

El modelo conceptual de las aplicaciones móviles



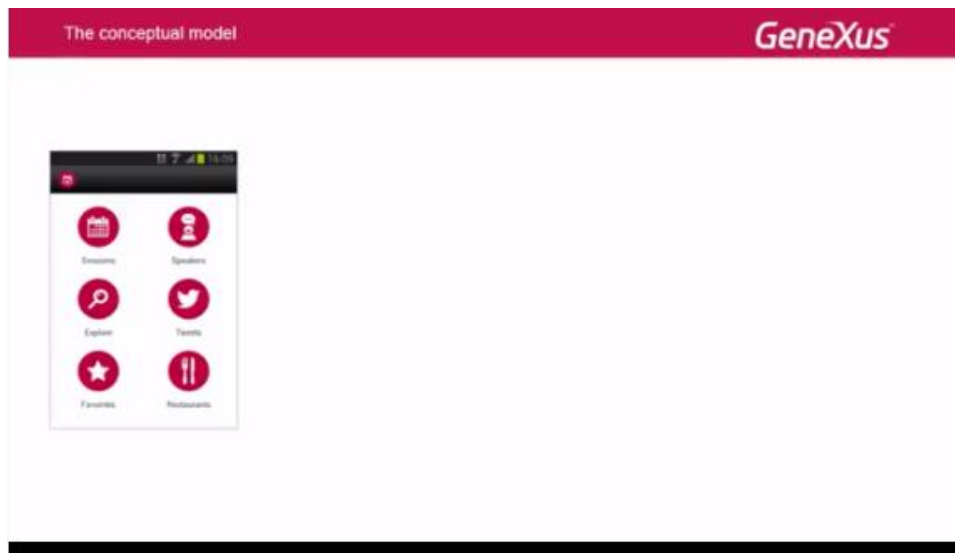
Conceptual model of mobile applications for Smart Devices

Cecilia Fernández | GeneXus Training

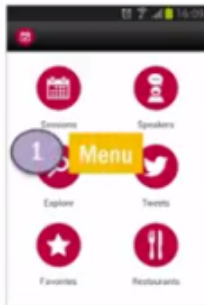
En videos anteriores, vimos la aplicación a la que queremos llegar y las características generales de las aplicaciones para Smart Devices.

Ahora nos introduciremos en el modelo conceptual correspondiente a estas aplicaciones.

Veremos los cuatro tipos de pantallas que se necesitan, y los objetos que las implementan, y sus relaciones.



Así, la primera pantalla que veremos es el menú.



Esencialmente contiene ítems que corresponden a las opciones que se le brindan al usuario.

La segunda pantalla:



es una Lista de elementos con los que trabajar. Suele corresponder al conocido “Trabajar con...” los datos de una transacción (en este caso “speakers”).

La manera de presentar la información de la lista es configurable.

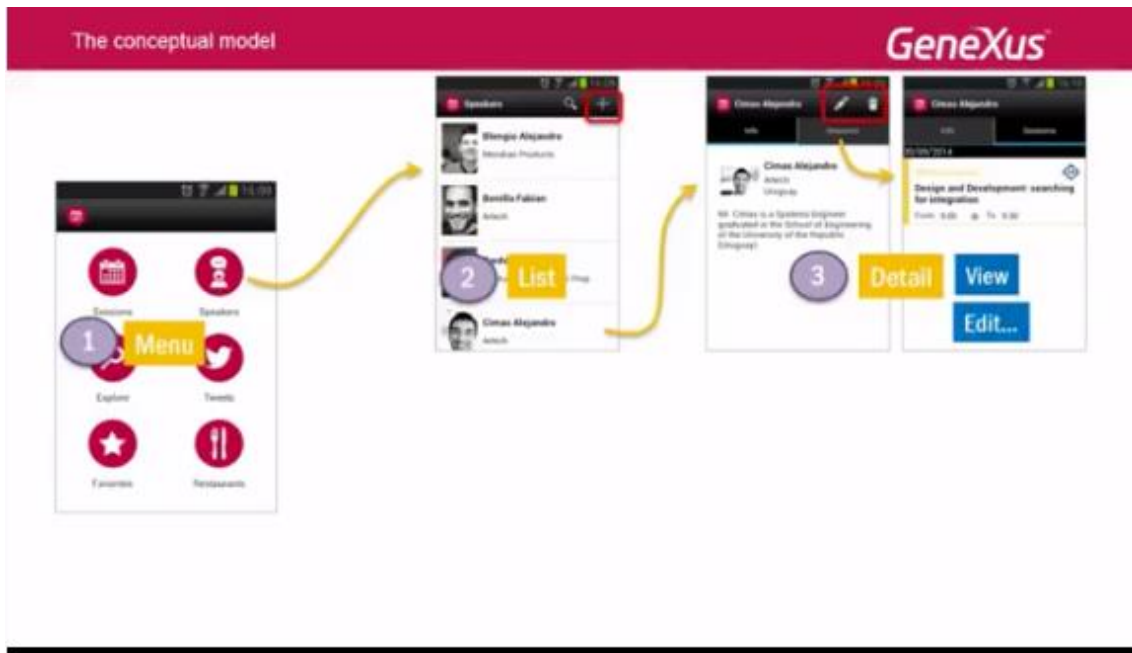
Lo esencial es que esta pantalla, en forma predeterminada, establece que al hacer tap sobre un elemento de la lista



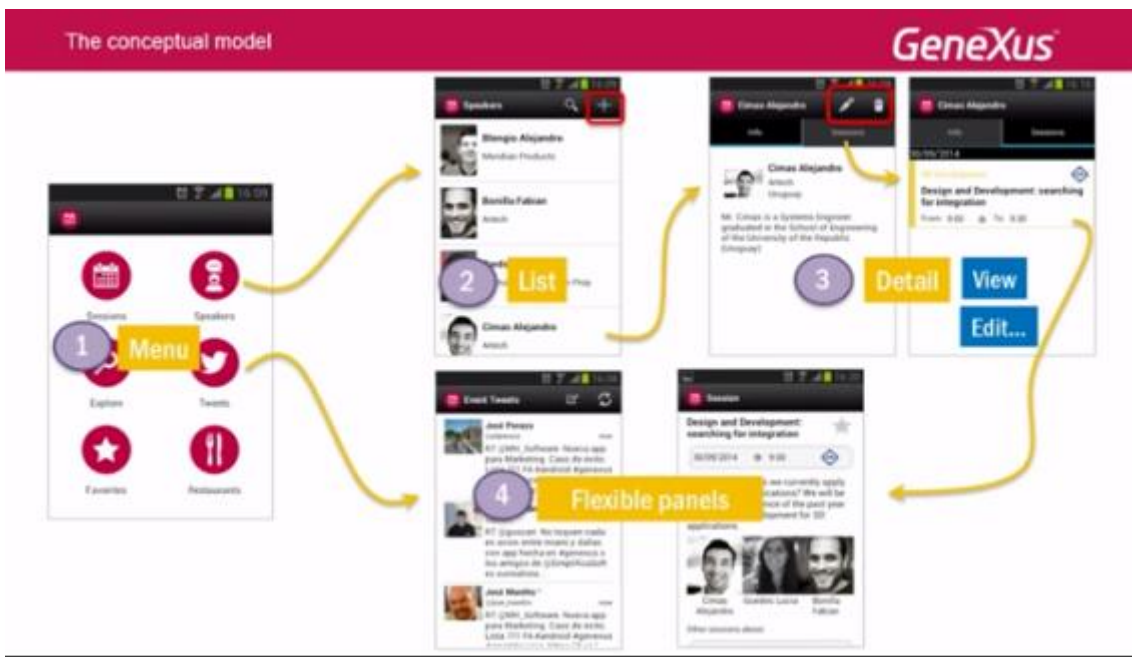
automáticamente se llame en modo View al tercer tipo de pantalla: la pantalla de Detail.



