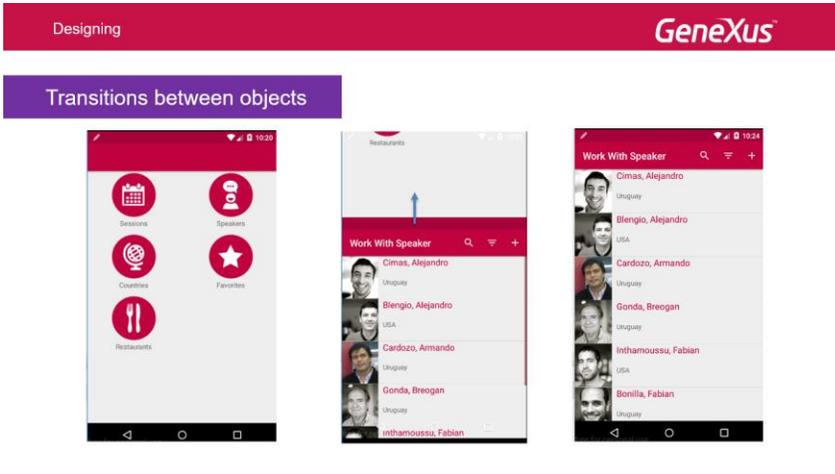
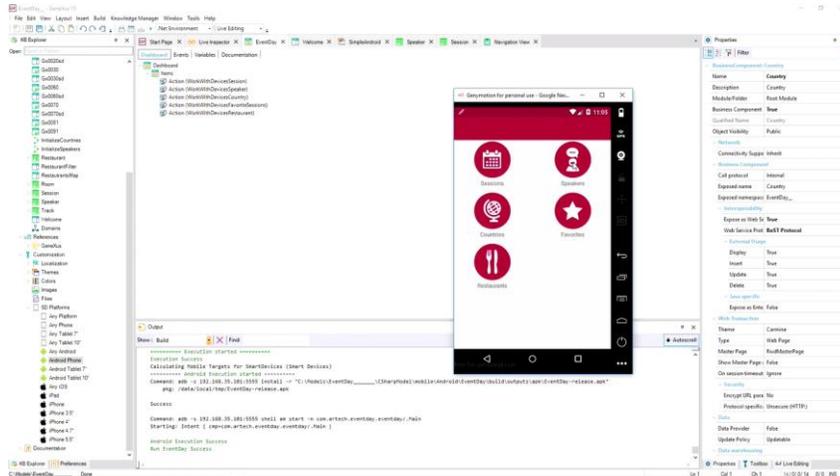


Transitions between objects

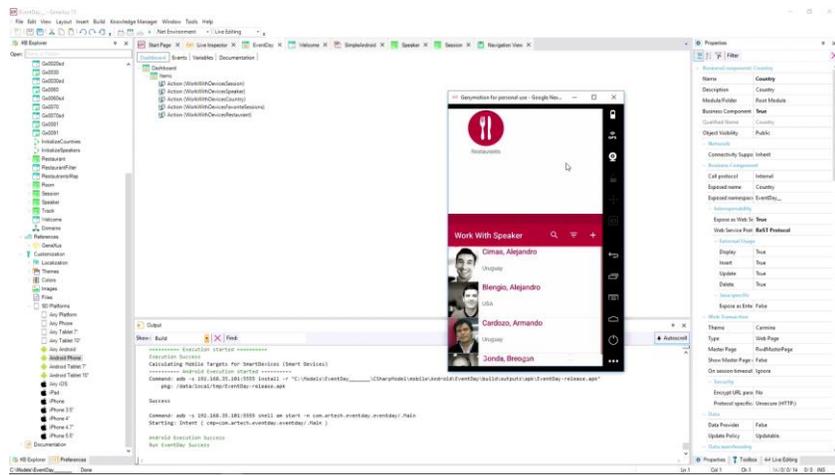


Ahora veremos la forma de incorporar distintos efectos de transición entre las diferentes pantallas invocadas, cuando se abre una pantalla y cuando se cierra.

