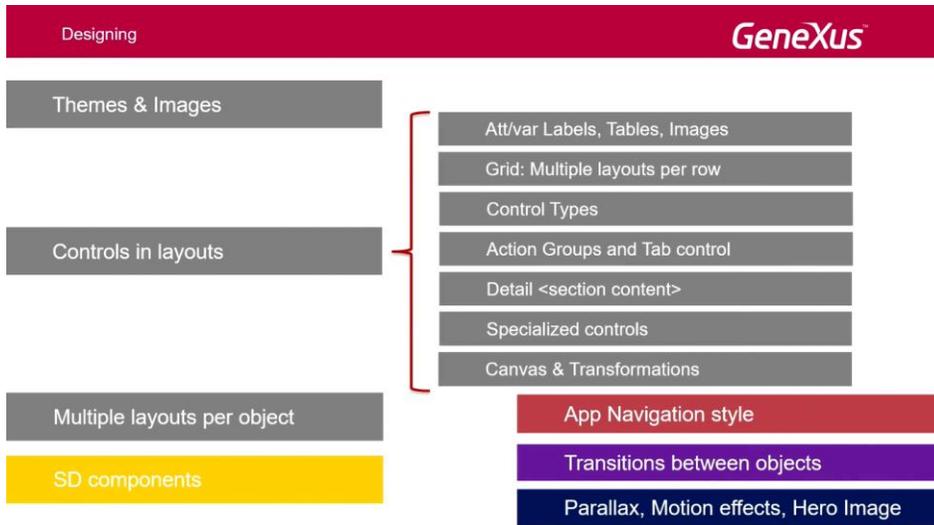


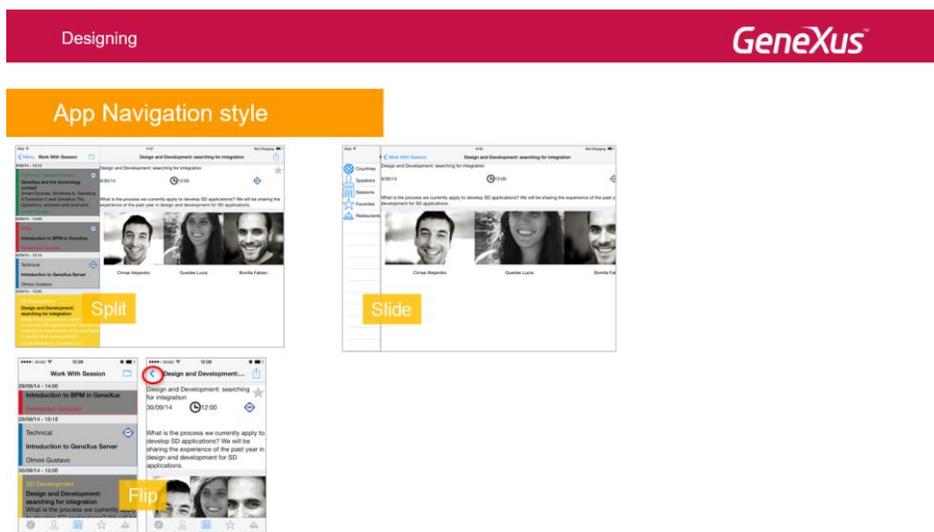
## App Navigation Style



Veremos los estilos de navegación posibles, entre el menú principal y las pantallas que se van invocando entre sí, y la forma de configurar el estilo deseado.

Dependiendo del tamaño de pantalla del dispositivo, podríamos querer mostrar la información de List y Detail a pantalla partida (**Split**), o no (**Flip**). En el caso de Flip, el List ocupará toda la pantalla y cuando desde él se llame al Detail, en este caso de las sesiones, se va a cargar ese Detail sobre la misma pantalla, de manera que se va a poder volver al List con el botón de back.

