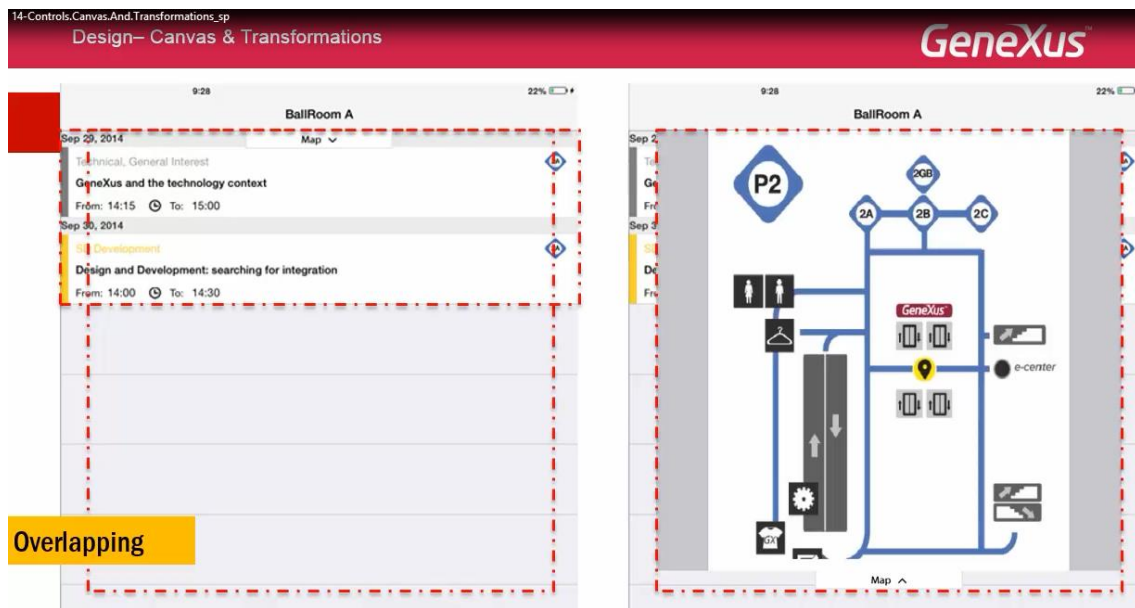
El del grid



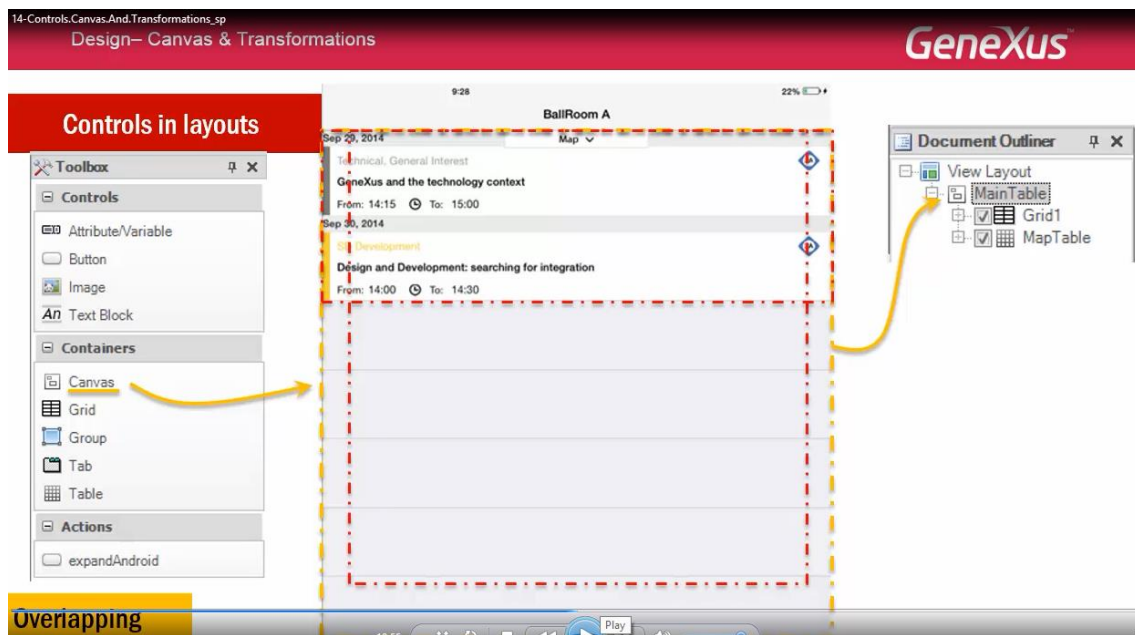
Y el del mapa con el botón



¿Cómo se consigue esto?



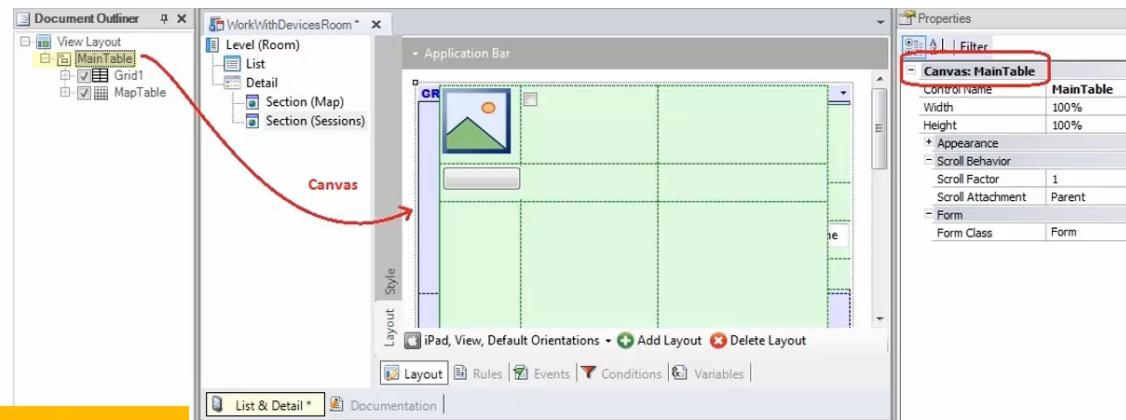
Con el control CANVAS como vimos en el ejemplo anterior



Aquí podemos ver el layout para ipad:



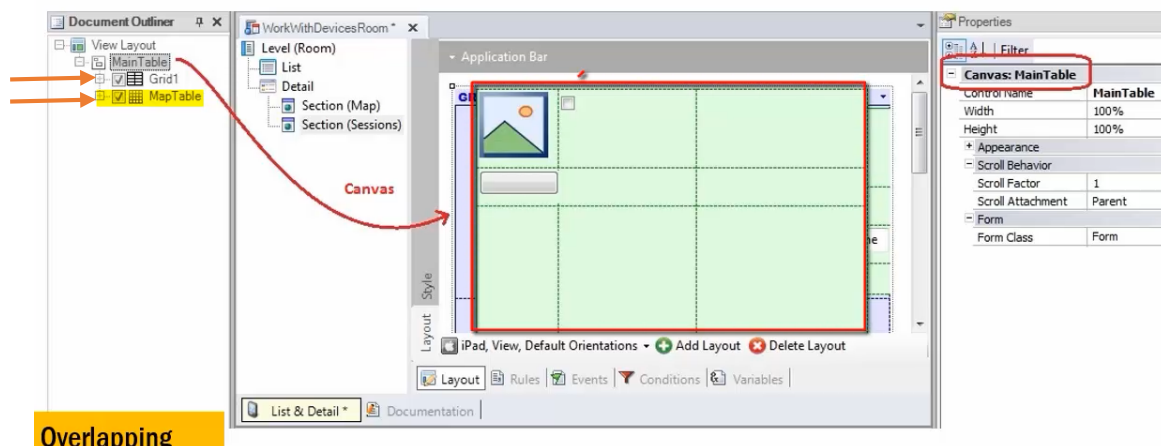
## Controls in layouts



## Overlapping

donde tenemos una tabla CANVAS, que tiene dentro 1 grid (el que implementa la carga de las sessions)... y una tabla de nombre **MapTable**, que implementa la pantalla superpuesta:

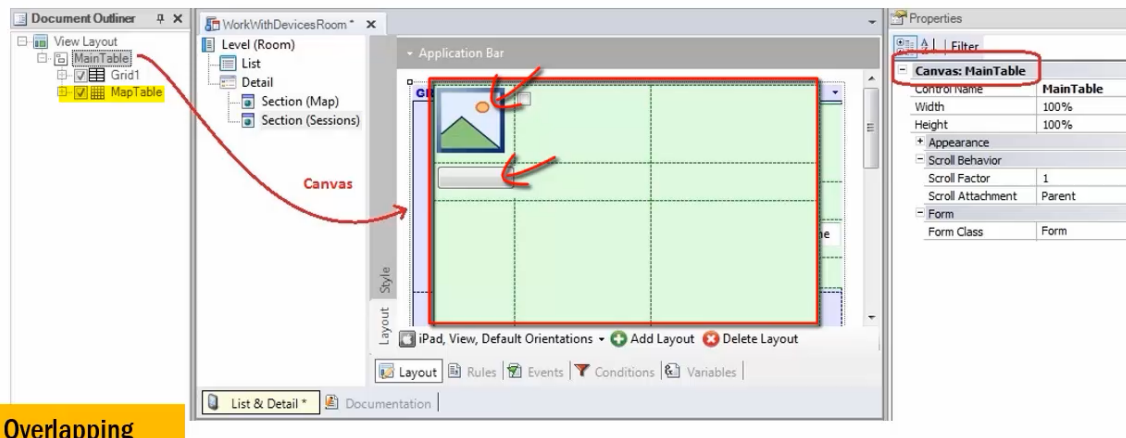
## Controls in layouts



## Overlapping

con el mapa y el botón:

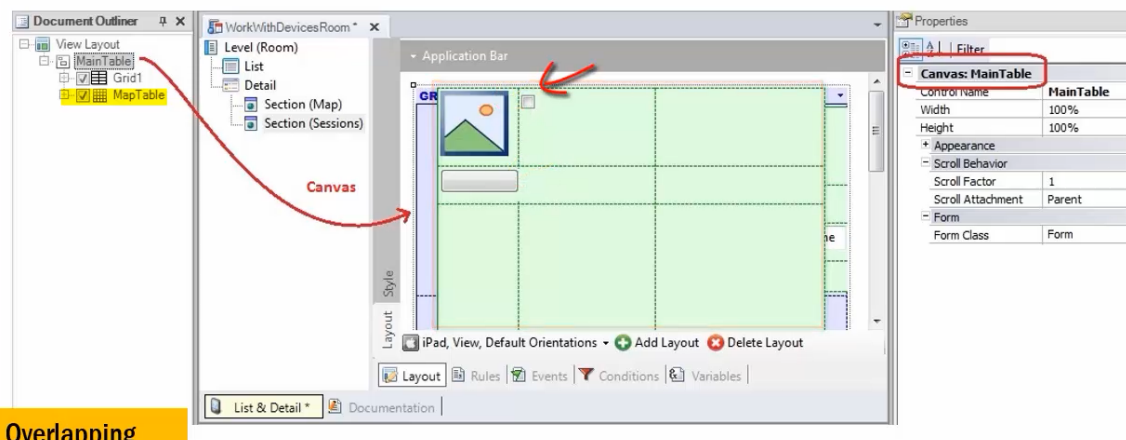
## Controls in layouts



Overlapping

Además podemos ver un checkbox

## Controls in layouts

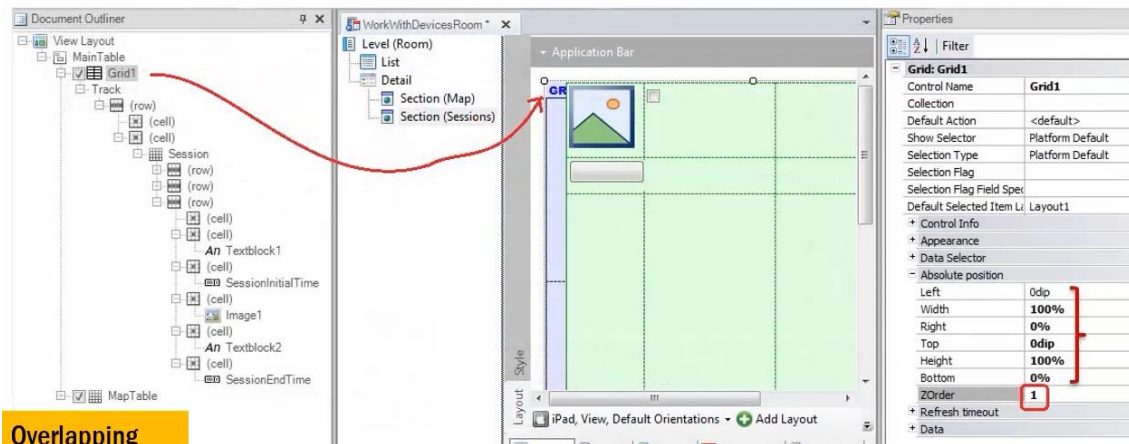


Overlapping

que no estará visible, que será utilizado para saber si está expandida o colapsada la tabla en cada oportunidad.

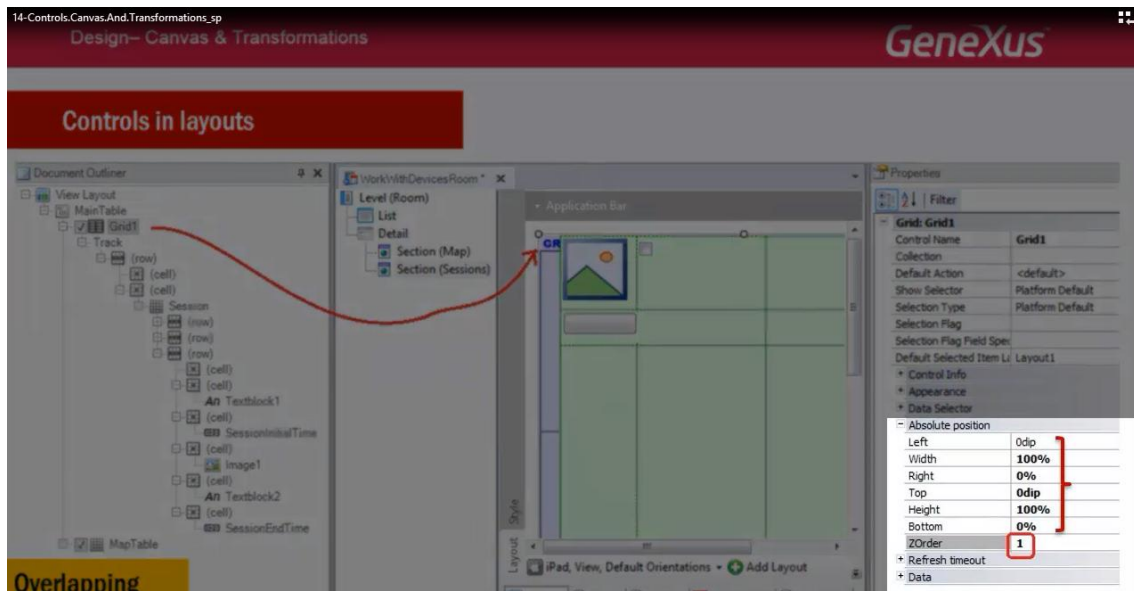
Si nos posicionamos en el grid y accedemos a sus propiedades

## Controls in layouts



## Overlapping

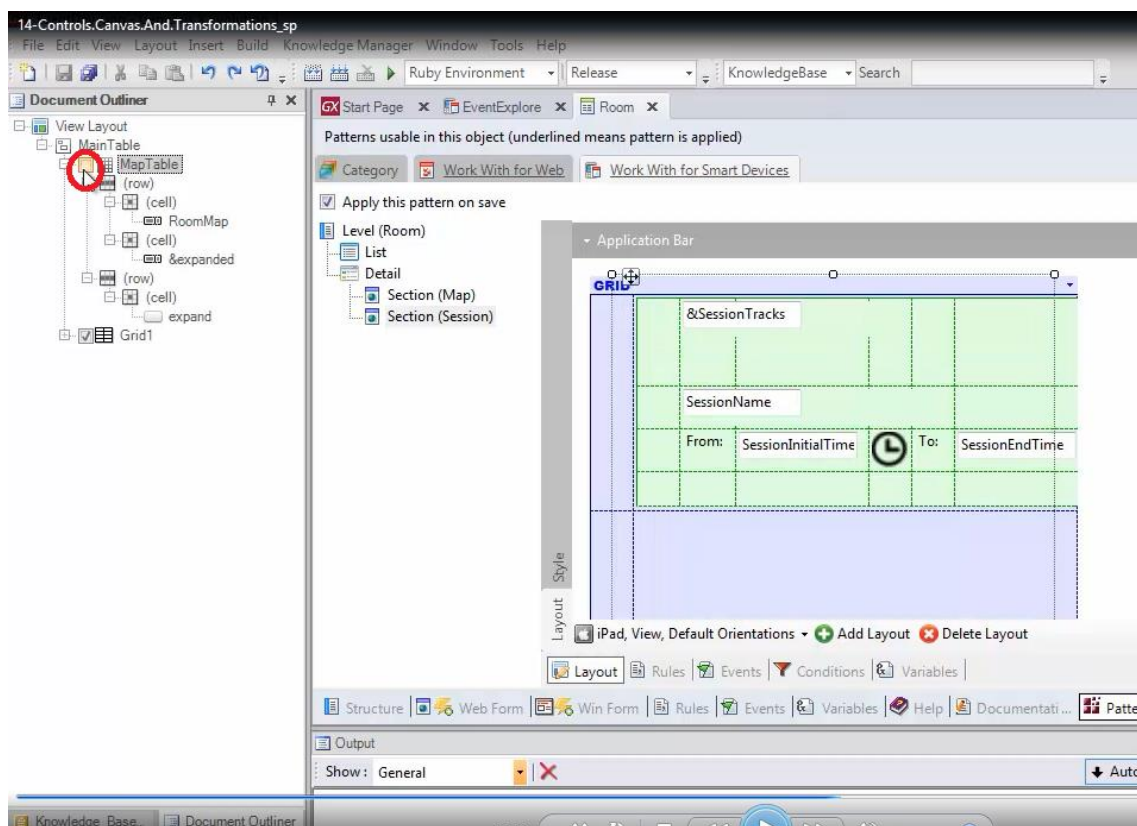
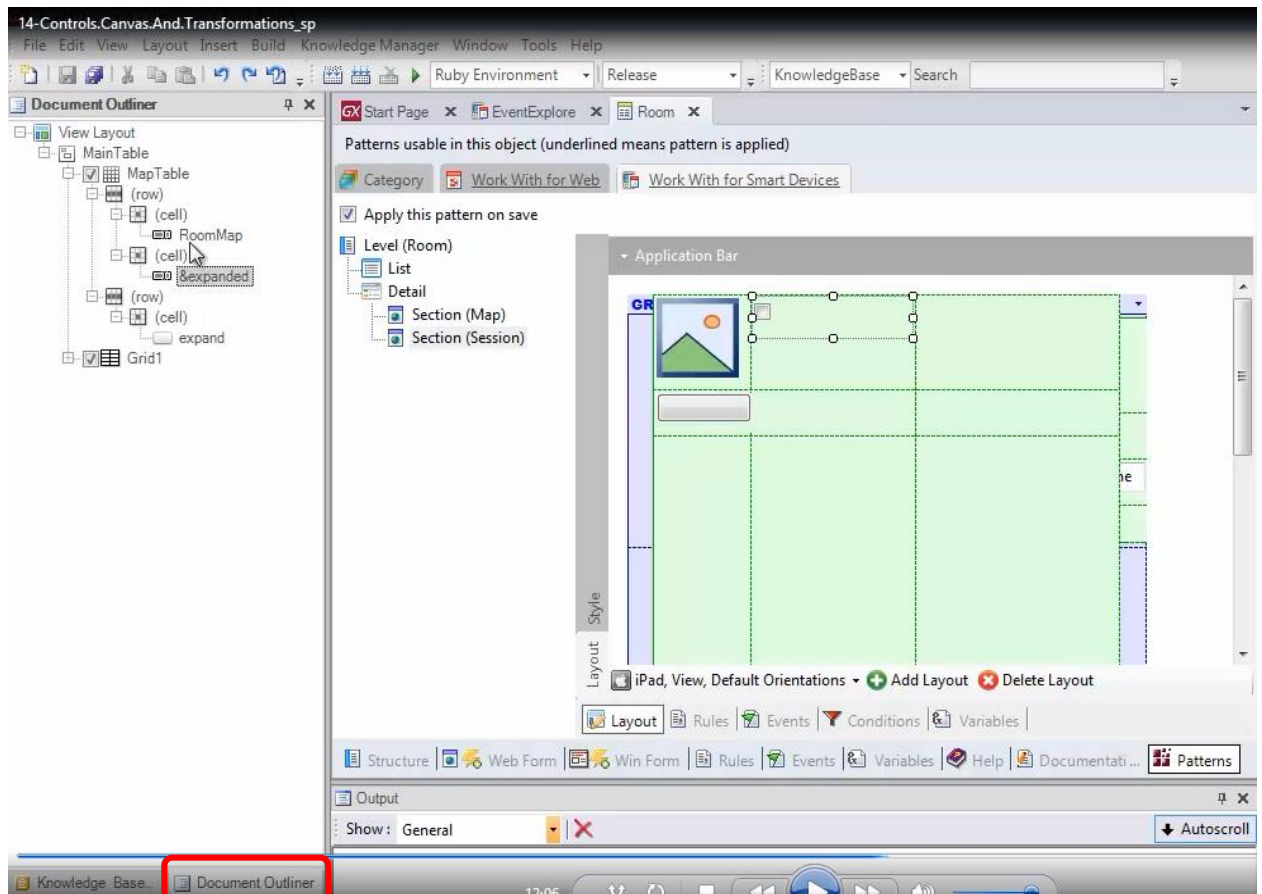
vemos los valores de las propiedades que hemos dado al grupo **Absolute Position**