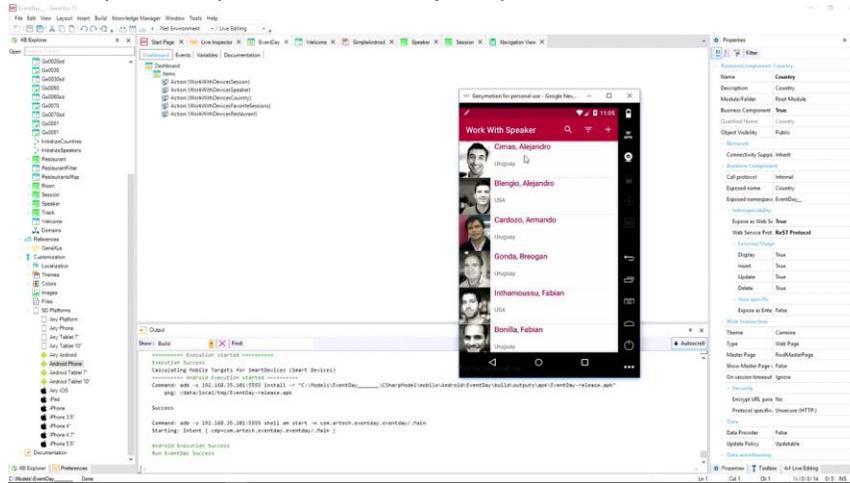
Vamos a verlo directamente en el emulador y en GeneXus.



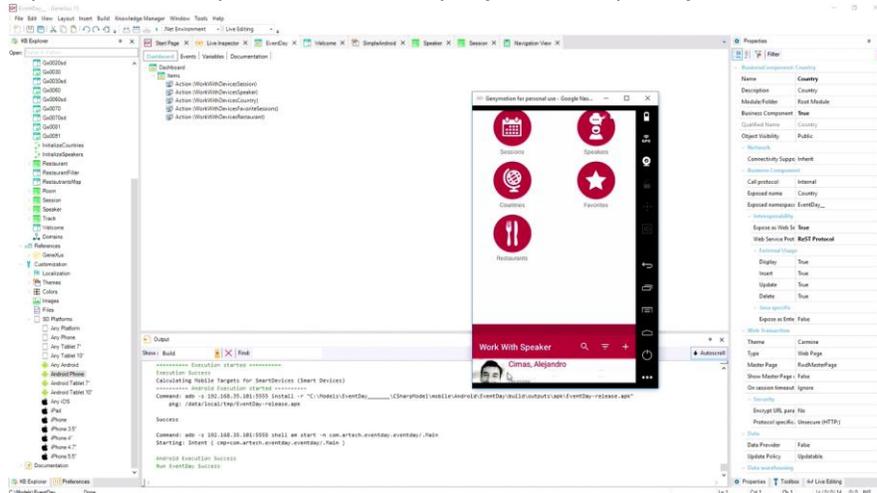
Vamos a ver qué sucede cuando hago "tap" sobre la opción de Speakers...



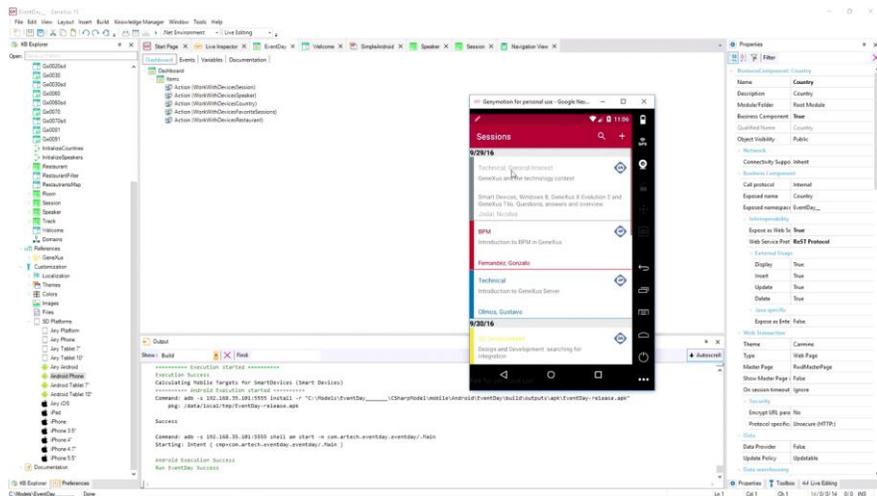
Vemos que está apareciendo de abajo, la pantalla del list



...y cuando me voy, cuando la cierro y bajo, vemos que baja, tiene ese efecto de transición.