Además, podríamos querer que, independientemente de cuán profundo estemos en el nivel de invocaciones (desde el Detail, por ejemplo, podríamos hacer tap sobre la imagen de uno de los oradores y entonces abrir el Detail del orador, y así sucesivamente; nos podríamos ir metiendo en profundidad en sucesivas pantallas), tener siempre accesible el menú principal (en nuestro caso el dashboard); eso se conoce como estilo de navegación **Slide**. De manera tal que va a poder aparecer siempre ese menú principal desplegable desde la izquierda o con un botón de hamburguesa, para tenerlo siempre accesible y entonces cortar la profundidad de las invocaciones, y agarrar por otra rama de ese árbol.



Tanto en Split como en Slide, tenemos las regiones Left y Content en que se va a dividir la pantalla en forma default. En el caso de Slide, la sección Left va a ser la ocupada por el menú que se va a ocultar y a desplegar a pedido del usuario; en nuestro caso va a ser el dashboard el que va a ocupar esa posición Left.

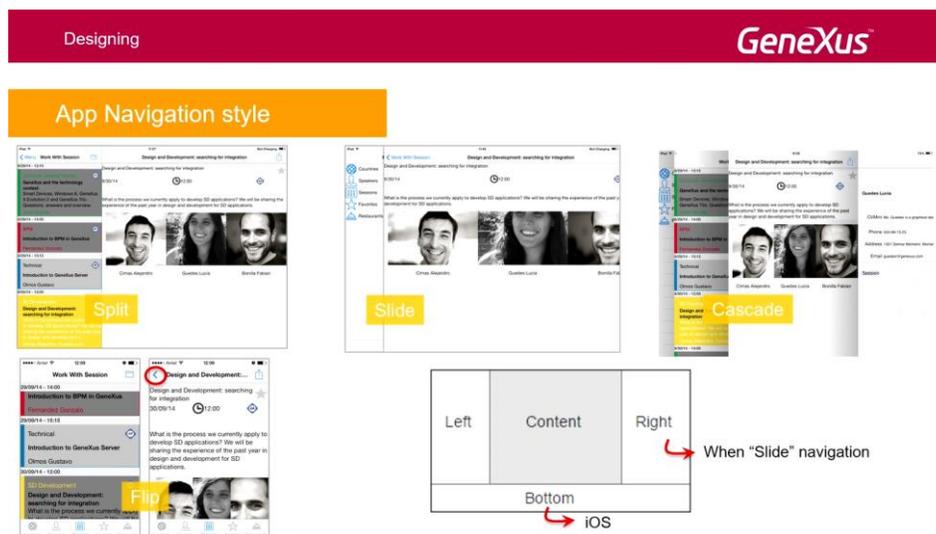
Lo importante es que un panel invocado desde la región Left va a ser desplegado en la región Content (reemplazando en el stack de invocaciones al objeto que estuviera allí antes cargado en esa ventana de Content, por lo que no habrá un back). Por otro lado, un panel llamado desde la región Content será desplegado en esa misma región (sin reemplazar en el stack de invocaciones al objeto que estuviera allí antes, por cual vamos a poder volver al objeto anterior; se va a agregar esa acción de back arriba a la izquierda, en la application bar).

Además, cuando el estilo de navegación es "Slide", se puede habilitar una región "Right". Esto ya no es default; se consigue haciendo uso de una api implementada como un External Object en GeneXus, de nombre Navigation, que vamos a ver más adelante cuando estudiemos las opciones de invocación, dentro del tema "comportamiento".

Y si estamos en iOS, en cualquier caso se puede habilitar una región Bottom.

Otra opción de navegación además de las tres anteriores (Split, Flip y Slide) será ir abriendo en cascada regiones a la derecha; podemos tener hasta tres niveles (**Cascade**).