Si se quieren permitir las operaciones de CRUD, entonces también existirá en este punto un layout para edición (no lo estamos mostrando aquí).



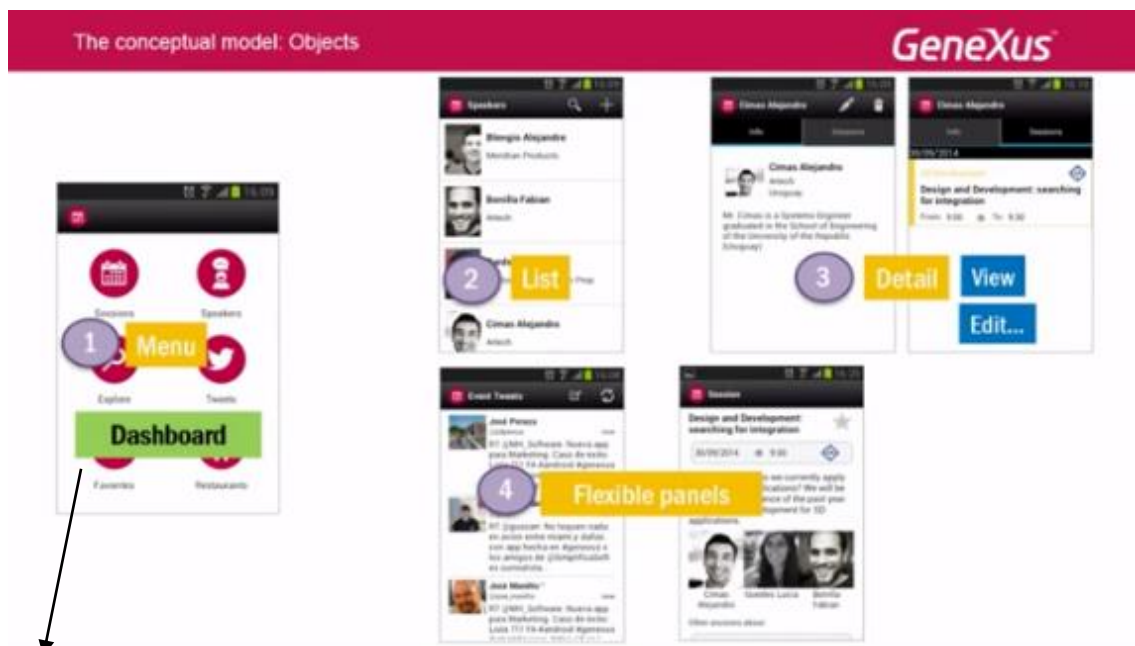
Por último tenemos los Paneles



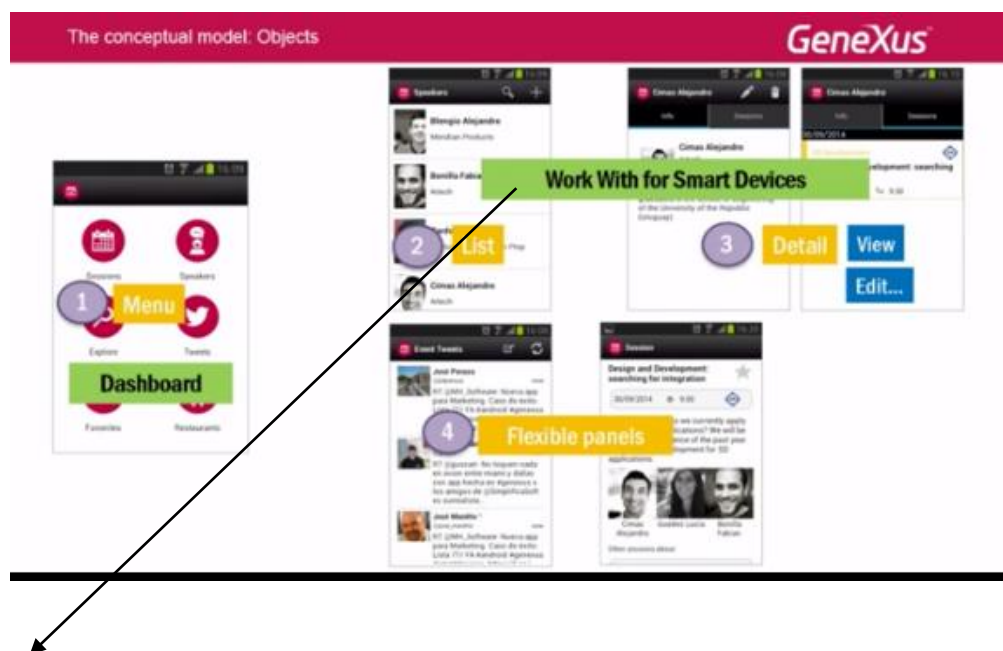
que proveen toda la libertad para diseñar el layout sin estructuración preestablecida. Todo lo implementa el desarrollador de cero. Por ejemplo, la pantalla de tweets, cuyos datos serán obtenidos a través de un web service.

Incluso el Detail de una conferencia, puede perfectamente implementarse como un panel.

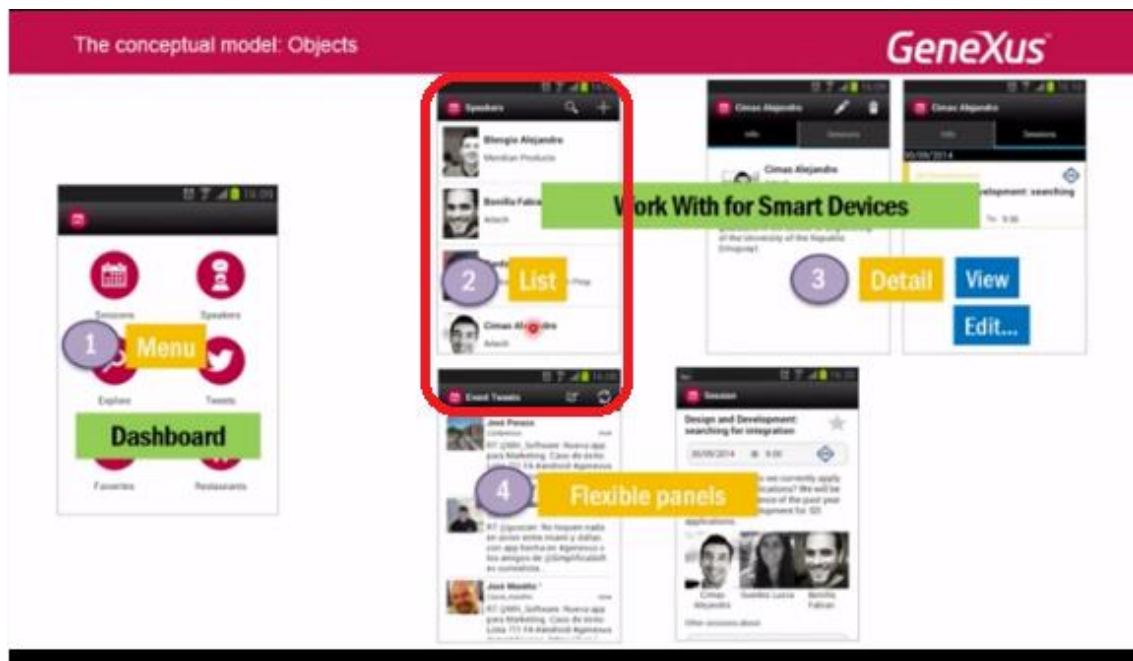
Para implementar estas pantallas y sus funcionalidades, tenemos tres objetos:



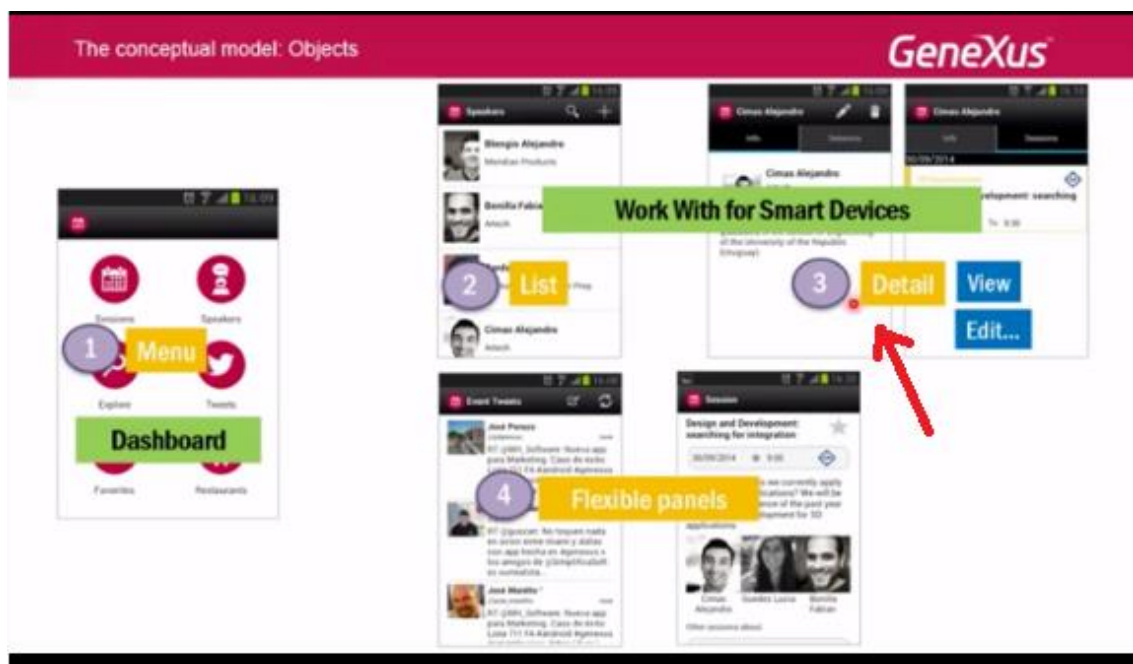
Dashboard: que implementa el menú.



El objeto “Work With for Smart Devices”: es tanto un objeto como un pattern, dos en uno. A diferencia del Work With for Web, no es un archivo de instancia a partir del cual se crean objetos que implementan el típico “trabajar con” los datos de una transacción, **sino que es un objeto que incluye en su interior**, esas especies de sub-objetos: los que implementan las pantallas de List

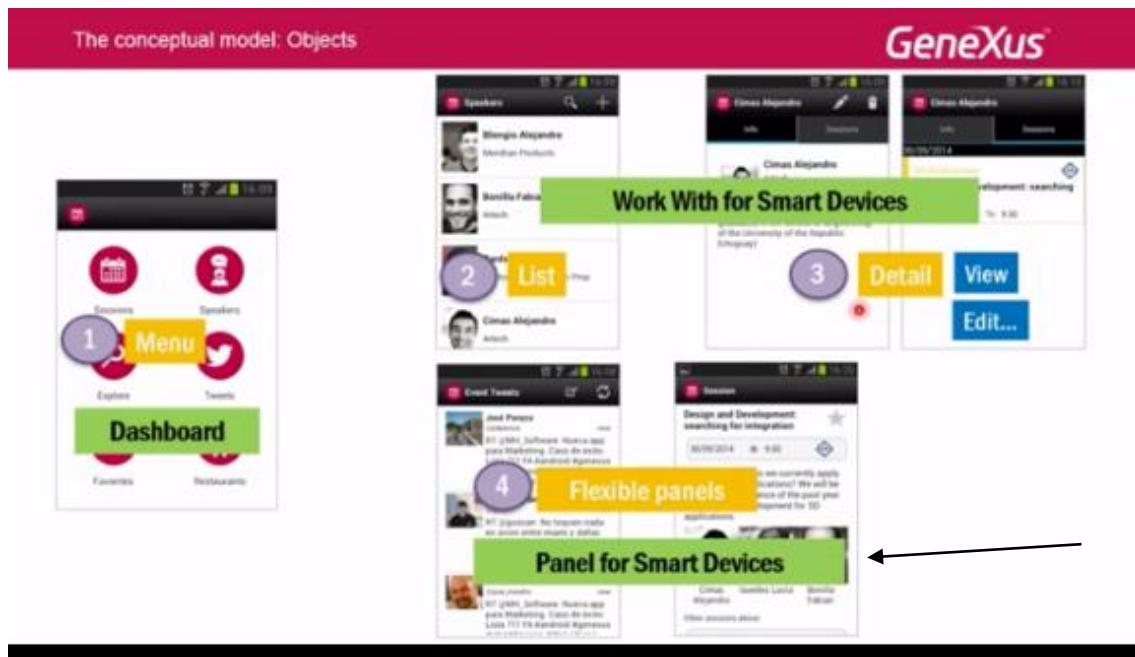


y de Detail (con todas sus secciones)



Generalmente este objeto estará asociado a una transacción, pero el hecho de que sea un objeto que puede crearse desde el diálogo de creación de objetos, le da la flexibilidad de poder crear todas sus pantallas (las de List y de Detail) de cero, cargando la información de donde se desee.

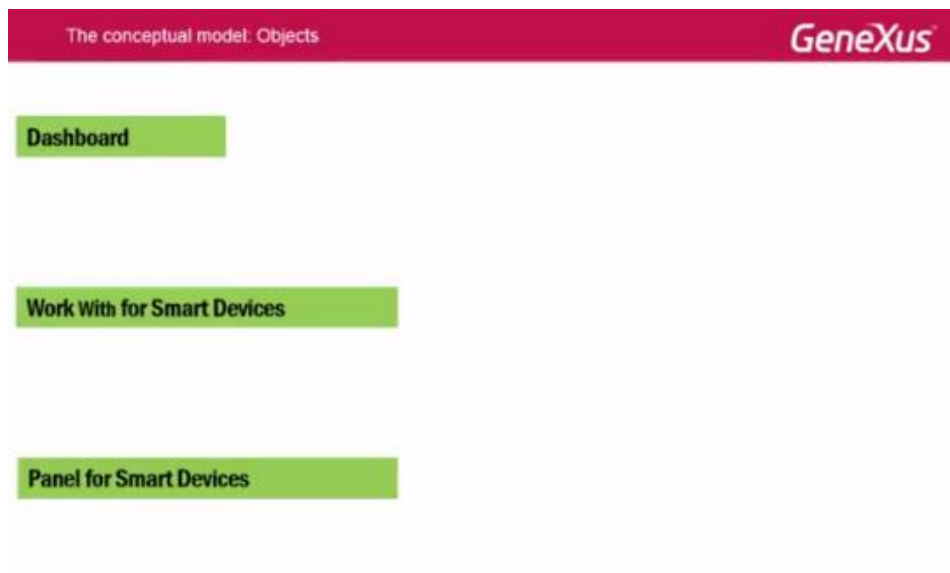
Luego, tenemos el objeto: **Panel for Smart Devices**



Es análogo al Web panel. También es análogo a cualquiera de las pantallas de List o de cada sección del Detail del Work With for Smart Devices.

