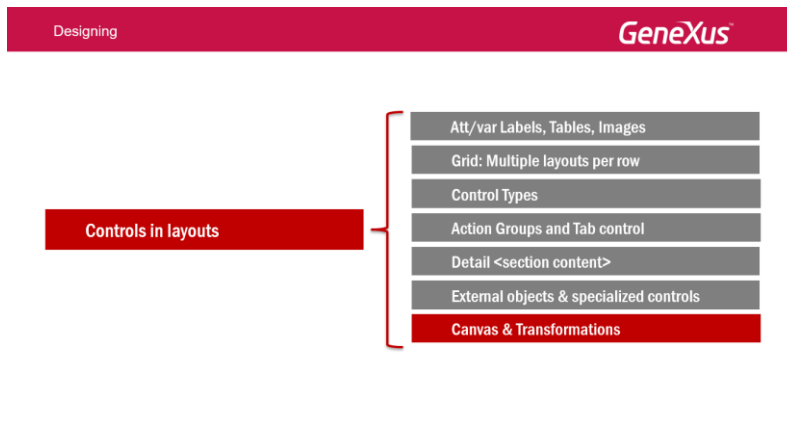


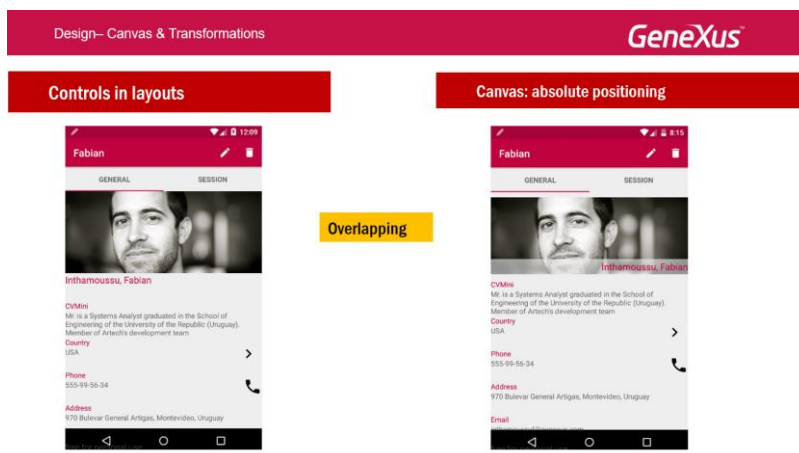
## Canvas and Transformations



Ahora pasemos al siguiente punto: cómo definir posicionamiento absoluto y cómo transformar un control moviéndolo, modificándole el tamaño, rotándolo, etc.

Así, supongamos por ejemplo, que deseamos que al visualizar la información de un orador el nombre completo se visualice **sobre** la foto, y no debajo como vemos en la pantalla de la izquierda.

Es decir, vamos a necesitar superponer el control atributo SpeakerFullName al control imagen SpeakerImage.

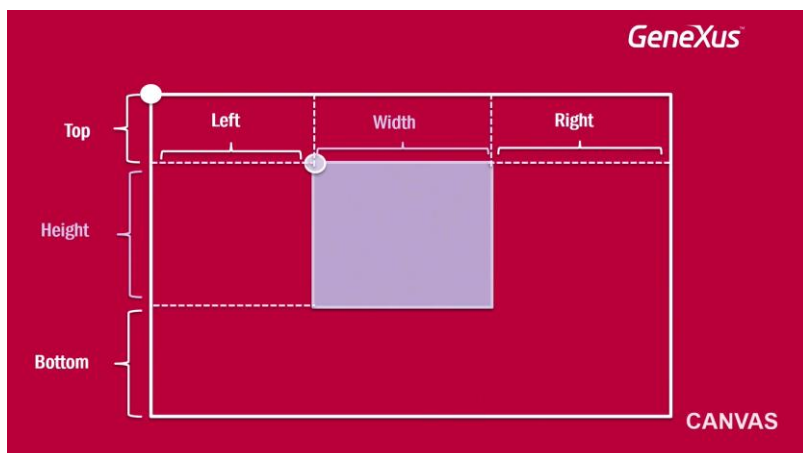


Para ello, necesitaremos definir un contenedor de controles con posicionamiento absoluto. Este contenedor es, en todo, como una tabla. Pero recibe un nombre especial: Canvas.

Teniendo un control Canvas en el layout, todos los controles que insertemos dentro, tendrán posicionamiento absoluto con respecto a los bordes del Canvas.

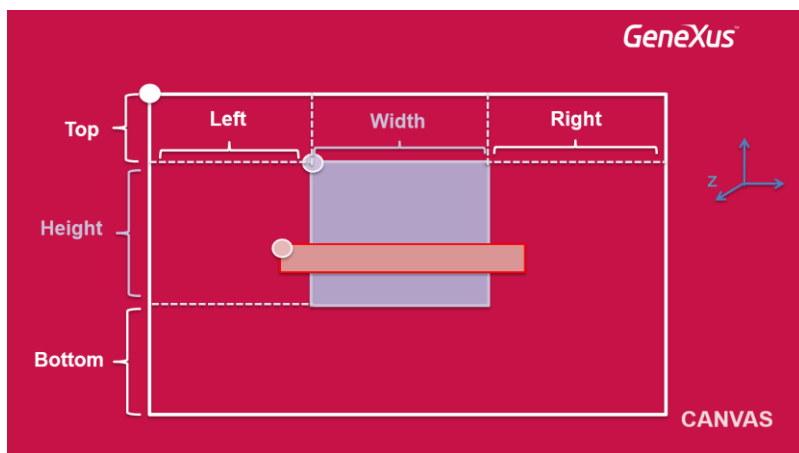
Así, si insertamos este control, tendremos que indicarle la distancia del borde superior (Top), el alto del control (Height), y la distancia del borde inferior (Bottom). Aunque ajustando dos de estas medidas, la tercera es automática, teniendo el Canvas un alto definido. Y por otro lado

tenemos que proporcionar la distancia de la izquierda (Left), el ancho del control (Width) y la distancia del borde derecho (Right).