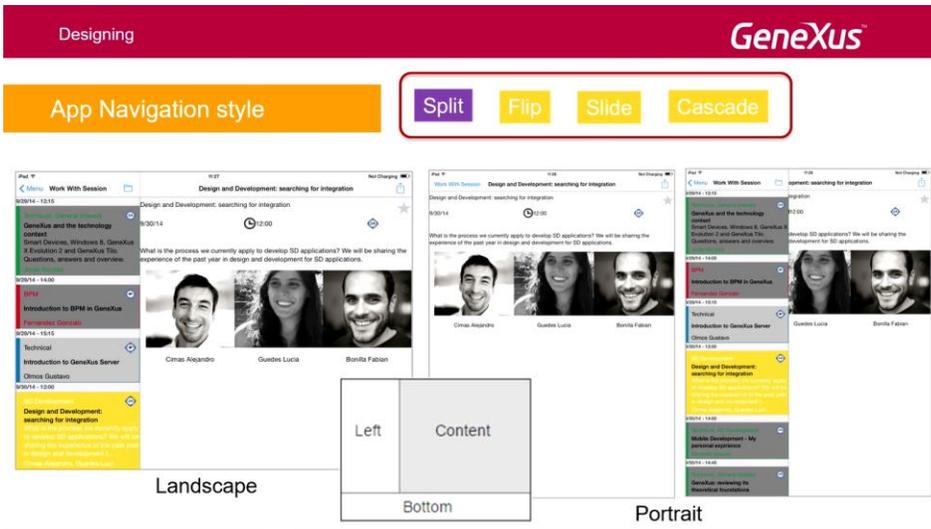
Vamos a ver brevemente cada una de estas cuatro opciones de navegación, y vamos a ver una pequeña demo en GeneXus.



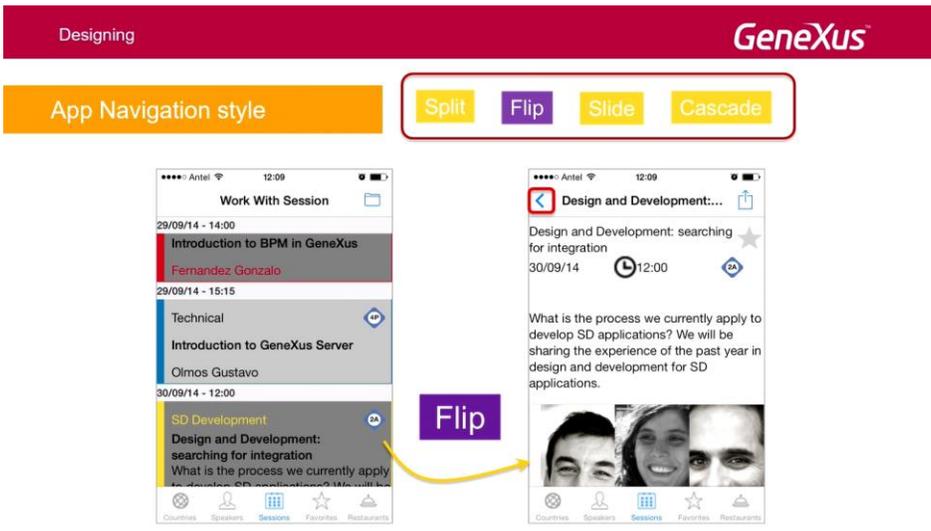
Por ejemplo, en el caso de Split, podemos ver un ejemplo con iPad, en ambas orientaciones (landscape y portrait). El valor default para tabletas es Split, lo que significa que siempre en landscape, es decir en orientación horizontal, se va a mostrar la pantalla partida; mientras que en orientación portrait, vertical, la pantalla de la izquierda se va a ocultar, pero el usuario podrá hacerla aparecer deslizando el dedo del lado izquierdo hacia el derecho o con la acción de la application bar (arriba a la izquierda).

La característica de este estilo de navegación es que todos los List de los Work With se van a abrir en el frame, en la región, de la izquierda, mientras que los Details se abrirán en el frame (más grande) de la derecha, en el "content". Estas opciones por defecto (dónde y cómo se abre el objeto invocado) pueden modificarse a través de las propiedades CallOptions de las invocaciones a objetos, como vamos a ver más adelante cuando veamos

el comportamiento; pero ya lo dejamos apuntado desde acá.



Bien. Por otro lado, en iPhone la opción por defecto para el estilo de navegación será Flip, lo que significa que siempre se mostrará un único layout; cada vez el objeto invocado se va a mostrar en ese único layout, el contenido entonces es único. Así, si se invoca al List de Sessions, se muestra íntegro en la pantalla, y si desde allí se quiere invocar al Detail, se abre entero en esta pantalla, ocupando toda la pantalla, y para poder ver el List, se va a contar con el back arriba a la izquierda, o con el gesto dactil de mover hacia la derecha el borde izquierdo, como siempre.