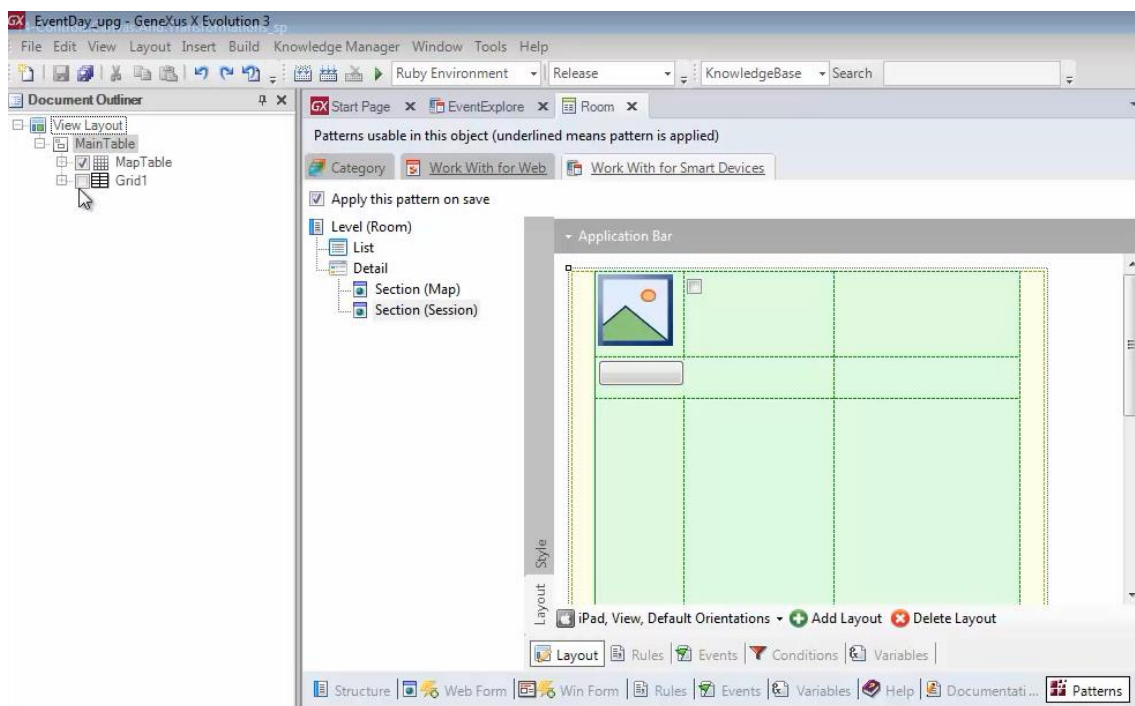
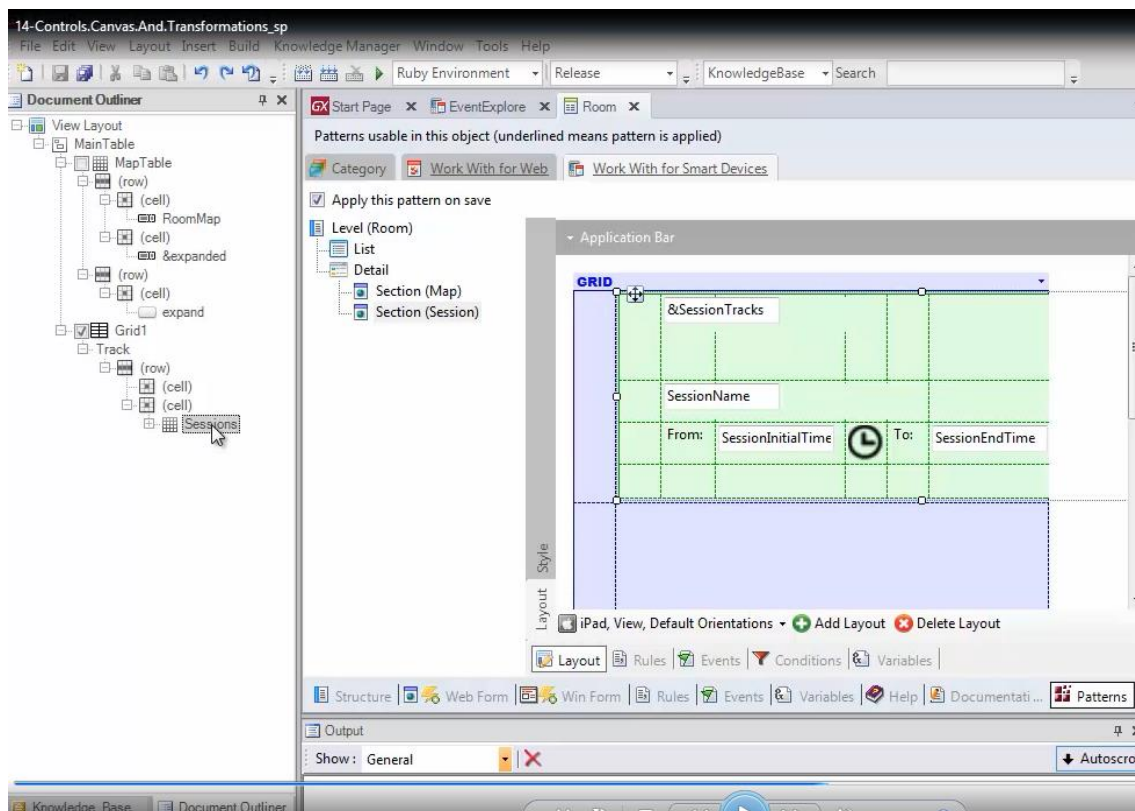
Recordemos que a mayor número Z, más cercano al usuario.

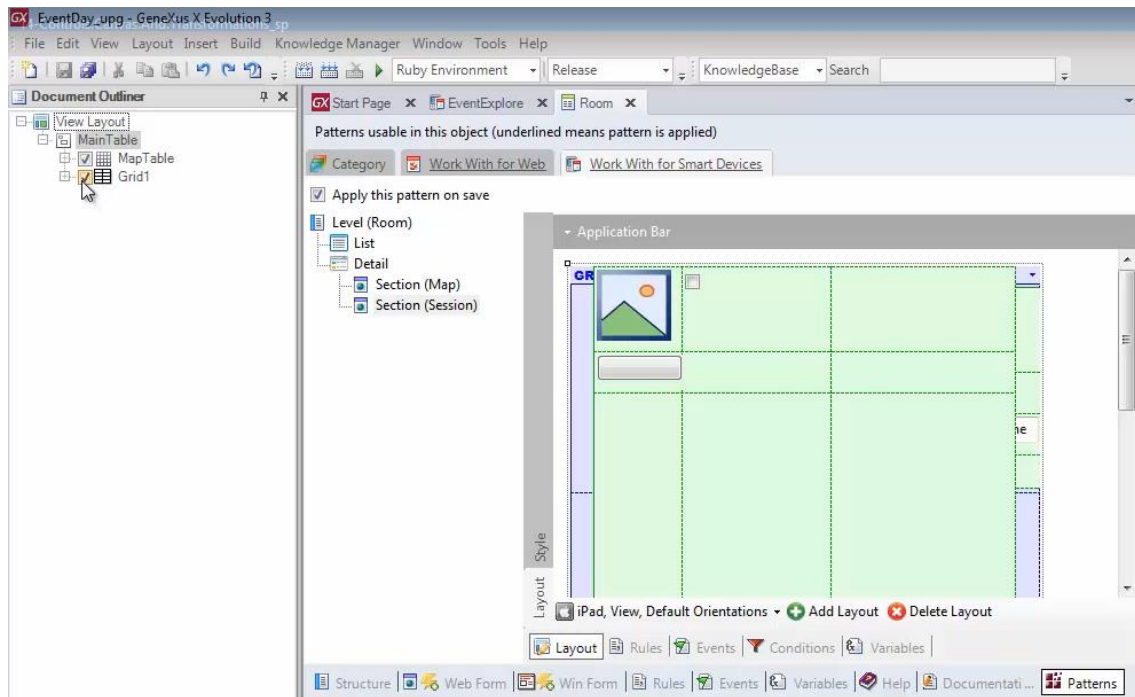
El número cero, indica el fondo de la pantalla, lo que estará más atrás.

Otra vez utilizamos el Document Outliner, para poder acceder a los controles que se encuentran debajo

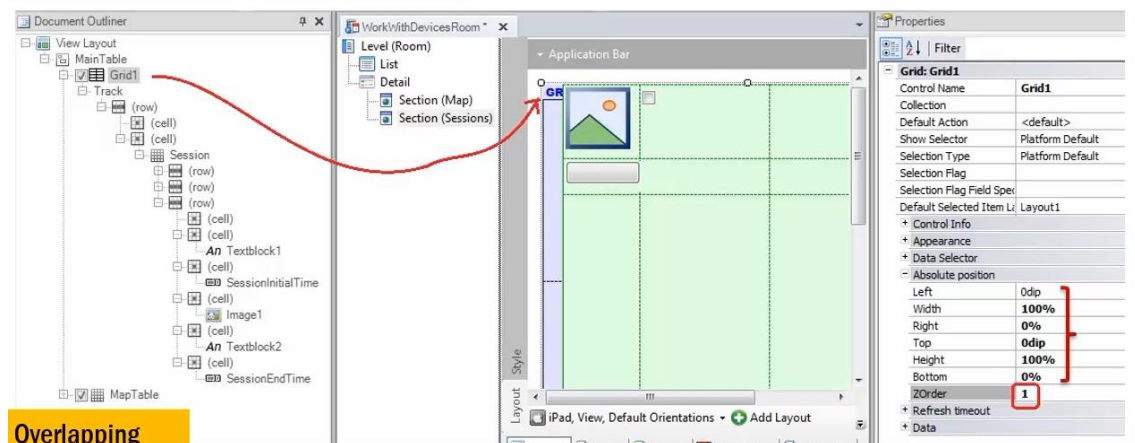








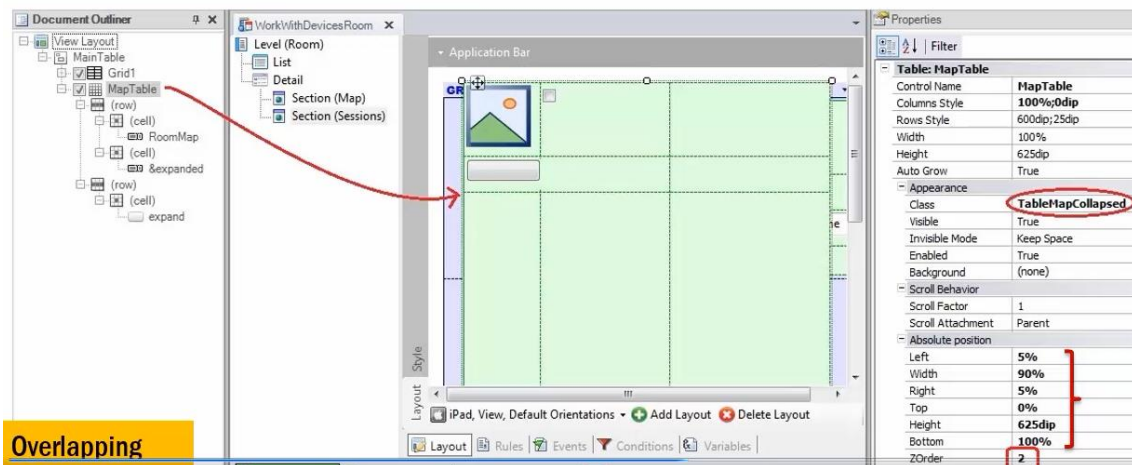
## Controls in layouts



Vemos entonces como el grid tiene dentro del CANVAS, el ZOrder=1

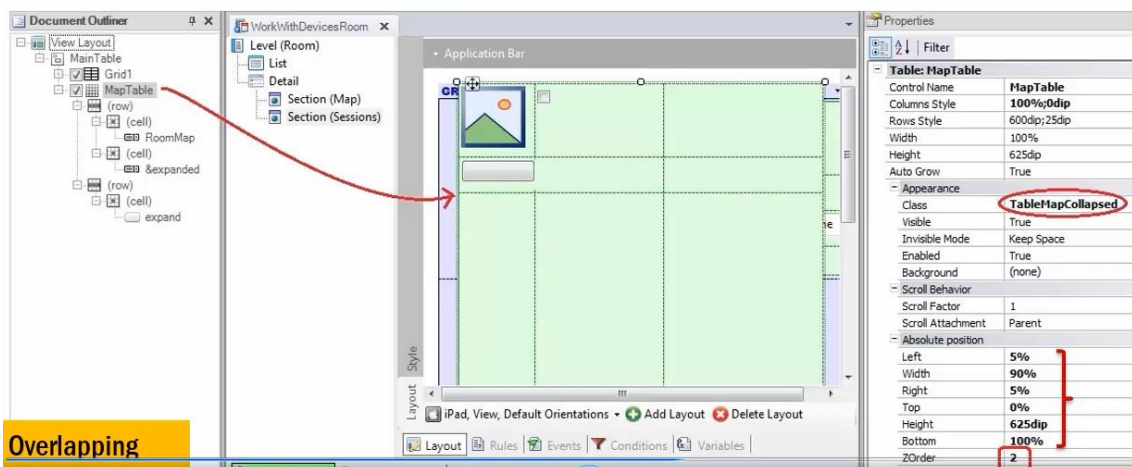
Y aquí vemos como la tabla de nombre MapTable

## Controls in layouts



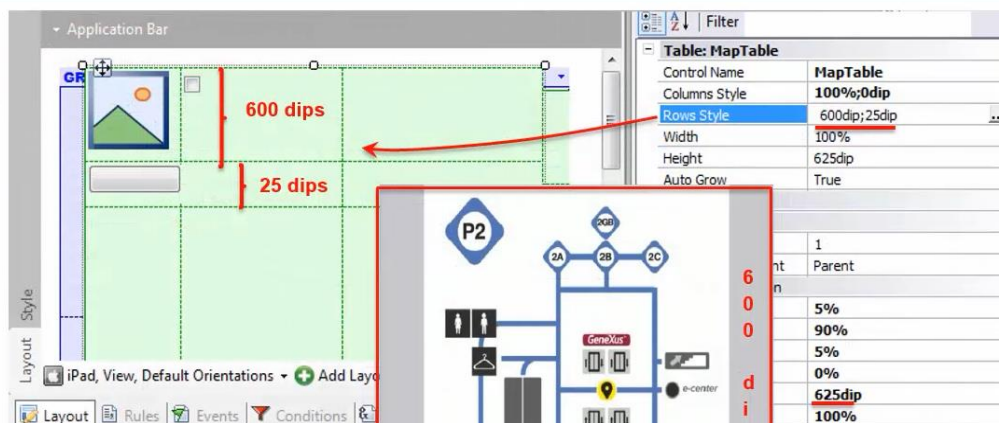
será la que se verá por encima, dado que su ZOrder es: 2

## Controls in layouts



También vemos cómo establece su posición absoluta: Left, Width, Right, Top, Height y Bottom.

Observemos que el Height es de 625 dips



que corresponde a los 600 dips de la primera fila + los 25 dips de la segunda.

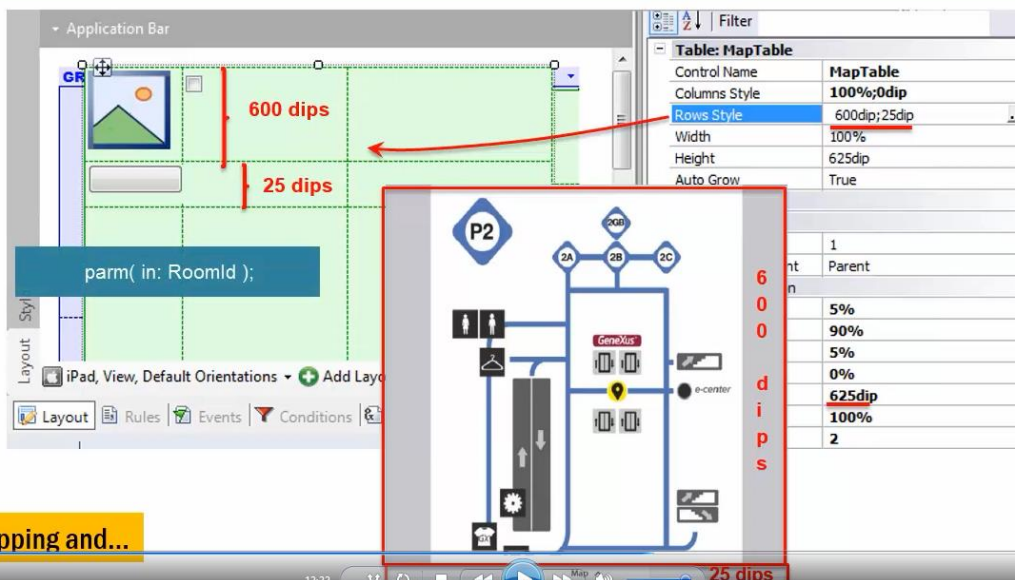
En la primera fila se encuentra el atributo RoomMap , que contiene la imagen del mapa de la sala.

