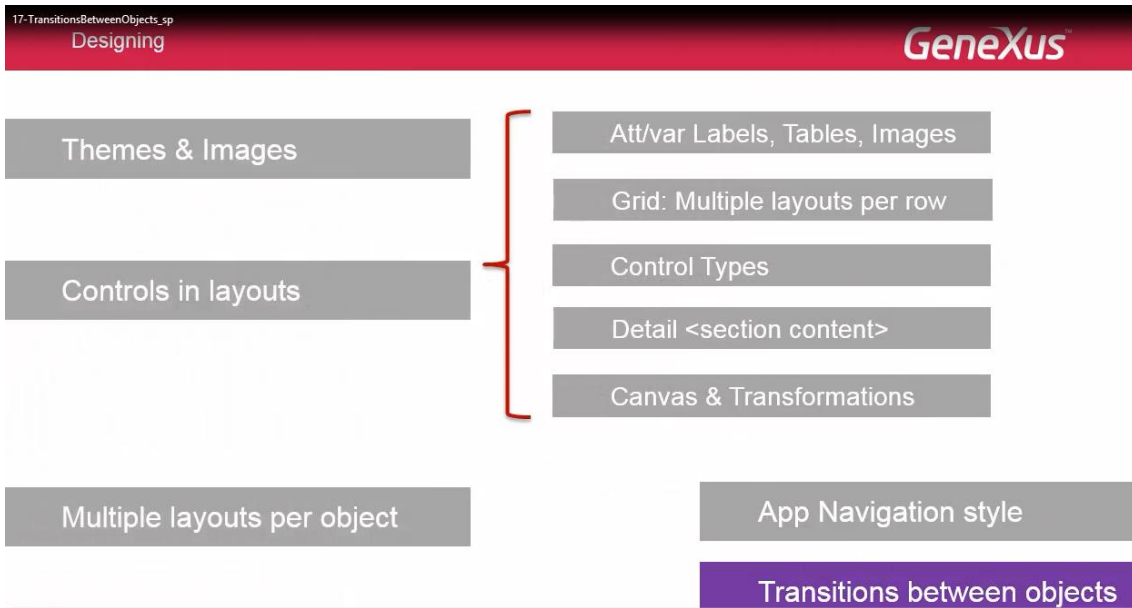


Designing: Transitions between objects



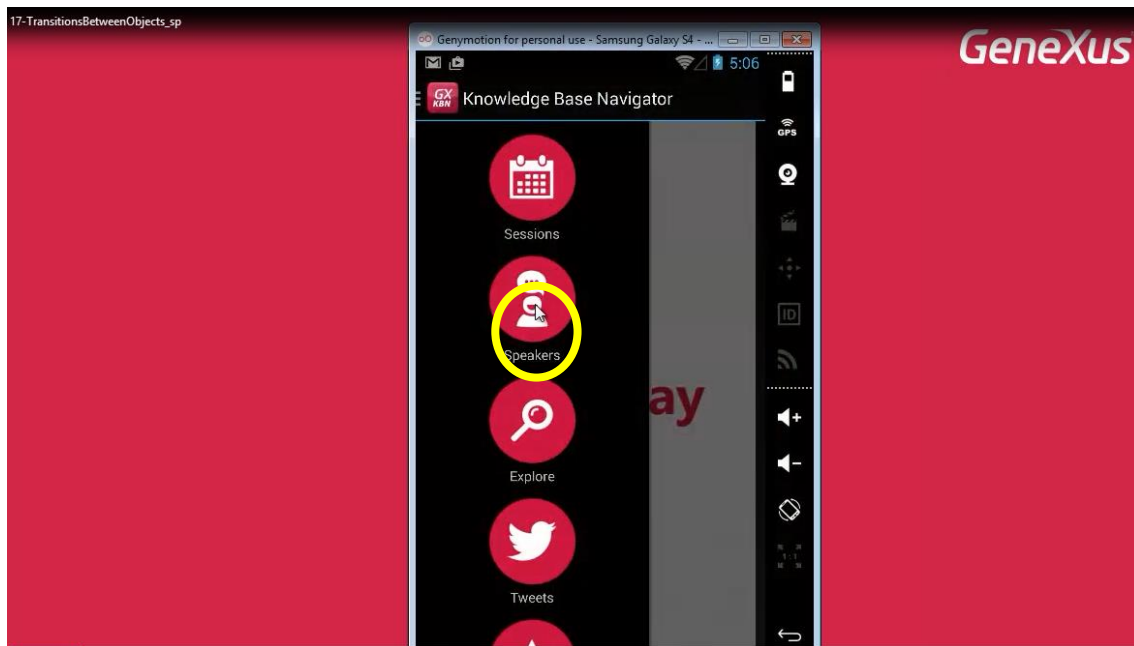
Designing: transitions between objects

Cecilia Fernández | GeneXus Training

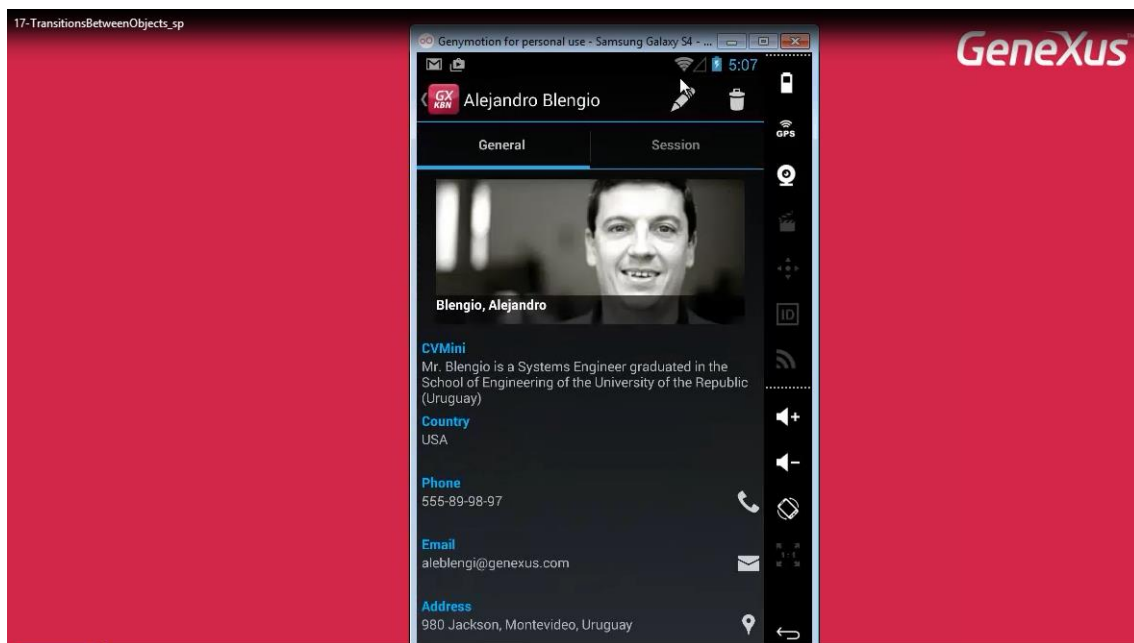
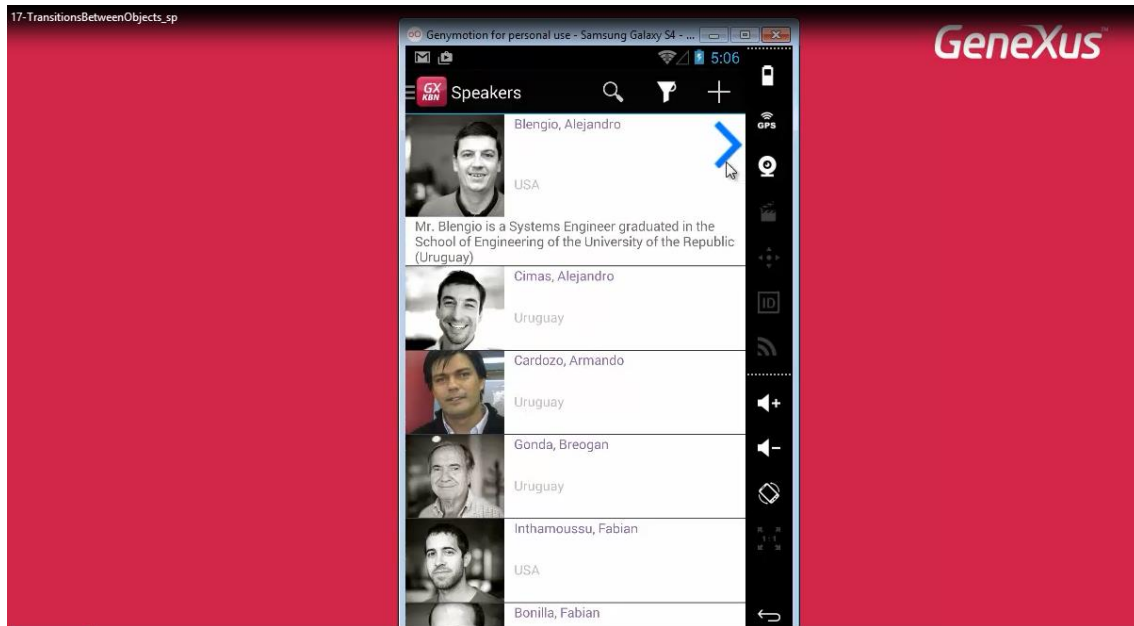


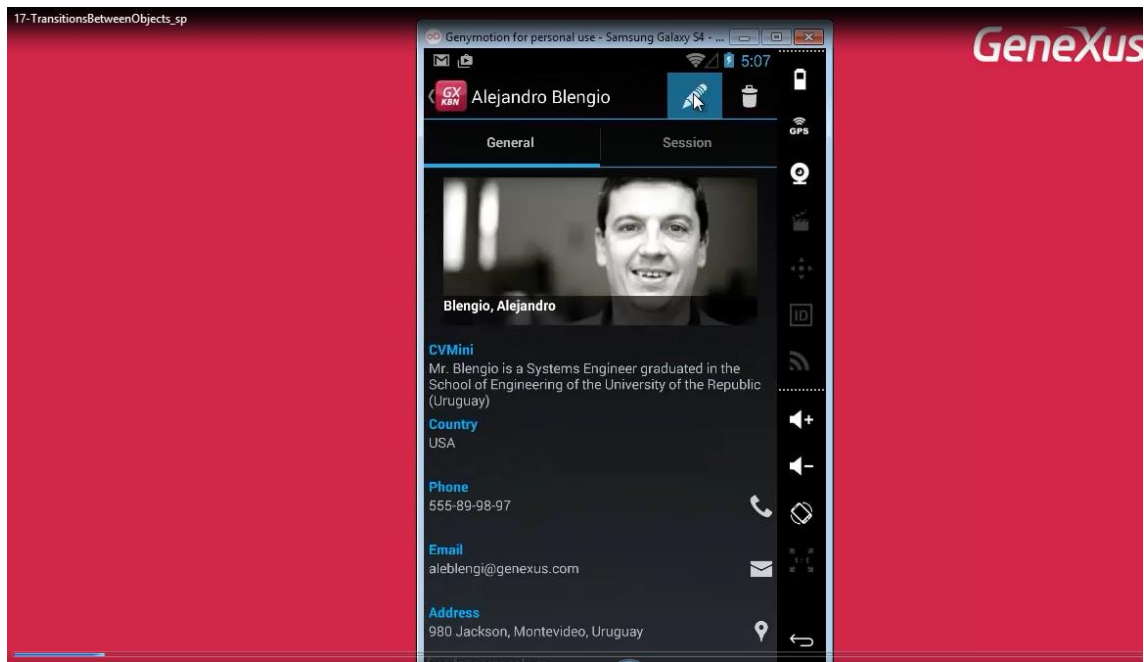
Nos queda por ver el último tema de diseño. Cómo lograr diferentes transiciones entre las pantallas de los objetos invocados.

Cada vez que desde una pantalla de la aplicación

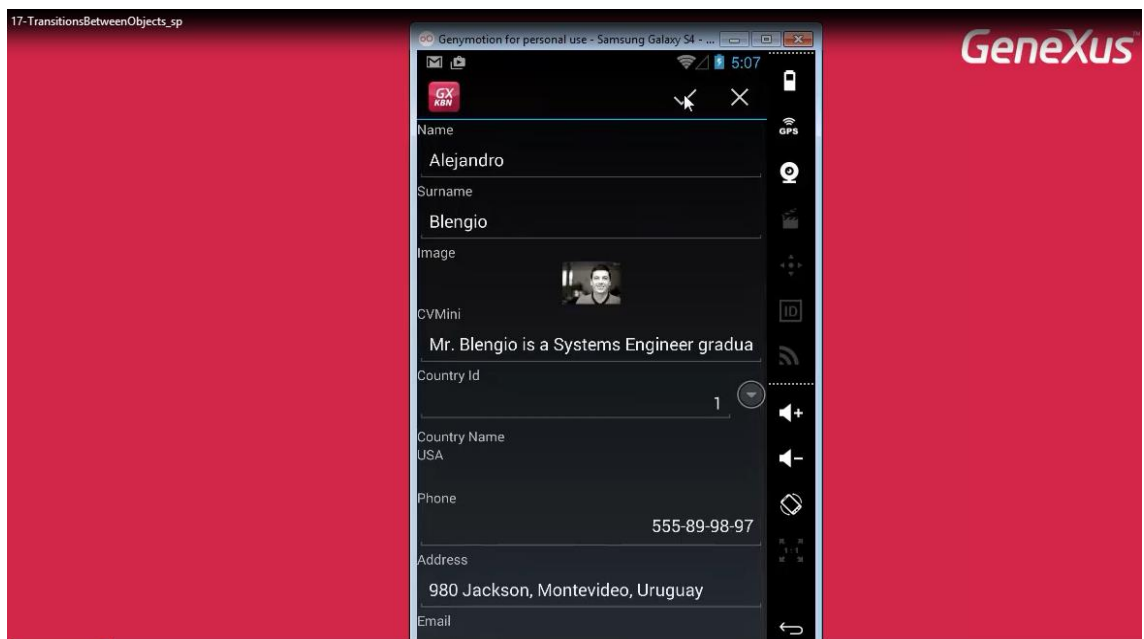


se llama a otra



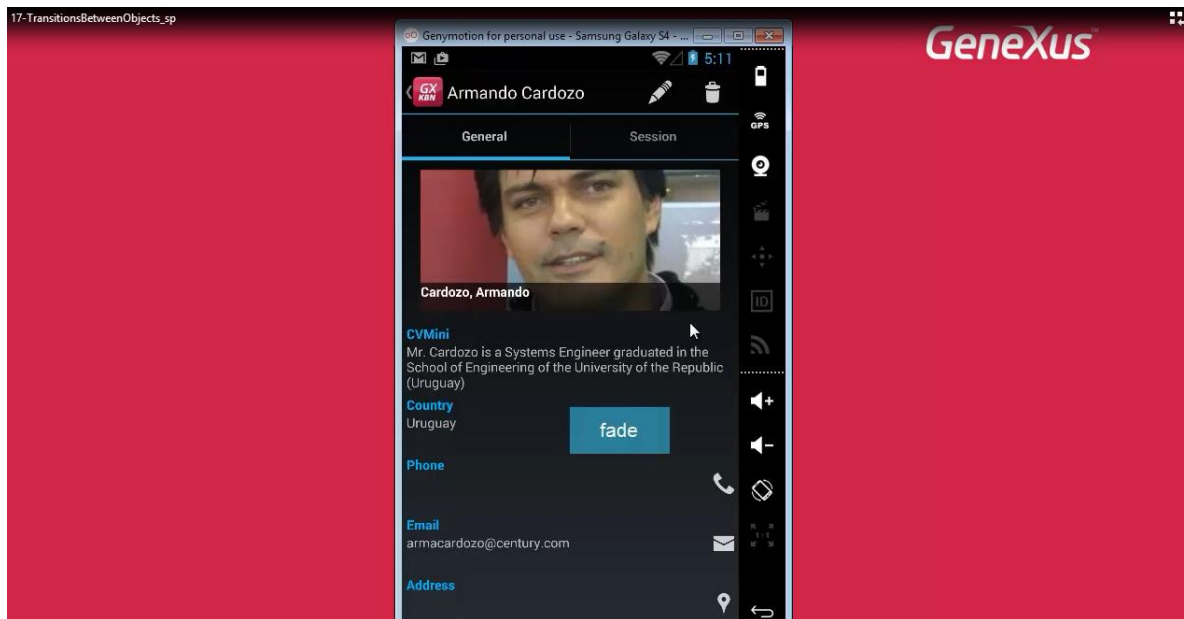


Tenemos la posibilidad de definir



con qué efecto de entrada y de salida, se producirá esa transición.