¿Qué pasa con el Slide, que mencionábamos hace un ratito?

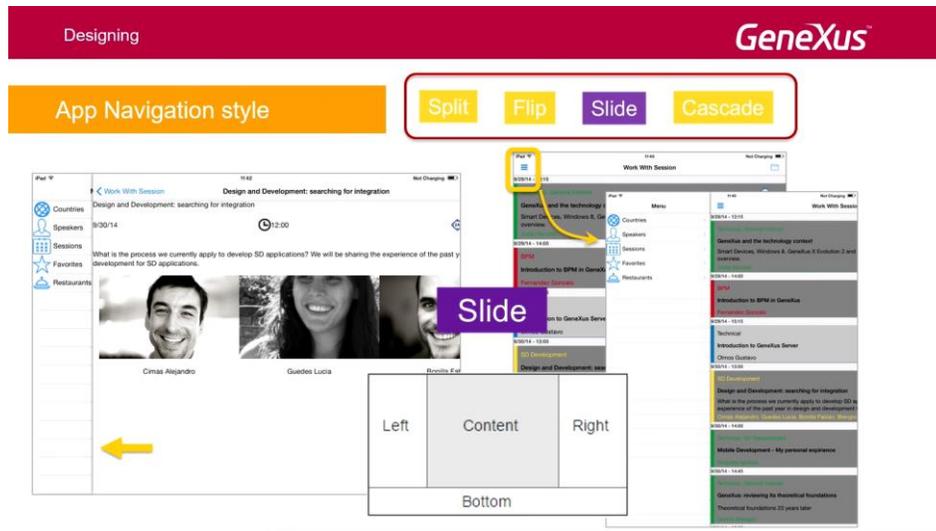
Este estilo de navegación va a permitir entonces tener siempre disponible el menú principal, no importa en qué nivel de profundidad nos encontremos, de modo de poder tomar por otro lado del árbol con un solo paso, ya sea deslizando con el dedo desde la izquierda o con el botón de hamburguesa, como vemos en las imágenes.

En la pantalla de la izquierda podemos ver la región Content, que tiene cargado el Detail de una Session, y tenemos el menú. ¿En qué momento estamos? Bueno, evidentemente el usuario presionó el Sessions, se cargó en la sección de Content el List de Sessions, y luego eligió una sesión, una conferencia determinada, y se abrió

el Detail de esa sesión. Y vemos que está apareciendo la posibilidad de volver al que había llamado a este. Entonces vemos cómo en esta pantalla se van a ir cargando los distintos objetos que se vayan llamando, y la manera entonces de cortar con todo ese árbol es, por ejemplo, elegir ver los Restaurants, y ahí se va a resetear todo lo que aparecía acá, y se va a mostrar el List de Restaurants, sin ningún back porque va a ser el invocado directamente desde el menú.

Podríamos, asimismo, querer abrir un objeto invocado en una región a la derecha del Content. Eso se logrará programáticamente, con las opciones de invocación y la api Navigation que habíamos mencionado, que viene con GeneXus.

Lo mismo para la región Bottom, que sólo válida para iOS; lo dejo apuntado otra vez, esto se verá más adelante.



Y bueno, tenemos también la opción Cascade, que vale tanto para tabletas como para phones. El menú principal se despliega en el área de la izquierda, el panel actual en el área derecha, y el que llamó a este último, en el medio.

En el ejemplo de la izquierda, vemos cómo desde el dashboard se llamó al List de Sessions, y desde éste se invocó al Detail de una session. Si luego, desde este Detail, nos metemos un nivel más en profundidad, por ejemplo haciendo tap sobre un speaker del grid, se abre una cuarta área a la derecha, con esa información. Si a partir de ahí se invoca a otro objeto (desde el nivel de profundidad 3 en adelante) se va a abrir en esta misma pantalla, la cuarta, respetando el stack; y ahí sí va a ir apareciendo entonces el botón de back, por cada pantalla nueva que se abra, para volver a la que la llamó.

Cuando se selecciona otro elemento desde el menú principal (el dashboard) los paneles de navegación en cascada son reseteados; es decir, los paneles 3 y 4 son eliminados y se vuelve a empezar.