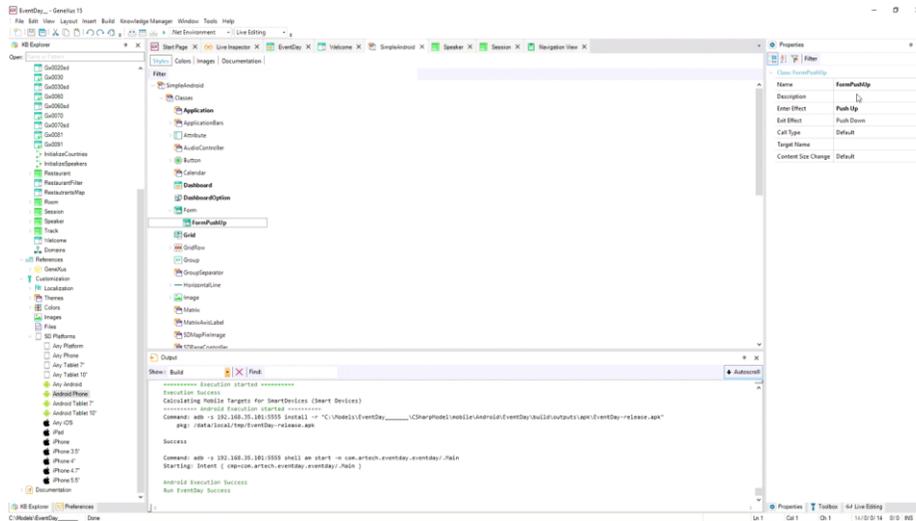


En cambio si abro las Sessions, no vemos ese efecto, no?



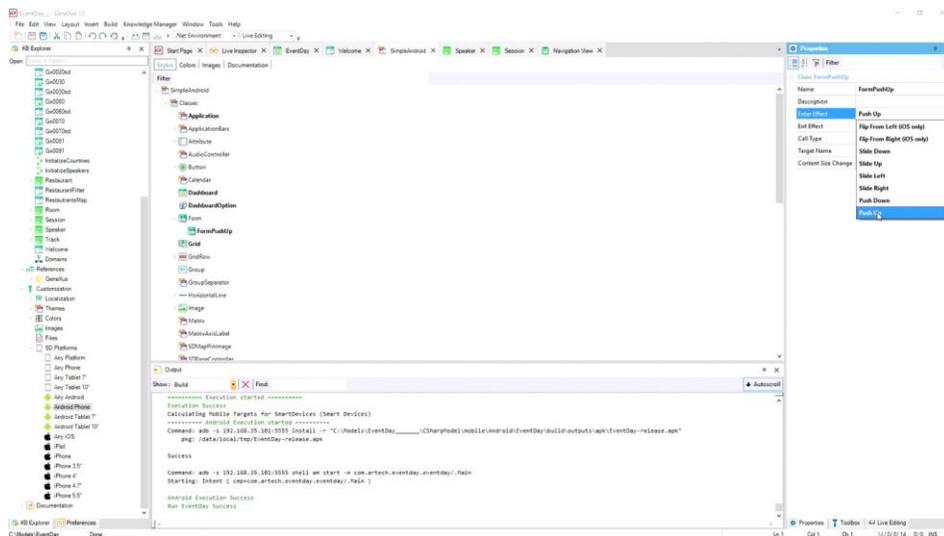
¿Cómo programé eso? ¿Cómo logré ese comportamiento?

Voy al Theme, SimpleAndroid en nuestro caso, y veamos que creé, bajo la clase Form, una subclase de esta, que llamé FormPushUp.



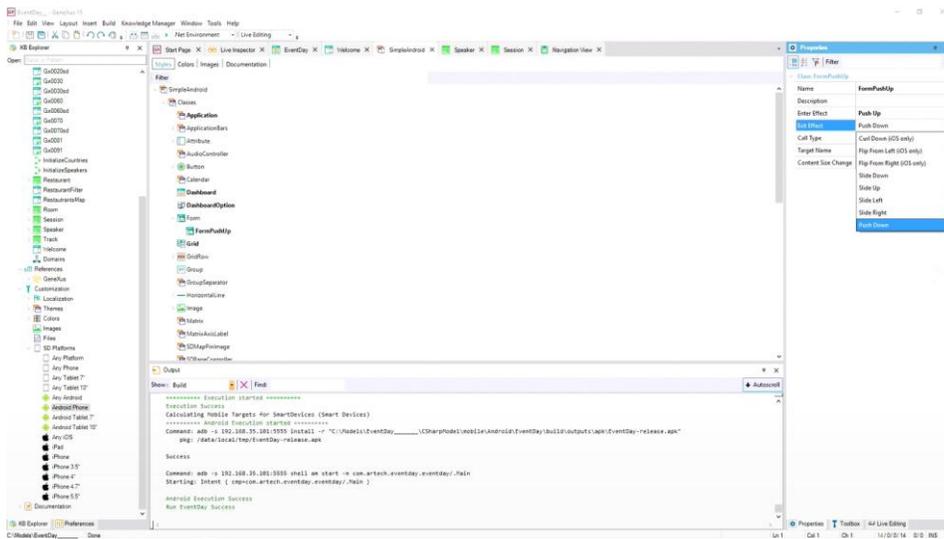
Si observamos, entre las propiedades de la clase Form, en este caso de esta subclase, tenemos las propiedades Enter Effect y Exit Effect. Y aquí es donde programamos cómo queremos que sea el efecto de apertura de la pantalla invocada, del form que contenga esta clase, y de cierre.