Aquí estamos ejecutando en el emulador de Genymotion una aplicación Android para el sistema operativo 4.3. y vemos entonces que el objeto default al invocar a otro objeto es: Fade



Sin embargo esto no es estándar para todas las aplicaciones Android. Va a depender de la versión del sistema operativo y del fabricante del dispositivo.

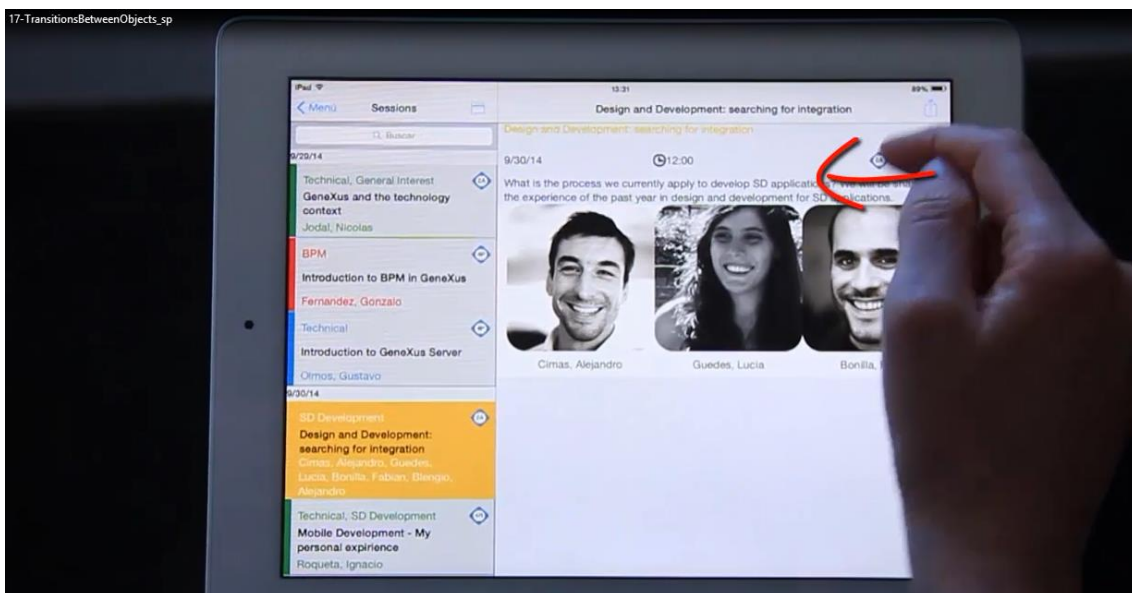
El valor por defecto en iOS es diferente dependiendo del método de invocación del objeto



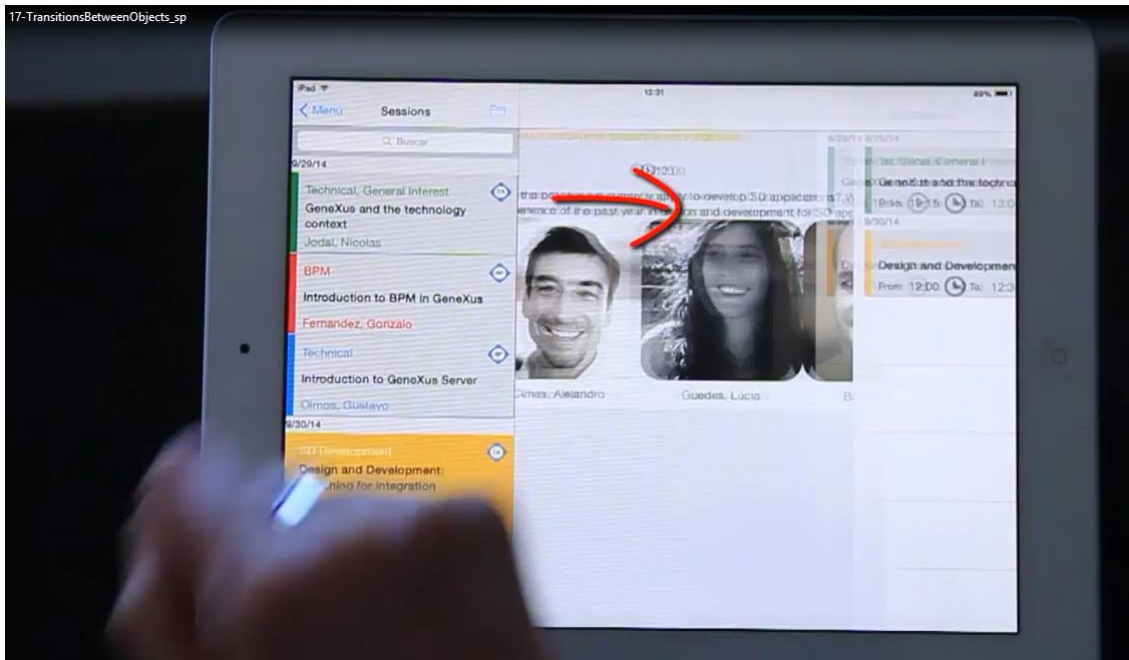


Es diferente si la invocación es Call o Popup.

En el caso de Call como vemos

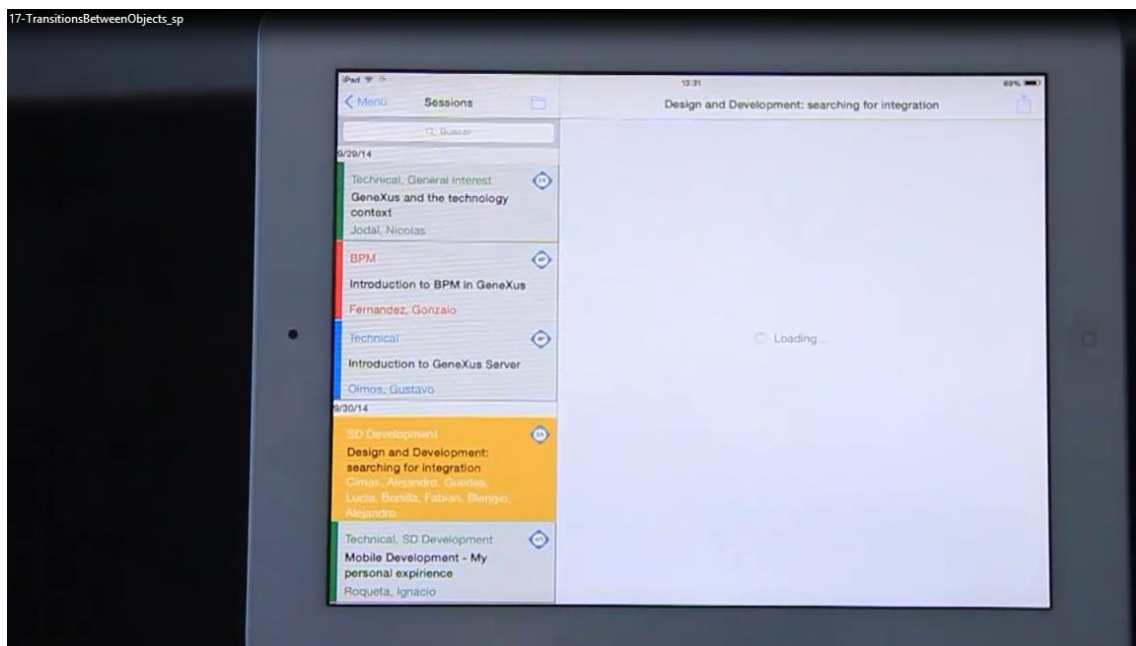


El default es: Push left



lo que significa que se abre desde la izquierda, desplazando lo que está abajo.

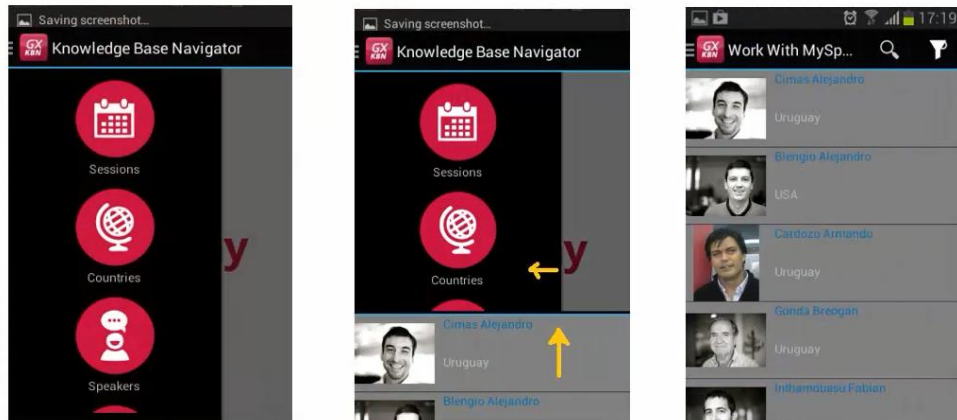
Si el objeto invocado se abre en otro target distinto



entonces se abre con Fade, no con Push Left.

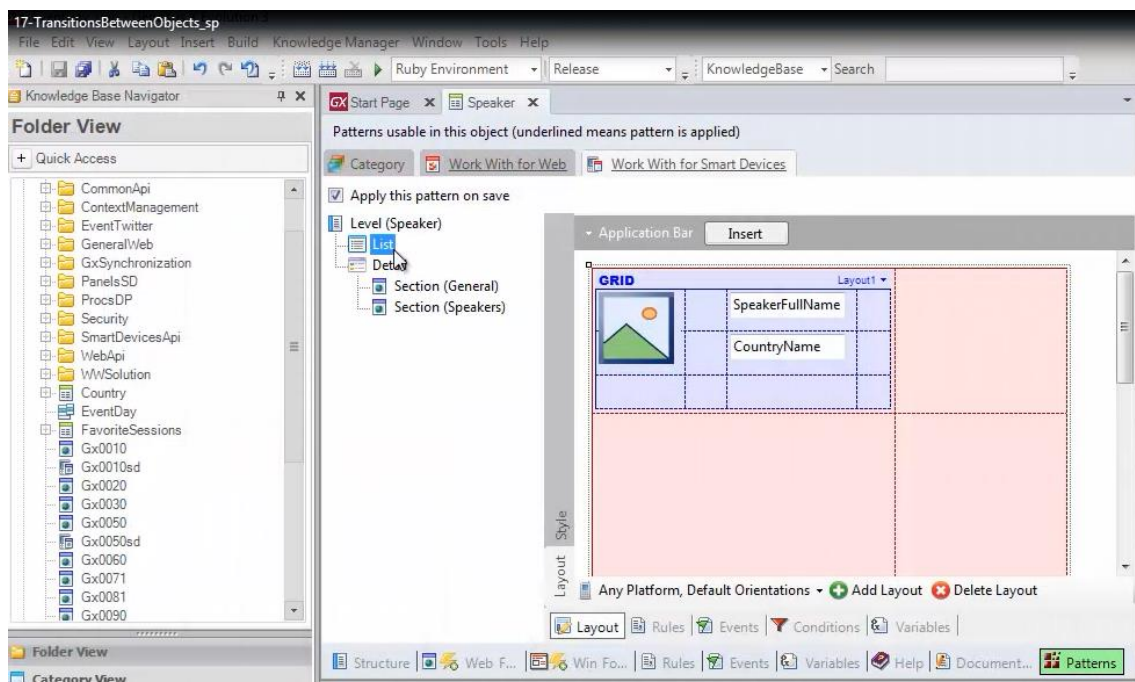
Si queremos que al invocar a la lista de speakers, esta pantalla aparezca desde abajo hacia arriba, Push Up, ¿cómo hacemos para configurar esa transición?

Transitions between objects

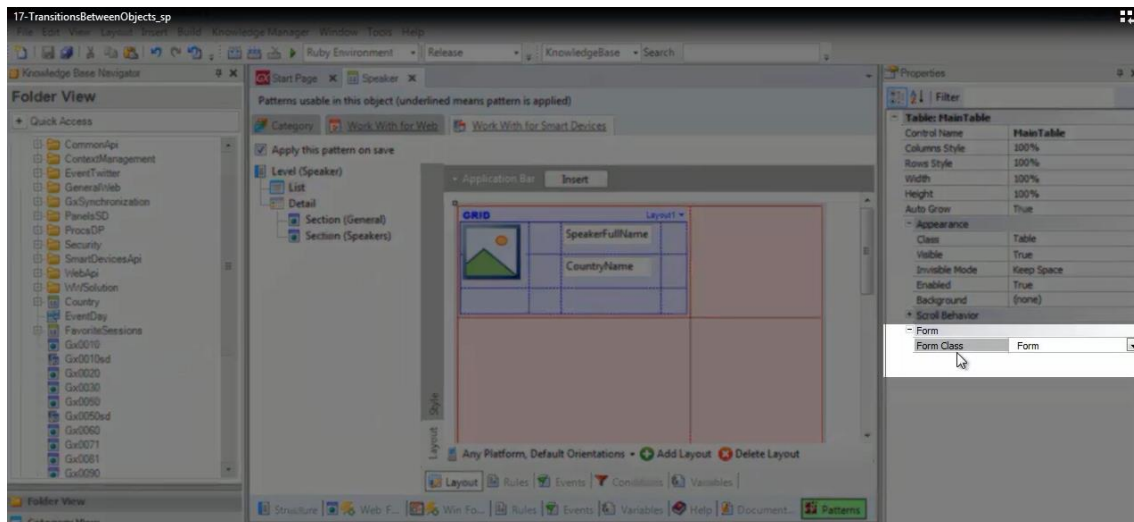


Dicho de otro modo: ¿dónde se encuentra seteado el efecto de entrada y de salida de la pantalla correspondiente al objeto invocado?