## App Navigation style

Split

Flip

Slide

Cascade



Esas son entonces las posibilidades de los estilos de navegación.

¿Dónde definimos cuál es el estilo de navegación de nuestra aplicación?

Se configura a nivel de plataforma. En el ejemplo, estamos a punto de especificar que para cualquier dispositivo iOS, el estilo de navegación de la aplicación va a ser Cascade, y eso vemos que es la propiedad **Navigation Style** a nivel de la plataforma.

Si queremos, por ejemplo, que esa sea la opción por defecto para todos los iOS, salvo para los iPhones, alcanza con ir a las propiedades del iPhone (en general o de alguno de los tamaños particulares), y en las propiedades del iPhone poner como Navigation Style el deseado, por ejemplo, Slide. Y ahí el nivel de precedencias de las propiedades observemos que es el usual; es decir, va a valer la más cercana a la plataforma que estamos eligiendo, y sino la del padre.

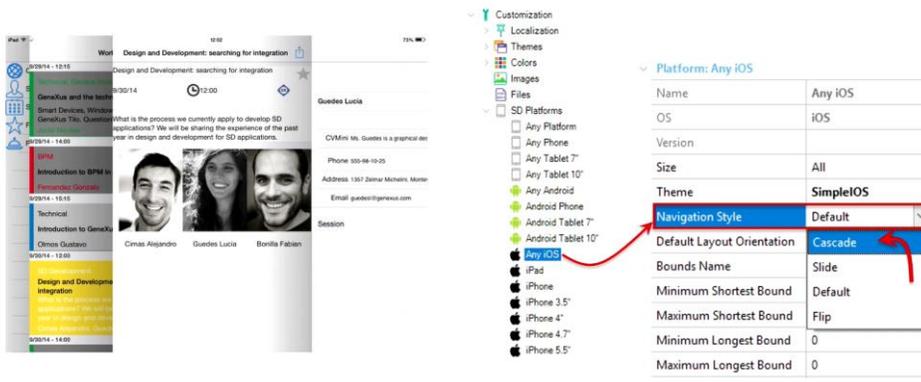
## App Navigation style

Split

Flip

Slide

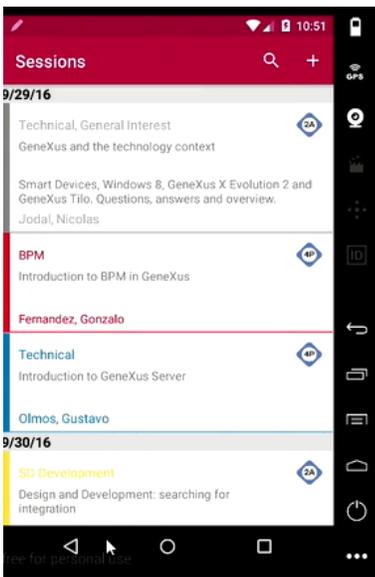
Cascade



Ahora vamos a hacer una pequeña demo en GeneXus; voy a abrir el emulador... aquí vemos que estamos con el Dashboard... estamos en el emulador, pero sería como un teléfono Android... y vemos que el estilo de navegación por defecto de nuestra aplicación para un teléfono Android es Flip: se cargan las pantallas en la única pantalla que tenemos.

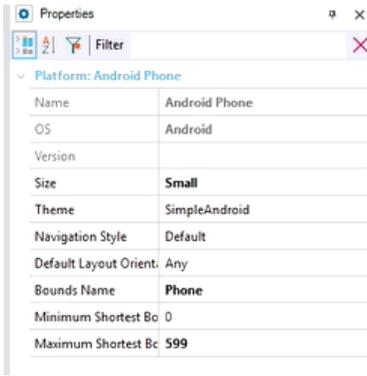


Por ejemplo si voy a las Sessions...se carga en toda la pantalla.