Recordemos que este panel es llamado cuando se invoca al Detail del Work With Devices Room, por lo que recibe por parámetro el identificador de sala desde que se obtiene el RoomMap

14-Controls.Canvas.And.Transformations\_sp

Design— Canvas & Transformations

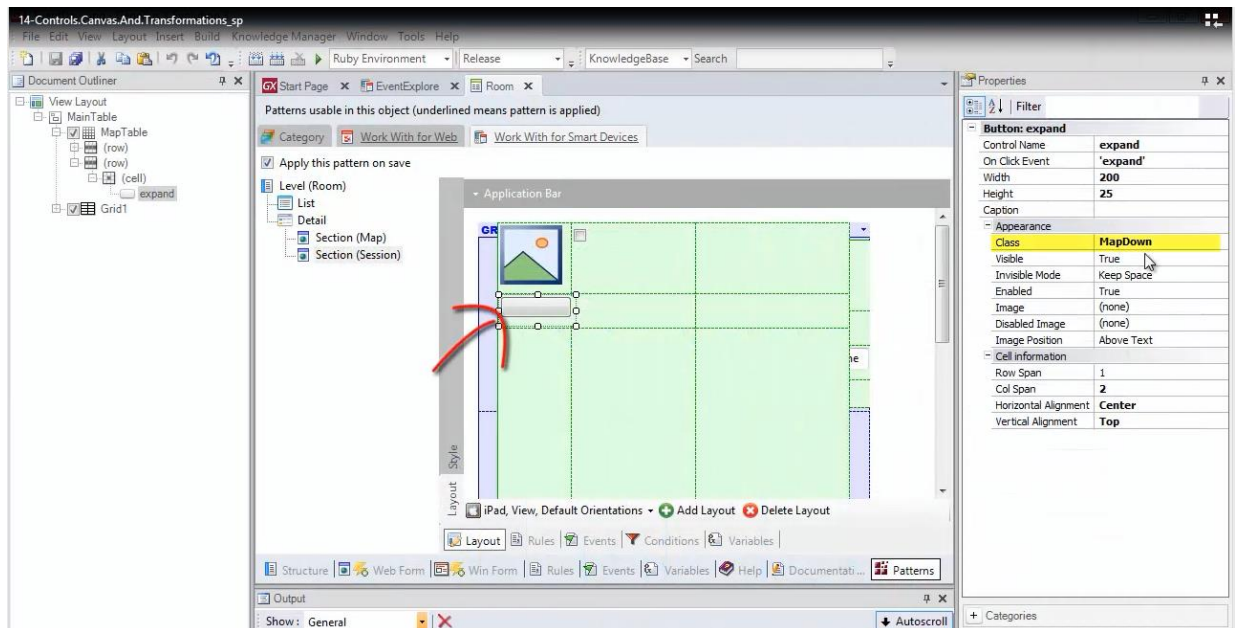
GeneXus



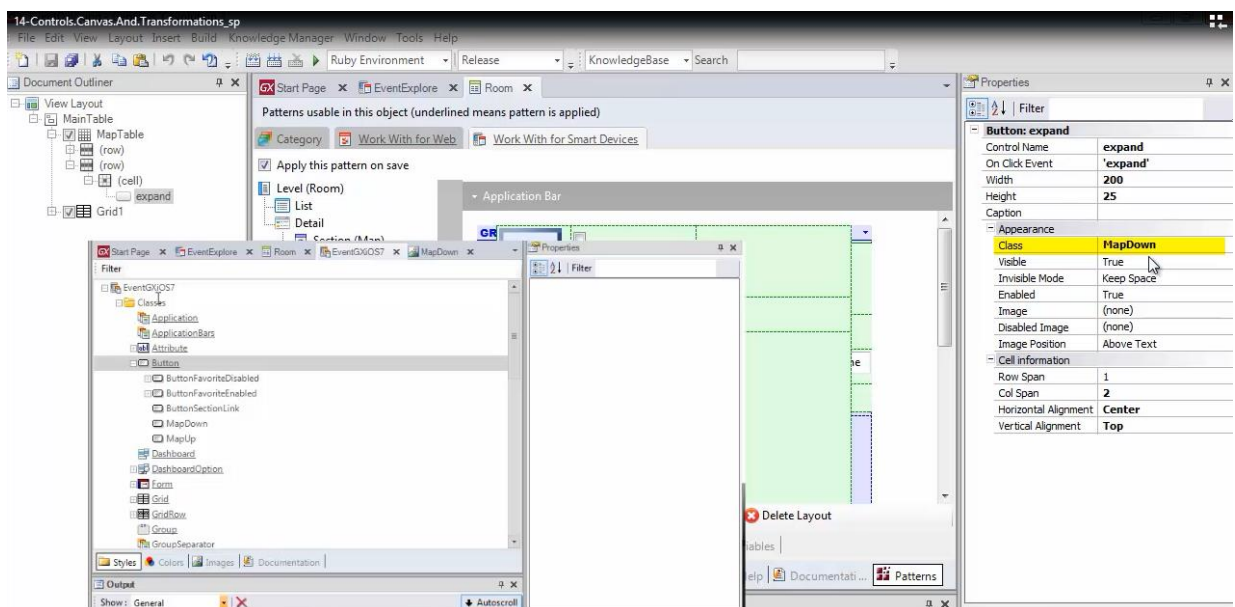
Overlapping and...

En la segunda fila se encuentra el botón, que tiene asignada una clase del Theme, que es la que configura la imagen que se mostrará en ejecución.

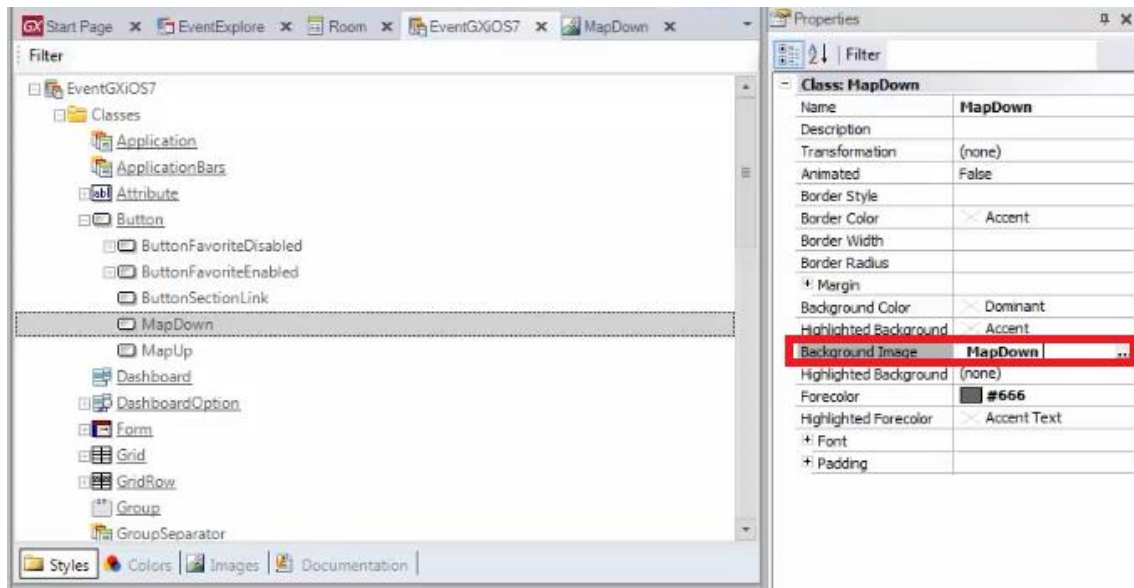




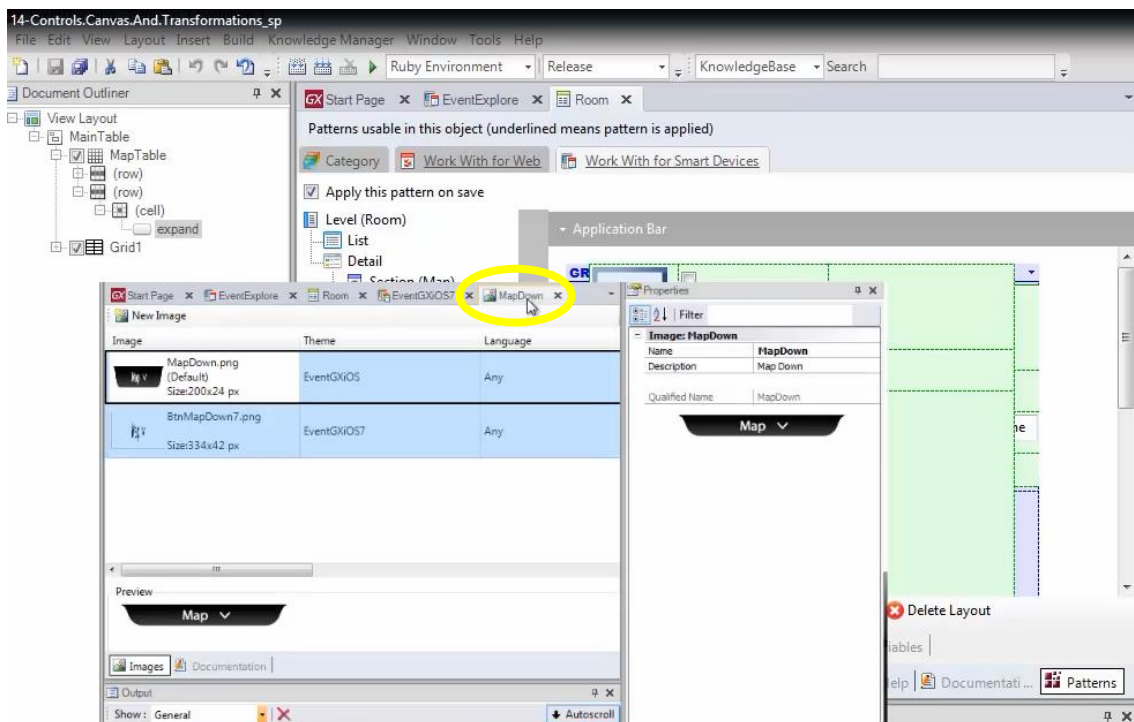
Aquí vemos el theme para iOS7



Y bajo la clase Button, tenemos la subclase: MapDown, que tiene como imagen de Background, la de igual nombre: MapDown

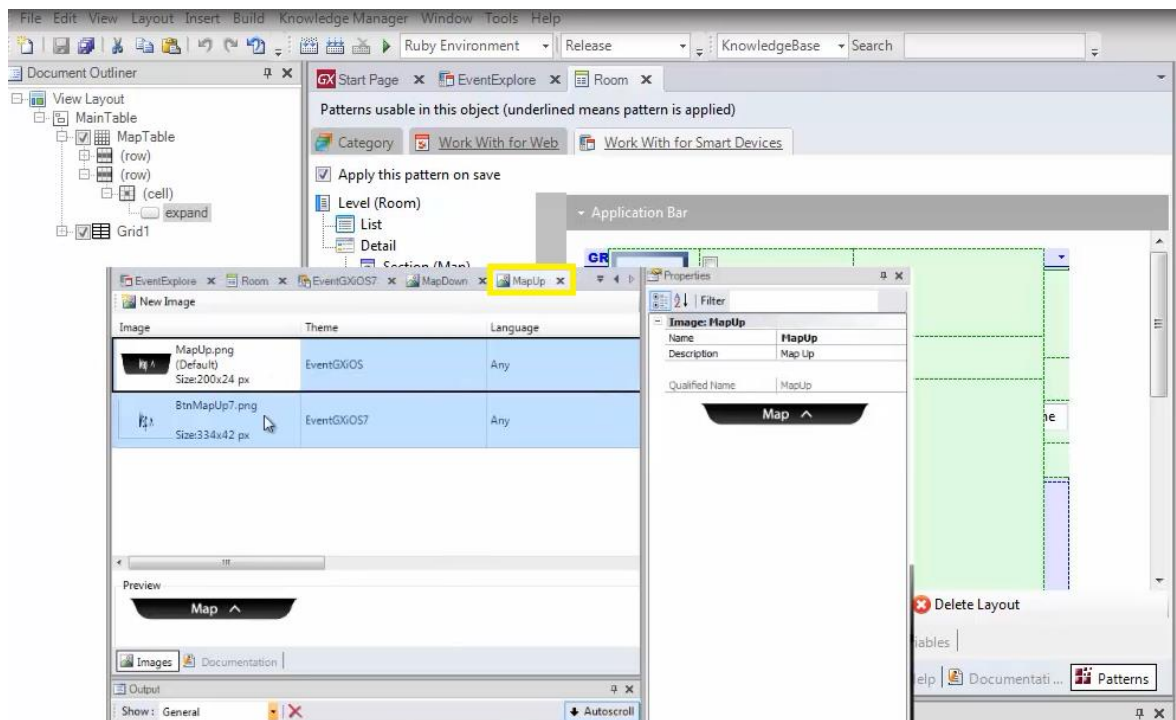


que si la vamos a ver

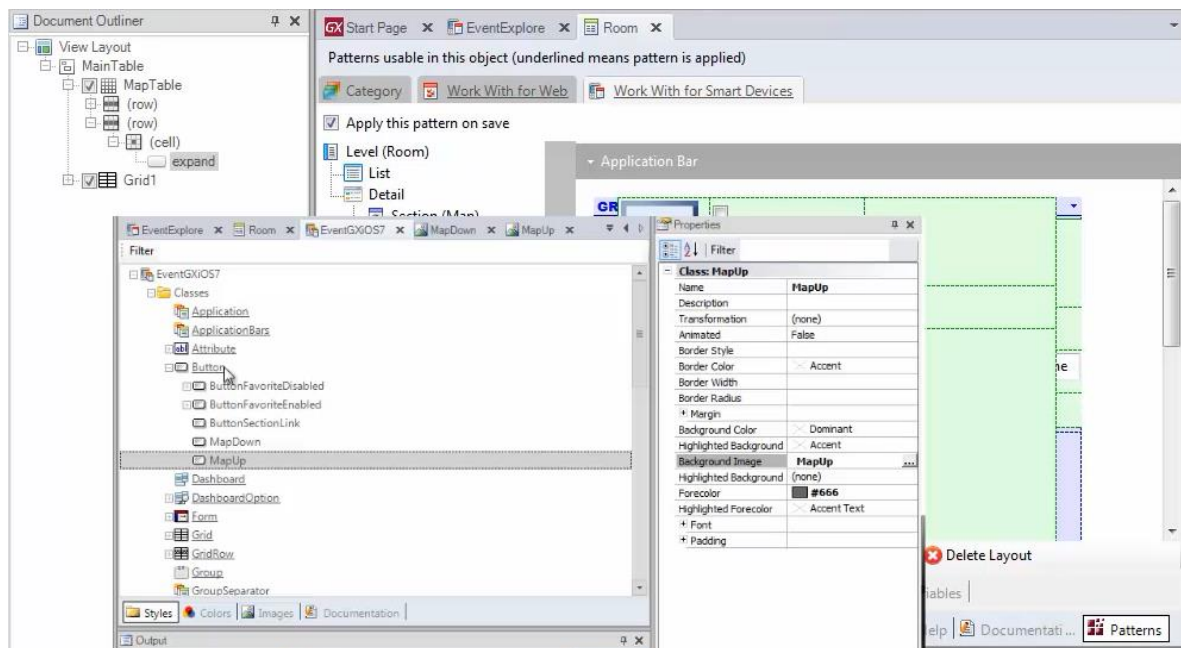


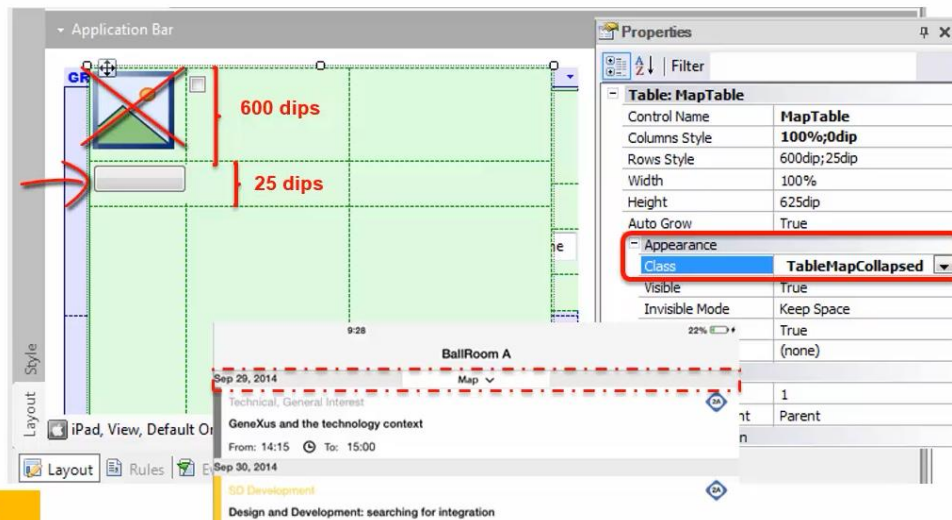
para el theme iOS7, es la que mostrábamos en ejecución.

Luego también tenemos la imagen MapUp



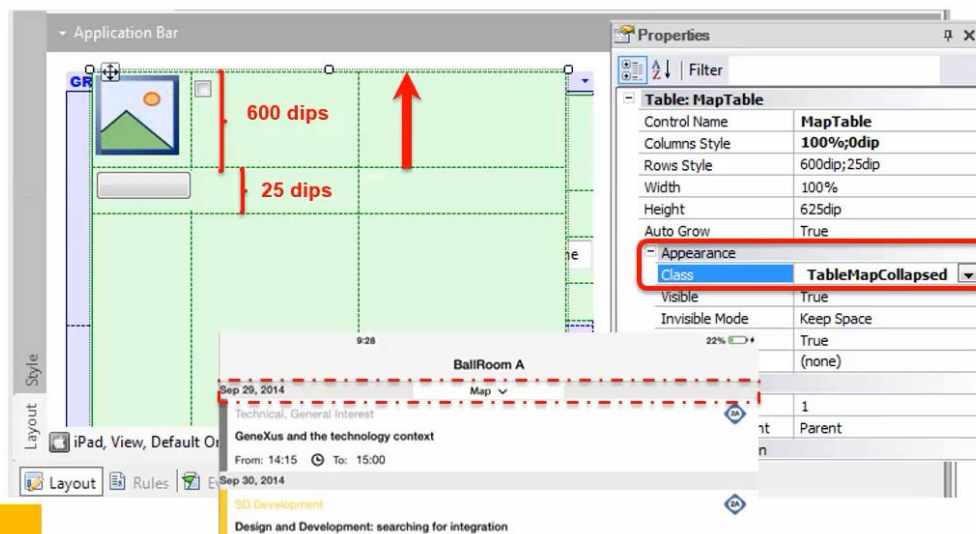
que estará asociada a la clase MapUp, bajo la clase Button





### Movement

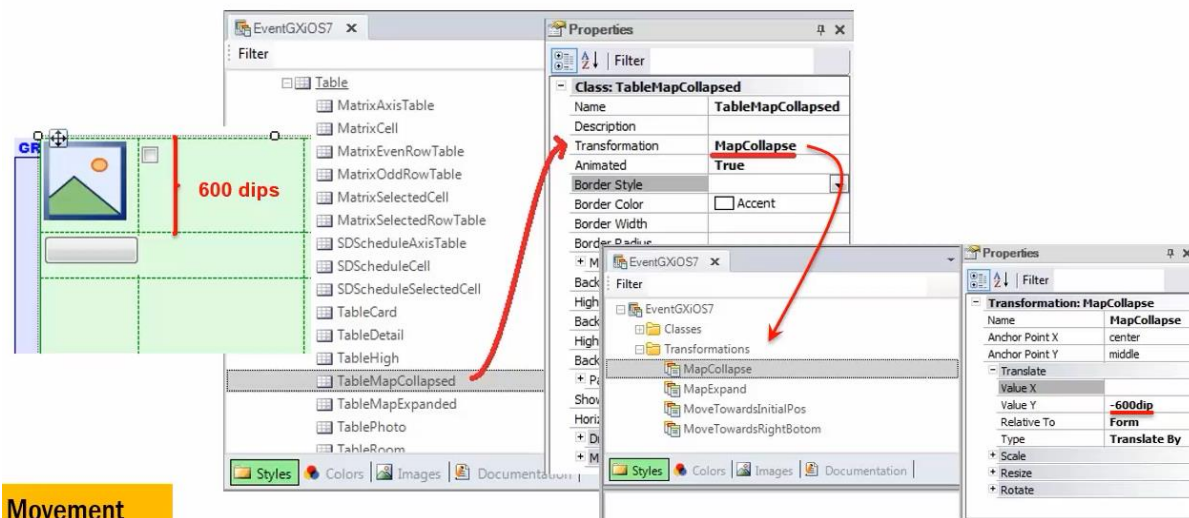
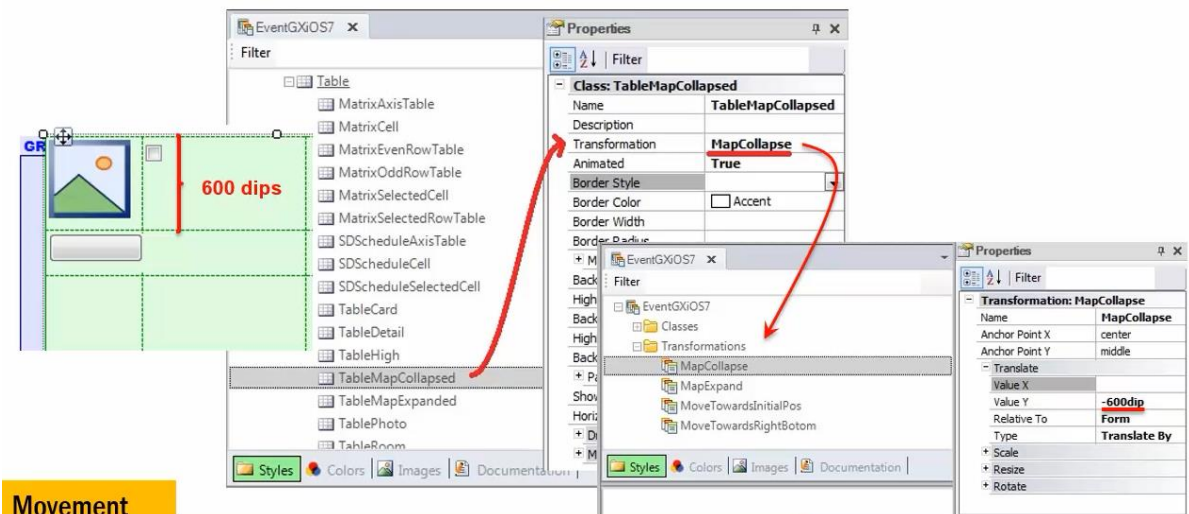
¿Por qué cuando invocamos al objeto por primera vez, vemos solamente el botón de expand MapDown y lo vemos arriba sin el mapa, como si toda la tabla se hubiera movido 600 dips verticalmente hacia arriba, dejando fuera del espacio del layout a la imagen del mapa?