Observemos en el List del Work With de speakers

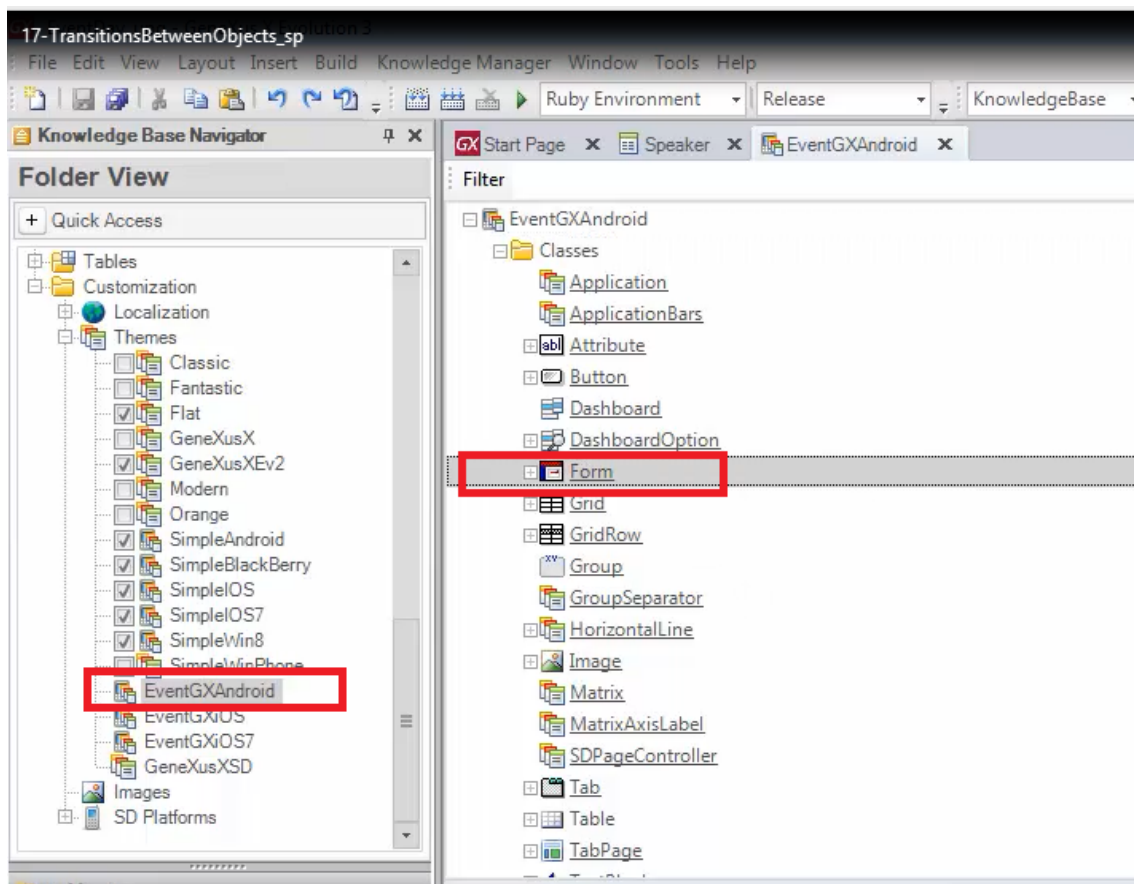


que la tabla principal tiene definida la propiedad

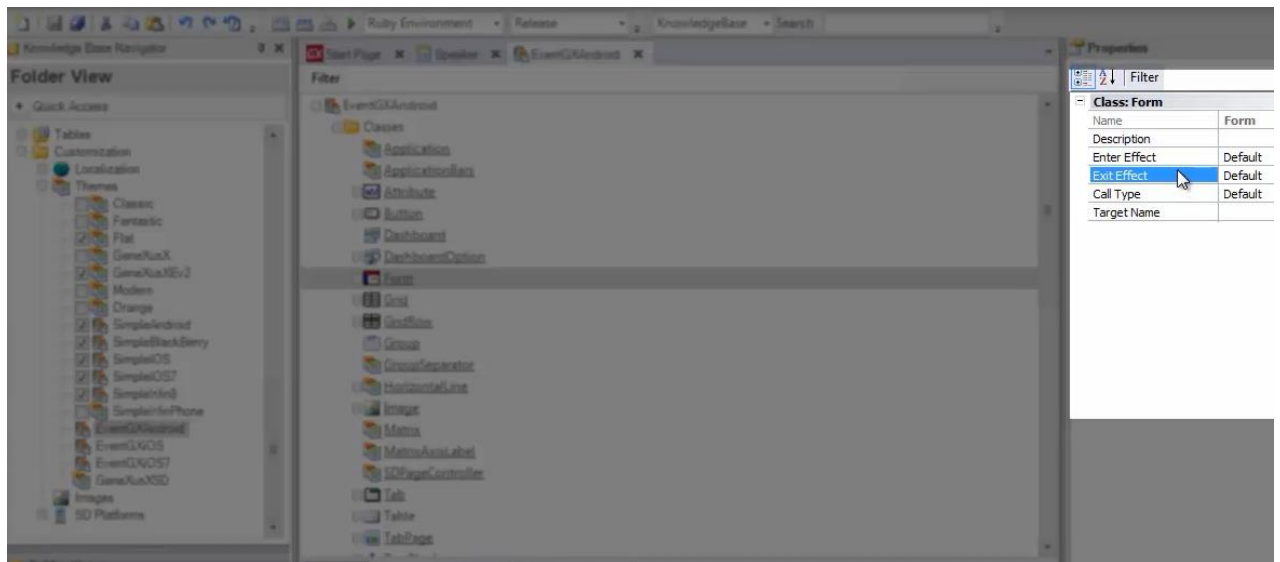


que está asumiendo por defecto la clase: Form, del theme correspondiente.

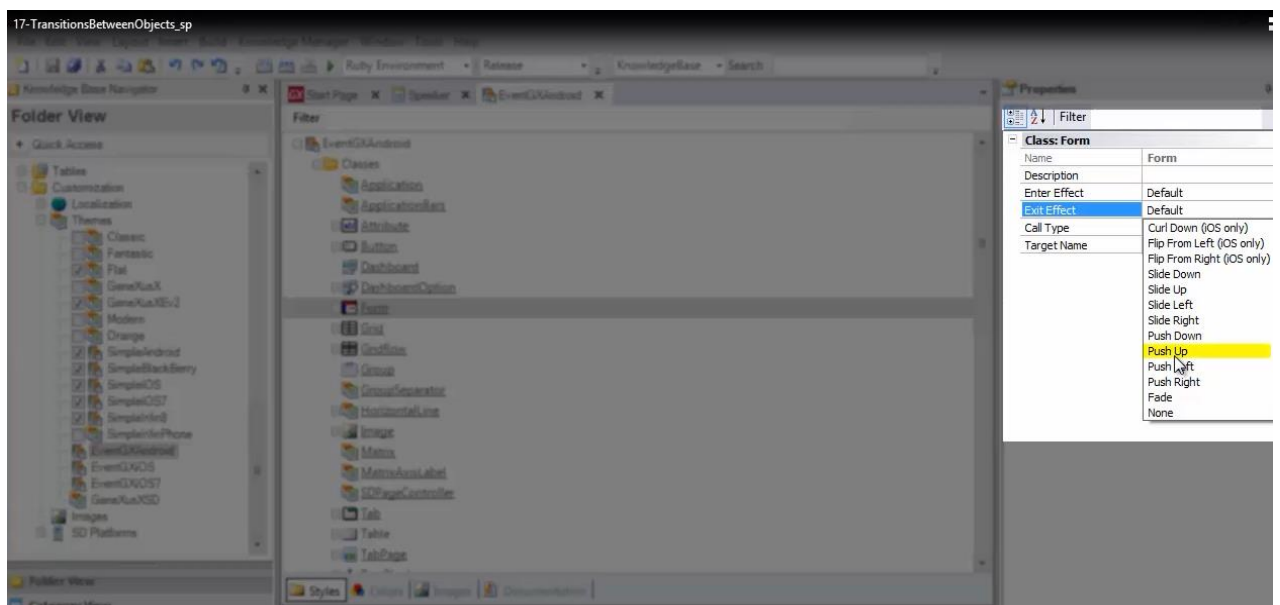
Si vamos a ver las propiedades de esa clase Form a ese theme (por ejemplo vamos a buscar el de Android..)



vemos que tenemos 2 propiedades: Enter Effect y Exit Effect, que justamente lo que hacen es definir esos efectos de transición, cuando se abre la pantalla y cuando se abandona



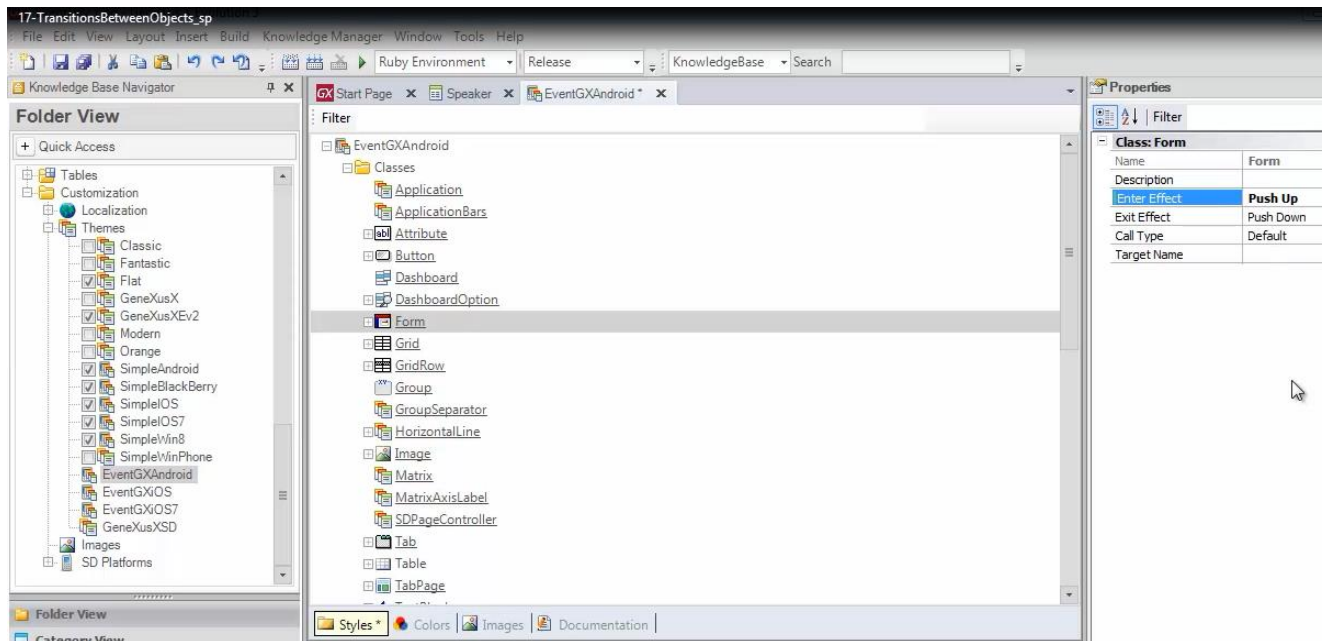
Observemos que entre todos los efectos posibles tenemos el Push Up por ejemplo



O el Push Left, que decíamos que era el Default para iOS cuando el objeto invocado se abría en el mismo target.

También veamos que algunos efectos sólo son válidos para iOS.

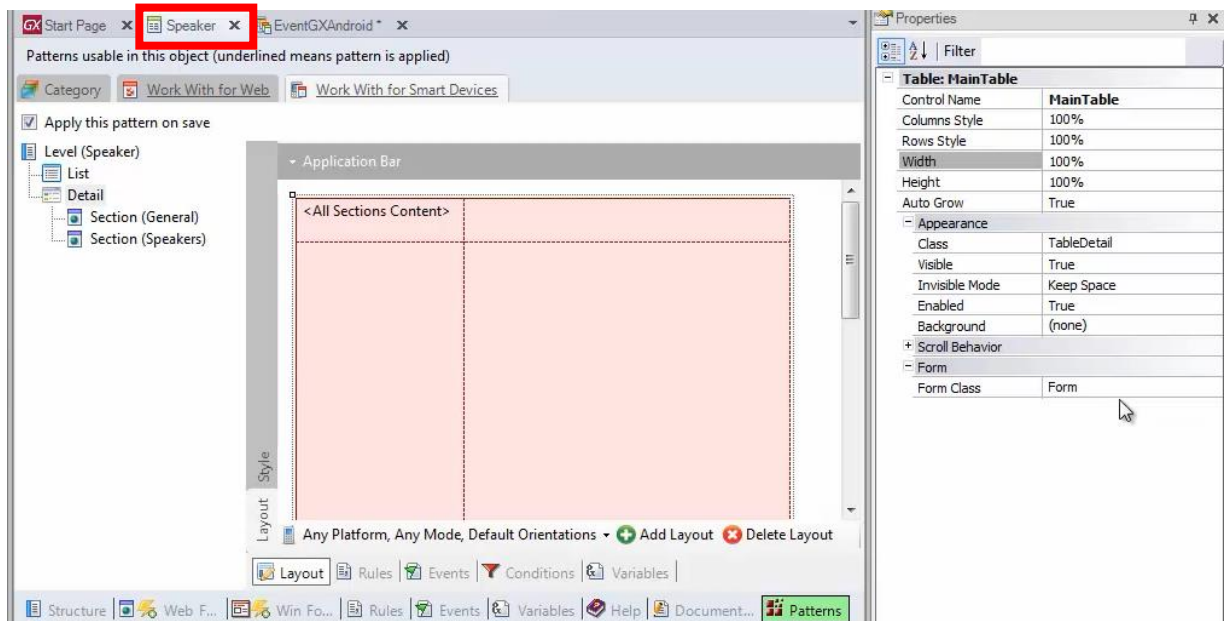
Por tanto, si en este form configuramos **Push up**, como efecto de entrada



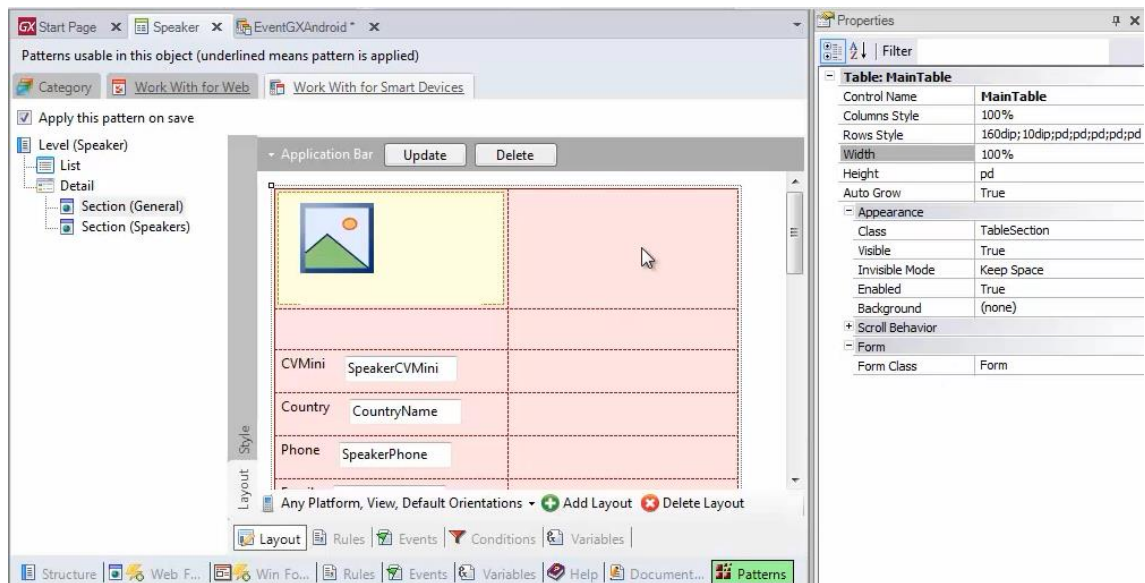
Observemos que simétricamente se configuró el efecto de salida como Push Down, es decir, su negativo; pero podemos modificarlo siempre y cuando tenga sentido.