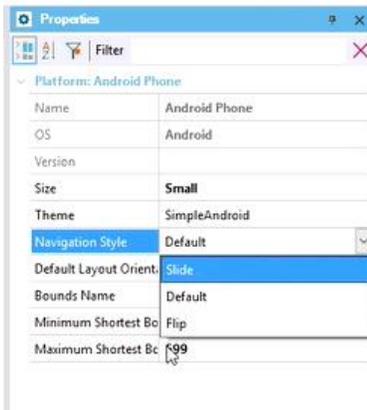


Vamos a querer que el estilo de navegación sea Slide, o sea que este menú esté siempre disponible desde la izquierda.

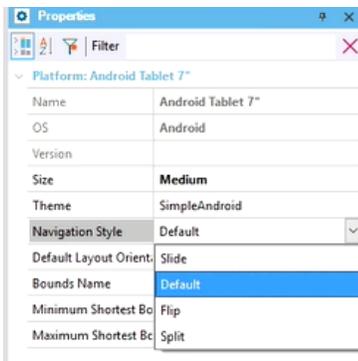
Vamos a ir a las propiedades de las plataformas, y vamos a observar que para Any Android, el Navigation Style dice Default. Bueno, si vamos a Android Phone en particular, vemos que también tiene el Default, está heredando el Default. Y el default para un teléfono decíamos que era Flip.



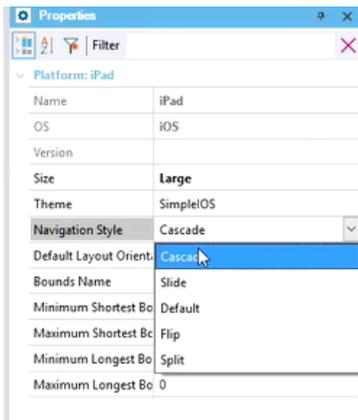
Entonces lo que podemos hacer, ya sea en Any Android o directamente en el Phone, es cambiar el estilo de navegación. Vean que acá aparecen sólo los disponibles para un teléfono Android, que son Slide o Flip. Vamos a cambiarlo a Slide.



Si vamos a Any Android, en cambio, vemos que los estilos de navegación que nos está indicando son Slide y Flip, porque tiene restricciones. Pero si vamos por ejemplo a una tableta Android, vemos que aparece Slid, Flip, y Split. Y si vamos a la tableta de 10" tiene esas mismas también.

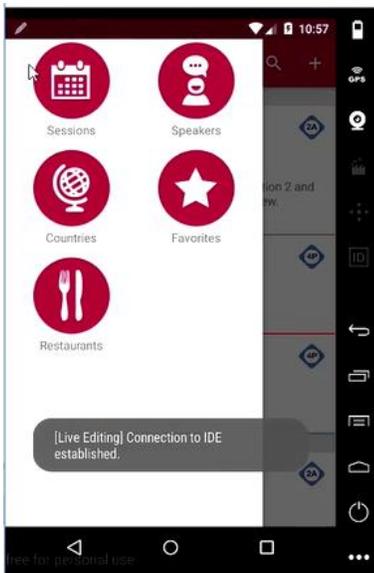


En cambio, si vamos por ejemplo a un iPad, vamos a ver aparecer el Cascade también.



Bien. Vamos a ejecutar. Botón derecho...Run... va a abrir el emulador...y vamos a ver qué sucede. Ya podemos ir pensando. Si el Dashboard se va a presentar como un menú deslizable, evidentemente va a tener que cargarse ya una pantalla, la del Content. ¿Qué contenido se va a cargar en esa región de Content? Bueno la respuesta la primera opción, se va a llamar al primer objeto de ese Dashboard. Entonces vamos a ver que va a aparecer el List de Sessions y el menú desplegable.

Y vemos cómo efectivamente levantó el List de Sessions... tenemos el menú de hamburguesa para mostrar entonces el menú desplegable. O podemos hacerlo directamente deslizándolo desde la izquierda.