Aquí es donde yo coloqué Push Up, vemos que tenemos todas estas posibilidades...



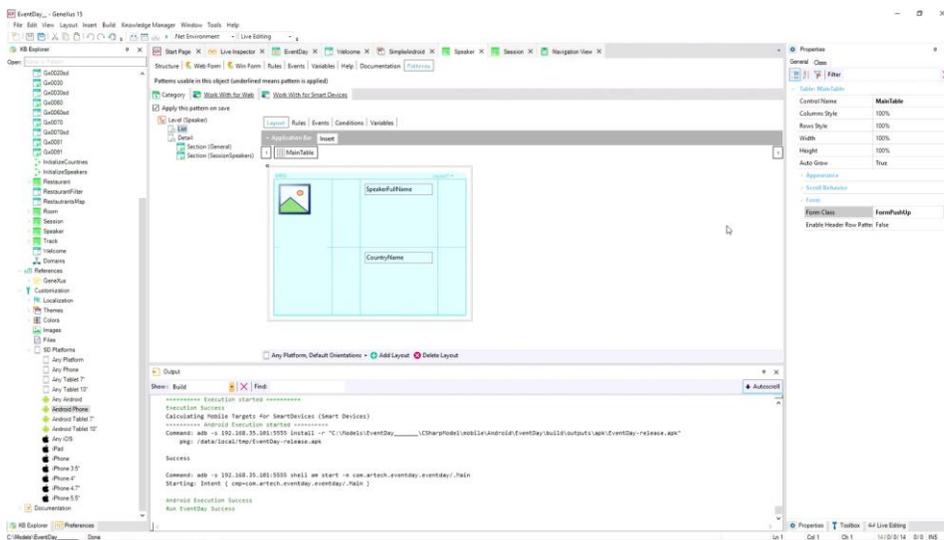
Por defecto está la opción Default, que va a depender de cada plataforma... algunas son solo válidas para iOS, otras valen para todas las plataformas... SlideLeft, bueno, etc.

Cuando seleccionamos una, la opción de Exit automáticamente se coloca a la opción contraria, el efecto contrario, acá sería Push Down.



¿Y cómo hice entonces para que el list de Speakers asumiera estos efectos de entrada y salida?

Bueno, simplemente yendo al list, y en la Main Table, configuré en la clase Form esta nueva clase:

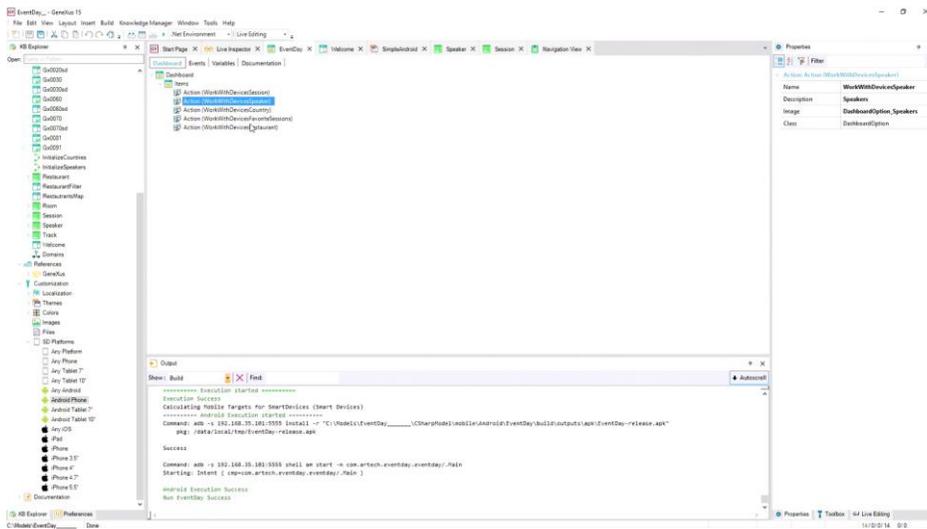


Y con esto conseguí el efecto deseado.