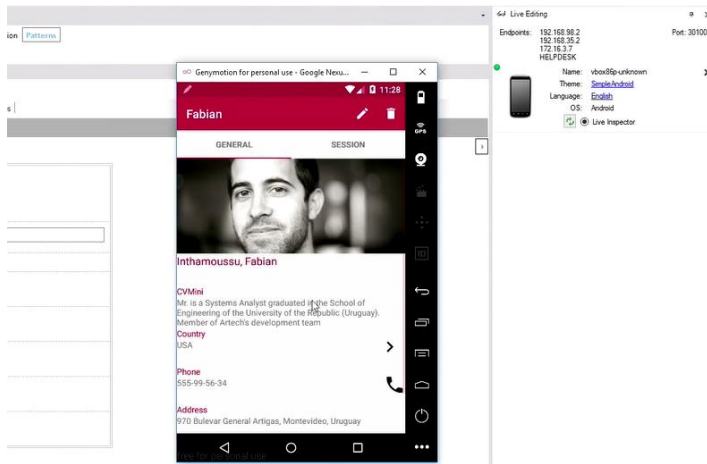


Asignando estas medidas de manera conveniente es que vamos a poder superponer los controles. Necesitaremos, por tanto, asignar también un valor para el eje Z, que va a determinar la capa de cada control, para saber cuáles estarán encima de cuáles. Esta propiedad, Z-order se va a llamar, va a asumir valores discretos. 0 va a ser la capa de más abajo, 1 la siguiente, y así.

Si estamos en plataforma Android, esto es un caso particular, y hay controles dentro del Canvas con propiedad elevation y también Z-order obviamente por estar adentro de un Canvas, la diferencia, en cuanto a la profundidad, entre dos controles con el mismo Z-order va a venir dada por su propiedad elevation. Así que ahí tenemos una manera de distinguir.

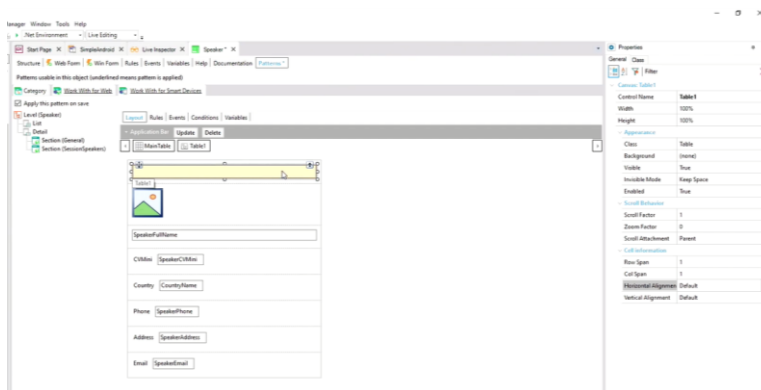


Bueno, entonces vamos a ir a GeneXus a hacer esto que mostrábamos en la slide anterior. Voy también a mostrar cómo se presenta ahora, antes de hacer nada, el detail de un orador... y vemos que el Full Name está apareciendo debajo de la foto.

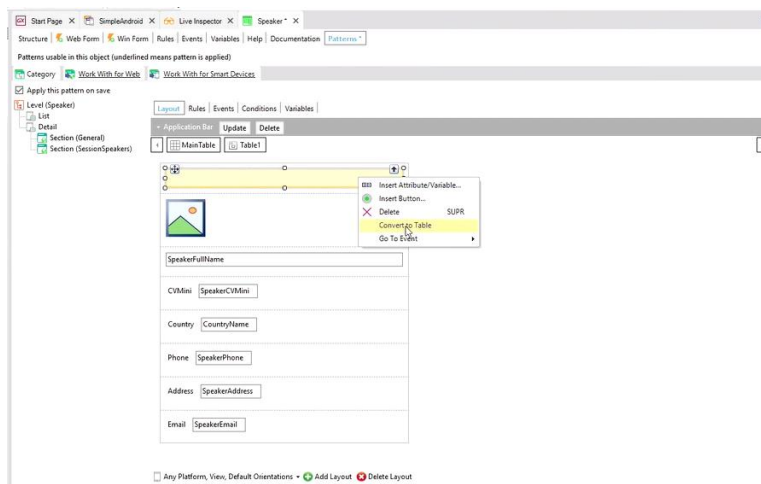


Acá estamos viendo el Layout... la foto va a ocupar una fila de (si vamos a las propiedades) 160 dips.