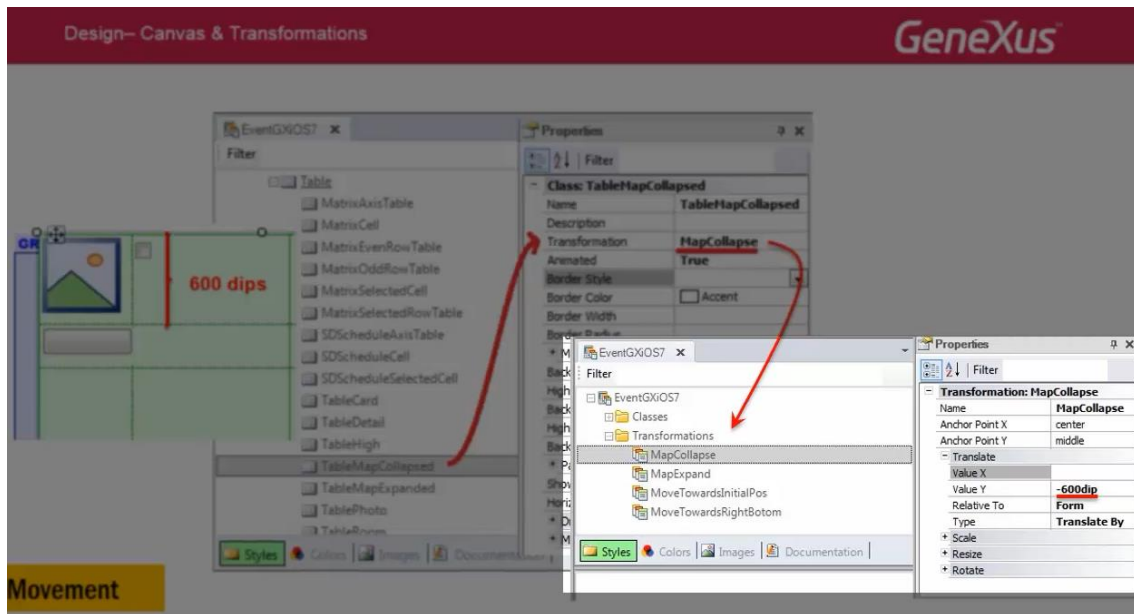
### Movement

Este comportamiento lo tiene especificado la clase que le hemos asignado a la tabla: **TableMapCollapsed**, clase que hemos creado nosotros los desarrolladores dentro del fin para ipad.

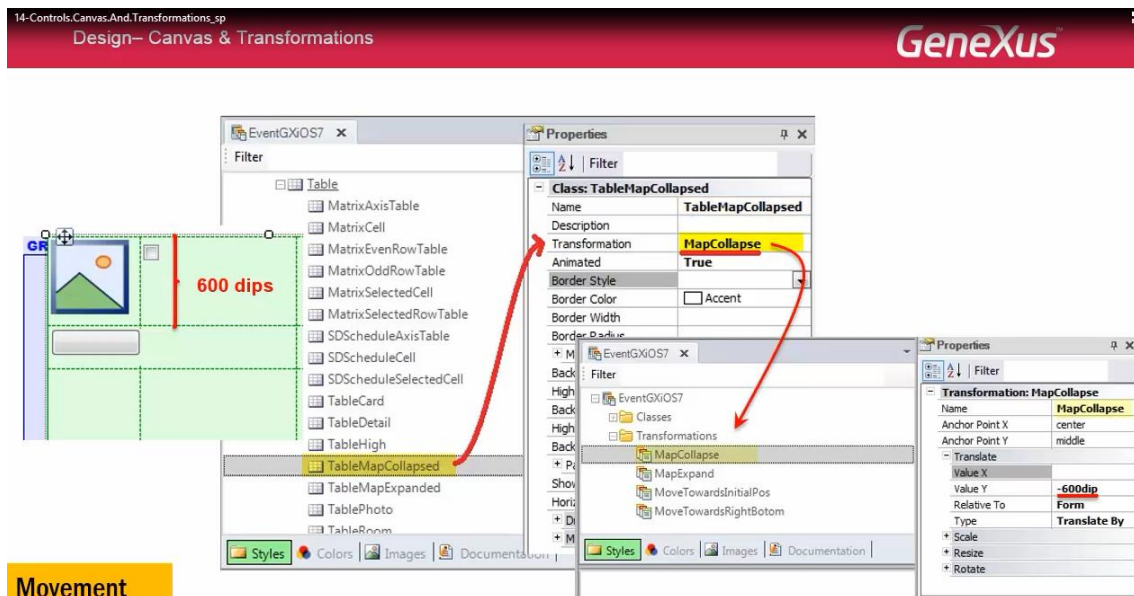




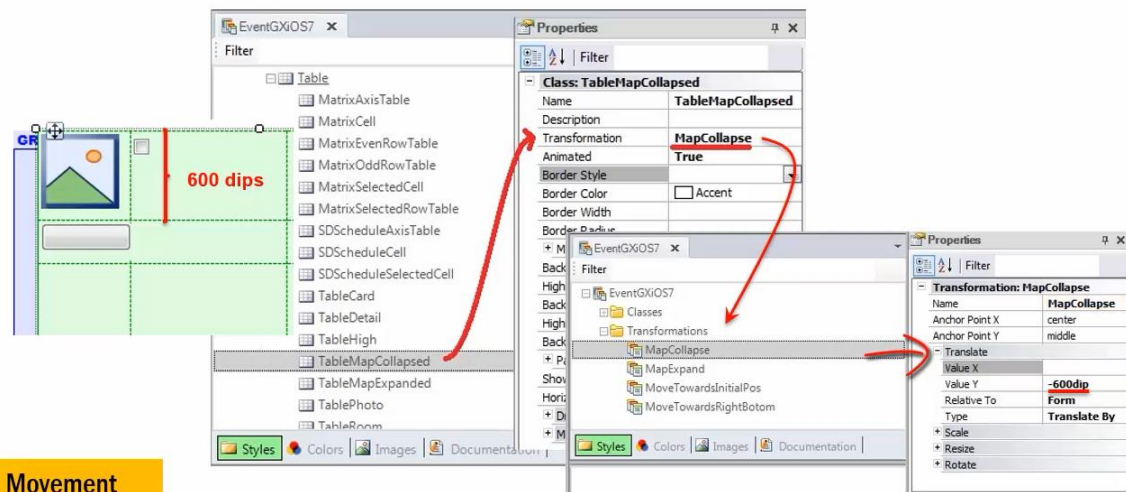
Si observamos, no sólo hemos creado esta clase, sino que hemos creado otra bajo el nodo **Transformation** del propio theme



y a la primera le asignamos entre sus propiedades, la segunda

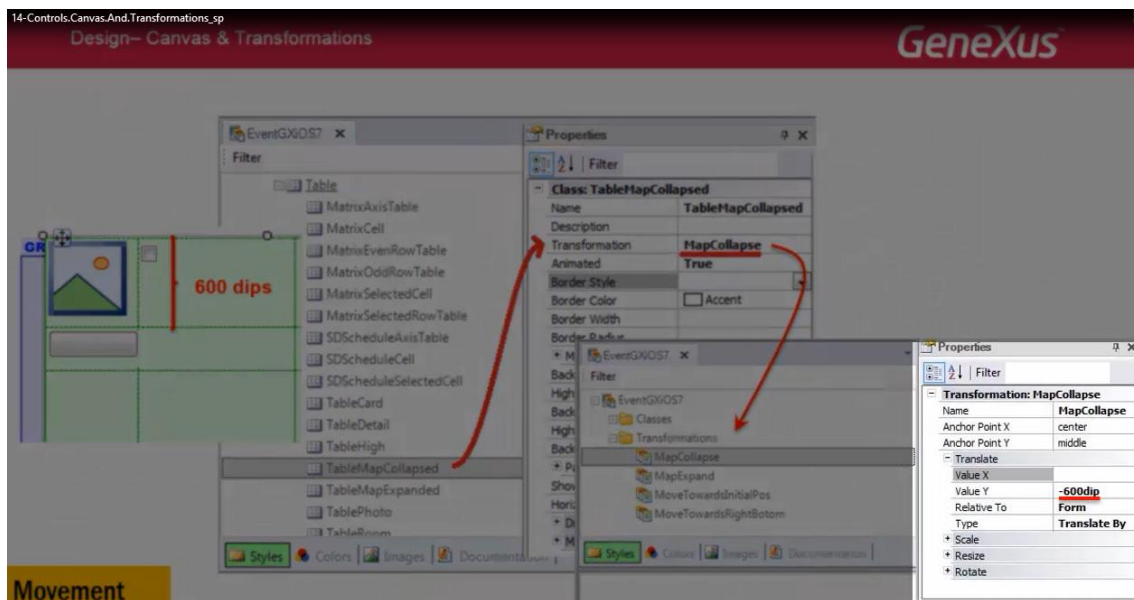


Es esta segunda la que especifica el comportamiento que vemos en ejecución, **de traslación** de la tabla entera

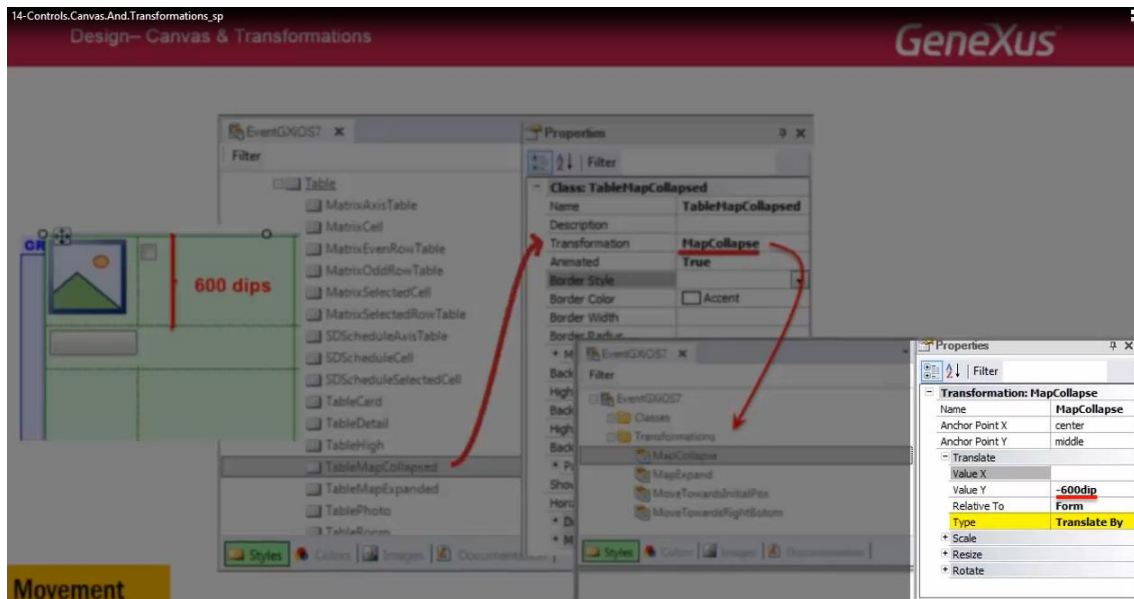


### Movement

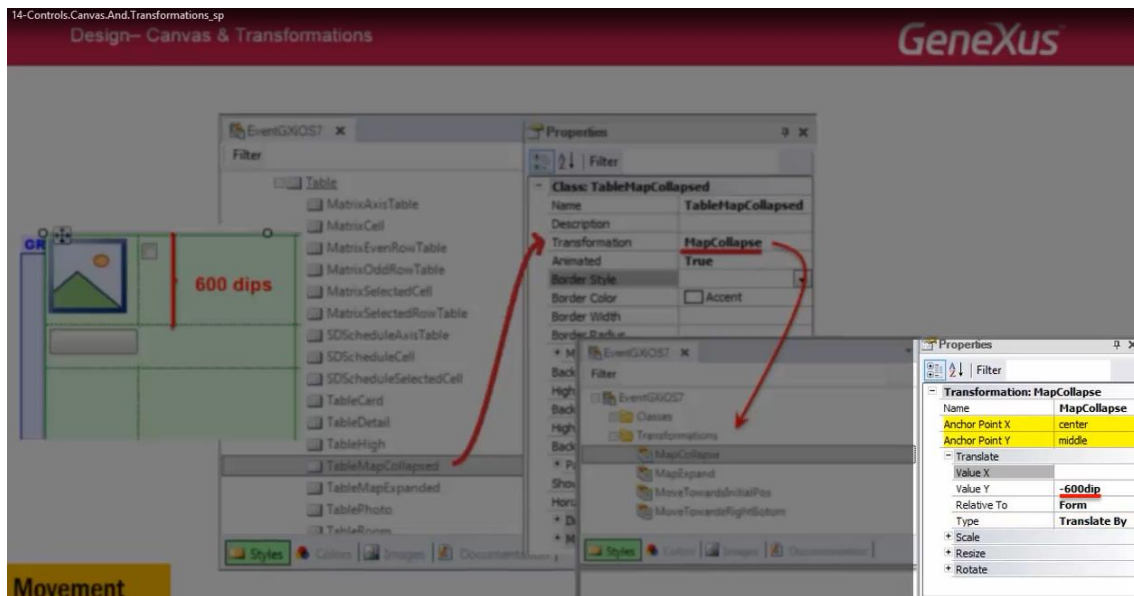
El control que tenga asociada esta clase, a través de otra, en nuestro caso, la tabla que tiene la clase **TableMapCollapsed**, que tiene la transformación **MapCollapse**



se moverá desde : **Translate By**



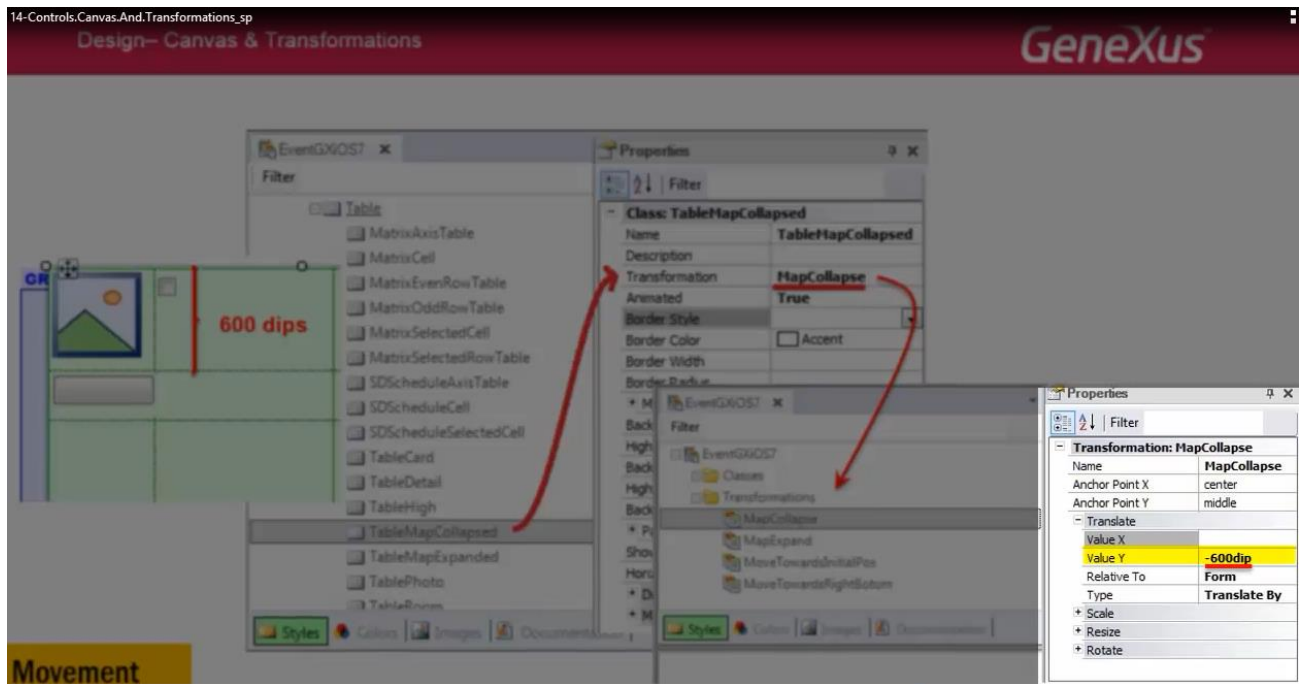
el punto X, Y indicado en las propiedades



Anchor Point X , Anchor Point Y .. en nuestro caso: el centro de la tabla..