Si hacemos esto, entonces todos los layouts de todos los objetos...

por ejemplo, vayamos al Detail y veamos que tiene la Form Class: Form

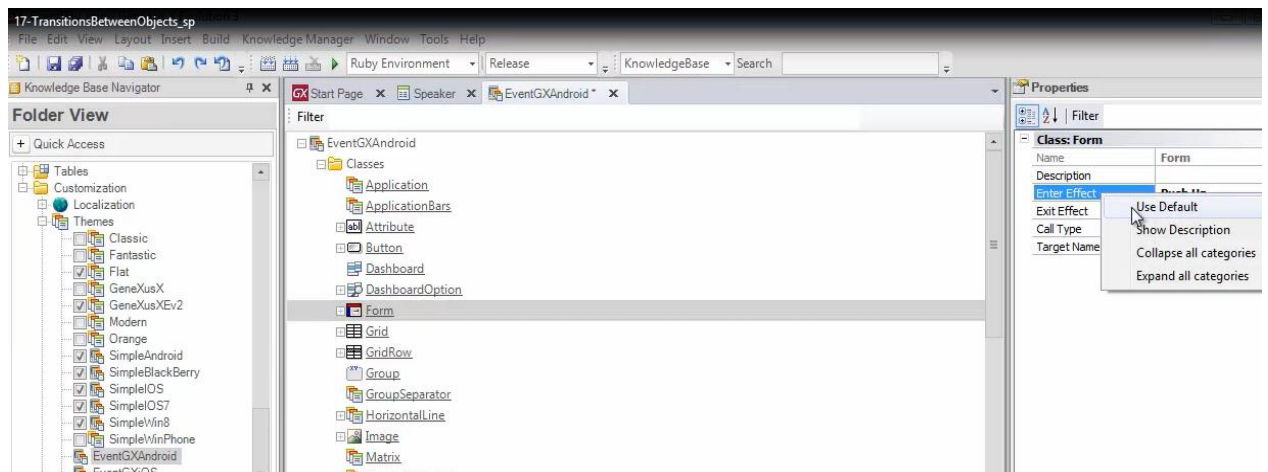


La Section General, también tiene la Form Class: Form

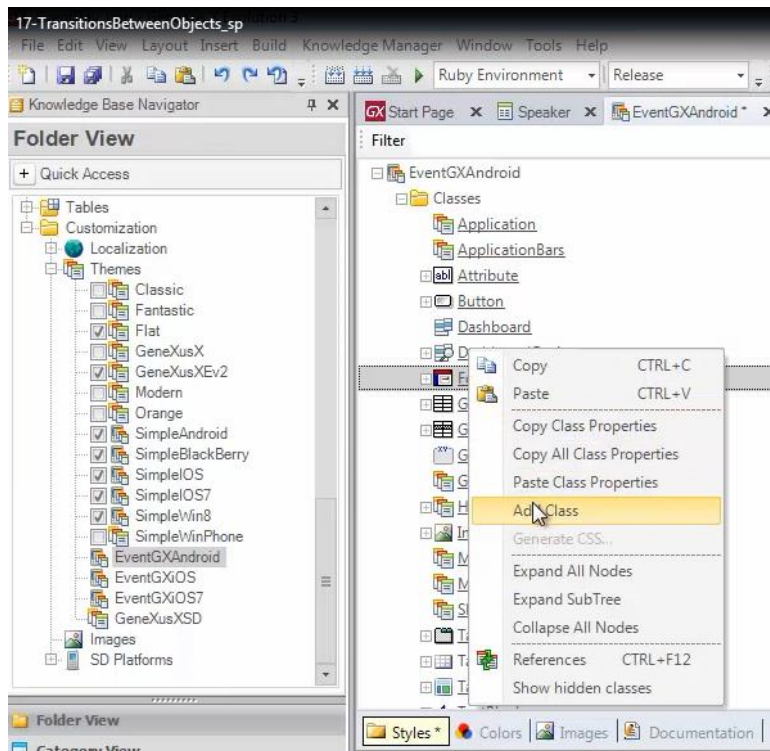


y así sucesivamente, entonces todos los objetos invocados que tengan esa clase van a abrirse con ese efecto de entrada y con ese efecto de salida, siempre que utilicen el theme: Android.

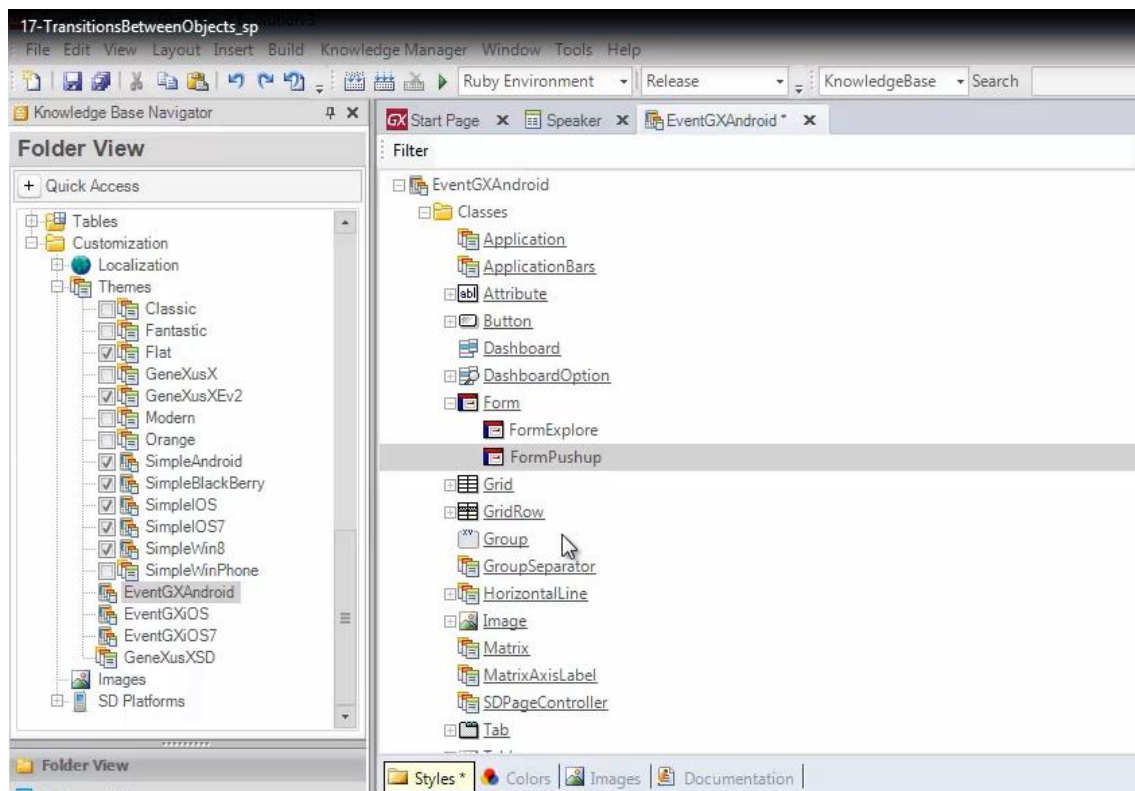
Pero si queremos que solamente algunos layouts se abran con Push Up y Push Down, como efectos de entrada y de salida respectivamente, entonces tendremos que definir una clase particular con esos efectos. A **Form** entonces podemos dejarle los efectos Default



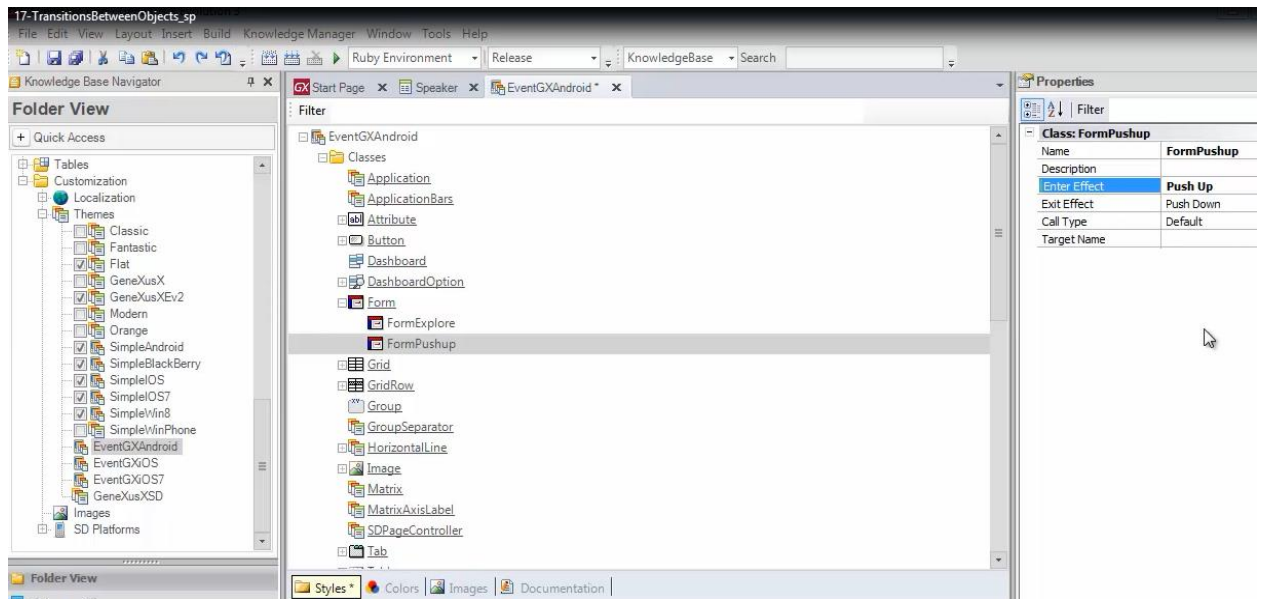
Y crear una subclase de la Form



A la que le vamos a llamar: FormPushup, por ejemplo:



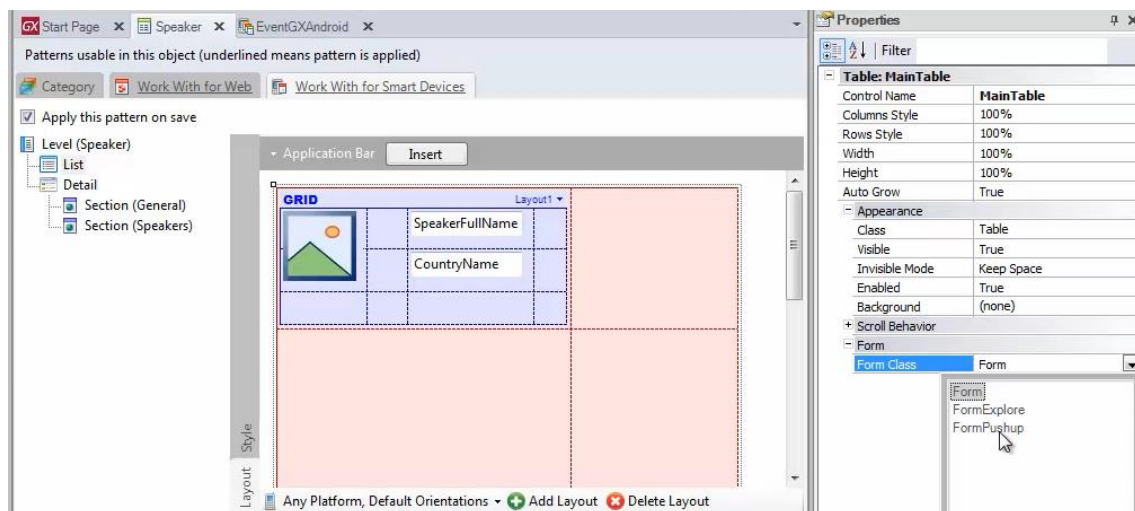
para poder usarla en todos los layouts en los que nos interesen esos efectos, y vamos a definir en esa clase, el efecto de entrada como: Push Up