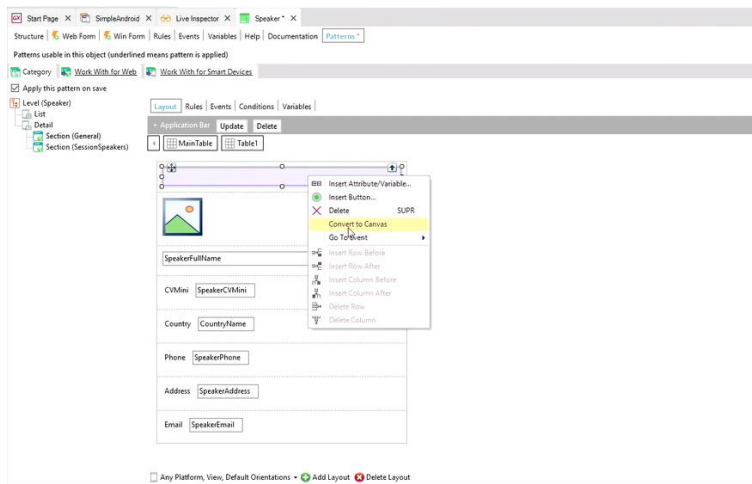
Bien, entonces lo que vamos a hacer es insertar un control Canvas. Voy a mover esto para abajo primero, para insertarlo en esa fila de 160 dips. Y voy a la Toolbox, y arrastro entonces el control Canvas dentro de esa fila. Si vamos a las propiedades, y editamos el control, vemos que dice Canvas, pero también dice como nombre del control Table 1; porque en todo va a ser como una tabla salvo en esta característica de permitir posicionamiento absoluto.



Tanto es así que podemos convertirlo a tabla...



Ahora dice tabla... o convertir la tabla a Canvas.

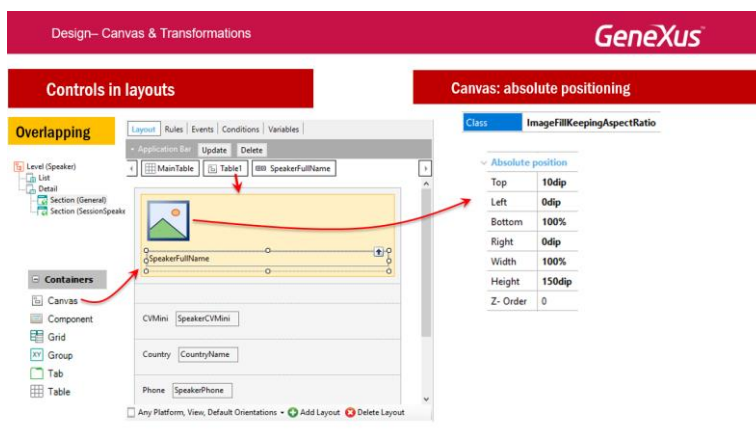


Bien. Entonces ahora vamos a hacer lo siguiente. Vamos a arrastrar primero que nada la imagen dentro de esa tabla Canvas, y vamos a observar que al haber hecho esto, si editamos las propiedades de la imagen, vemos que aparece este grupo, Absolute Position, con todas las propiedades que veíamos antes: Top, Left, Bottom, Right, Width, Height, y Z-order. Entonces, especificando estas propiedades es que vamos a poder decir dónde se va a ubicar, dónde va a comenzar esta imagen, es decir en qué posición relativa a los bordes del Canvas.

Entonces, por ejemplo, como nosotros vamos a querer que la imagen ocupe todo el ancho, como veíamos en el emulador... estamos con Live Editing, por eso este cambio se está viendo de esta manera, todavía no lo terminamos. Así que el ancho va a ser el completo de la fila. De arriba del Canvas, vamos a querer que esté a 10 dips. De la izquierda, a 0 dips, porque vamos a querer que ocupe todo el ancho, como decíamos; y lo mismo de la derecha, 0 dips. Así que el ancho vamos a dejar el 100% del Width, para que se expanda en toda la fila.

Ahora, el alto vamos a querer que sea de 150 dips, porque entonces  $150 + 10$  va a dar 160 que era el alto de la fila entera. Y de esa manera entonces, teniendo ese Height, vemos que Bottom quedó con 100%, pero el 100% de 0 va a ser 0. Es decir que va a estar sobre el fondo del Canvas.

Bien. Z-order vamos a dejarle en 0 porque vamos a querer que la imagen sea lo que esté más atrás.

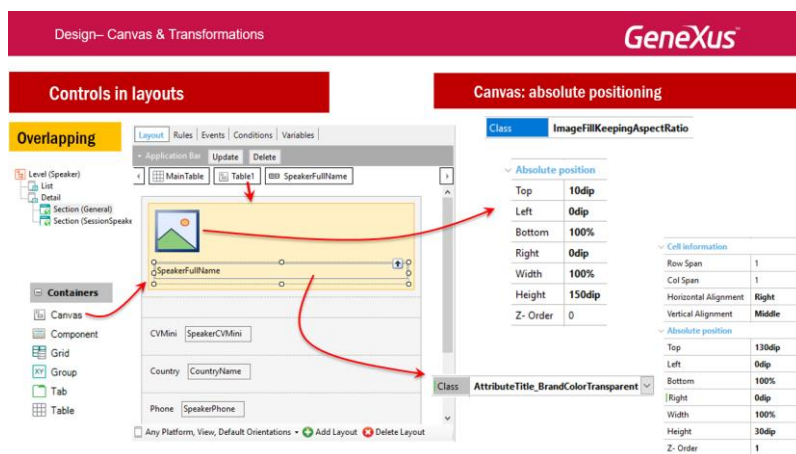


Tenemos que arrastrar entonces ahora el atributo SpeakerFullName, que como vemos, no tiene por ahora, como está afuera del Canvas, el grupo Absolute Position. Ahora, cuando lo arrastremos dentro del Canvas, vemos que sí está apareciendo ese grupo. Y vamos a tener que otra vez especificar la posición absoluta que va a ocupar ese control.