En el eje Y, 600 dips hacia arriba



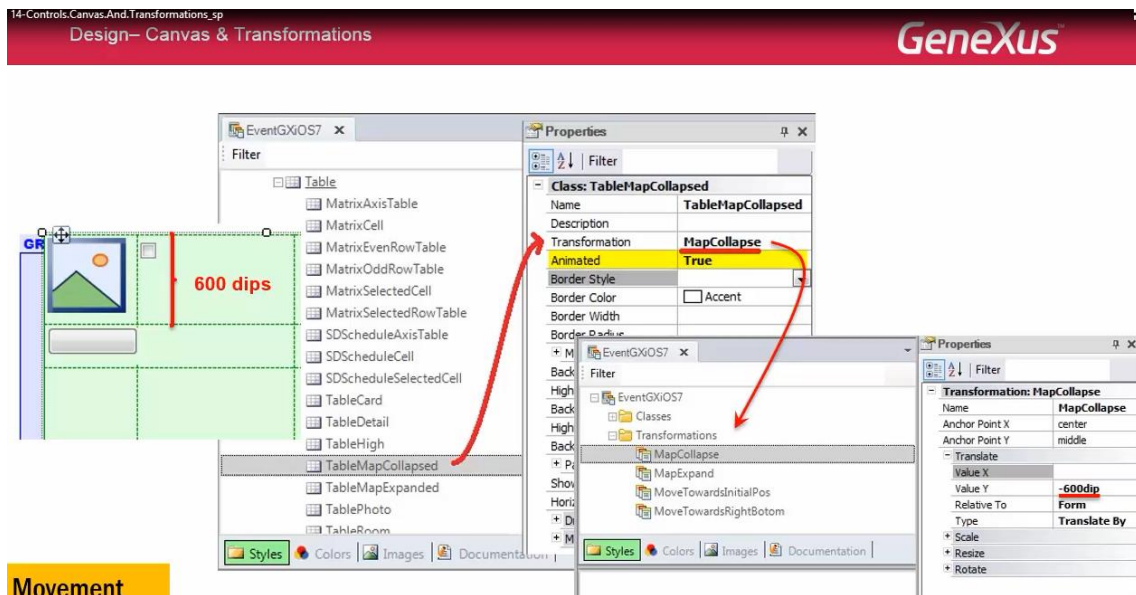


tomando como origen de coordenadas **Relative To: el From.**

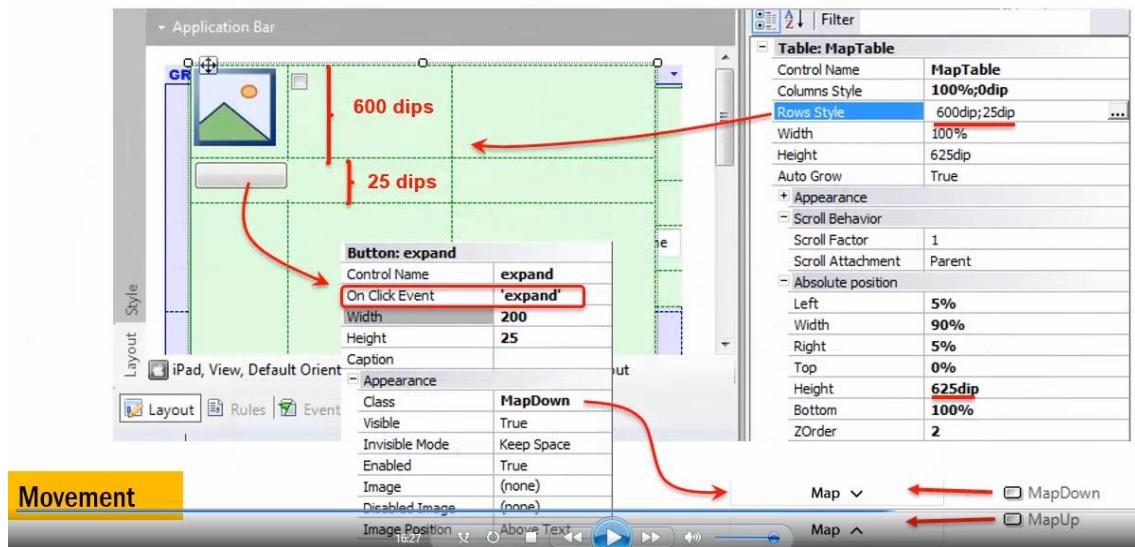
El origen cero cero, equivale al vértice superior izquierdo del form.

En nuestro caso, la tabla se moverá hacia arriba 600 dips. Como está al borde de la pantalla, ese movimiento dejará afuera la imagen.

Obsérvese que la clase de la tabla, tiene prendida la propiedad: Animated



lo que hará que ese desplazamiento se vea como tal cuando la tabla esté extendida y se colapse. Lo veremos a continuación.



Cuando la tabla está trasladada hacia arriba (colapsada), vamos a querer que cuando el usuario presione el botón Map, ésta se expanda; lo que no es otra cosa que decir, que aparezca tal cual la vemos en diseño, pero de forma animada dando la sensación de movimiento.

¿Cómo conseguimos que la tabla tenga el comportamiento anterior? (e traslade en forma animada algunas veces, y otras... cuando el usuario presiona el botón, que aparezca tal cual la vemos en diseño solapada con el grid?)

Reparemos también en que el botón debe cambiar su imagen en un caso y en el otro.

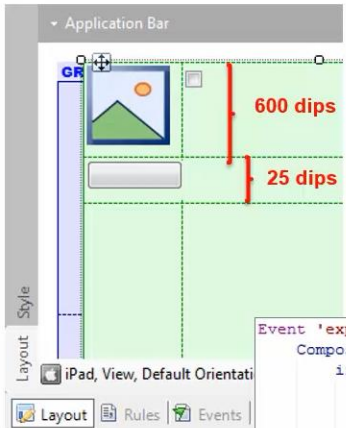
Si observamos el botón Expand, tiene configurado en Diseño la clase **MapDown** del theme, que a su vez, tiene configurada la imagen.

Pero recordemos que vimos otra clase: MapUp, que tiene configurada la otra imagen. Será en el evento Expand, asociado al botón, donde programaremos el comportamiento en un caso y otro (cuando está expandida la información y cuando está colapsada)

Cómo simplemente cambiándoles las clases a la tabla y al botón, de acuerdo al caso que se trate.

Cuando la tabla está trasladada, es decir, colapsada, es decir: not expanded

Design– Canvas & Transformations
GeneXus™



**EventGXOS x**  
 Filter  
 TableHigh  
 TableMapCollapsed  
 TableMapExpanded  
 TablePhoto  
 TableRoom  
 TableRounded  
 TableSection

Event 'expand'  
 Composite  
 if &expanded  
   MapTable.Class = "TableMapCollapsed"  
   expand.Class = "MapDown"  
   &expanded = false  
 else  
   MapTable.Class = "TableMapExpanded"  
   expand.Class = "MapUp"  
   &expanded = true  
 endif  
 Endcomposite  
 Endevent

Class: TableMapCollapsed	
Name	TableMapCollapsed
Description	
Transformation	MapCollapse
Animated	True

Class: TableMapExpanded	
Name	TableMapExpanded
Description	
Transformation	MapExpand
Animated	True

Map ▾  
 Map ▲

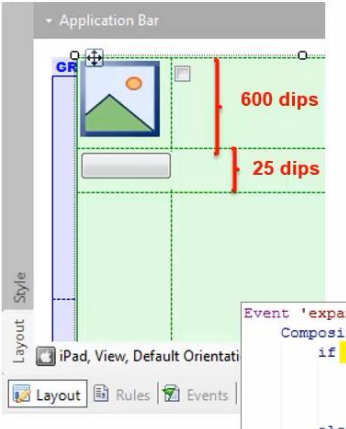
Event Start  
   &expanded = false  
 Endevent

**Movement**

se le asigna a la tabla, la clase “TableMapExpanded” que tiene la propiedad: Animated y tiene definida una traslación vacía, que no hace nada. Y al botón, se le cambia la clase, para que ofrezca la posibilidad de colapsar la información.

Obsérvese que se hace uso de una variable que hace de switch

Design– Canvas & Transformations
GeneXus™



**EventGXOS x**  
 Filter  
 TableHigh  
 TableMapCollapsed  
 TableMapExpanded  
 TablePhoto  
 TableRoom  
 TableRounded  
 TableSection

Event 'expand'  
 Composite  
 if &expanded  
   MapTable.Class = "TableMapCollapsed"  
   expand.Class = "MapDown"  
   &expanded = false  
 else  
   MapTable.Class = "TableMapExpanded"  
   expand.Class = "MapUp"  
   &expanded = true  
 endif  
 Endcomposite  
 Endevent

Class: TableMapCollapsed	
Name	TableMapCollapsed
Description	
Transformation	MapCollapse
Animated	True

Class: TableMapExpanded	
Name	TableMapExpanded
Description	
Transformation	MapExpand
Animated	True

Map ▾  
 Map ▲

Event Start  
   &expanded = false  
 Endevent

**Movement**

Prende / Apaga :

Application Bar

600 dips

25 dips

EventGXIOS x

Filter

- TableHigh
- TableMapCollapsed
- TableMapExpanded
- TablePhoto
- TableRoom
- TableRounded
- TableSection

Class: TableMapCollapsed

Name	TableMapCollapsed
Description	
Transformation	MapCollapse
Animated	True

Class: TableMapExpanded

Name	TableMapExpanded
Description	
Transformation	MapExpand
Animated	True

Event 'expand'

```

Composite
if &expanded
  MapTable.Class = "TableMapCollapsed"
  expand.Class = "MapDown"
  &expanded = false
else
  MapTable.Class = "TableMapExpanded"
  expand.Class = "MapUp"
  &expanded = true
endif
Endcomposite
  
```

MapDown

MapUp

Event Start

```

&expanded = false
Endevent
  
```

Movement

Endevent 18:06

Se inicializa en false

Application Bar

600 dips

25 dips

EventGXIOS x

Filter

- TableHigh
- TableMapCollapsed
- TableMapExpanded
- TablePhoto
- TableRoom
- TableRounded
- TableSection

Class: TableMapCollapsed

Name	TableMapCollapsed
Description	
Transformation	MapCollapse
Animated	True

Class: TableMapExpanded

Name	TableMapExpanded
Description	
Transformation	MapExpand
Animated	True

Event 'expand'

```

Composite
if &expanded
  MapTable.Class = "TableMapCollapsed"
  expand.Class = "MapDown"
  &expanded = false
else
  MapTable.Class = "TableMapExpanded"
  expand.Class = "MapUp"
  &expanded = true
endif
Endcomposite
  
```

MapDown

MapUp

Event Start

```

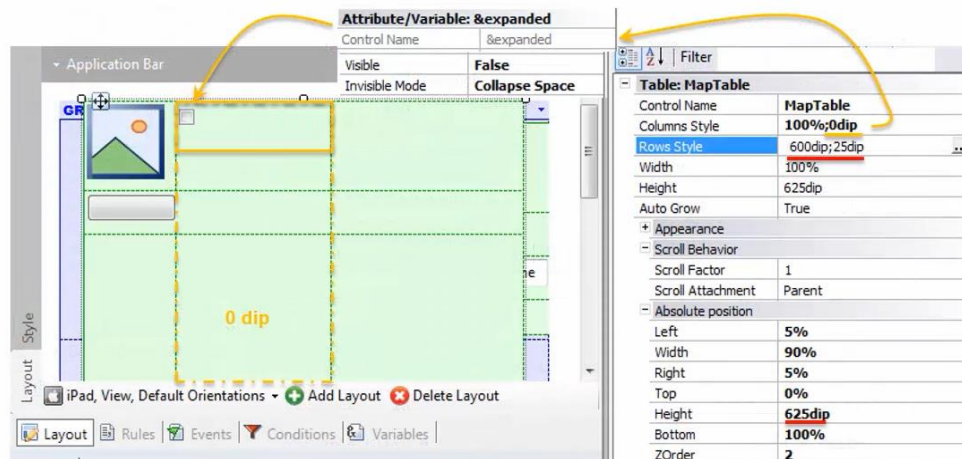
&expanded = false
Endevent
  
```

Movement

Endevent 18:10

indicando que inicia colapsada y por ello el valor en diseño de la clase de la table es: "TableMapCollapsed". Para que el valor de esta variable se mantenga entre ejecuciones, es que se coloca en el layout, invisible.





### Movement

Así, la segunda columna está solo a los efectos de mantener en pantalla el valor de la variable &expanded, que servirá entonces para que el evento asociado al botón sepa, cada vez que este se presiona, si lo que se necesita es expandir o colapsar.

Como el control está invisible y su espacio no se mantiene (observe el valor de la propiedad: Invisible Mode), que puede tomar el valor Collapse Space o Keep Space... y además, como especificamos cero dips para el tamaño de la columna, no afectará en nada la visualización.

The screenshot shows the GeneXus design tool interface with the table control expanded. The dimensions are indicated as 600 dips for the first column and 25 dips for the second column. The EventGXGOS x pane shows the following event logic:

```

Event 'expand'
Composite
  if &expanded
    MapTable.Class = "TableMapCollapsed"
    expand.Class = "MapDown"
    &expanded = false
  else
    MapTable.Class = "TableMapExpanded"
    expand.Class = "MapUp"
    &expanded = true
  endif
Endcomposite
Endevent

```

The right pane shows the class definitions for the table states:

Class: TableMapCollapsed	
Name	TableMapCollapsed
Description	
Transformation	MapCollapse
Animated	True

Class: TableMapExpanded	
Name	TableMapExpanded
Description	
Transformation	MapExpand
Animated	True

Below the class definitions, there are two buttons: 'Map v' and 'Map ^'. The 'Map v' button is associated with the 'MapDown' event, and the 'Map ^' button is associated with the 'MapUp' event. The 'Event Start' pane shows the initial state: &expanded = false.

### Movement

Si ahora tenemos expandida la tabla y presionamos el botón

14-Controls.Canvas.And.Transformations\_sp  
Design- Canvas & Transformations

GeneXus

Application Bar

Layout Style

iPad, View, Default Orientation

Layout Rules Events

EventGXOS

Filter

- TableHigh
- TableMapCollapsed
- TableMapExpanded
- TablePhoto
- TableRoom
- TableRounded
- TableSection

Styles Colors Images De

Class: TableMapCollapsed

Name	TableMapCollapsed
Description	
Transformation	MapCollapse
Animated	True

Class: TableMapExpanded

Name	TableMapExpanded
Description	
Transformation	MapExpand
Animated	True

Map v MapDown

Map ^ MapUp

Event Start

```

expanded = false
Endevent

```

Event 'expand'

```

Composite
if &expanded
  MapTable.Class = "TableMapCollapsed"
  expand.Class = "MapDown"
  &expanded = false
else
  MapTable.Class = "TableMapExpanded"
  expand.Class = "MapUp"
  &expanded = true
endif
Endcomposite
Endevent

```

Movement

Endevent 18:58

se le cambiarán las clases a la tabla y al botón.

El cambio en la clase de la tabla

14-Controls.Canvas.And.Transformations\_sp  
Design- Canvas & Transformations

GeneXus

Application Bar

Layout Style

iPad, View, Default Orientation

Layout Rules Events

EventGXOS

Filter

- TableHigh
- TableMapCollapsed
- TableMapExpanded
- TablePhoto
- TableRoom
- TableRounded
- TableSection

Styles Colors Images De

Class: TableMapCollapsed

Name	TableMapCollapsed
Description	
Transformation	MapCollapse
Animated	True

Class: TableMapExpanded

Name	TableMapExpanded
Description	
Transformation	MapExpand
Animated	True

Map v MapDown

Map ^ MapUp

Event Start

```

expanded = false
Endevent

```

Event 'expand'

```

Composite
if &expanded
  MapTable.Class = "TableMapCollapsed"
  expand.Class = "MapDown"
  &expanded = false
else
  MapTable.Class = "TableMapExpanded"
  expand.Class = "MapUp"
  &expanded = true
endif
Endcomposite
Endevent

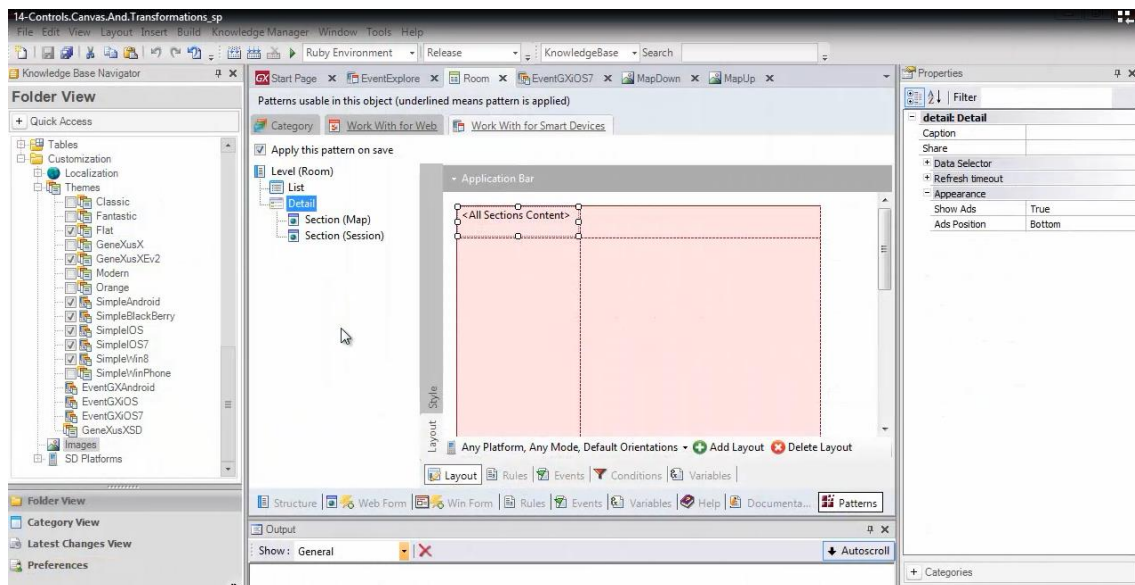
```

Movement

Endevent 19:04

Press and hold to fast-forward

es la que provocará que en forma animada, se mueva hacia arriba 600 dips, dando la impresión de colapsarse.