Se crean de cero, por lo que el desarrollador tiene la absoluta libertad de colocar y quitar lo que desee, y cargar los datos de la base de datos o de cualquier otro lugar.

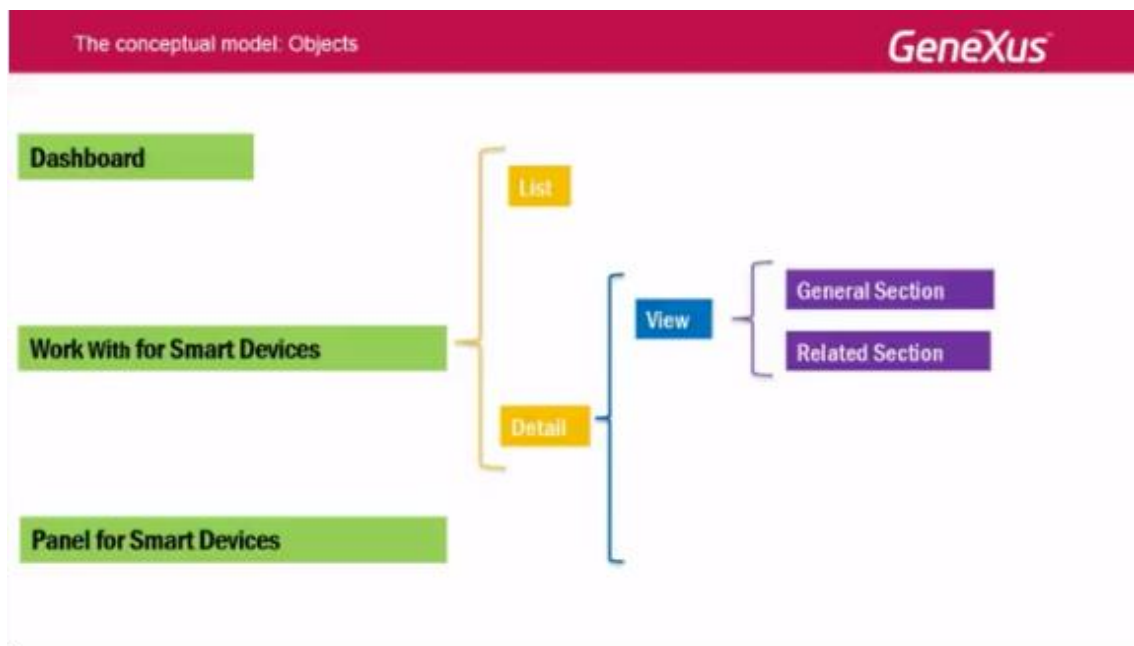
En resumen tenemos estos tres tipos de objetos



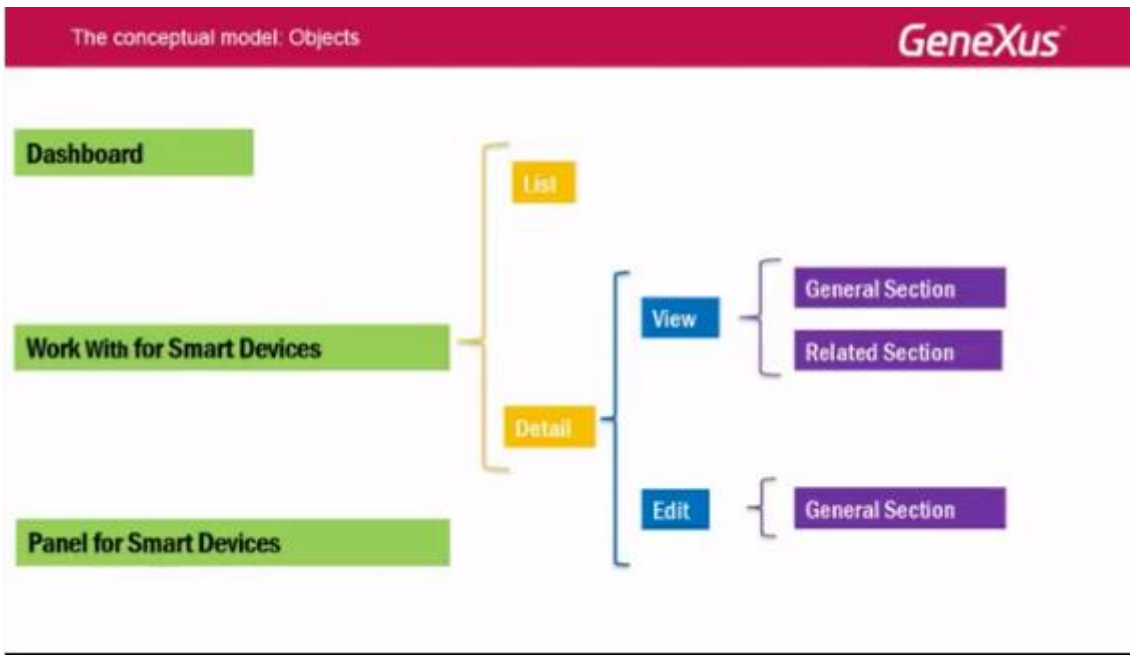
donde el Work With for Smart Devices encapsula el List y el Detail



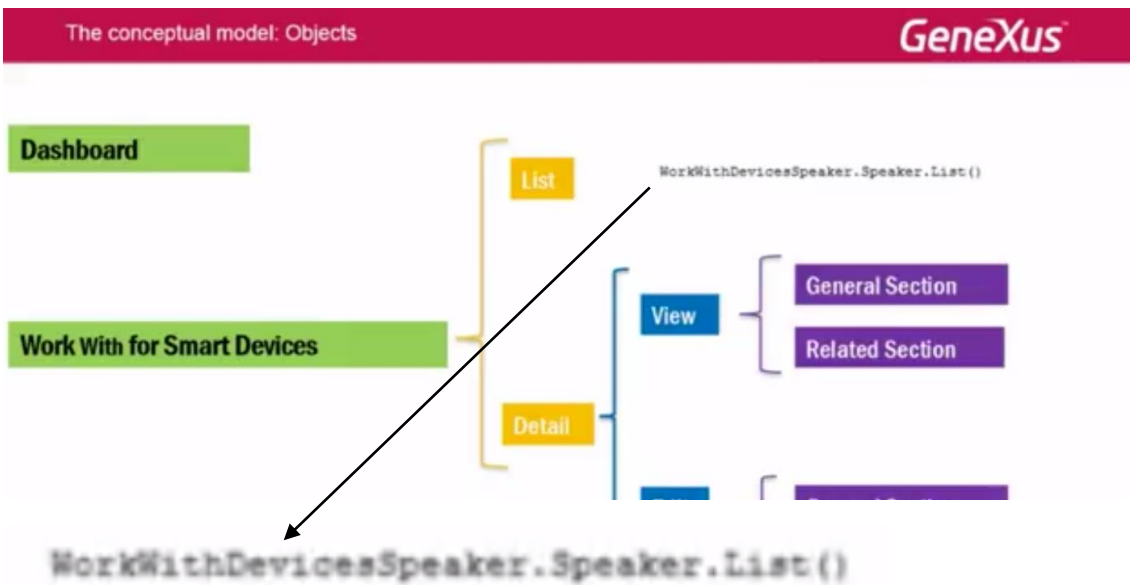
y a nivel de este último, las pantallas de visualización de la información general y relacionada



y la pantalla de edición de la información general para las operaciones de CRUD



Tomando como ejemplo el pattern Work With for Smart Devices aplicado a la transacción Speaker, la sintaxis para invocar **al List** será esta:

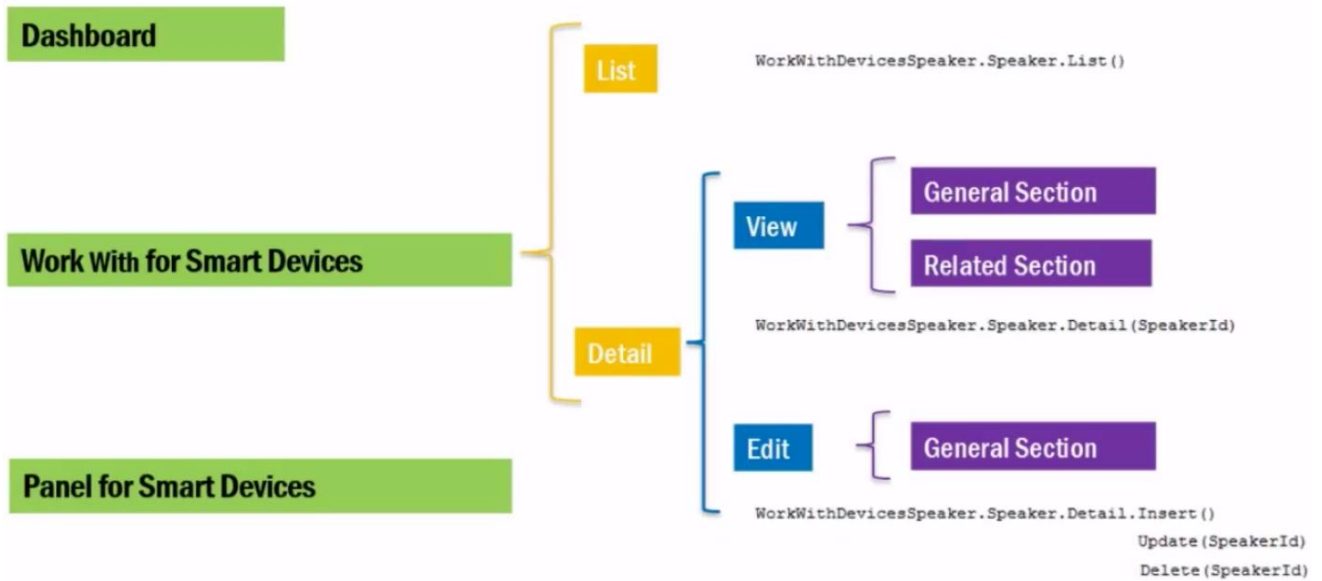


Para el Detail, el modo de visualización será esta:

`WorkWithDevicesSpeaker.Speaker.Detail(SpeakerId)`

Observemos que le tenemos que pasar la clave primaria..

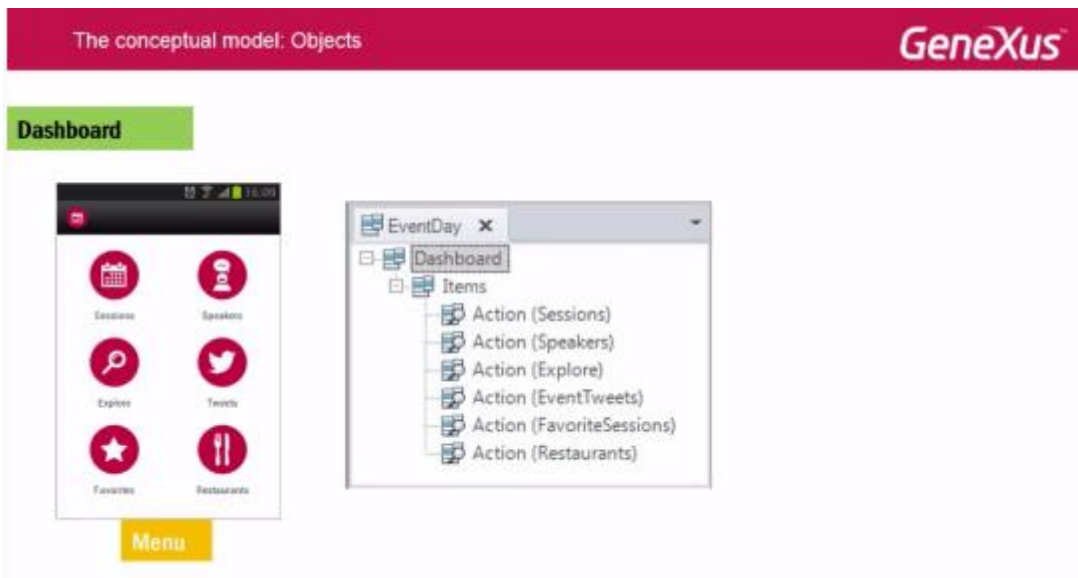
Y para el Detail en modo de Edición, la navegación es indicando la operación: Insert, Update o Delete:



Volveremos sobre esto luego.

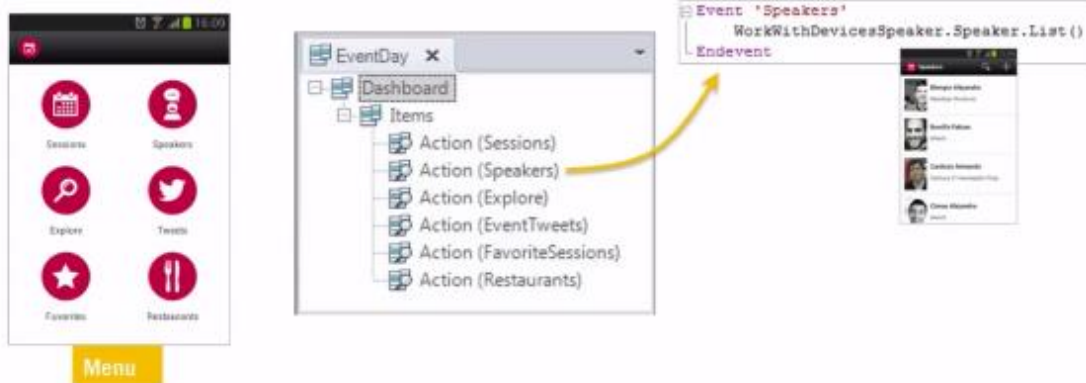
Ahora veamos brevemente las particulares de cada objeto.

El objeto Dashboard consistirá esencialmente en ítems que especifican las acciones que serán ofrecidas al usuario.