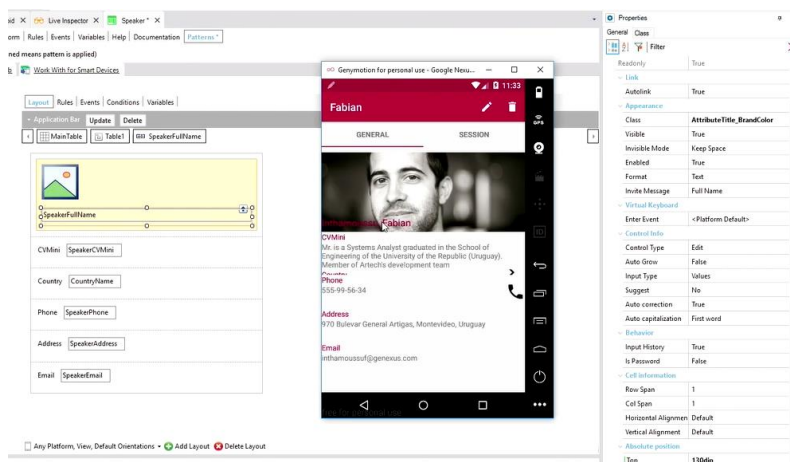
Vamos a especificar que el control va a tener un alto de 30 dips. Entonces, ya venimos acá abajo y ponemos 30 dips; ese va a ser el alto de este control. Bien. Pero al ser 30 y vamos a querer que esté recostado sobre el fondo del canvas, entonces de la posición Top cuánto va a ocupar, ¿a cuánto va a estar del Top? A 130, porque  $130 + 30$  me da 160 que es el alto de la fila. Entonces aquí voy a poner 130.

Luego, de la izquierda va a estar a 0 dips porque vamos a querer que también ocupe exactamente el mismo ancho que la imagen. Así que 0 de la izquierda y 0 de la derecha. 100% va a ser el ancho, y otra vez Bottom me está quedando el 100%, pero en Bottom, el 100% de 0 (porque  $130 + 30$  ya me da 160, por tanto le va a quedar 0 dips para abajo) va a ser 0.

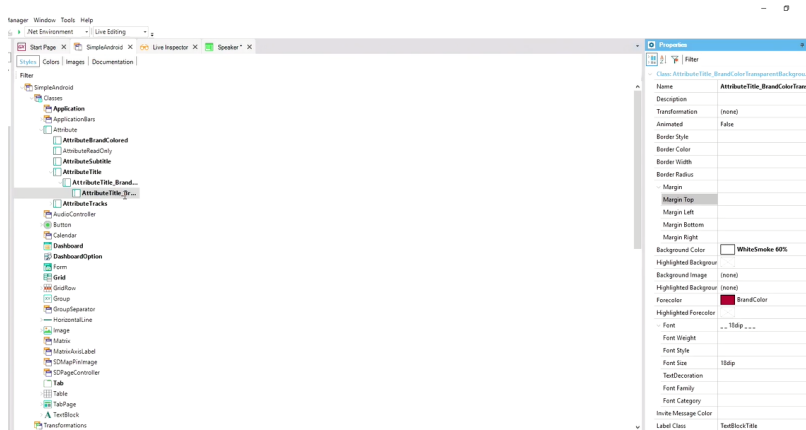
Y acá lo importante es que el Z-order va a tener que ser mayor que el de la imagen, porque queremos que el atributo esté por encima, así que le voy a poner 1.



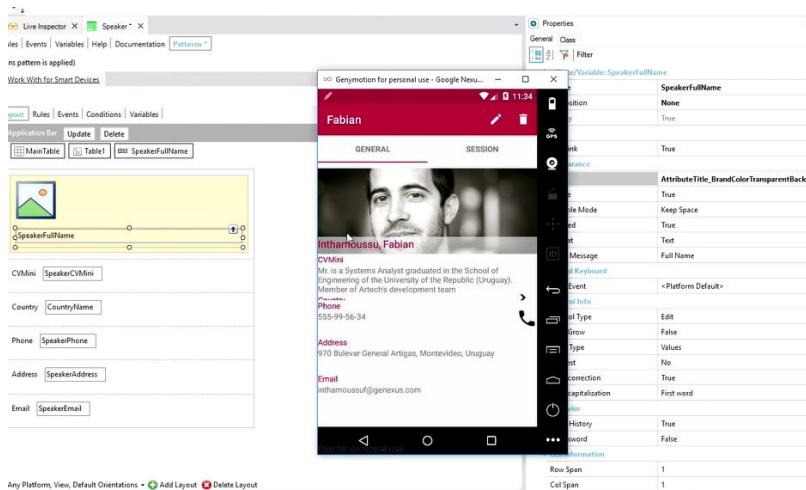
Si abro el emulador, vemos como efectivamente está apareciendo como queríamos.



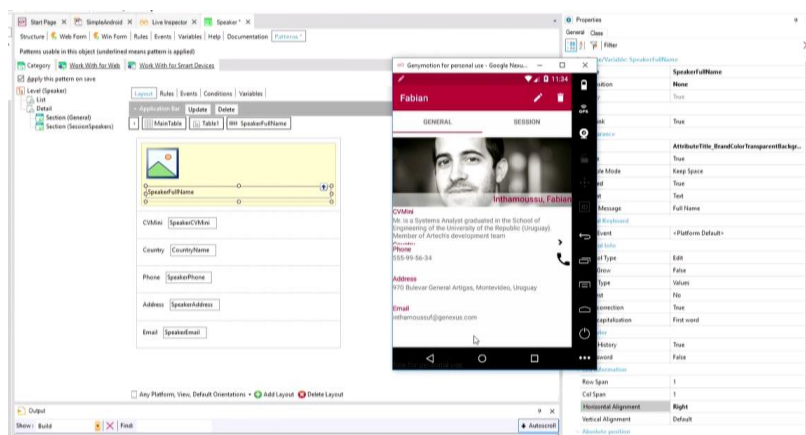
Le puedo cambiar también la clase al atributo, hemos creado una nueva clase en el tema, bajo esta otra...una clase que hemos llamado de esta manera... a la que le hemos configurado como Forecolor Brand Color, pero además le pusimos un background color que tiene transparencia...