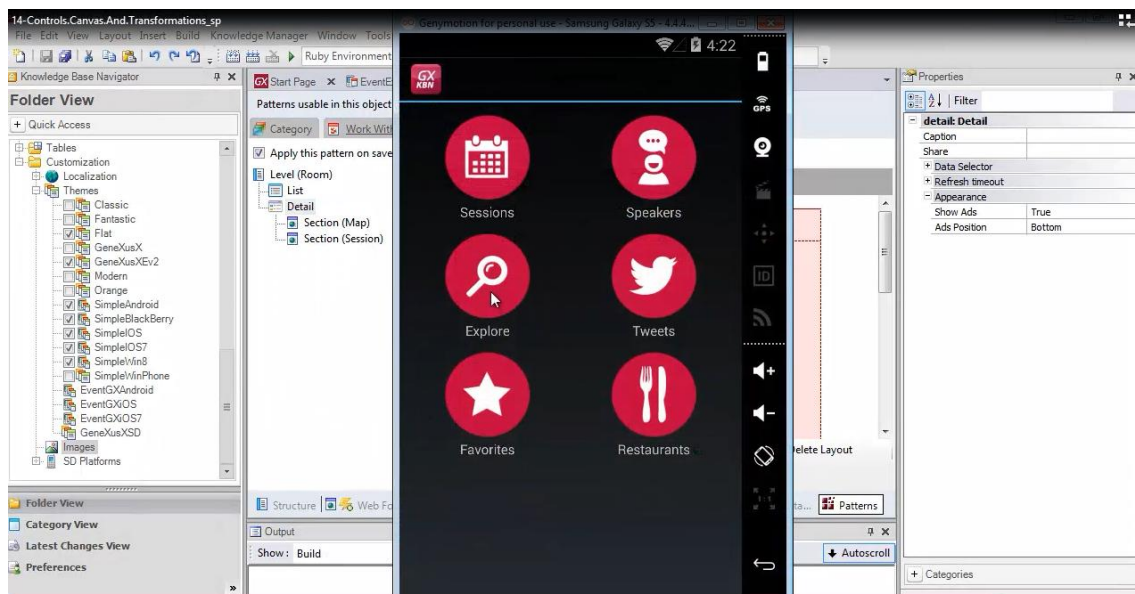


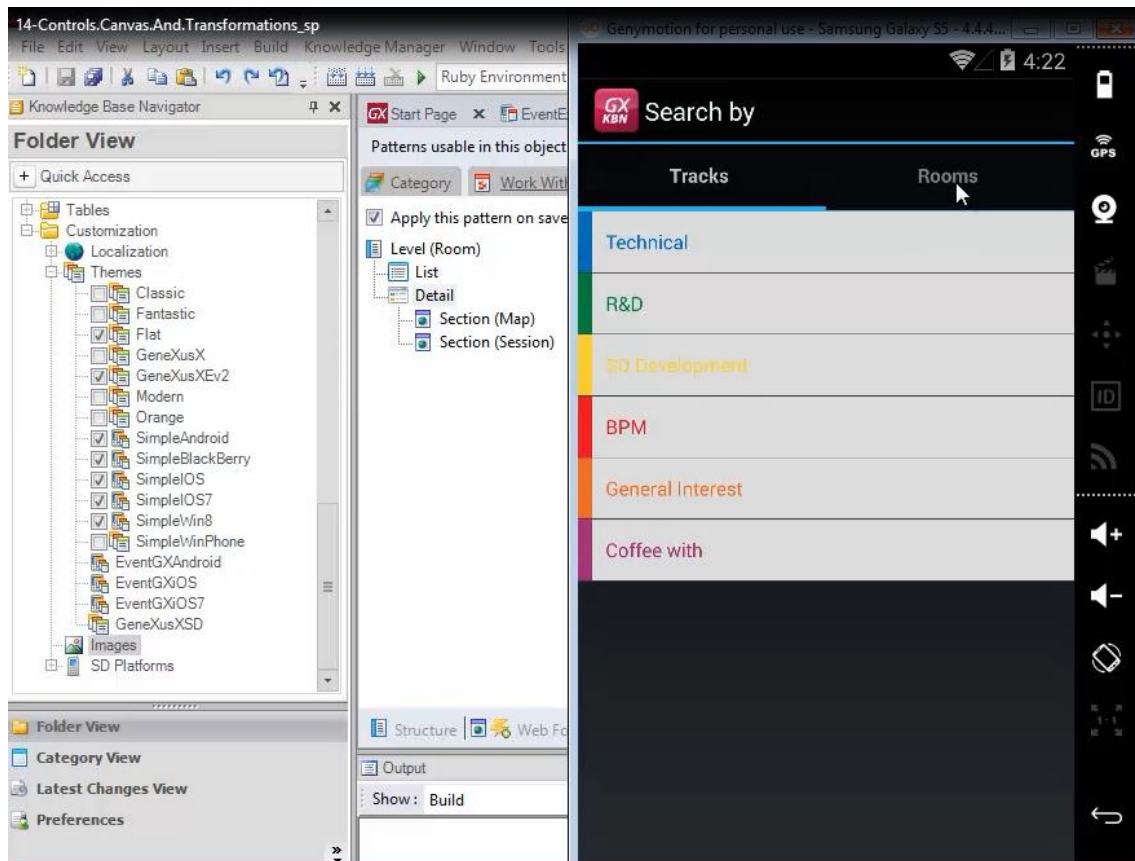
Por último, identifiquemos todo lo visto en GeneXus.

Estamos en la transacción Room, en la sección Patterns, Work With for Smart Devices, nodo Detail.

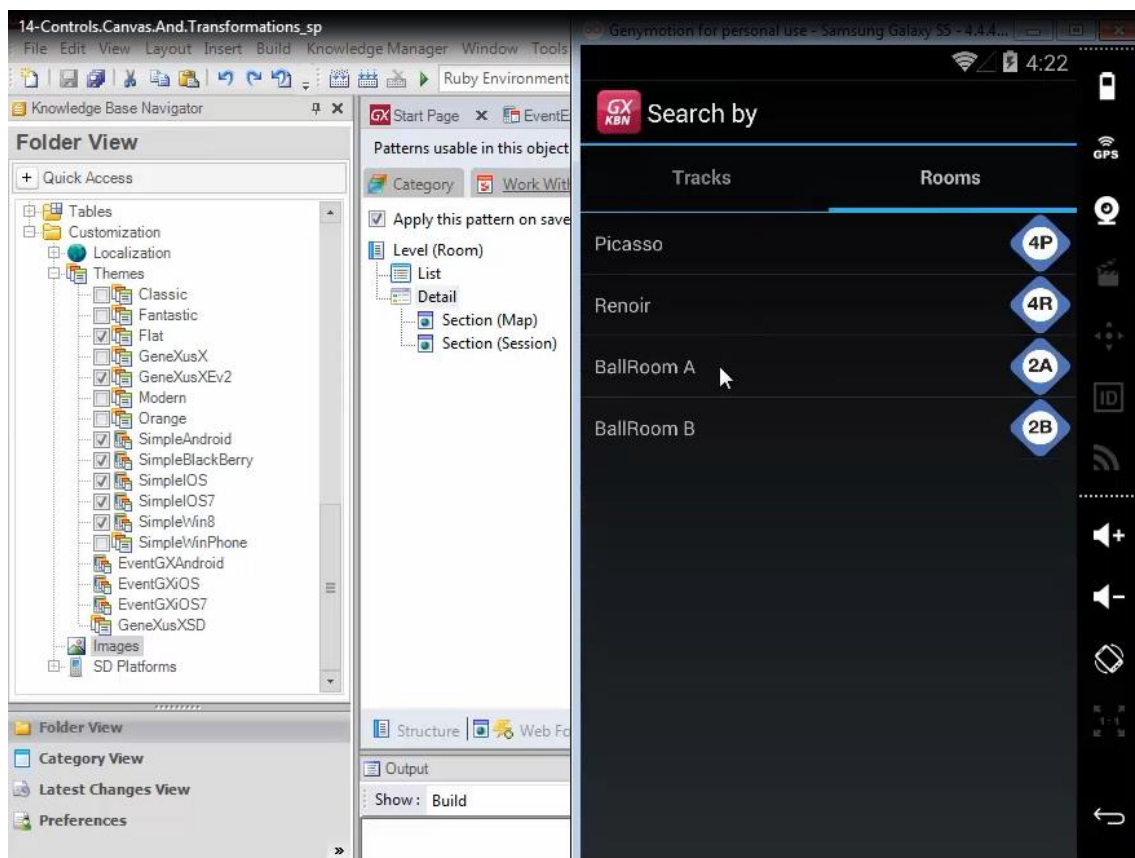
Aquí vemos el <All Sections Content> donde se van a cargar cada una de las secciones que tenemos. Pero esto no es lo que deseamos pues necesitamos que ambas secciones (la que muestra el mapa y la que muestra el grid con las conferencias) aparezcan solapadas.

Sin embargo, si ejecutamos la aplicación Android



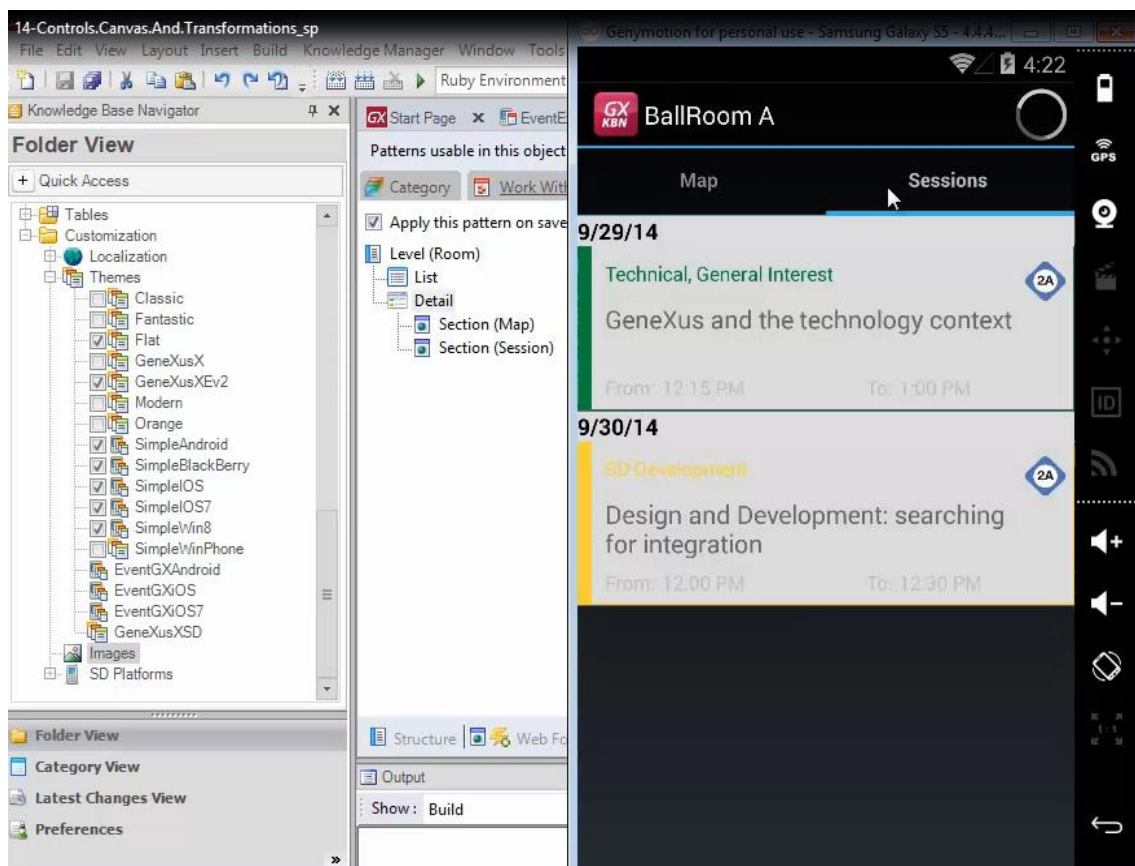
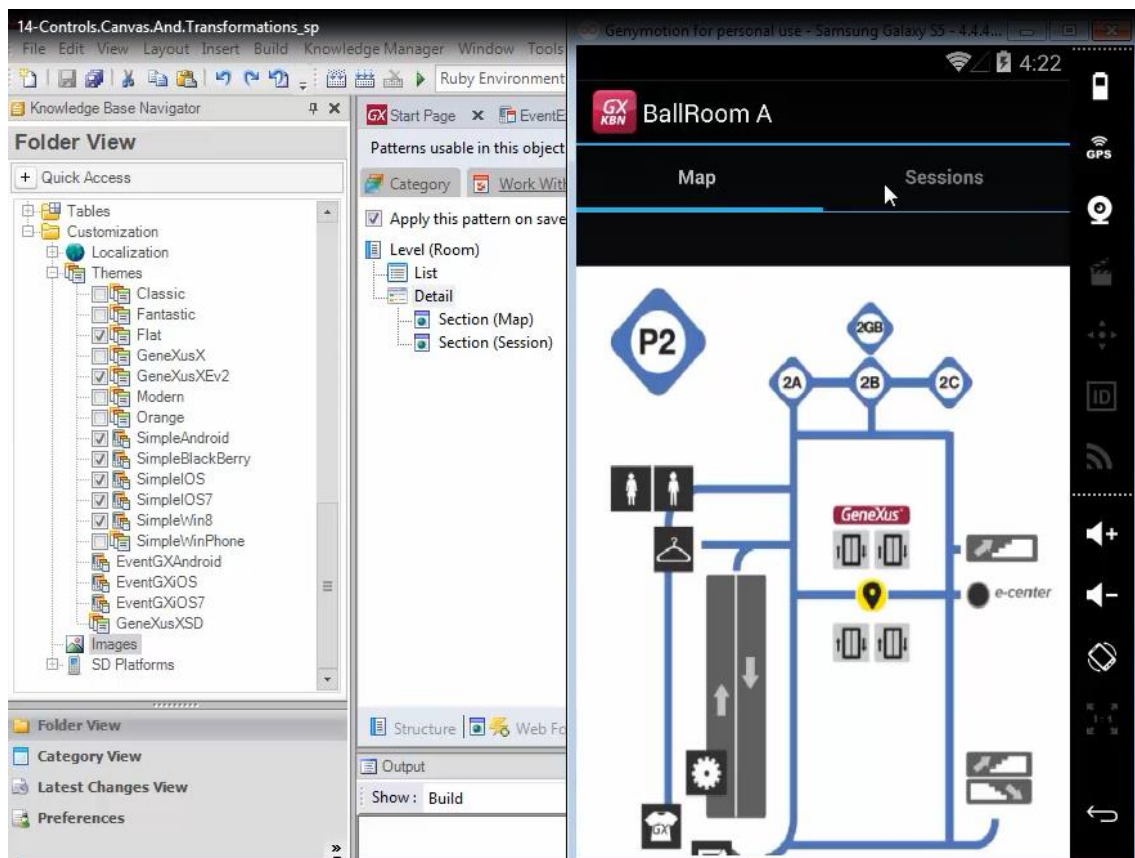


vamos a los Rooms y elegimos por ejemplo: BallRoom A

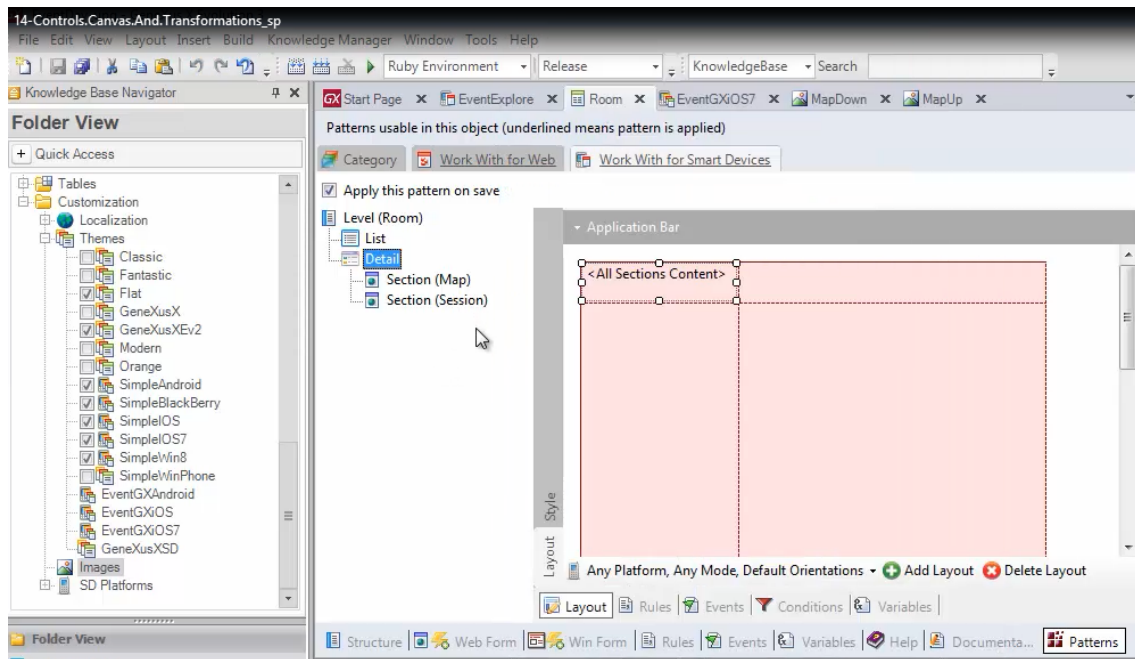


Vemos que el Detail se está mostrando como 2 TABS



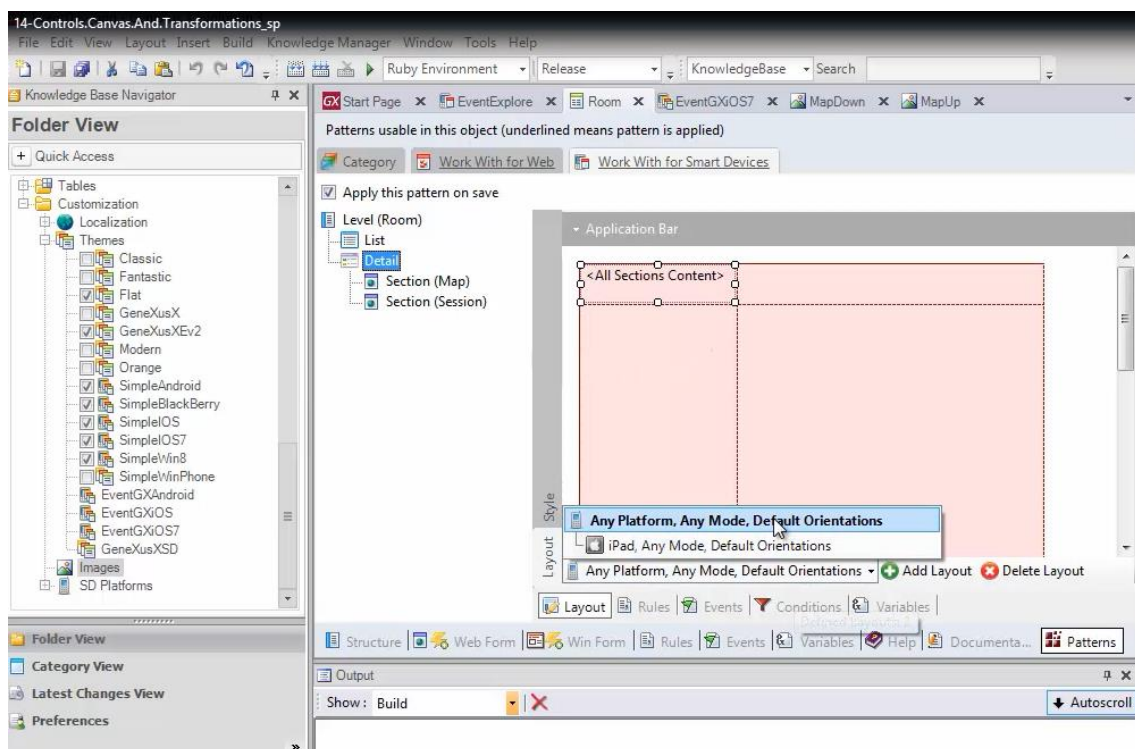


Así como los estamos viendo en GeneXus

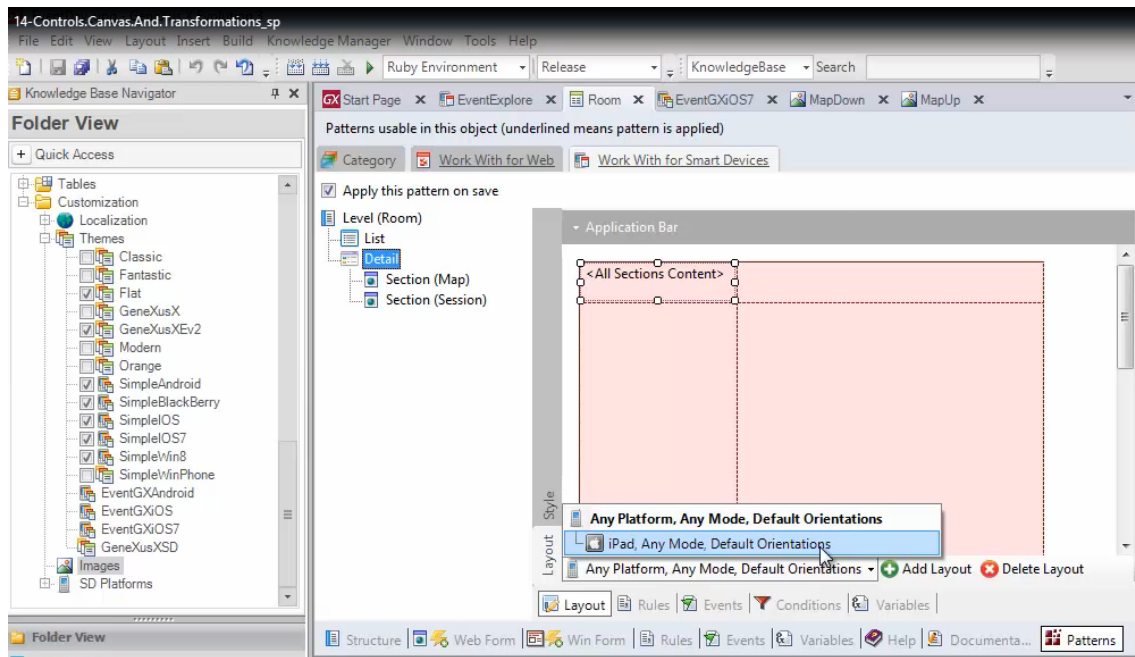


¿Dónde está programado que para iPad, se muestre del modo que estudiamos?