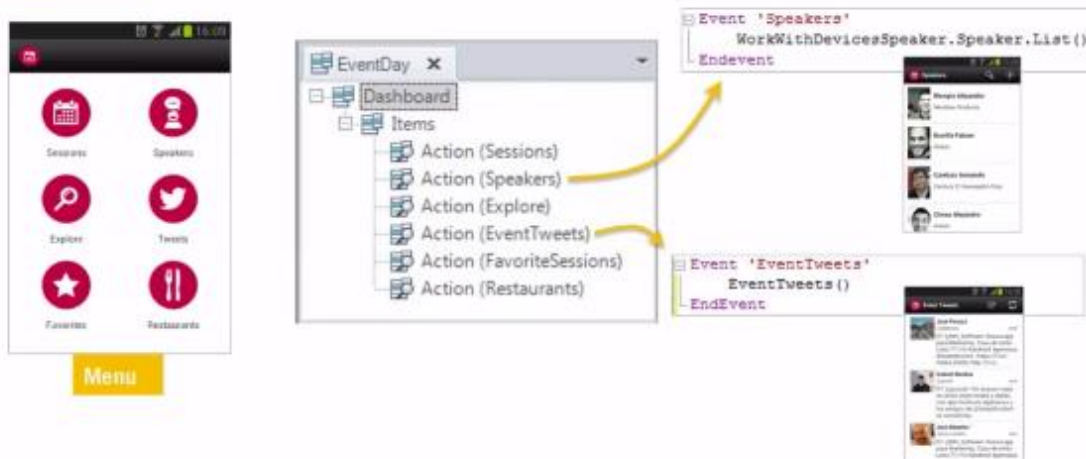
Cada una de ellas corresponderá a un evento donde se programa a quién invocar como consecuencia de elegir el ítem.

Dashboard



Al insertar cada acción en el dashboard, automáticamente se abrirá la ventana "Select Objects" para que elijamos el objeto al que queremos invocar como resultado de esa acción.

Dashboard



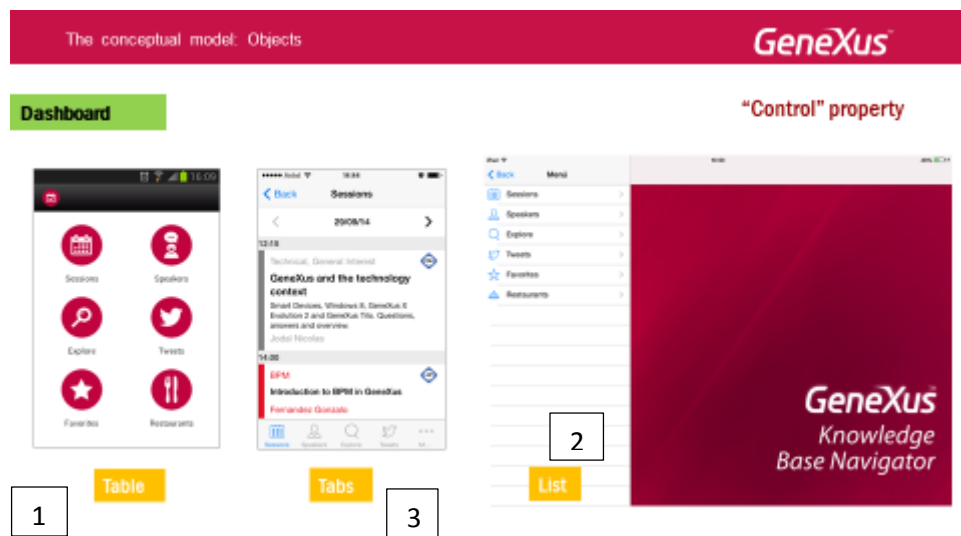
Automáticamente quedará programa da la invocación en los eventos, que por supuesto, podemos modificar.

Puede haber muchos objetos dashboard dentro de la aplicación (todas las veces que se necesite utilizar un menú).

A diferencia de las aplicaciones Web, que permiten múltiples puntos de entrada (a partir de URLs), las aplicaciones para Smart Devices, que estarán instaladas en el dispositivo, tienen un único punto de entrada. El punto de entrada suele ser un menú, por tanto un objeto Dashboard.



Es por ello que los dashboard, por defecto, son objetos Main.



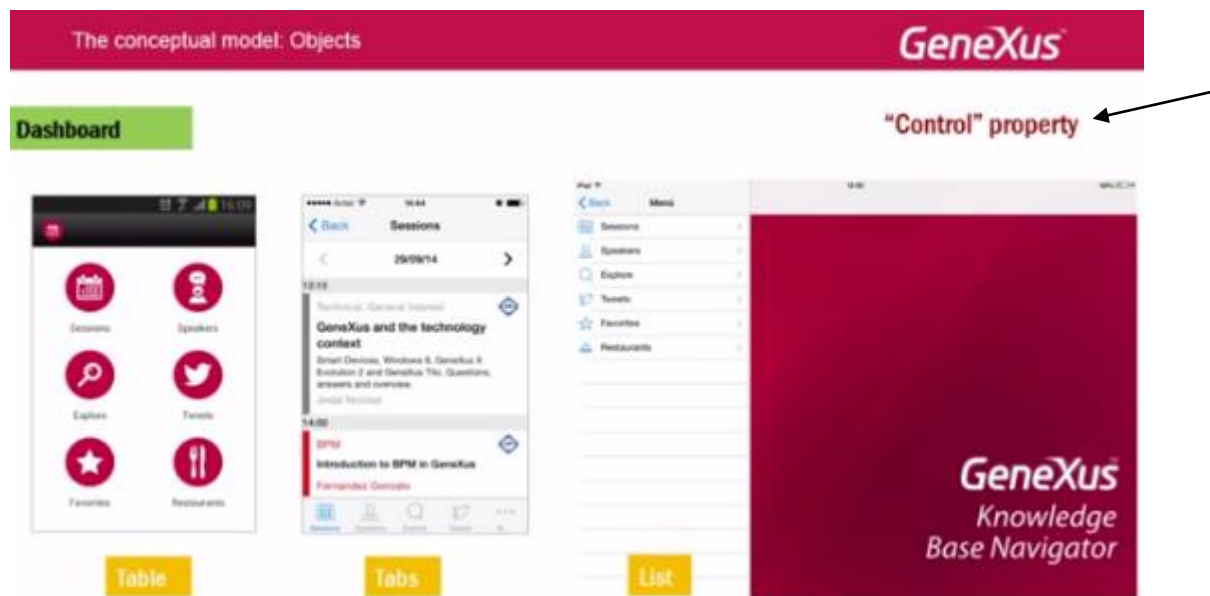
Si bien cada plataforma tiene su estándar, podemos elegir que los ítems del menú se muestren:

- 1) en forma de **tabla** (que es el default de Android)
- 2) en forma de **List** (que es el default de las tabletas)
- 3) o en forma de **Tabs** (que es el default de los iPhones) donde el área central aparece cargada con el objeto llamado por el primer ítem (en nuestro caso: Sessions).