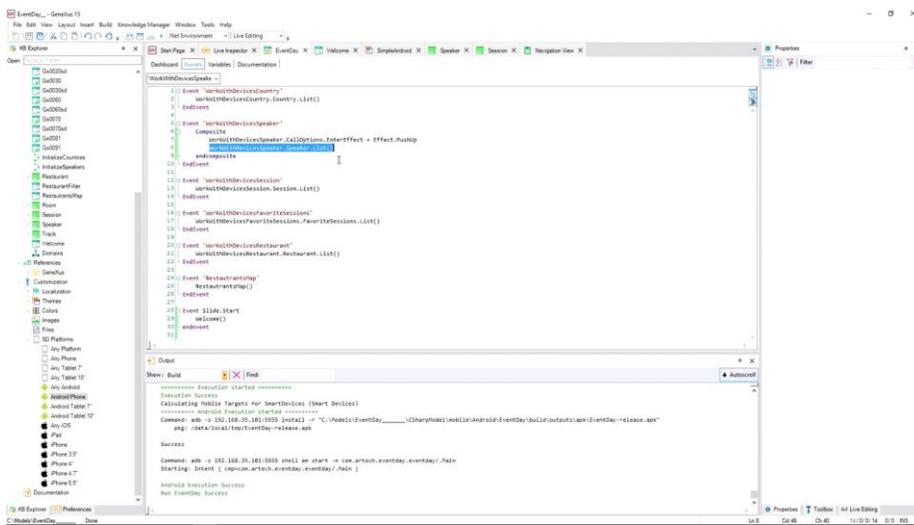
Esto lo hacemos entonces en forma estática, es decir, como lo configuramos aquí, toda vez que cualquier objeto llame a este otro se va a abrir y cerrar de esa manera.

Pero podríamos hacerlo de manera especial, dinámica, es decir donde programemos la invocación; es decir que en ese caso solamente, en esa invocación se despliegue de esta manera, y en otras no.

Bueno, eso lo podemos hacer entonces, por ejemplo, si vamos al Dashboard...

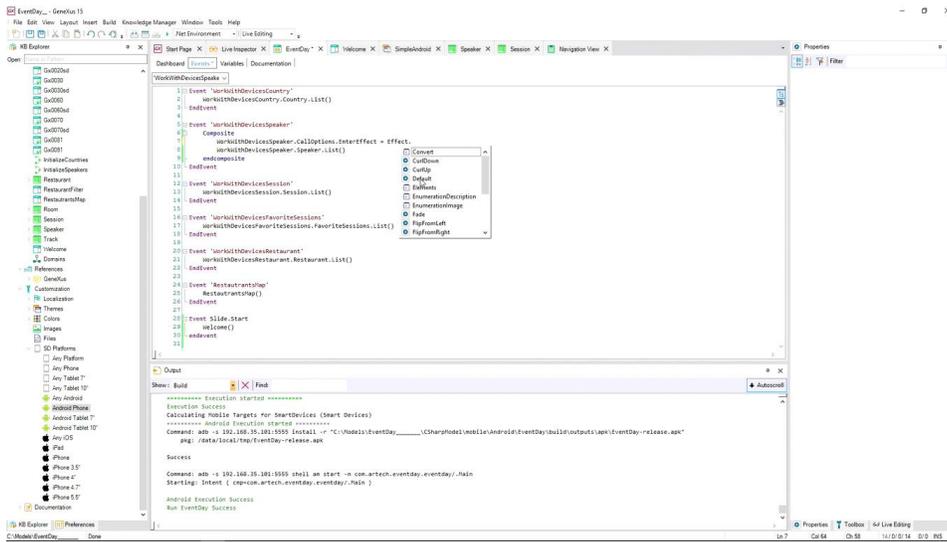


yo ya lo dejé programado, si vamos a los eventos, al evento asociado al tab del Work With Devices Speakers... si voy aquí, vemos que agregué, además de la invocación (esto era lo que estaba por defecto), en las Call Options, que las vamos a ver más adelante, el Enter Effect como Effect.PushUp:

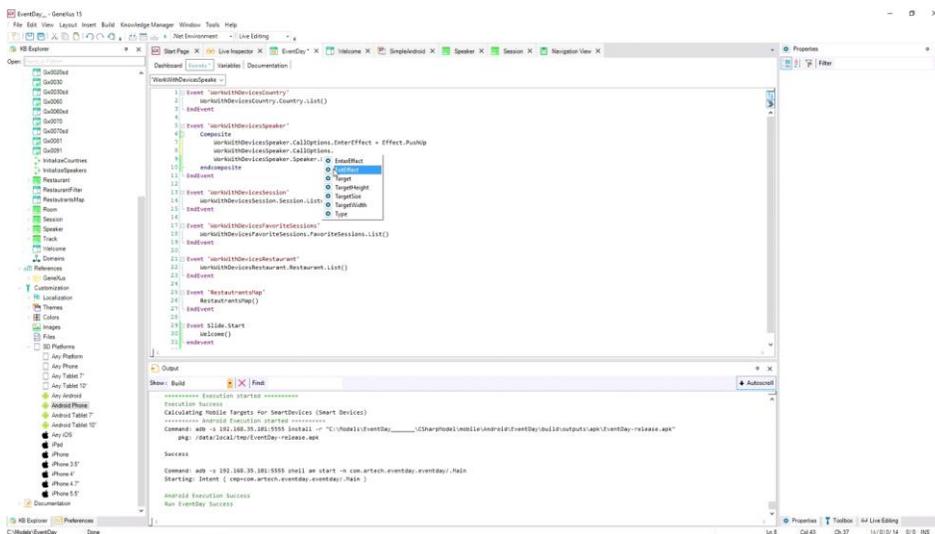


¿Qué es Effect?

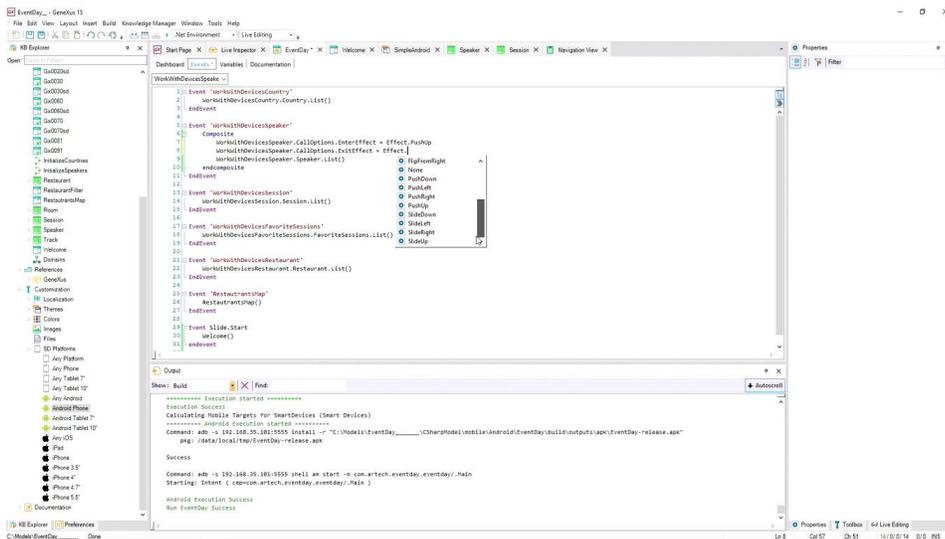
Es un dominio predefinido que viene con GeneXus; vamos a borrar esto, vamos a ver que si pongo punto ("."), ya me muestra todas las posibilidades, los efectos de entrada que puedo elegir, que son propios de ese dominio:



Bueno, habíamos elegido PushUp. No tenemos porqué, podríamos poner el evento, por ejemplo, podríamos escribir “WorkWithDeviceSpeaker” seguido de un punto, y vemos que podemos directamente llamar al nivel del Work With, que era lo que hacíamos hasta ahora, o podemos usar la opción Call Options. Y ahí entre las Call Options tenemos unas cuantas...



...podríamos programar también el efecto de salida. No tuvimos que hacerlo porque en realidad, ya vemos que por defecto tiene el contrario. Pero podríamos acá haber programado directamente el efecto contrario, PushDown, u otro, que tiene que ser consistente con el que escribamos como efecto de entrada:



Bueno, esto del comando Composite después vamos a ver cuándo es necesario escribirlo, por ahora no nos vamos a preocupar de esto.

Así que esta es la otra manera entonces de poder programar esos efectos de transición entre las diferentes pantallas. Con esto entonces terminamos este tema y vamos a pasar al siguiente.