...de manera tal de que... vamos a ver ahora lo que sucede, cuando le cambio entonces la clase al control por esa nueva... vamos a mostrar que sale entonces con ese blanco pero difuminado, se ve un poco del fondo.



Lo otro que podemos hacer es decirle que este atributo se alinee, por ejemplo, a la derecha.

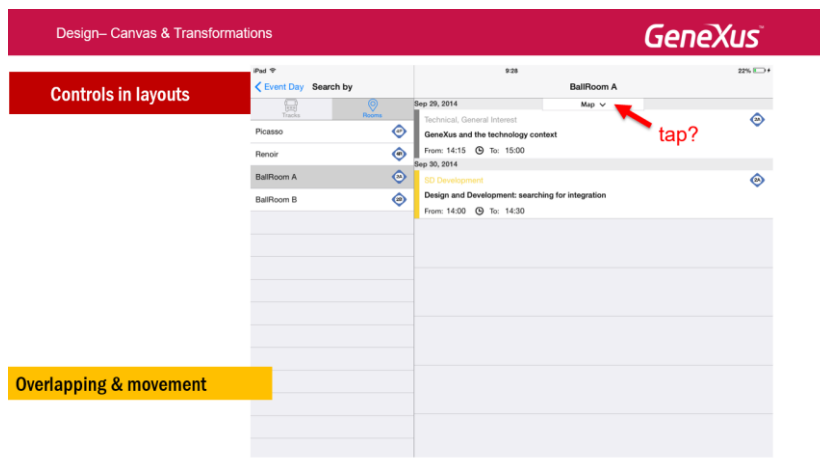




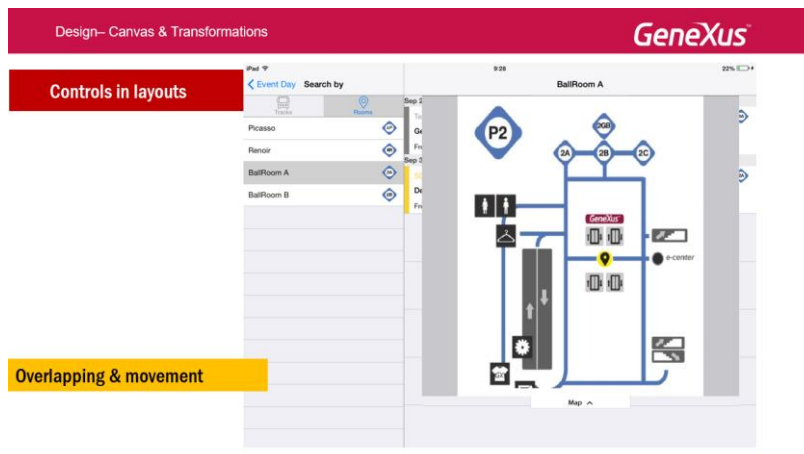
Esto era en un iPad, desde el menú invocábamos al objeto EventExplore (un Work With for Smart Devices creado en forma manual, del que solo se utiliza su nodo Detail, con las dos secciones: Tracks y Rooms, que muestran las listas de tracks y las salas, respectivamente). Si desde la lista de salas se elige una –en el ejemplo que estamos viendo, BallRoom A–, se invoca al Detail del WorkWithDevicesRoom para ver los datos de esa sala elegida.

Si tenemos esta aplicación instalada entonces como decíamos en un iPad, vamos a ver que se nos despliega la lista de conferencias agendadas en esa sala. Pero además, vemos arriba lo que parece un botón para expandir, ese botón Map que estamos viendo acá.

Si hacemos tap sobre ese botón, ¿qué es lo que va a pasar en ejecución?

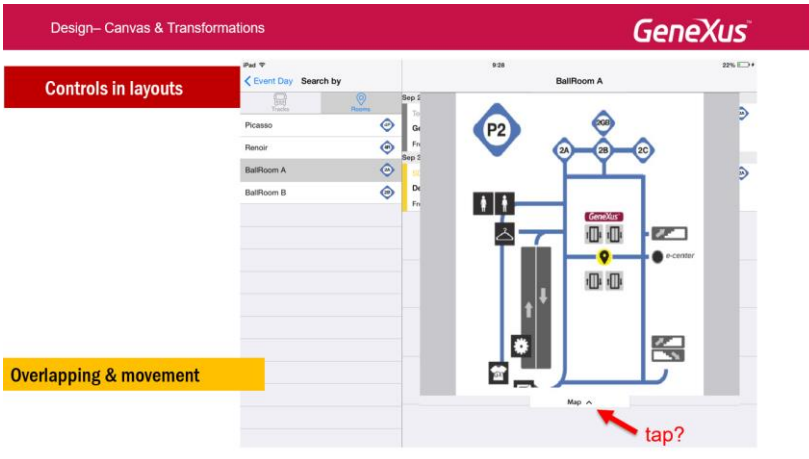


Vamos a ver que se despliega el mapa que muestra dónde se encuentra la sala en el plano del piso, del edificio donde se está desarrollando el evento.