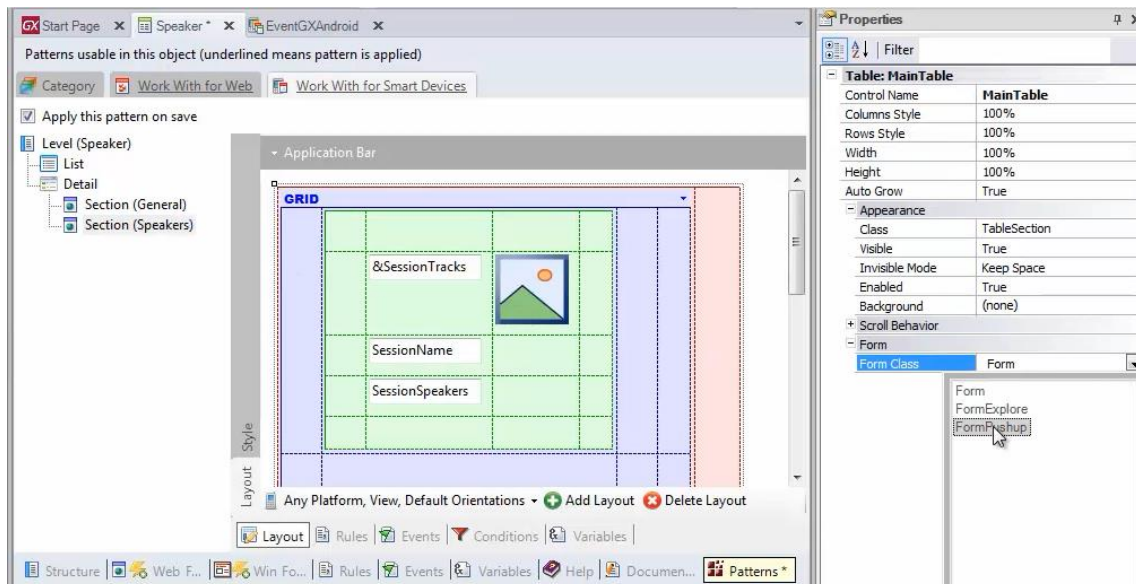
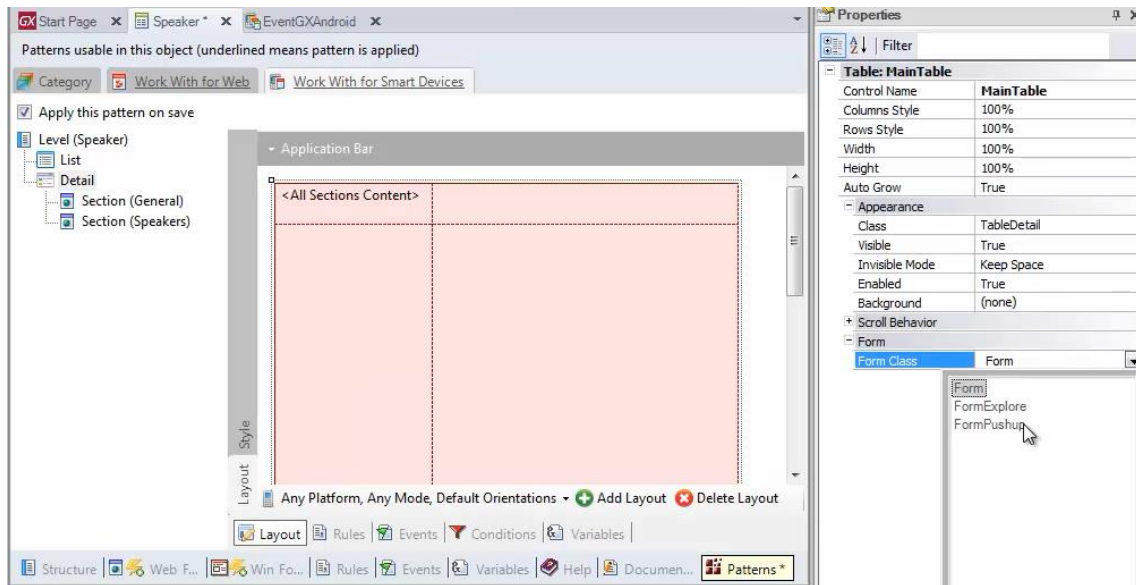
y el efecto de salida, dejarlo: Push Down

Grabamos.. y ahora lo que tenemos que hacer entonces, es ir a los objetos donde nos interesa que las pantallas se abran con esos efectos

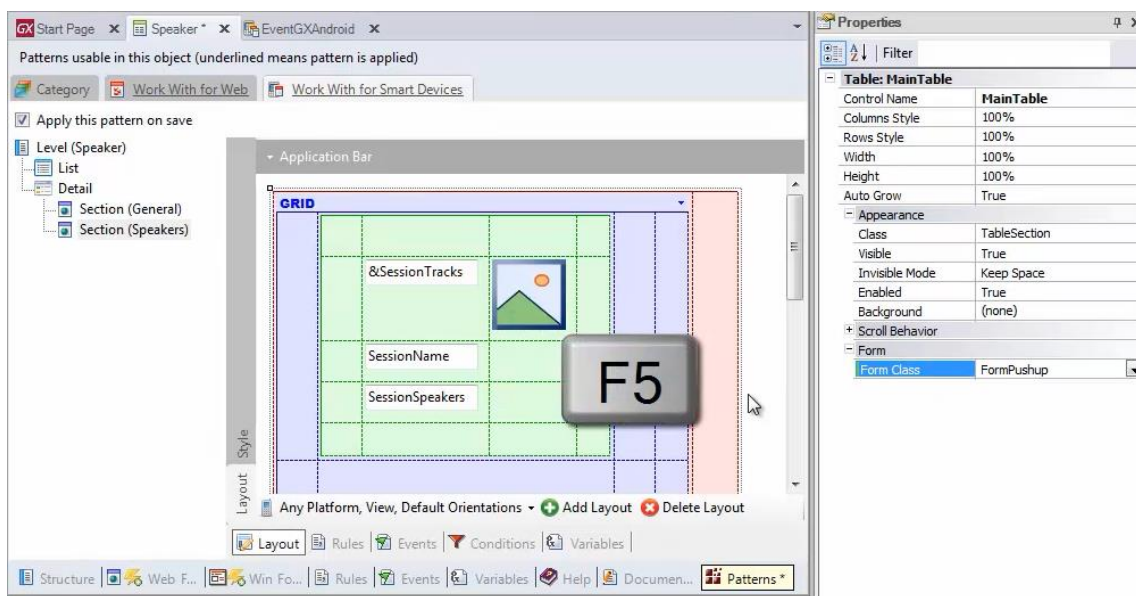
y configurarle la Form Class → por esta nueva



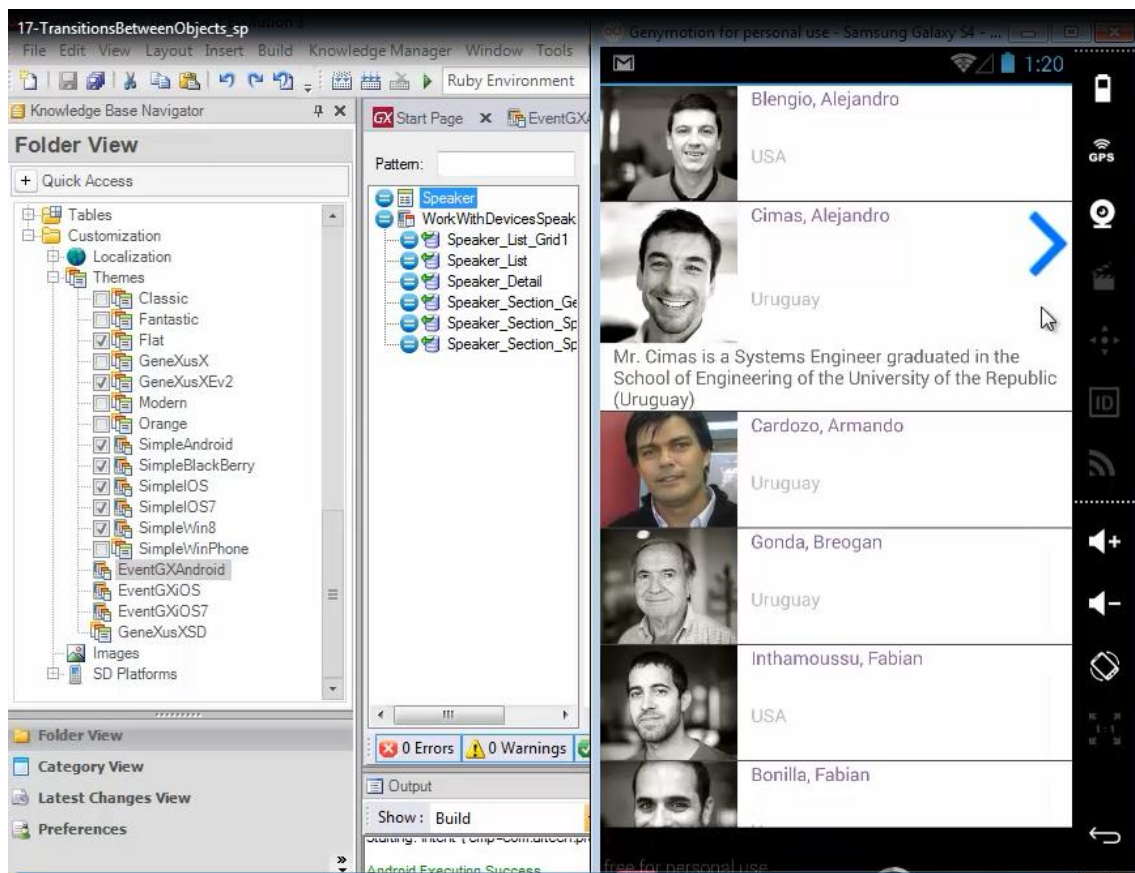
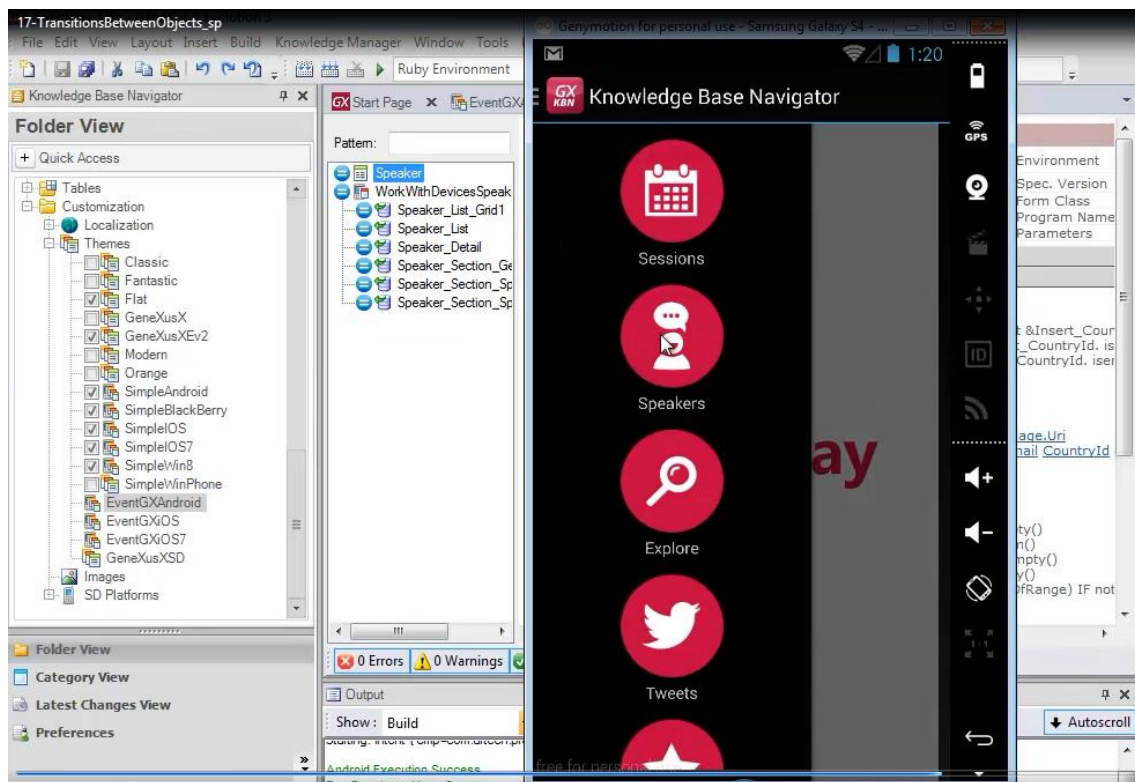
Vamos a hacerlo para todas las pantallas de este work with