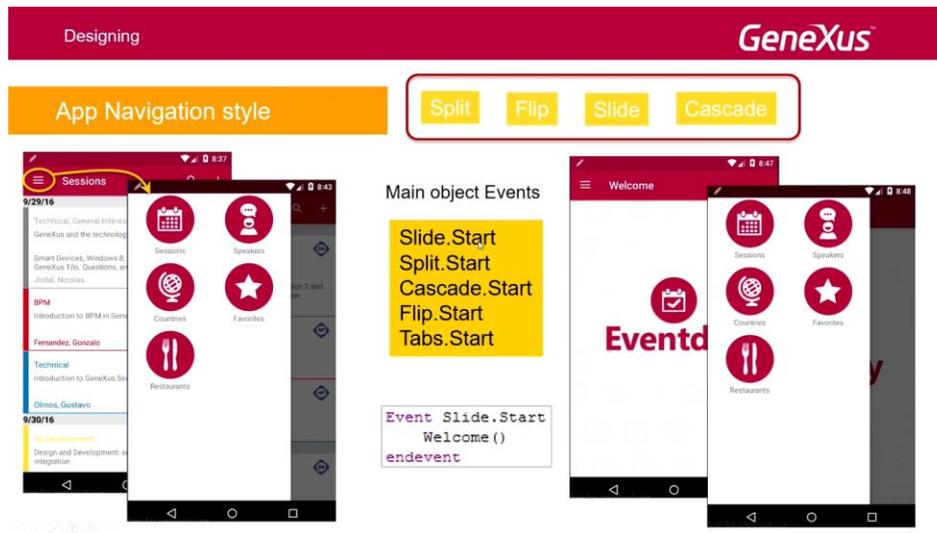


Podríamos querer no llamar a este List, sino invocar a otra pantalla cuando se está abriendo por primera vez la aplicación. ¿Cómo hacemos eso?

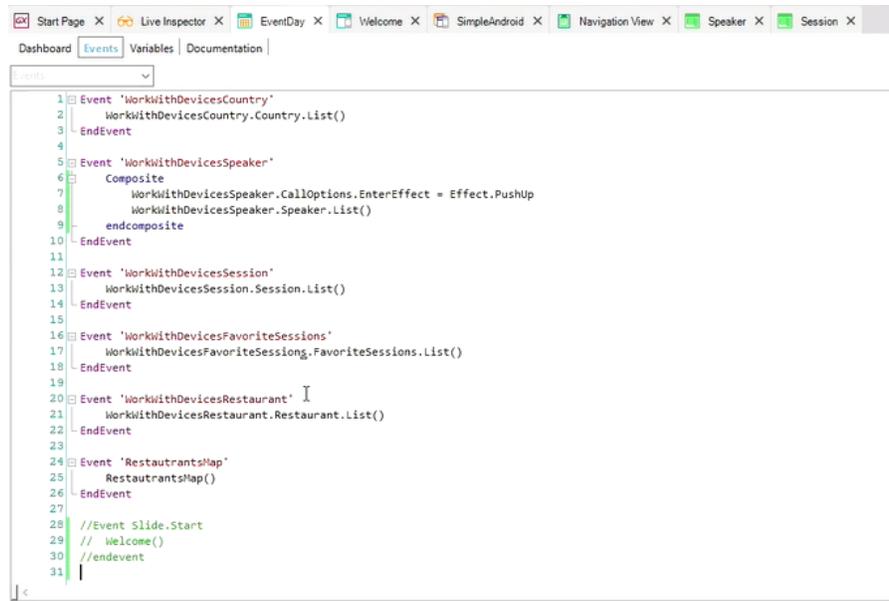
Tenemos ciertos eventos que podemos programar a nivel del objeto main, del objeto principal, que en nuestro caso es un Dashboard, pero podría ser un panel por ejemplo; que son estos que aparecen acá: Slide, Split, Cascade, Flip, y Tabs punto (".") Start. Es el evento Start asociado a los estilos de navegación.

Esto hace que entonces nosotros, por ejemplo, si programamos Slide.Start y la aplicación se está ejecutando para un iPad que tiene estilo de navegación Cascade, ese evento, en ese caso para esa plataforma, no se va a ejecutar; se va a ejecutar solamente cuando el estilo de navegación sea Slide.

Entonces, por ejemplo, podemos crearnos un Panel para Smart Devices “Welcome”, con una imagen, de manera tal que ese panel solamente se ejecute en el evento Start del estilo de navegación Slide; solamente en ese caso.