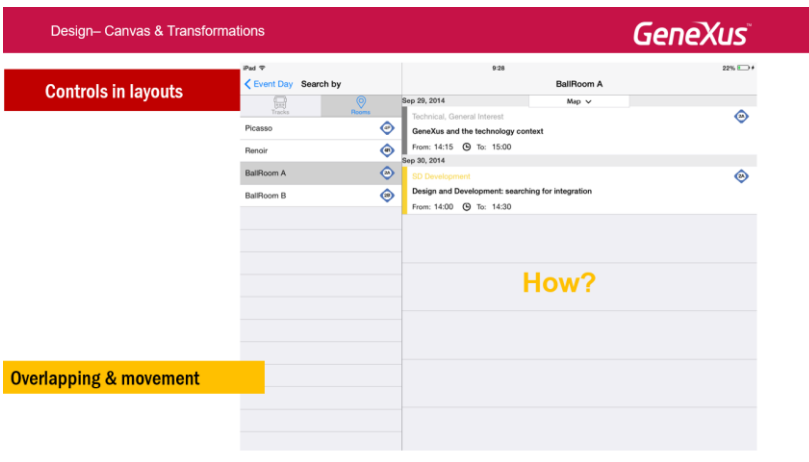


Si ahora volvemos a hacer tap en el botón, que vemos que cambió la orientación de la flechita... ¿qué es lo que sucede?



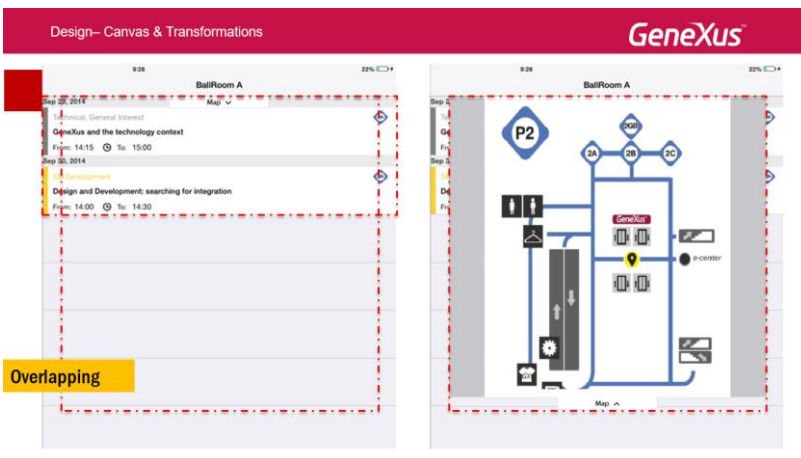


Lo que vamos a ver es que se colapsa el mapa, y estamos viendo solamente el botón.

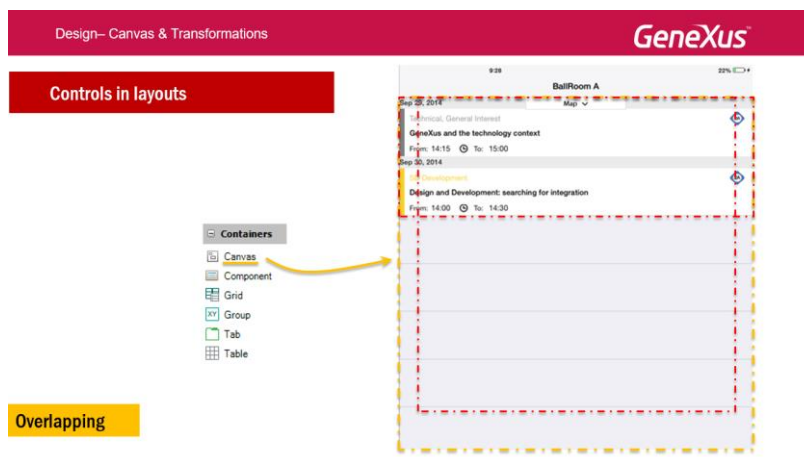


¿Cómo hacemos esto? ¿Cómo lo implementamos?

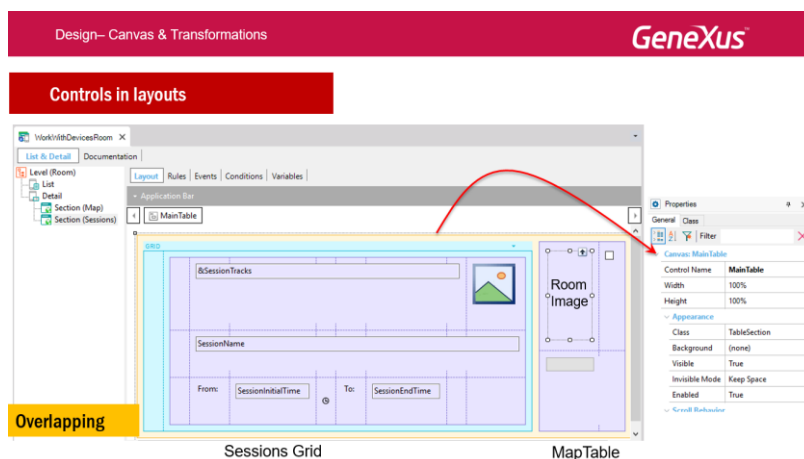
Bueno, la parte del solapamiento ya la sabemos, la acabamos de ver, y es con un control Canvas. En este layout tenemos, en una misma área de pantalla, la superposición de un grid, que es el que muestra las conferencias, con una tabla que contiene una imagen con el mapa y un botón; entonces está ese solapamiento.



Entonces por lo tanto lo que vamos a tener que hacer para implementar esto es insertar una tabla Canvas...