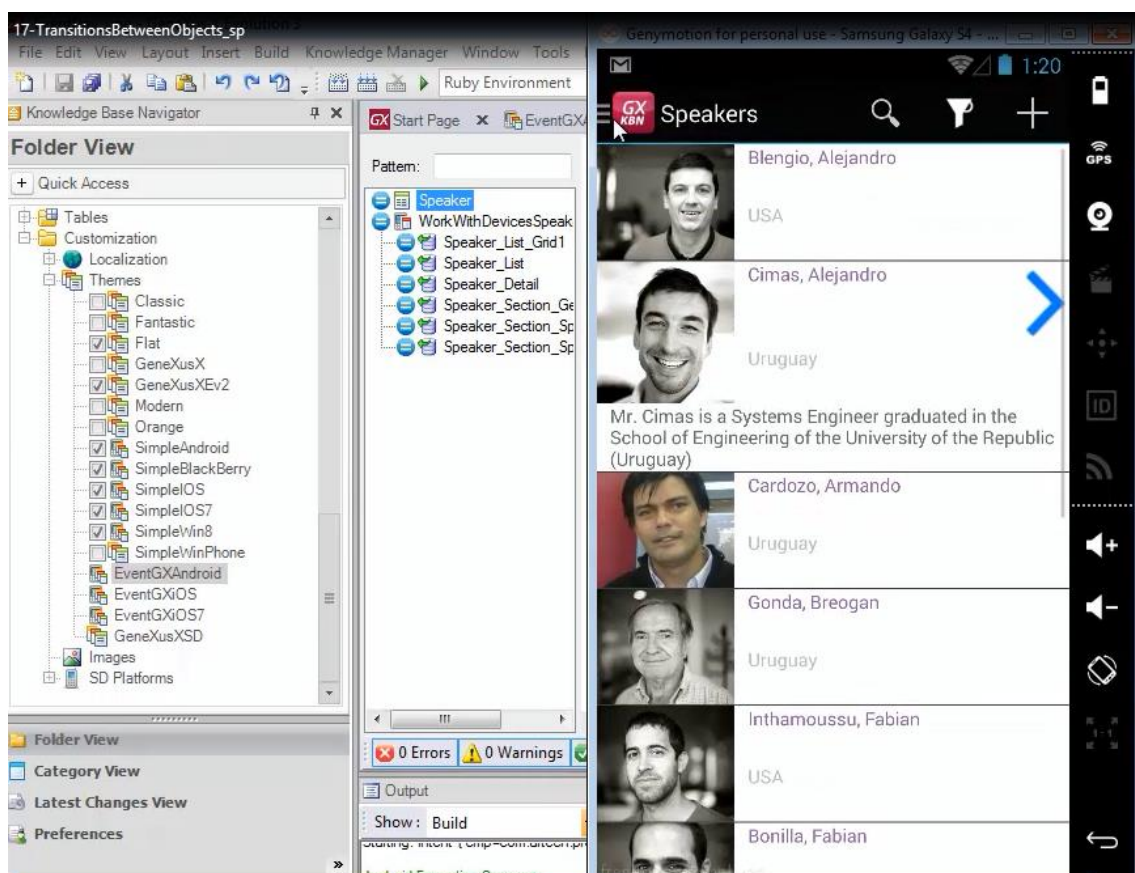
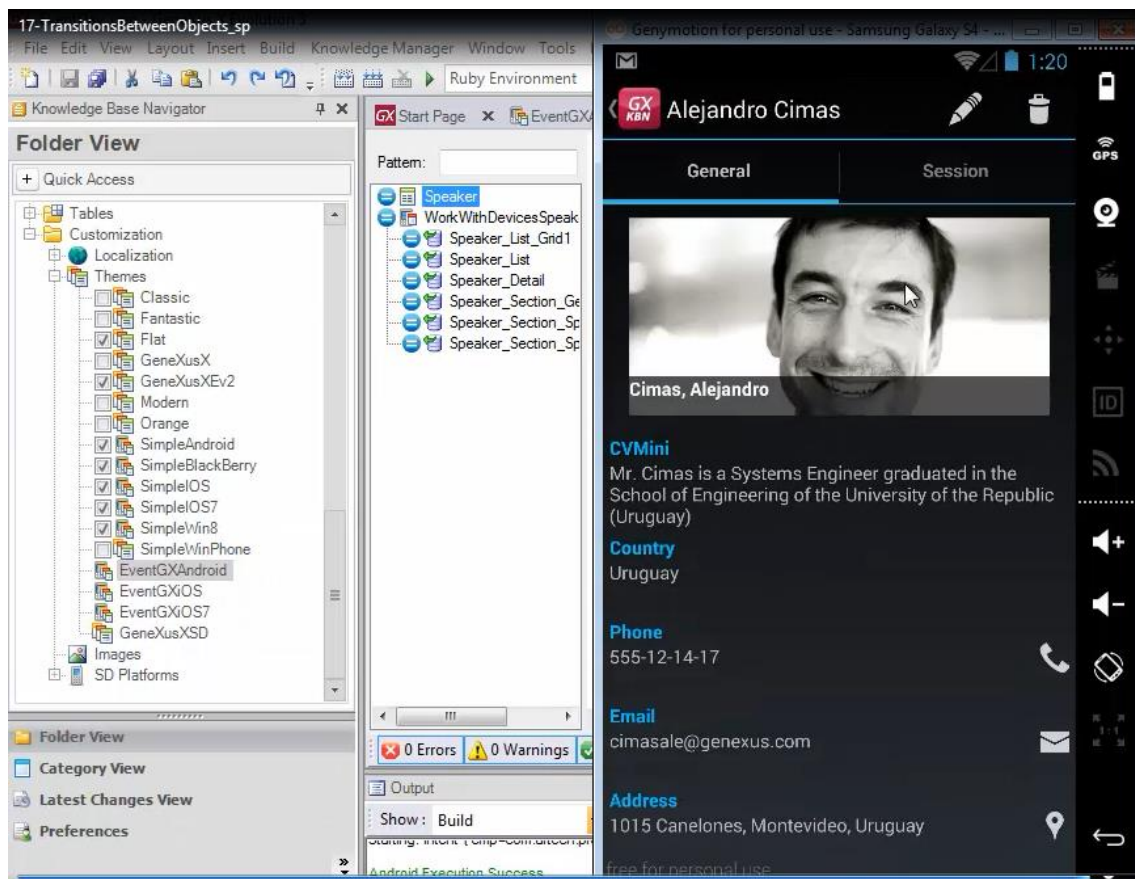


Hagamos F5



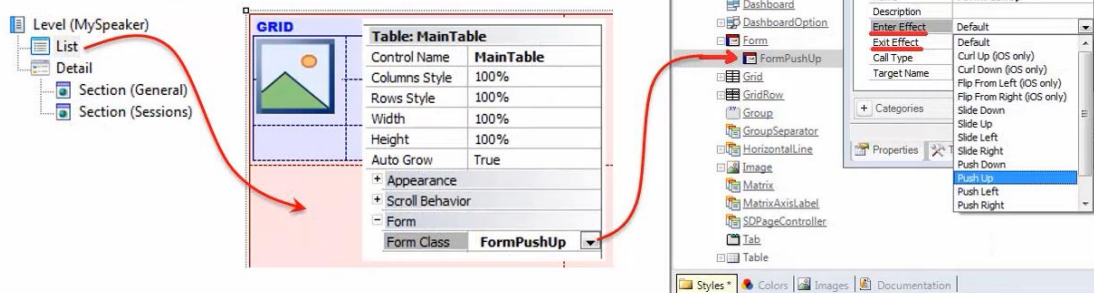
Video filmado con GeneXus X Evolution 3





Transitions between objects

At design time

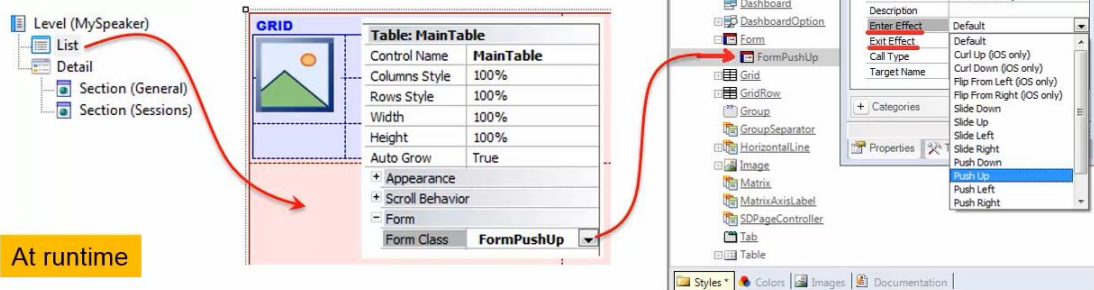


Acabamos de ver cómo configurar el efecto de entrada y de salida para la pantalla invocada, dejándolo fijado en tiempo de diseño.

Pero también podemos hacerlo en tiempo de ejecución

Transitions between objects

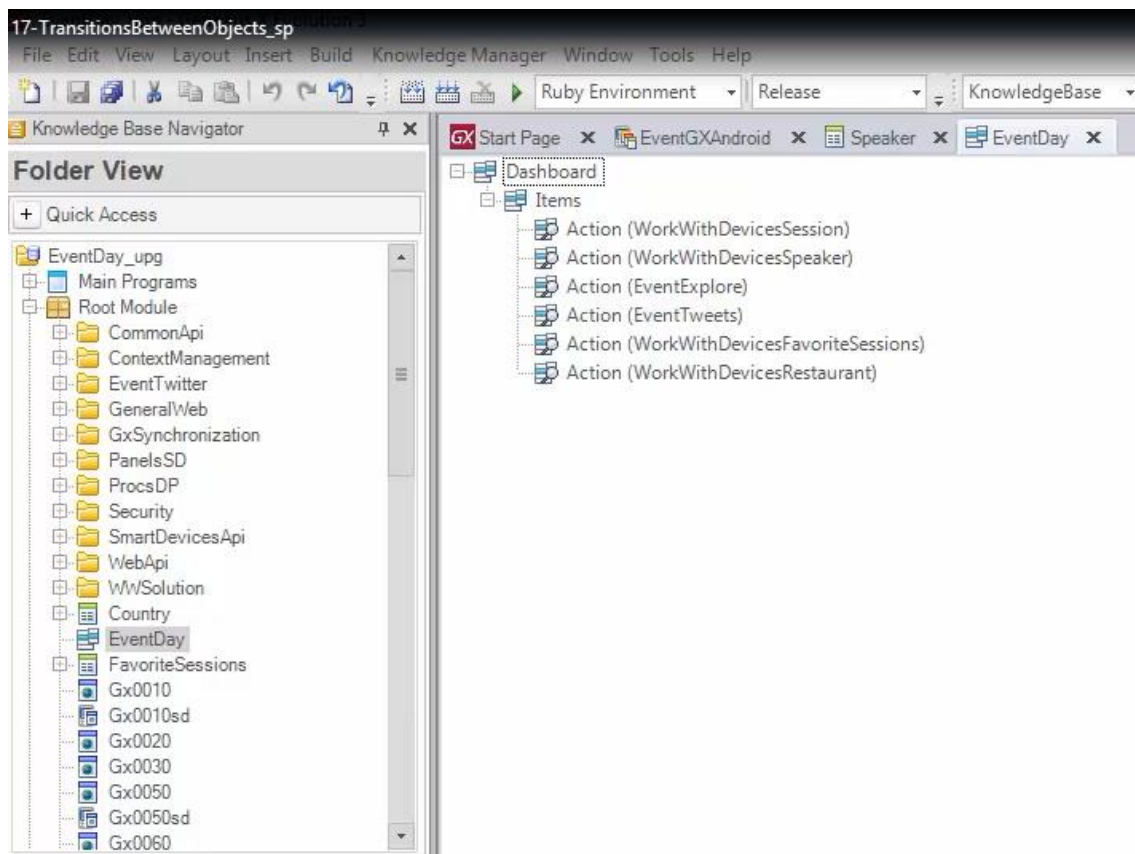
At design time



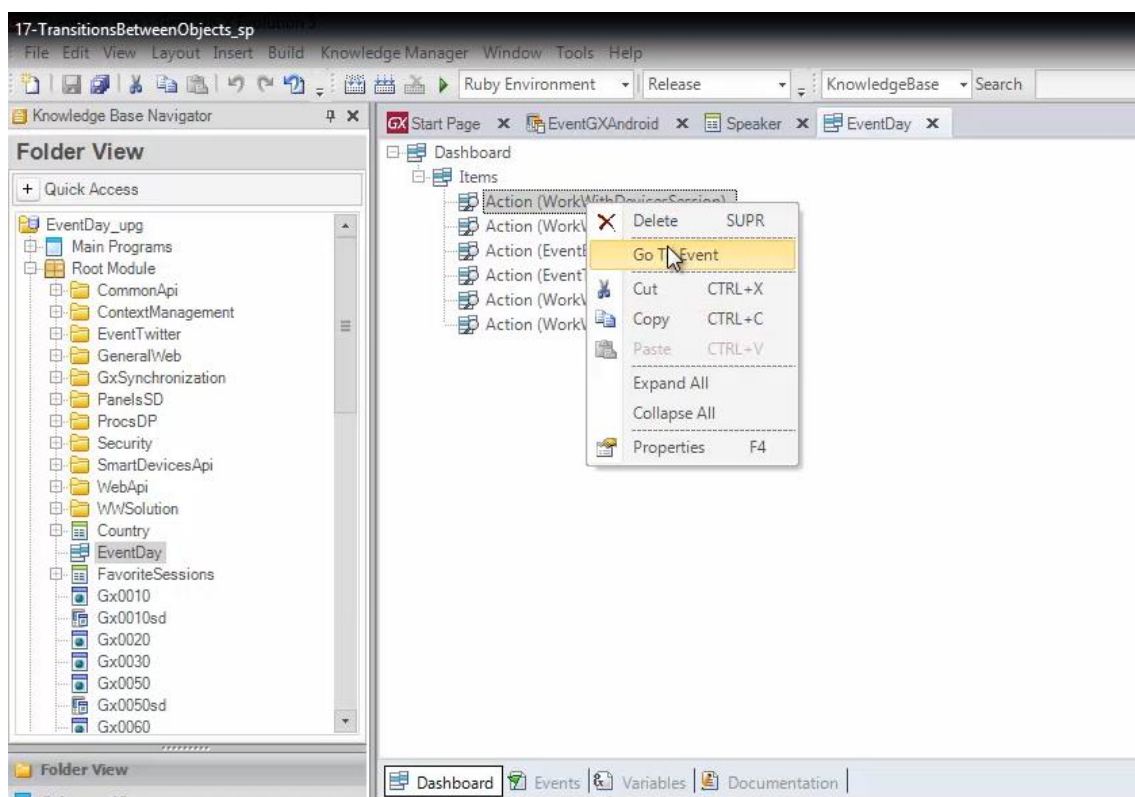
At runtime

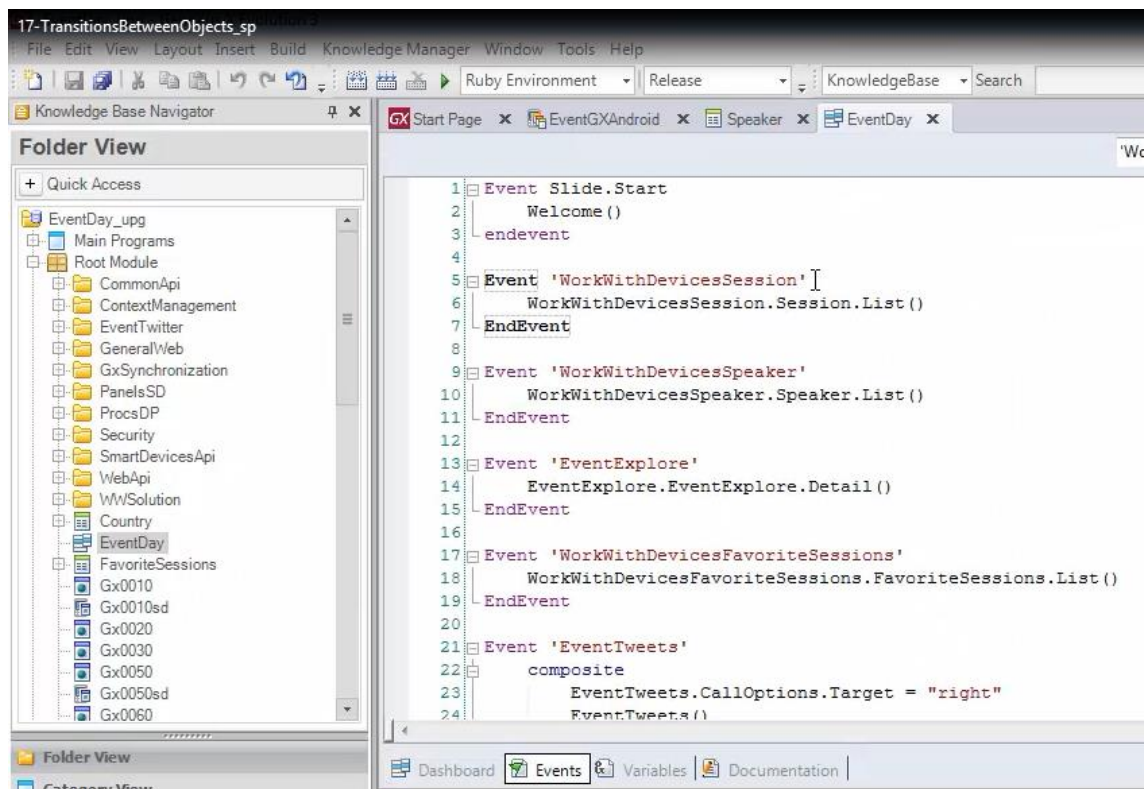
lo que nos dará más libertad.