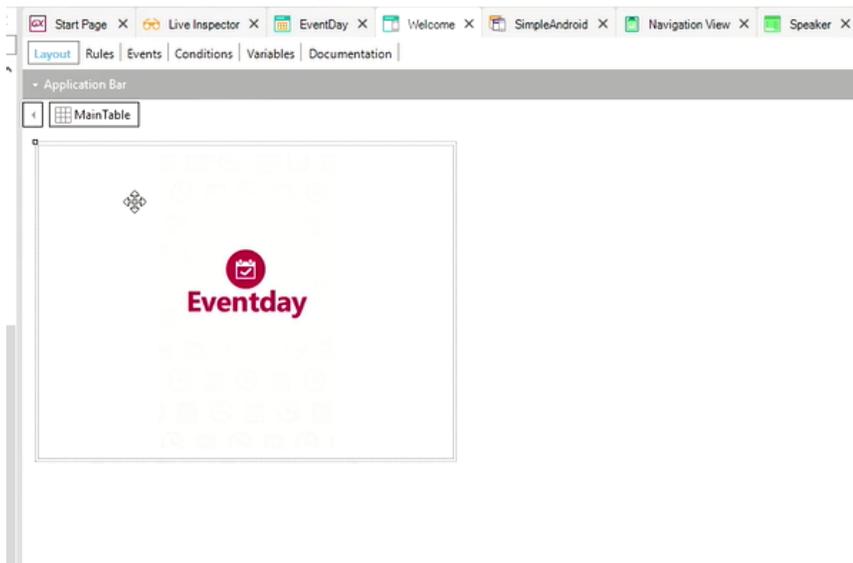


Venimos a los eventos de nuestro objeto principal, y vemos que lo dejé comentado porque yo ya lo había hecho...



ya me creé un Panel for Smart Devices de nombre Welcome; vemos que un Panel for Smart Devices es análogo a un Web Panel, y tiene todas las secciones esperables.

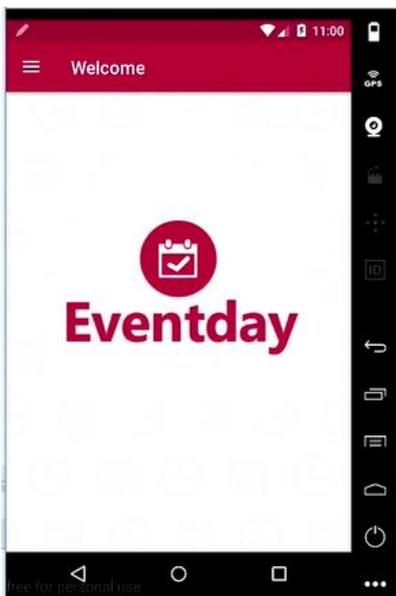
Entonces acá coloqué una imagen...



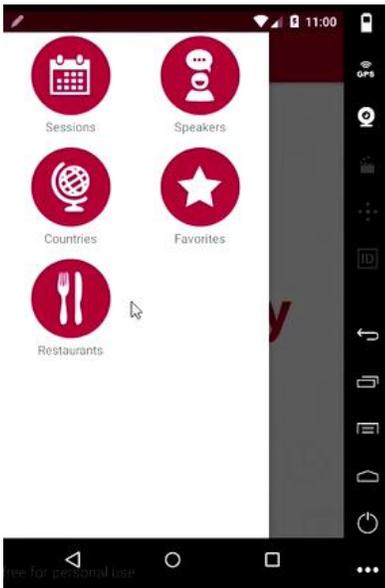
...y lo que vamos a hacer entonces es descomentar este evento, que veamos que lo que estoy haciendo simplemente es invocar a ese panel, cuando se trata de un estilo de navegación Slide y cuando este Dashboard es ejecutado por primera vez, en el Start.

```
27 |  
28 | Event Slide.Start  
29 |     Welcome();  
30 | -endevent  
31 |
```

Entonces, vamos a hacer Run... y ahí vemos como está funcionando como queríamos, como está invocando a ese panel.



Y acá tenemos el menú:



Con esto terminamos entonces este tema de los estilos de navegación, y vamos a pasar al siguiente tema, donde vamos a ver la forma de incorporar distintos efectos de transición entre las pantallas invocadas.