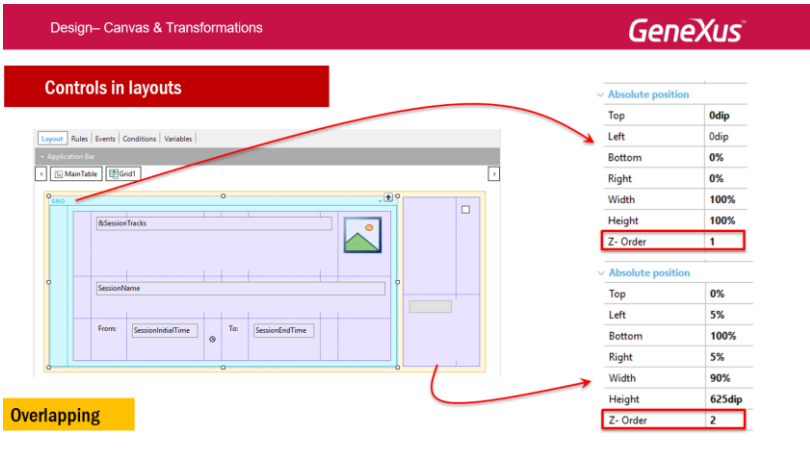


...dentro de la cuál ubicaremos al grid y a la tabla que contiene el mapa y el botón. A esa tabla que contiene el mapa y el botón le vamos a llamar MapTable, ahora vamos a ver para qué la necesitamos.

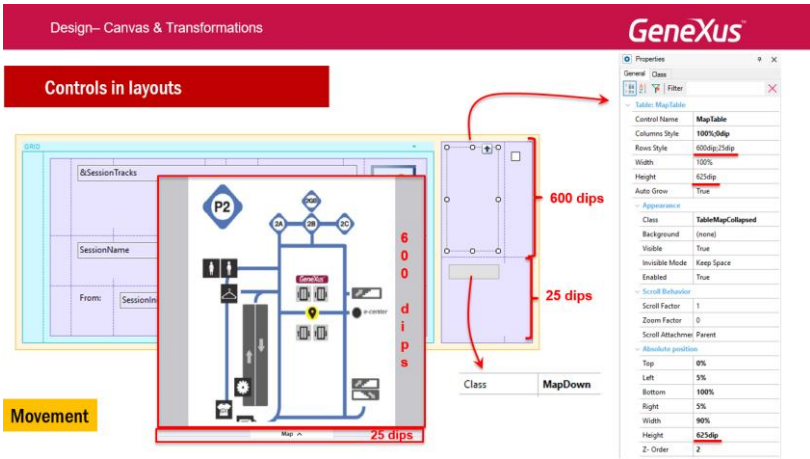


En diseño, no vemos los controles solapados, no es un editor WYSIWYG, justamente para poder trabajar con cada uno en diseño. Pero bueno, como vimos en ejecución, sí aparece como tiene que aparecer, y tenemos Live Editing entonces para ir probando, así que no es necesario tenerlos superpuestos aquí.

Bien. Entonces, además de brindar las posiciones absolutas de grid y tabla, con las propiedades del grupo Absolute Position, es con la propiedad Z-Order que indicamos que el grid estará abajo de la tabla, como vemos aquí. Le pusimos Z-order = 1 al grid, y Z-order = 2 a la tabla.

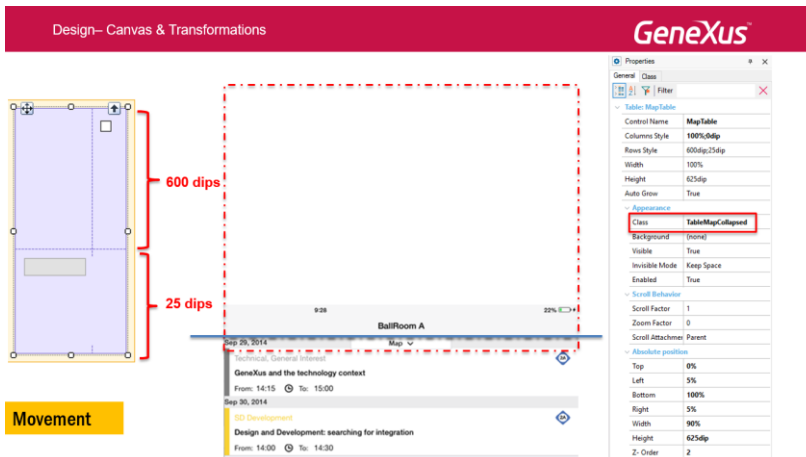


La tabla superpuesta tiene dos filas: una de 600 dips, donde se encuentra el atributo RoomMap que contiene la imagen del mapa de la sala y... otra de 25 dips, donde se encuentra el botón, que tiene asignada una clase del theme, MapDown, que es la que tiene configurada la imagen que se muestra en ejecución.



¿Por qué vemos solamente el botón de expand (Map) cuando invocamos al objeto por primera vez, y lo vemos arriba, como si toda la tabla se hubiese movido 600 dips verticalmente hacia arriba, dejando afuera del espacio del layout a la imagen? En esta línea azul estamos indicando la posición Top del Canvas.

Este comportamiento que acabamos de nombrar lo tiene especificado la clase que le hemos asignado a la tabla: TableMapCollapsed (clase que creamos nosotros, los desarrolladores, dentro del Theme para iPad).