Así, desde el objeto llamador, por ejemplo el Dashboard

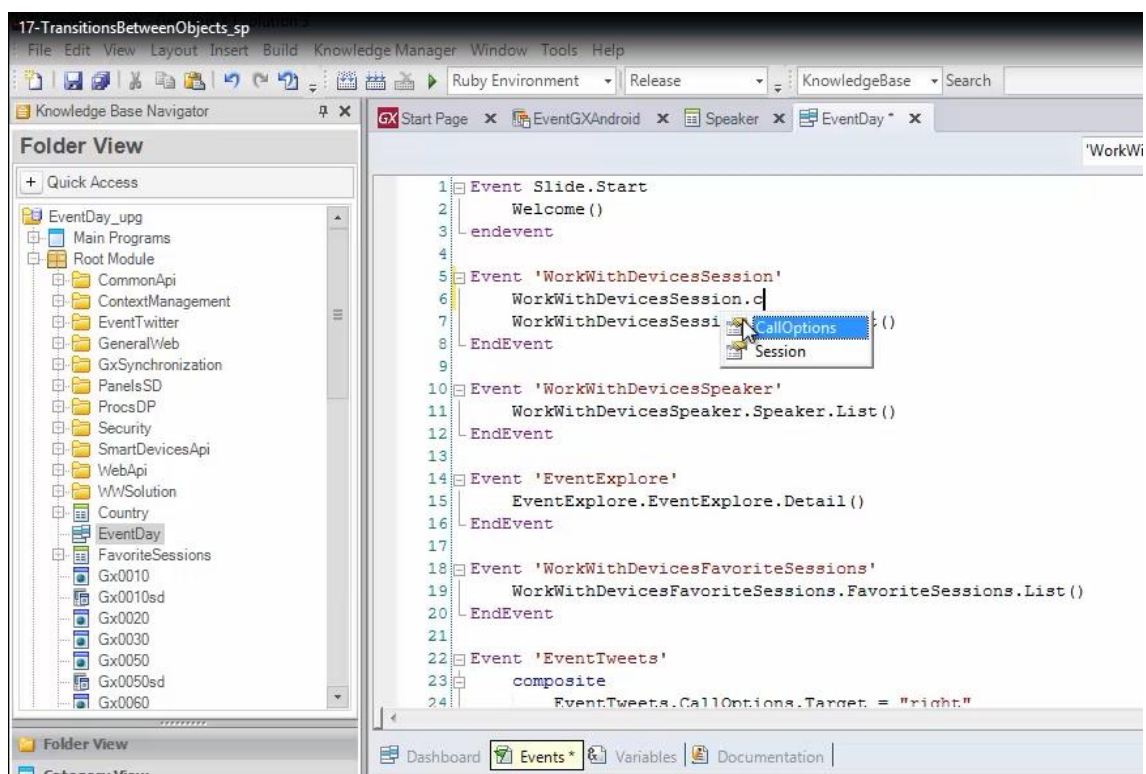


antes de hacer la invocación al List de Sessions,



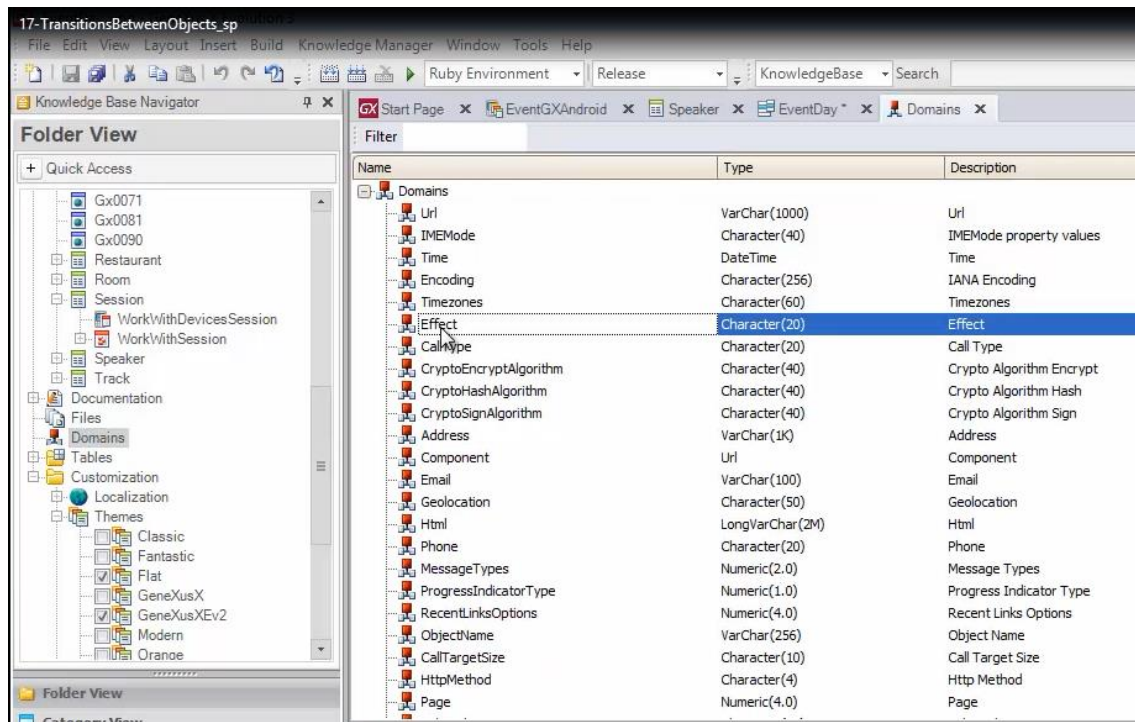


mediante la propiedad: CallOptions dinámica, del objeto que estamos invocando,

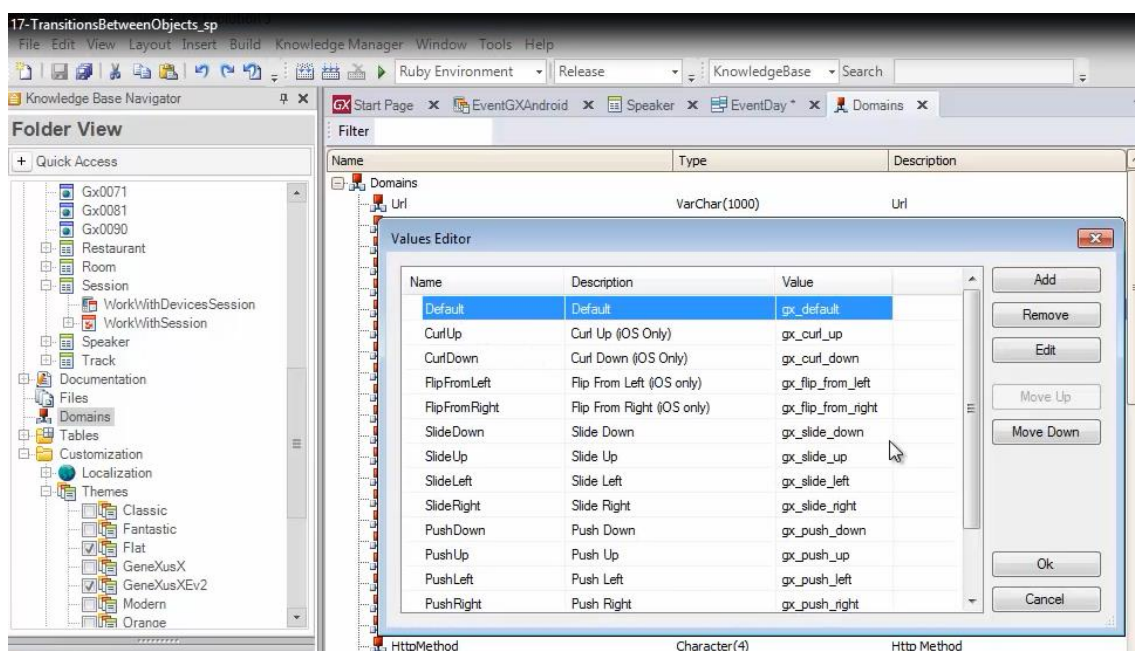


(a la que volveremos en otro video), configuramos el efecto de entrada, igual al que nos interesa.

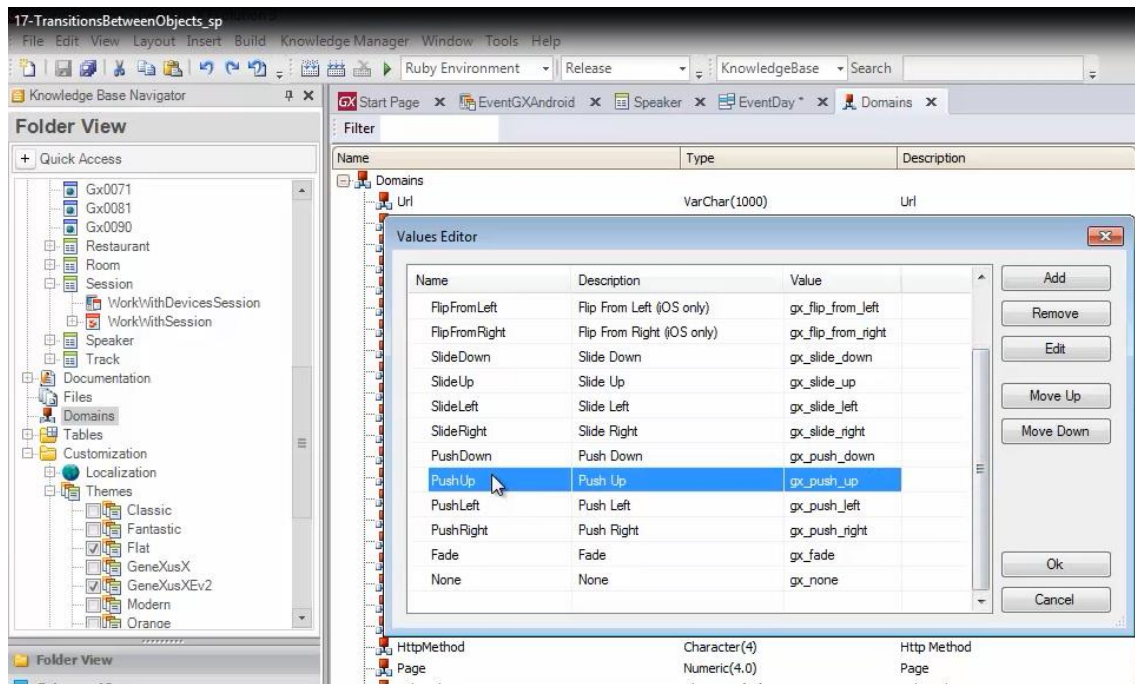
Los efectos vienen codificados en un dominio predefinido, que se llama: Effect



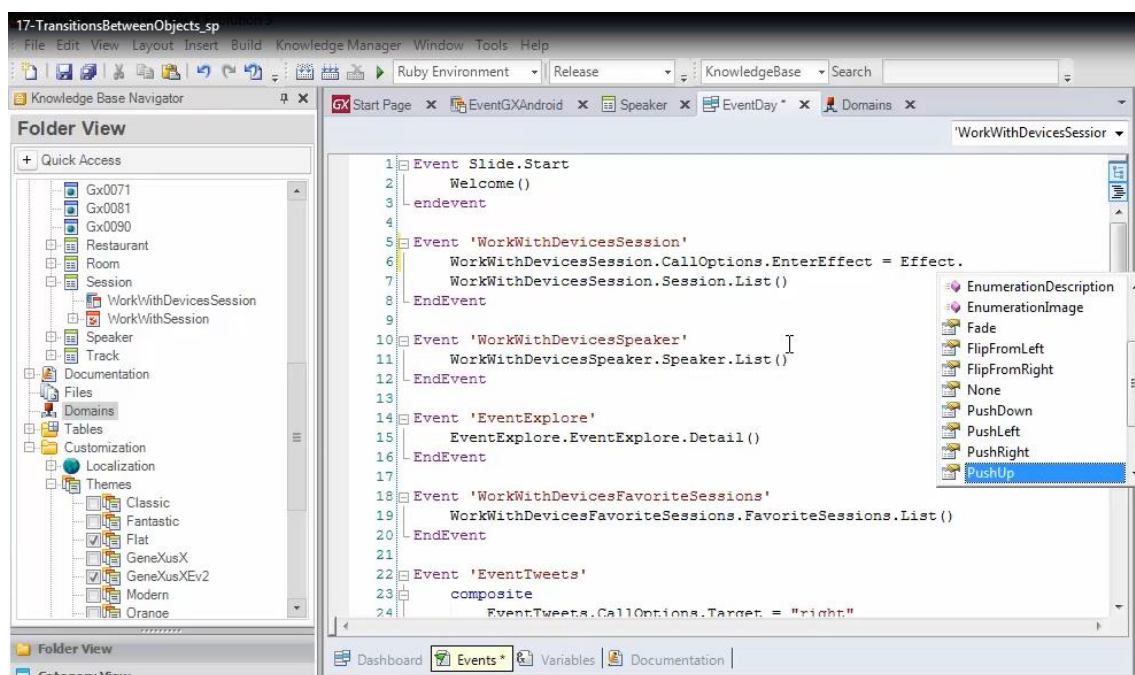
que si vemos, es un enumerado, con todas las posibilidades ofrecidas



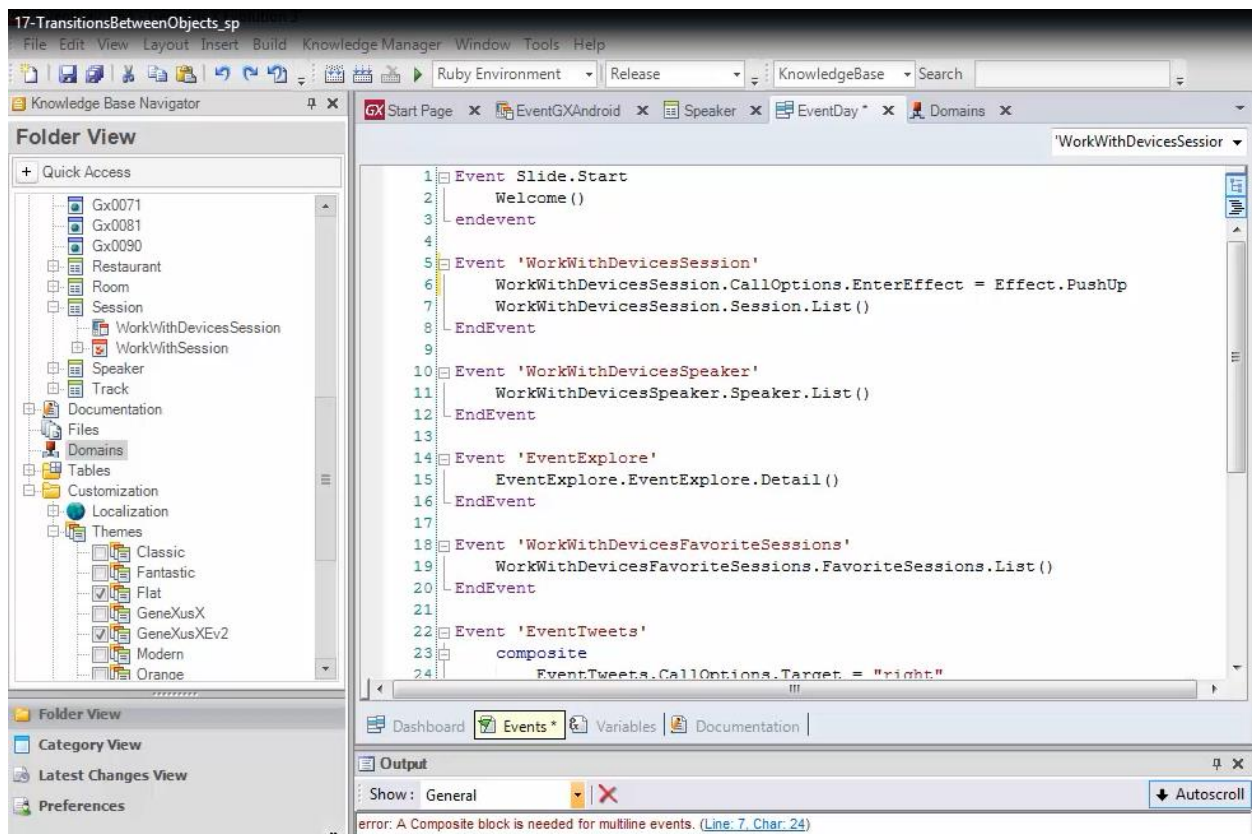
entre ellas: Push Up, Push Down



Entonces lo que hacemos es utilizar ese dominio numerado.. y elegir la opción que nos interesa: Push Up