Si observamos el Layout que estamos viendo

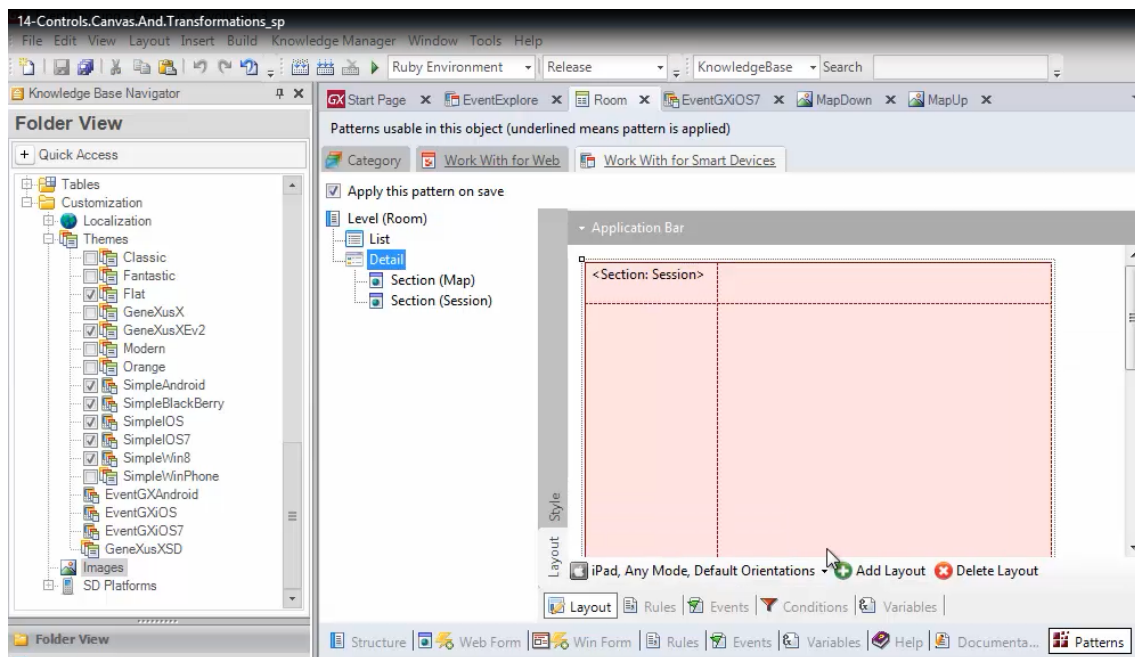


corresponde a cualquier plataforma, cualquier modo y las orientaciones Default... pero tenemos otro layout definido

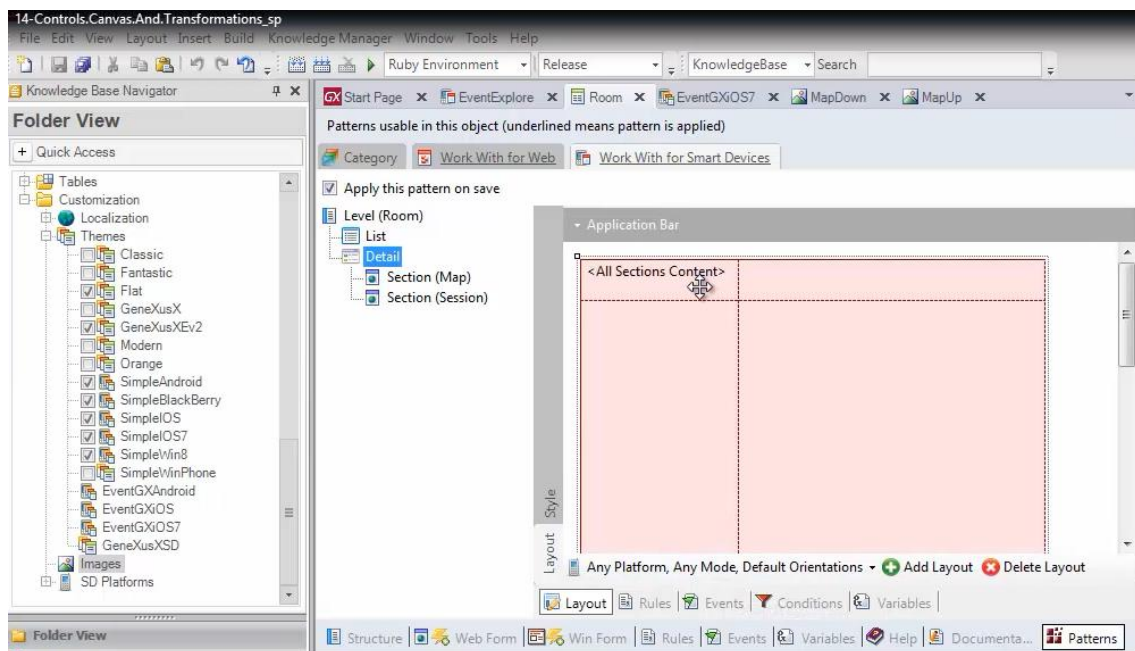
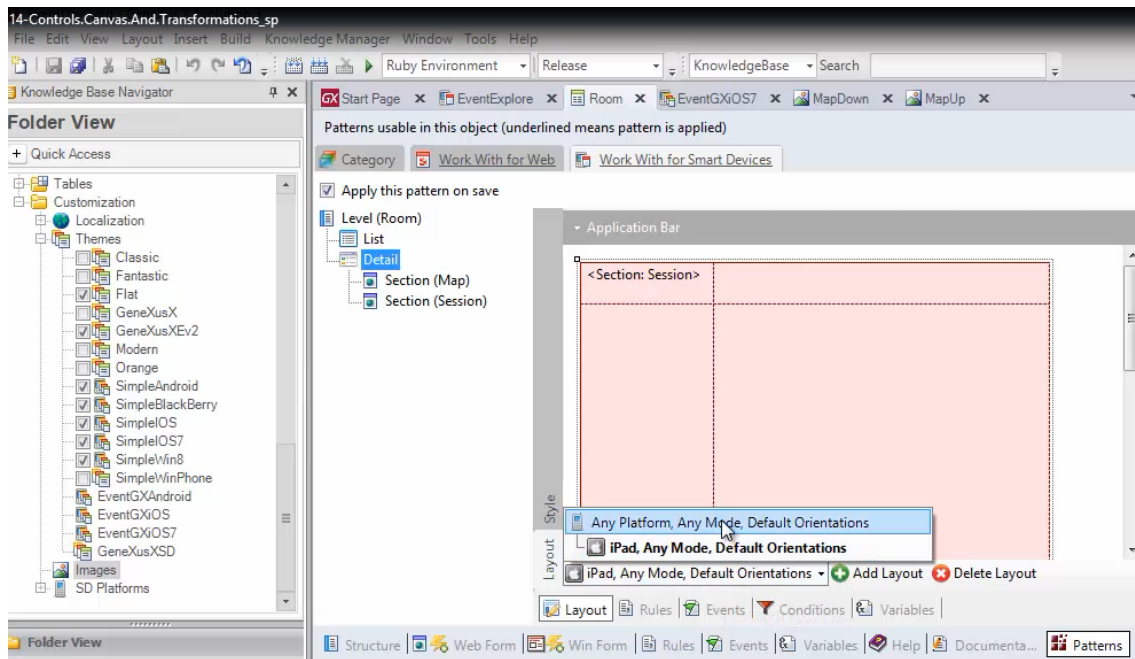


esta vez para iPad, en cualquier modo y con las orientaciones Default.

Vamos a editarlo.

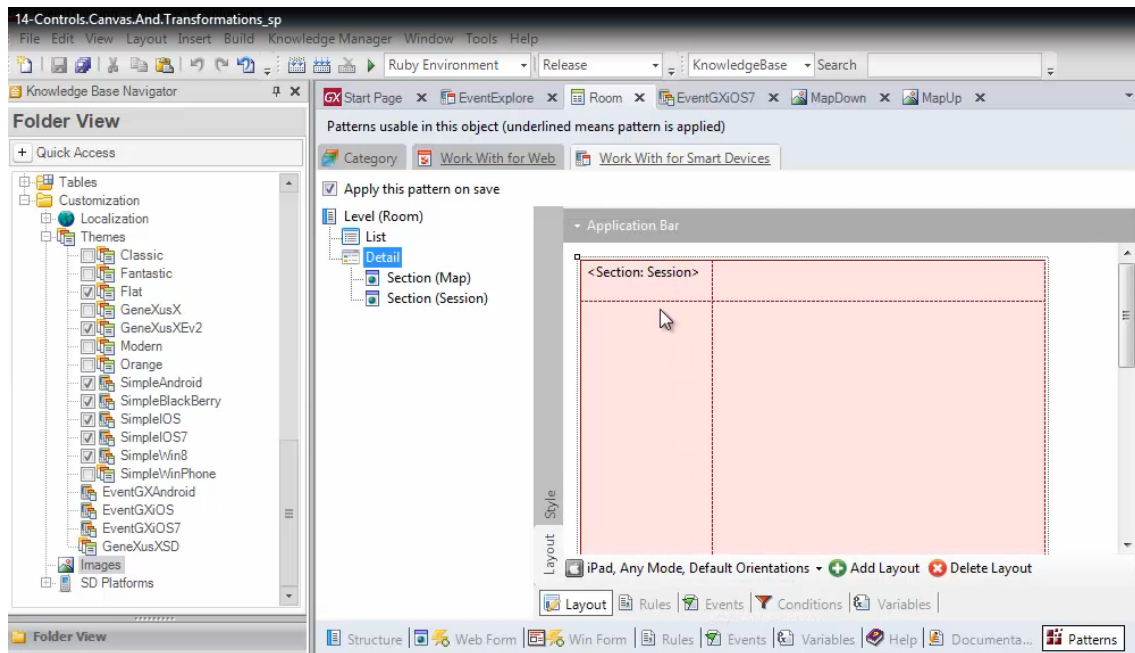


Aquí la diferencia del layout anterior

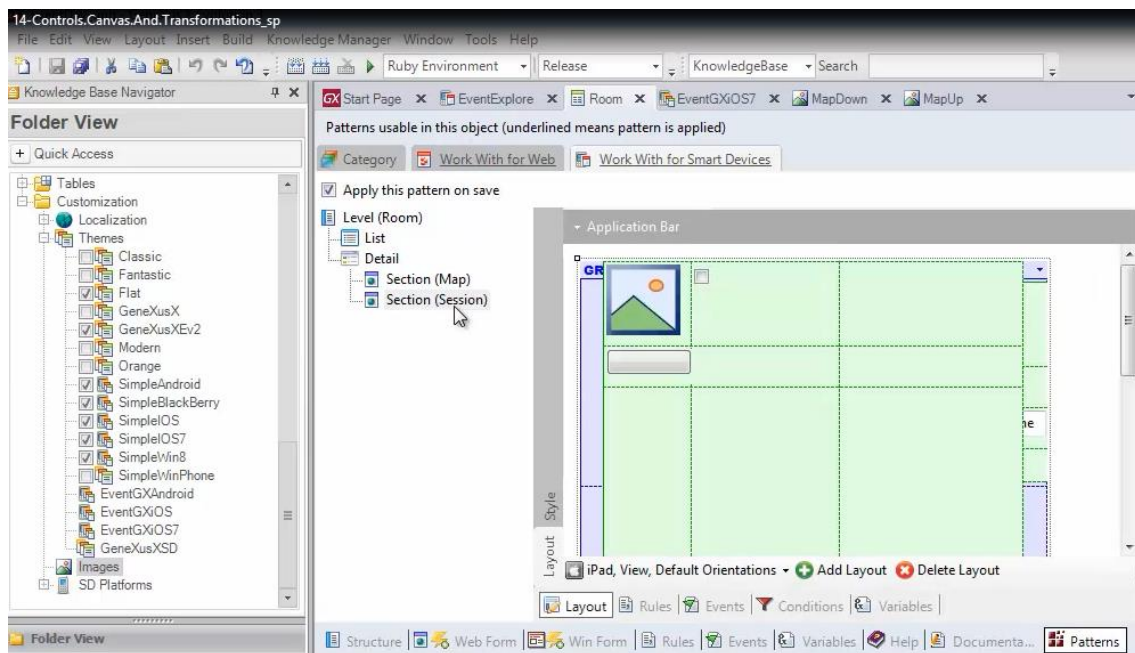


que tenía <All Sections Content>

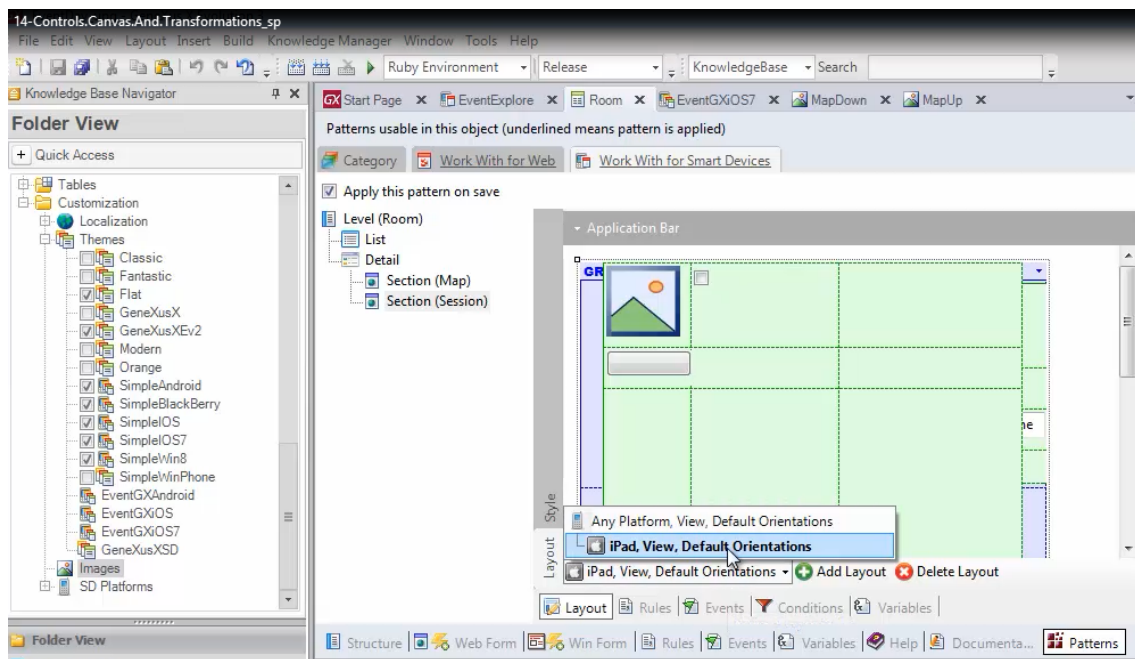




Vemos que para iPad tenemos solamente la sección: Session cargada... y si vamos a ver la sección Session:

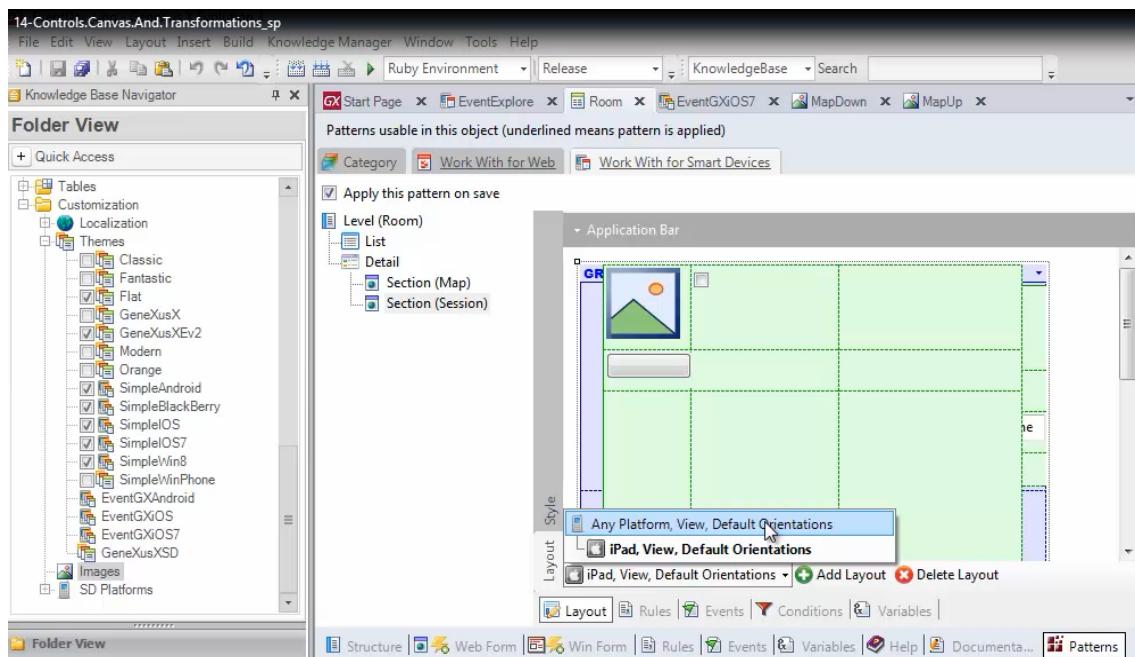


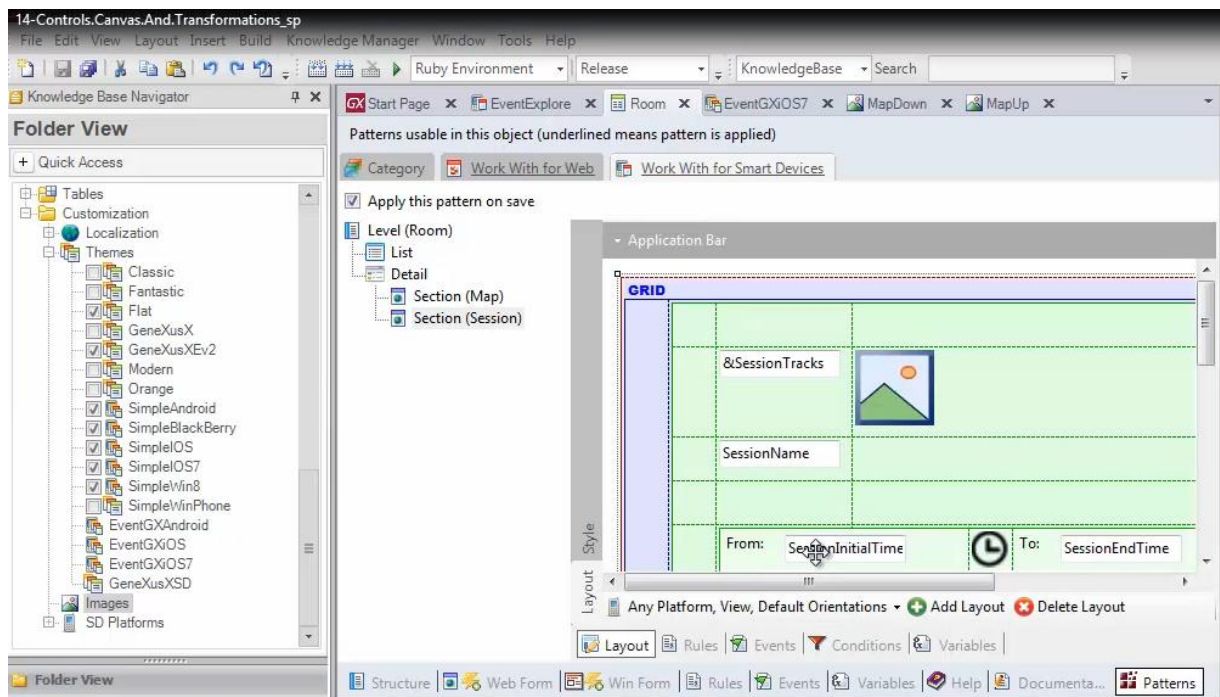
encontramos que para este layout:



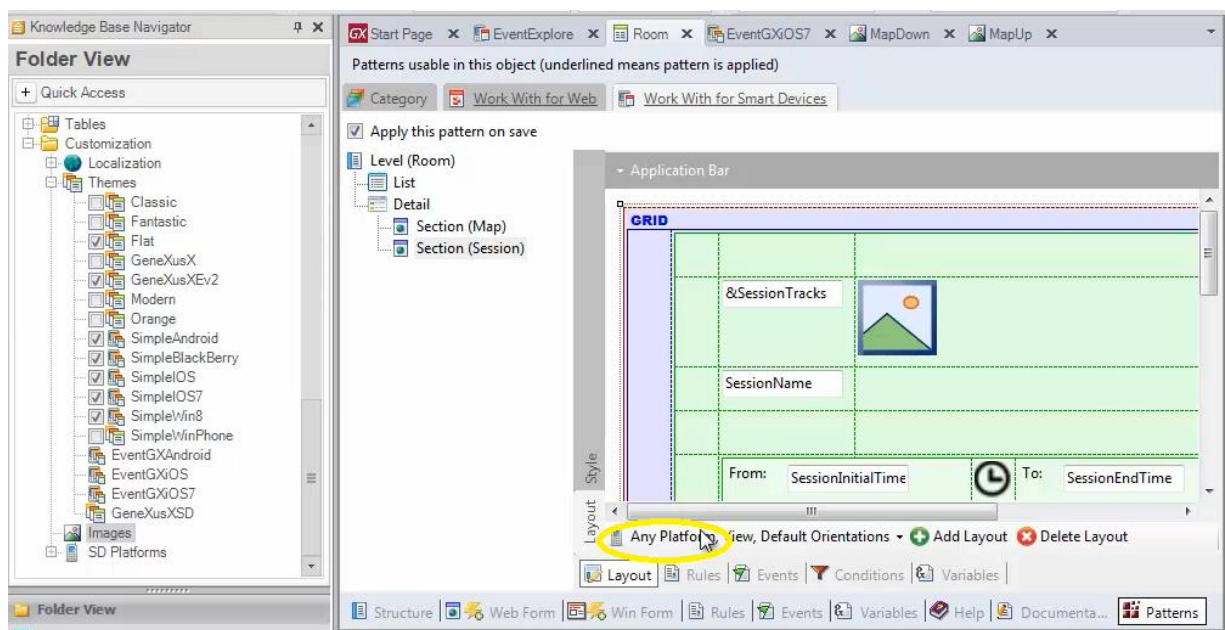
para iPad, en modo View, es que tenemos programado el comportamiento anterior.

Si editamos el layout para Any Platform

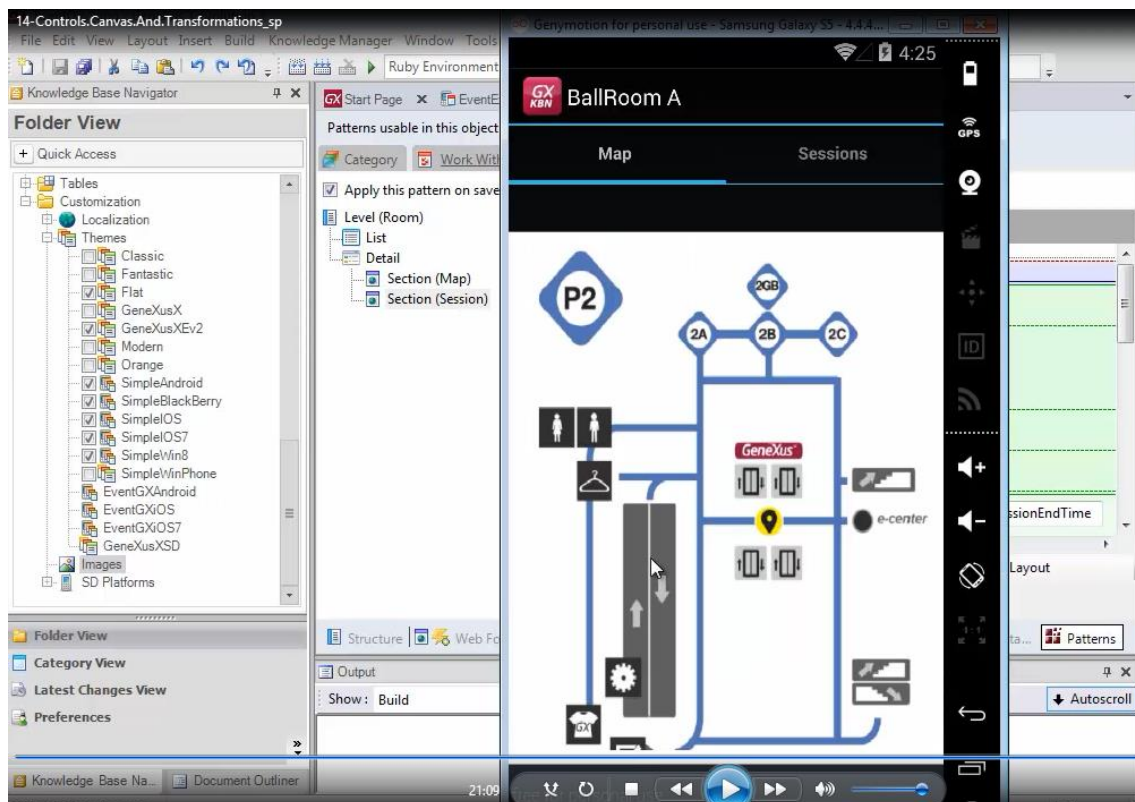




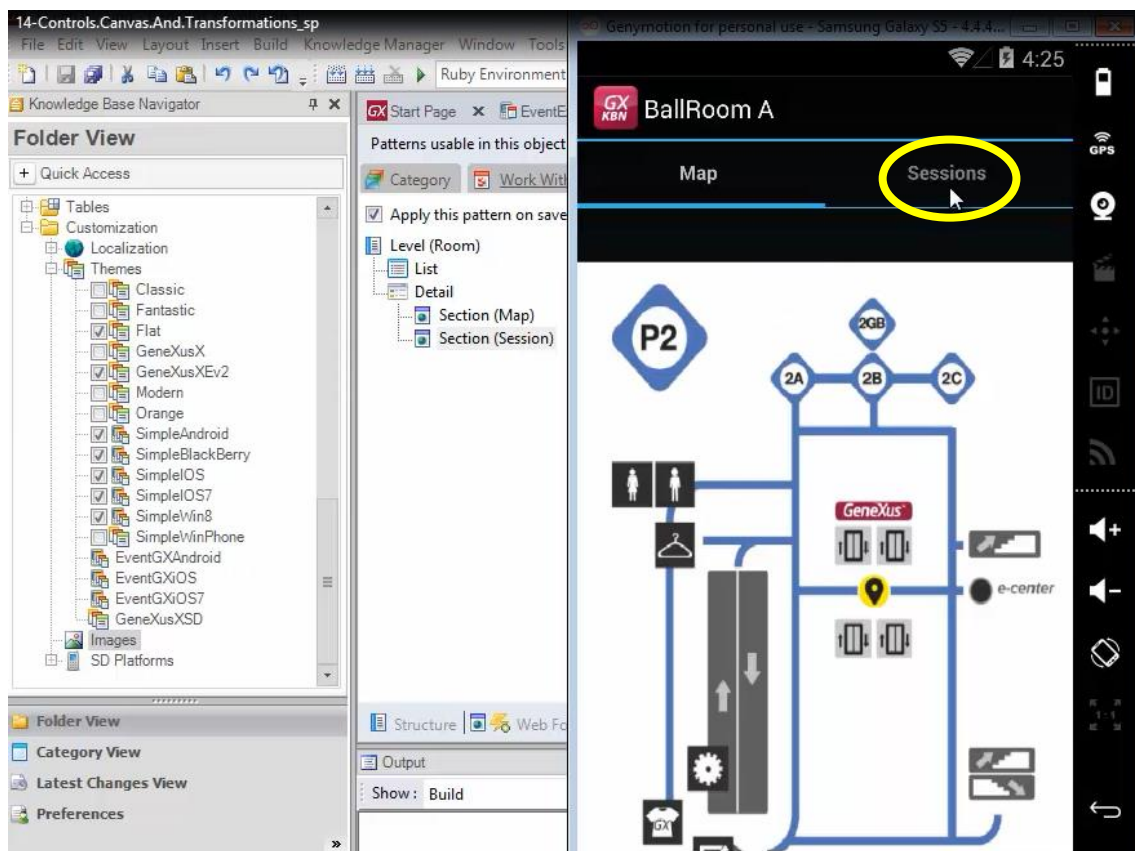
veremos que corresponde únicamente al grid, por tanto para cualquier plataforma que no sea iPad





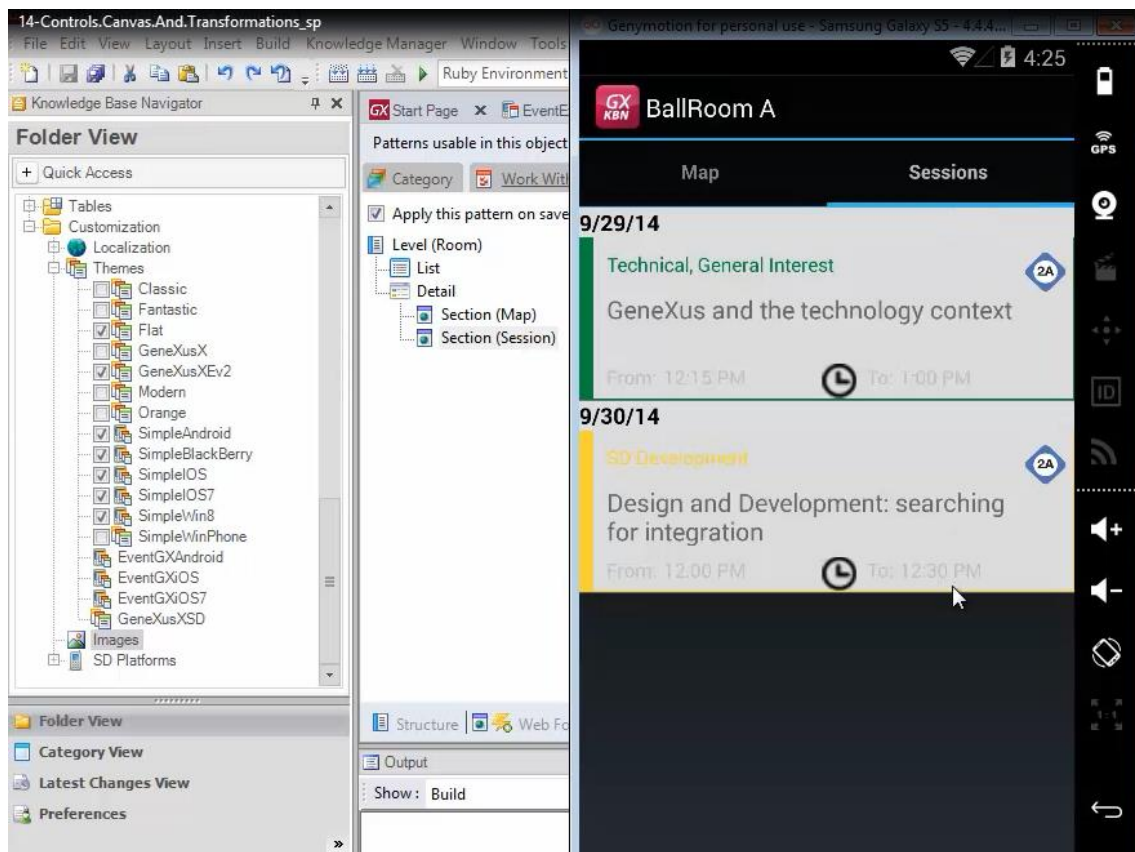


El Detail

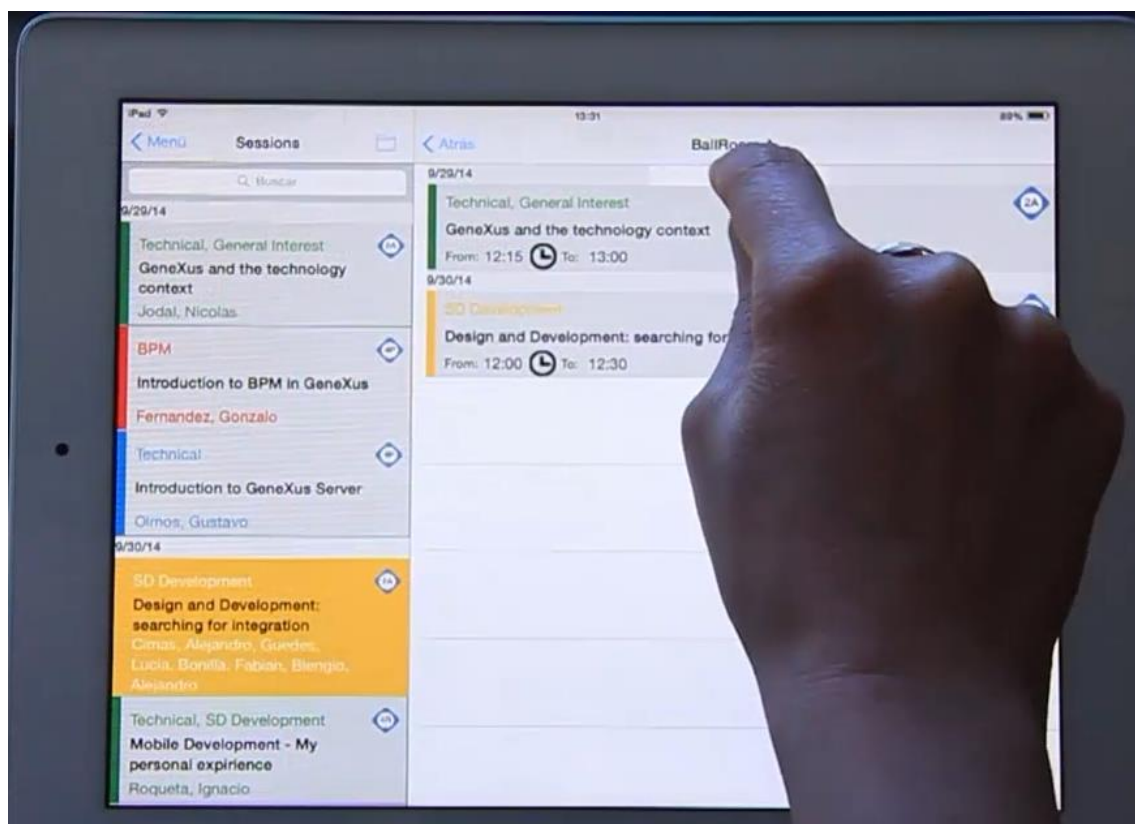


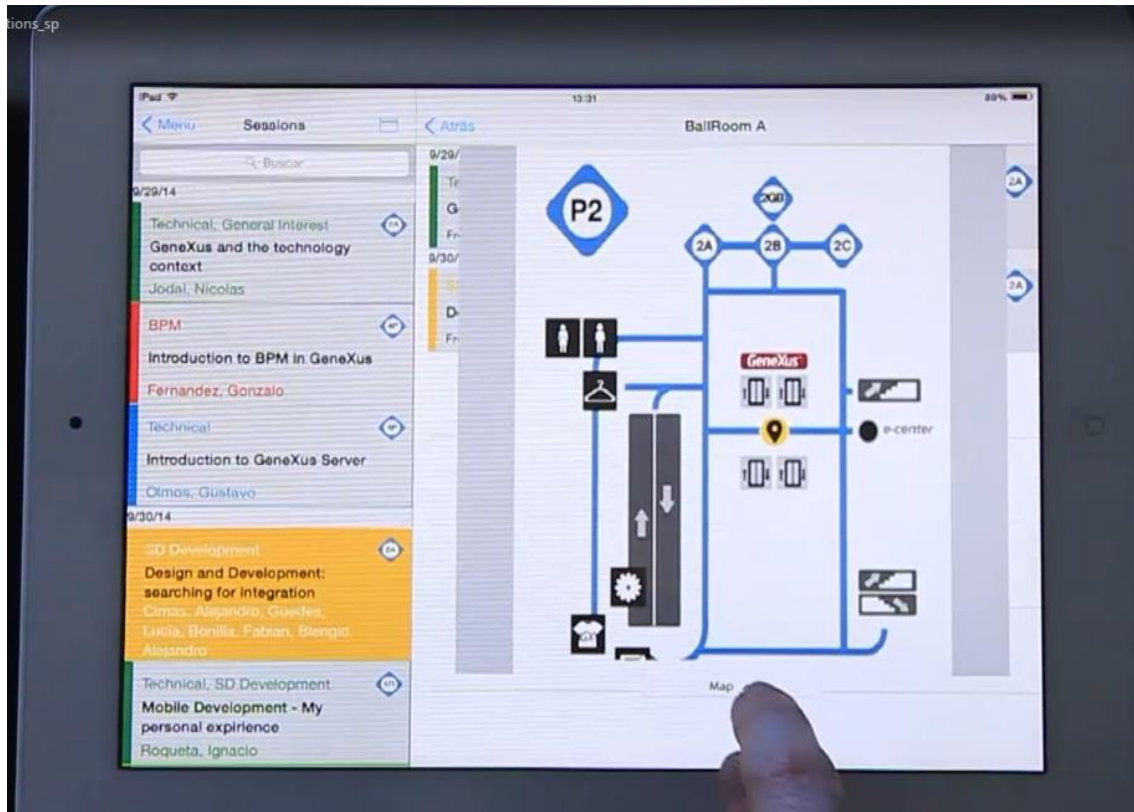
Se va a mostrar de esta manera





Y solamente para dispositivos iPad, se verá de la otra





Sobre este tema de los diferentes layouts por pantalla del objeto, se volverá en el siguiente video, si ud. Está haciendo el curso de Smart Devices.



Summarizing



#### Canvas

- Container providing absolute positioning

#### Transformations

- Classes that allow to change the shape, size and/or position of a layout element.
  - Translate
  - Scale
  - Resize
  - Rotate

Resumiendo: el control Canvas, es un contenedor de controles, que permite que todo lo contenido pueda sumir posicionamiento absoluto. Esto permite entre otras cosas, superponer controles.

Vea más información sobre este control, en nuestro wiki

**Controls in layouts****Summarizing****Canvas & Transformations****Canvas**

- Container providing absolute positioning

<http://wiki.genexus.com/commwiki/servlet/hwikibypageid?22452>

**Transformations**

- Classes that allow to change the shape, size and/or position of a layout element.
  - Translate
  - Scale
  - Resize
  - Rotate

Las transformaciones permiten que un control del layout, pueda ser trasladado, escalado, redimensionado o incluso rotado. Para ello, se definen clases especiales, bajo el nodo Transformations del theme.

Vea más información sobre este tema, así como en las plataformas en las que está implementada esta funcionalidad, en nuestro wiki

**Controls in layouts****Summarizing****Canvas & Transformations****Canvas**

- Container providing absolute positioning

**Transformations**

- Classes that allow to change the shape, size and/or position of a layout element.
  - Translate
  - Scale
  - Resize
  - Rotate

<http://wiki.genexus.com/commwiki/servlet/hwikibypageid?23694>