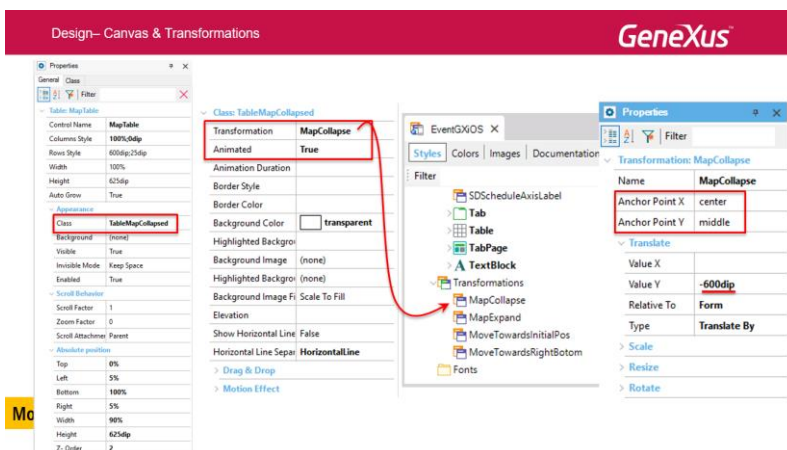


Entonces vemos que esta clase tiene asignada una “transformación”. Esa “transformación” es la que la va a definir el desarrollador dentro del mismo Theme, y es donde se especifica el comportamiento que vemos en ejecución, de **traslación**.

El control que tenga asociada la clase que tiene esta transformación, se va a mover **desde** (Translate by, lo vemos aquí abajo) el punto X, Y indicado (Anchor Point X, Anchor Point Y), en el eje Y, 600 dips hacia arriba (Value Y = -600dip bajo el grupo Translate), tomando como origen de coordenadas (Relative to) el Form (el punto (0, 0) del form equivale al vértice superior izquierdo).

En nuestro caso, la tabla se moverá hacia arriba, 600 dips. Como está al borde de la pantalla, ese movimiento va a dejar afuera a la imagen, como veíamos en la slide anterior. Eso es lo que hace que desaparezca la imagen del mapa de la sala.

Observemos que la clase de la tabla tiene prendida la propiedad Animated, como vemos aquí, en True, y esto es lo que va a hacer que el desplazamiento se vea como tal cuando la tabla esté extendida y se colapse.



Como vemos aquí, es decir cuando está extendida, se hace Tap sobre el botón, y vamos a ver cómo se desplaza de esa manera de forma animada.

Entonces, ¿cómo se programa todo esto?

Cuando la tabla está trasladada hacia arriba (es decir colapsada, lo que equivale a decir **no expandida**), 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).

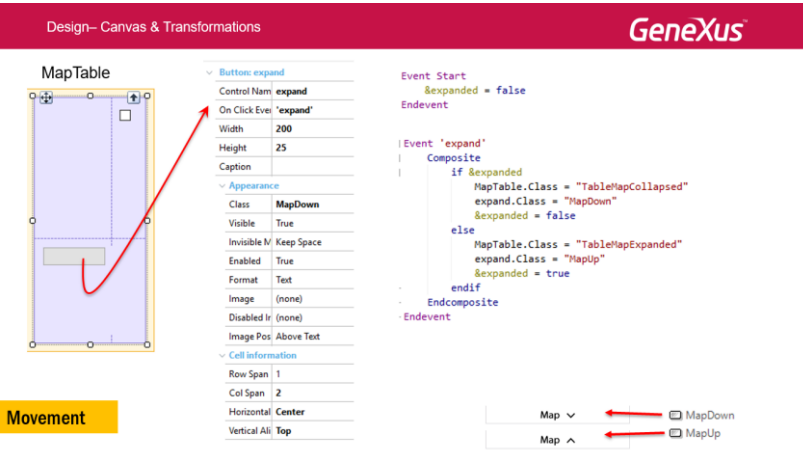
De la misma manera, el botón debe cambiar su imagen en un caso y el otro. El botón, tiene configurado en diseño la clase MapDown del theme, que podemos ver que es la que muestra la imagen de esta manera, pero existe otra clase, MapUp, que tiene configurada la otra imagen.

Entonces será en el evento “expand”, asociado al botón, que llamamos también expand, donde se va a programar entonces 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 de que se trate.

Por ejemplo, cuando la tabla está expandida y se presiona el botón, se va a entrar en este if, y lo que vemos es que se cambia la clase de la tabla que contiene a la imagen de la sala y al botón por TableMapCollapsed, que es la que tiene la transformación de traslación que vimos antes. Y allí es donde se va a dar el efecto buscado, es decir que va a parecer que se traslada hacia arriba, que sube y desaparece la imagen de la sala.

También observemos que se cambia la clase del botón para que salga el que tiene la flechita hacia abajo. Por otro lado, se va a utilizar una variable invisible en la tabla como switch, para prender y apagar, para indicar cuando está colapsada y cuando está expandida.

Para dar el movimiento de expansión de la tabla, que es el inverso, el que veníamos viendo, también vamos a tener que definir una transformación, en sentido inverso, y es análogo.



The screenshot displays the GeneXus IDE interface for a control named 'MapTable'. On the left, the design view shows a table with a 'Map' button. A red arrow points from the 'Map' button to the 'expand' event handler in the code view. The code view shows the following logic:

```
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
```

At the bottom, there are two buttons: 'Map' with a downward arrow (labeled 'MapDown') and 'Map' with an upward arrow (labeled 'MapUp').

En resumen, el control Canvas es un contenedor de controles, que permite que todo lo contenido pueda asumir posicionamiento absoluto, y de esa manera, eso nos va a permitir, entre otras cosas, superponer controles.

Por otro lado, las transformaciones permiten que un control del Layout pueda ser trasladado, escalado, redimensionado, o incluso rotado, en runtime. Para ello se definen clases especiales bajo el nodo Transformations del theme.

Design— Canvas & Transformations **GeneXus**

**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

---

Con esto terminamos con lo que queríamos ver de los controles en los Layouts. Pasemos al siguiente tema.

Designing **GeneXus**

**Controls in layouts**

- Att/ var Labels, Tables, Images
- Grid: Multiple layouts per row
- Control Types
- Action Groups and Tab control
- Detail <section content>
- External objects & specialized controls
- Canvas & Transformations

---