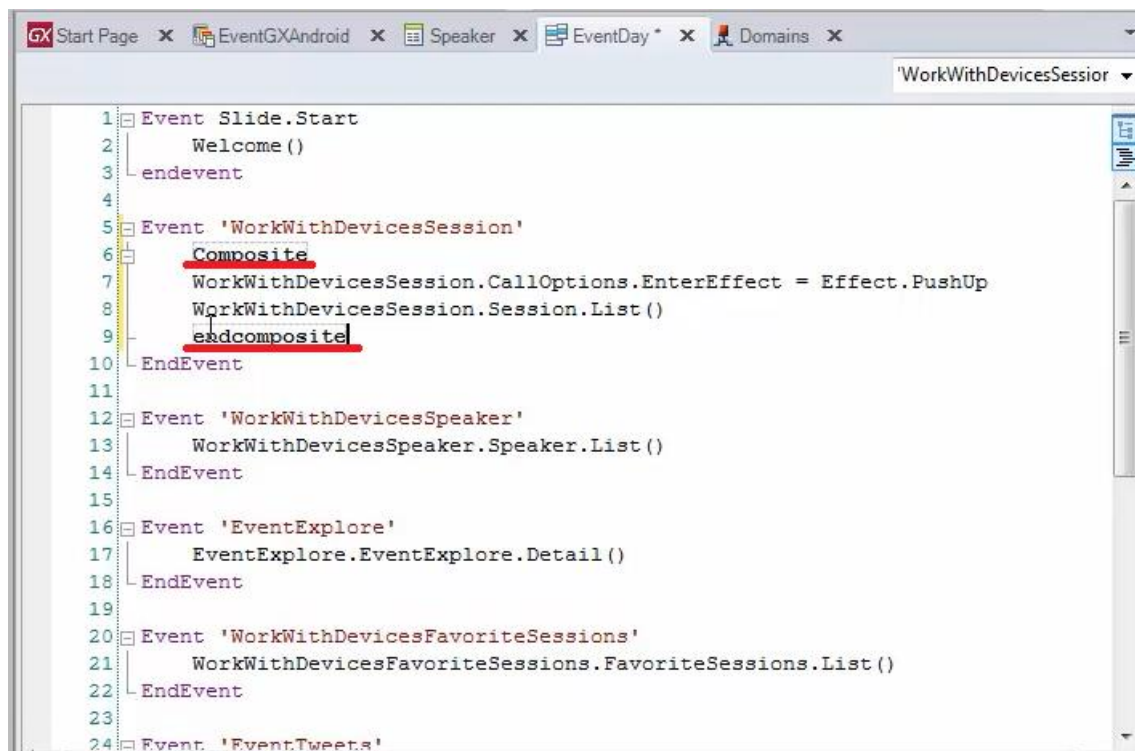
Si intentamos grabar, así, nos va a dar un error que nos informa



que nos está faltando el bloque: Composite, necesario para eventos multi-línea.

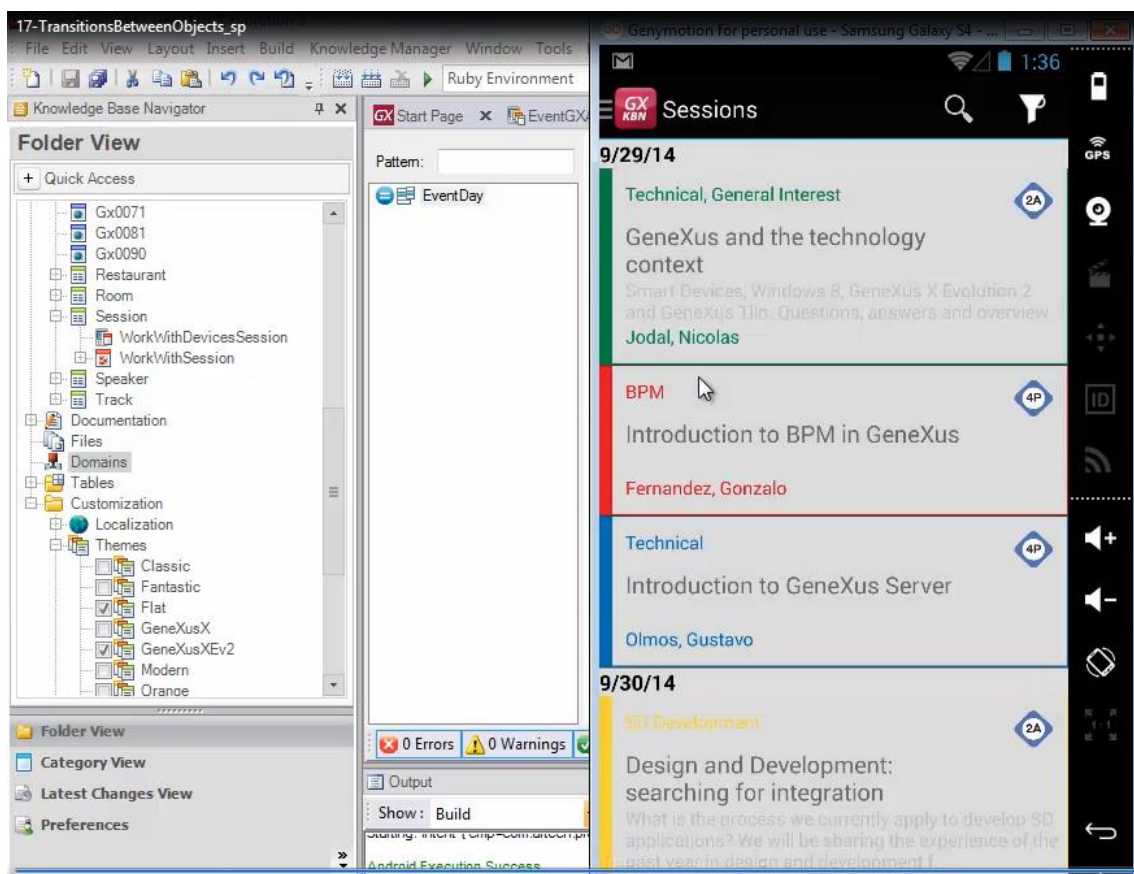
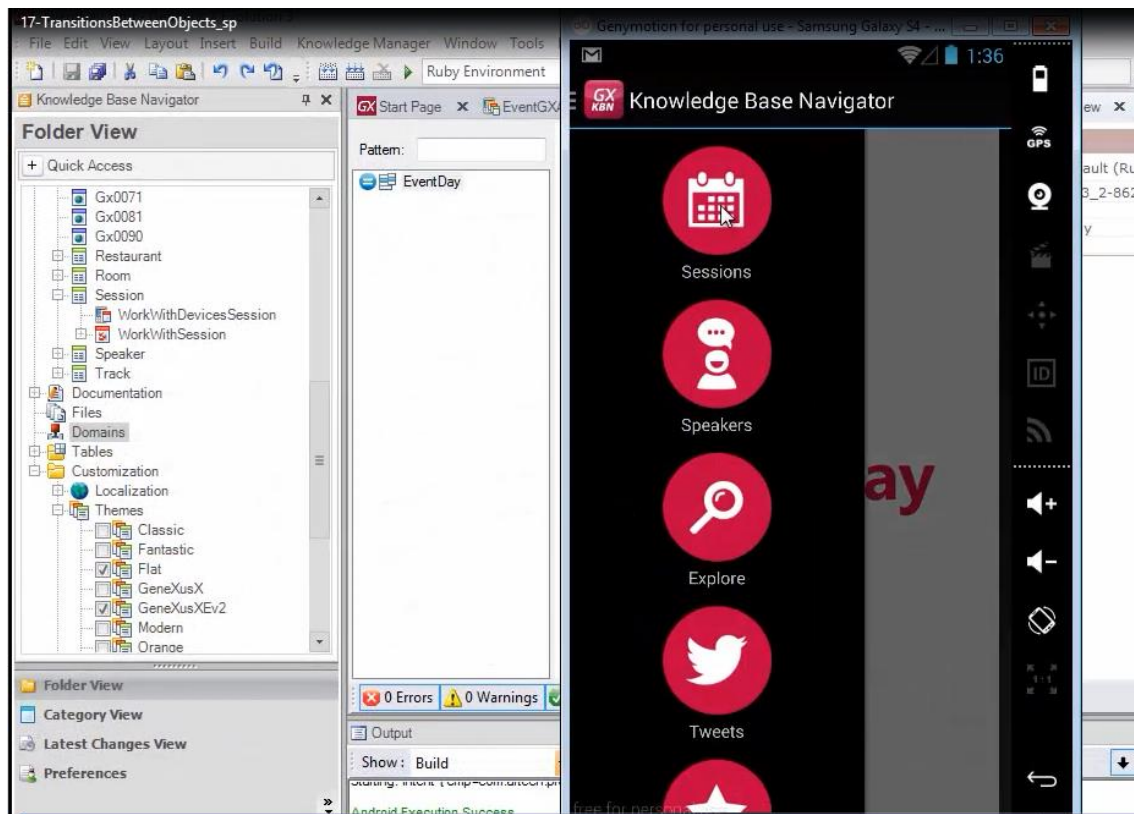
Sobre esto volveremos más adelante.

Vamos a colocarlo simplemente

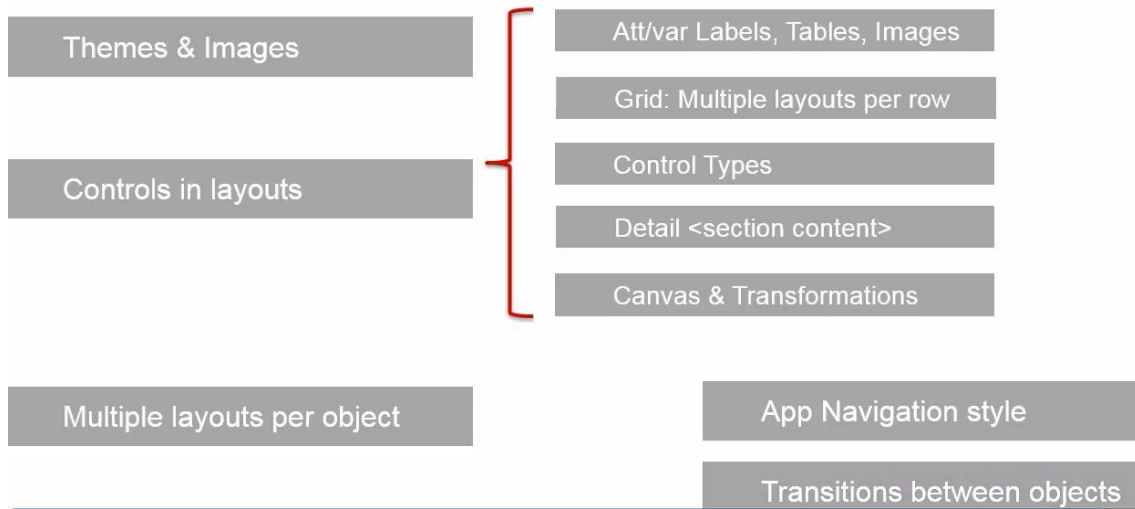


Y ahora sí, hacer F5

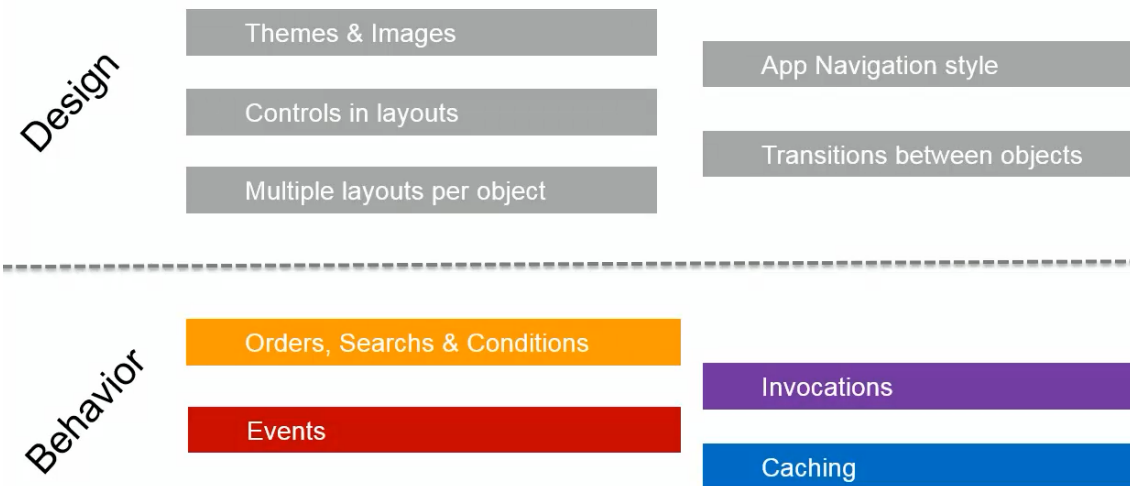
Vamos a ejecutar ahora el List de Sessions



y vemos cómo se abrió desde abajo



Terminemos de estudiar con las transiciones, los aspectos que hacen al diseño de la aplicación.



En lo que sigue, nos abocaremos a estudiar el comportamiento, aunque en algunos aspectos, diseño y comportamiento, se mezclan.