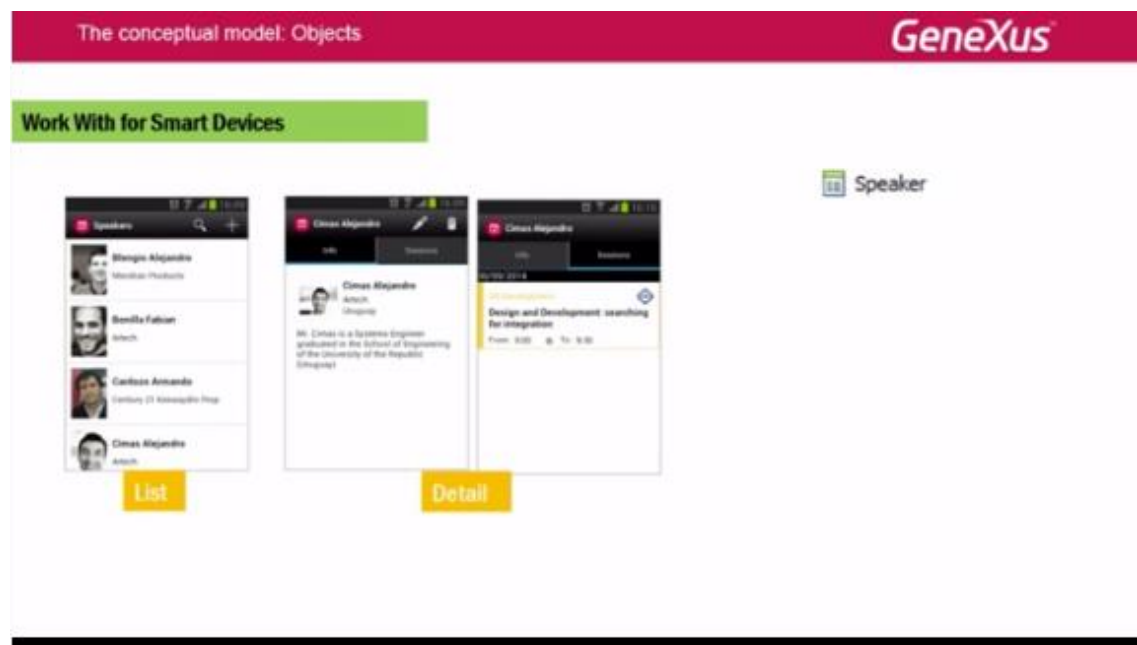
Para modificar el comportamiento default, **existe una propiedad del Dashboard** que permite seleccionar alguno de estos tres valores:

- Table
- Tabs
- o List

La propiedad de nombre: **Control**



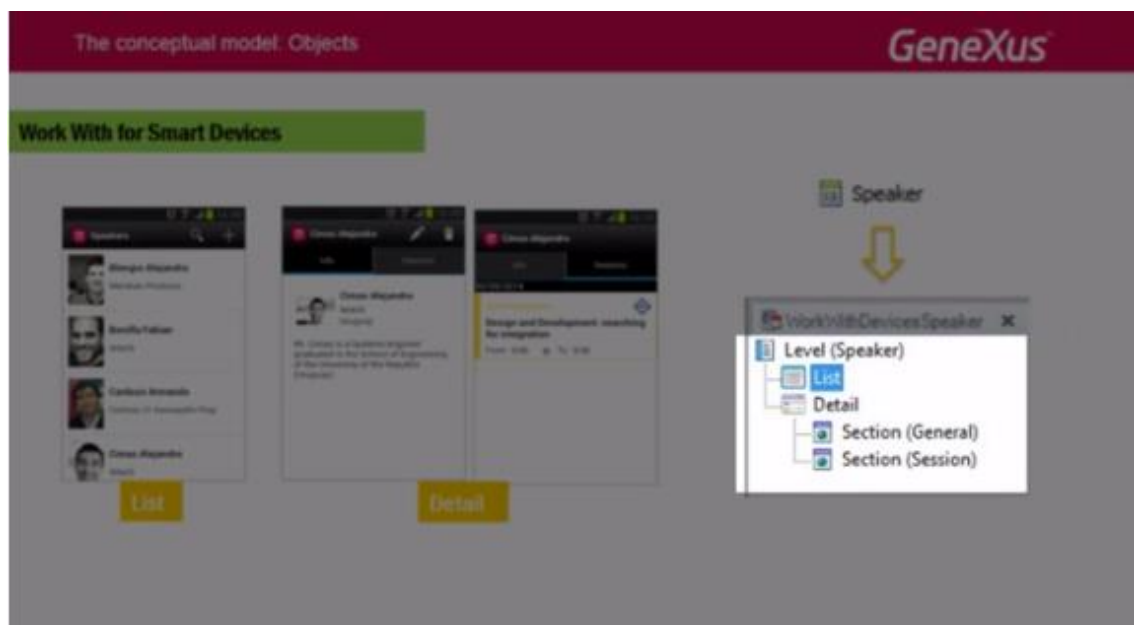
Partiendo de la transacción Speaker:



Aplicarle el pattern “Work With for Smart Devices”:



dará como resultado la creación de un objeto de este tipo, que presentará una **estructura jerárquica** que representa tanto el **List** como el **Detail** :

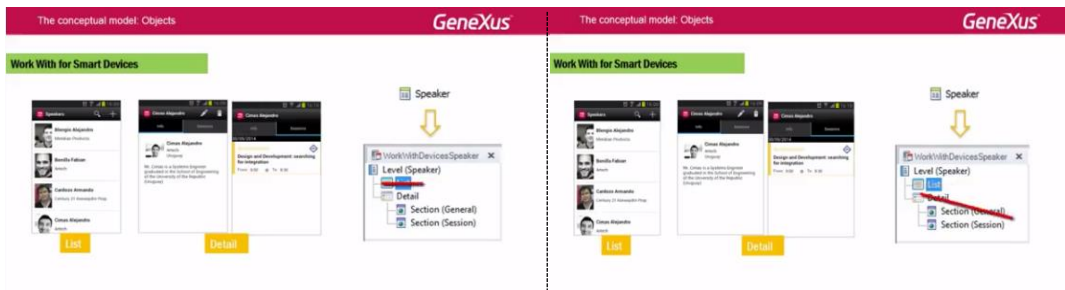


El desarrollador podrá personalizarlo, trabajando directamente sobre sus pantallas –lo que veremos en lo que sigue–, agregando o quitando niveles

Work With for Smart Devices



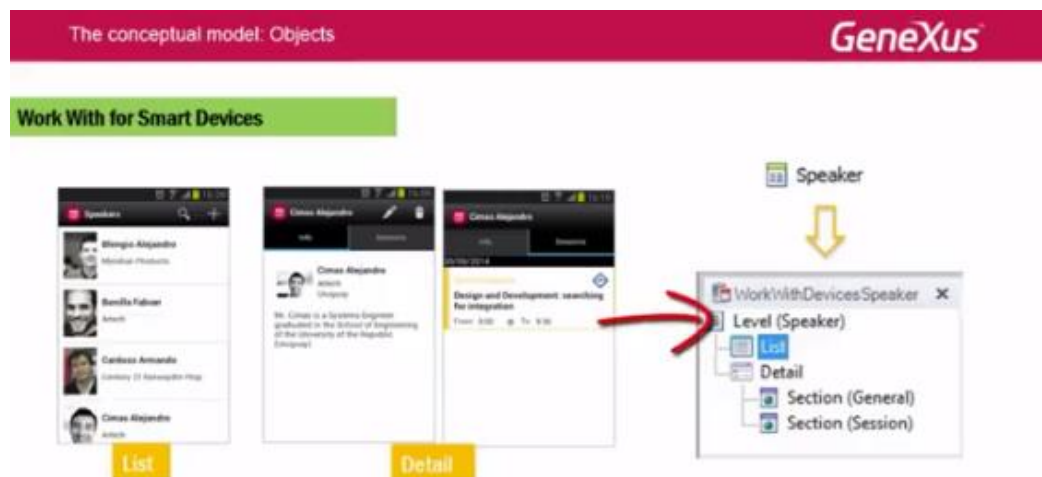
o, incluso, eliminando el List o el Detail



o quitando o agregando sections a este último: al Detail:



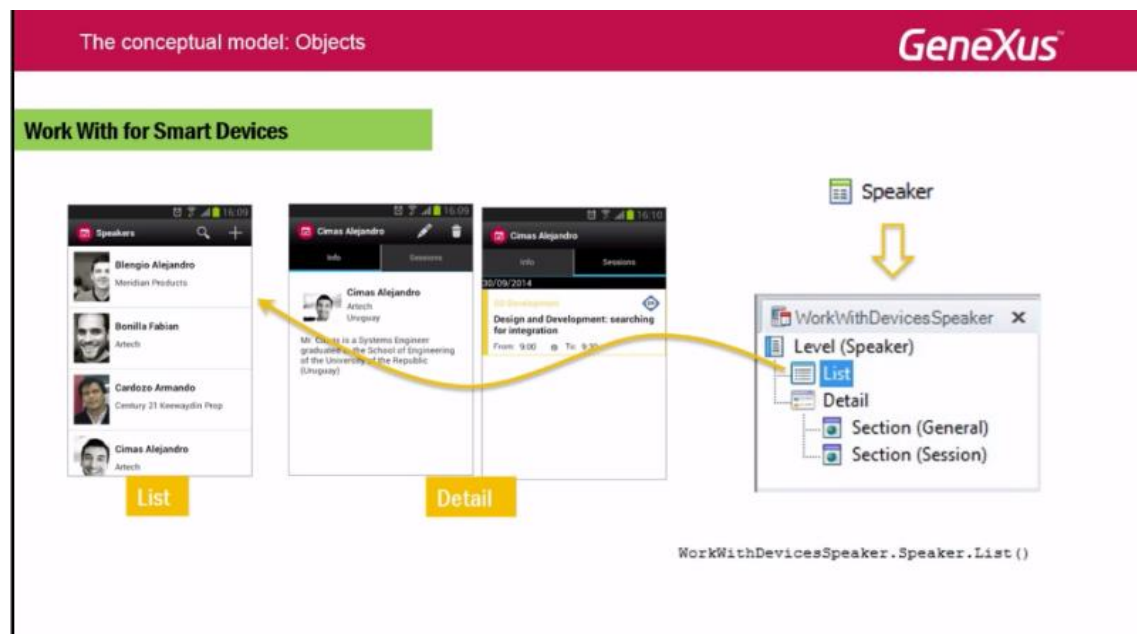
Al aplicar el pattern a la transacción de un nivel (Speaker), podemos ver que aparece un nivel



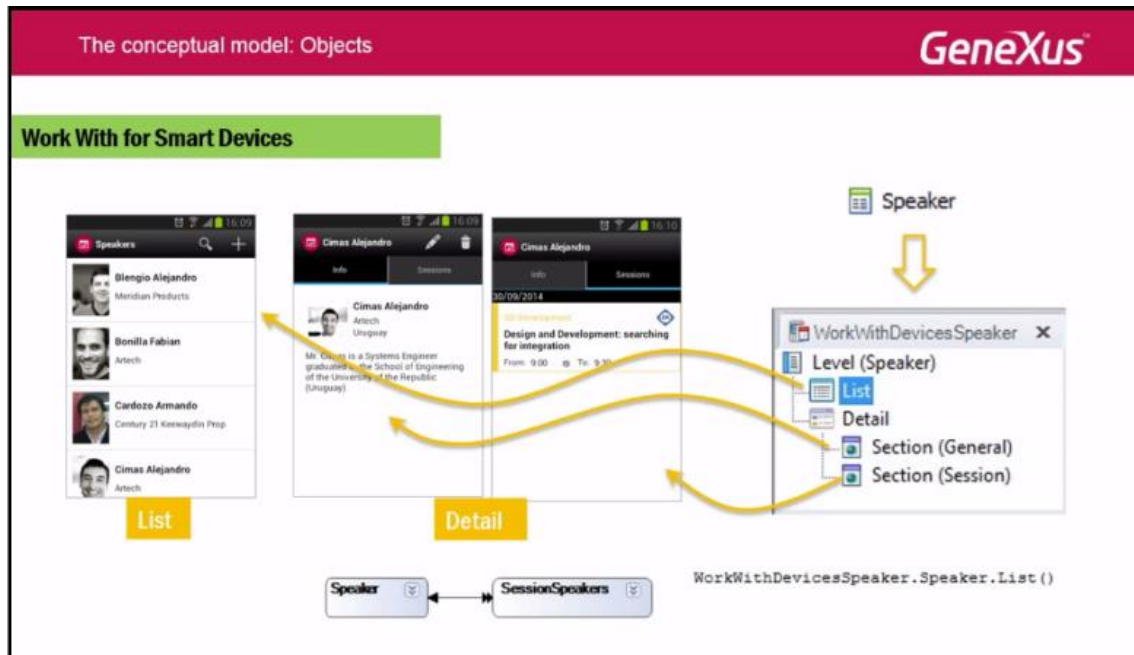
que corresponde a ese nivel de la transacción, con la mencionada estructura jerárquica.

Esto ya nos da la pauta que podemos tener más niveles paralelos.

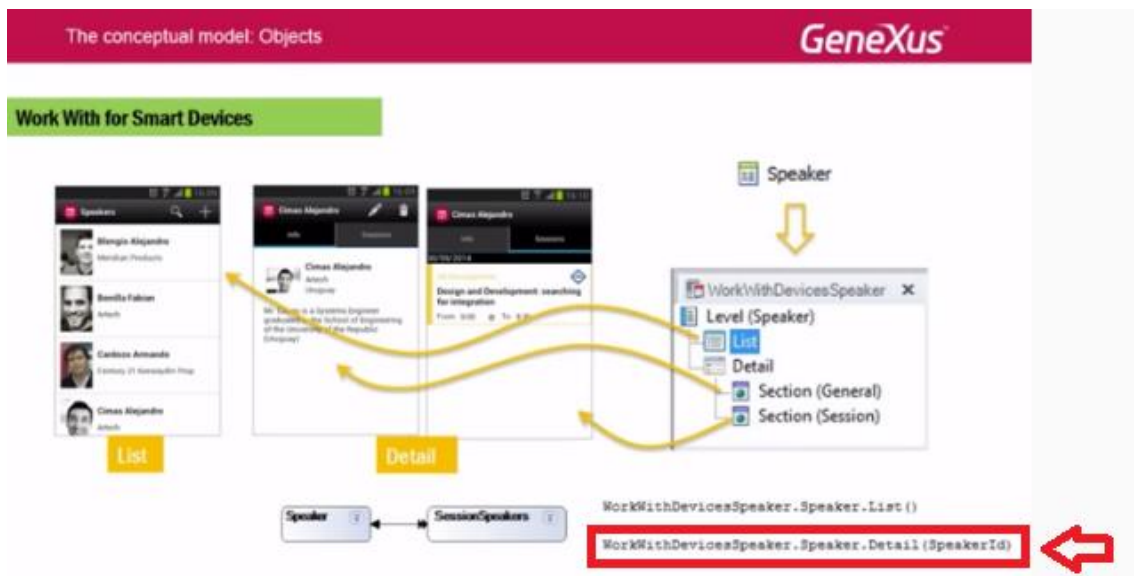
Aquí vemos como se invoca el List del nivel Speaker, el único que tenemos por el momento:



El **Detail** consiste de varias secciones: la **general**, correspondiente a la información de ese nivel de la transacción, y una sección por cada tabla de la base de datos con información directamente subordinada (en este caso, la de las conferencias en las que el speaker participa).



La invocación se hace al detail **como un todo**, pasándole por parámetro la clave primaria:



Work With for Smart Devices

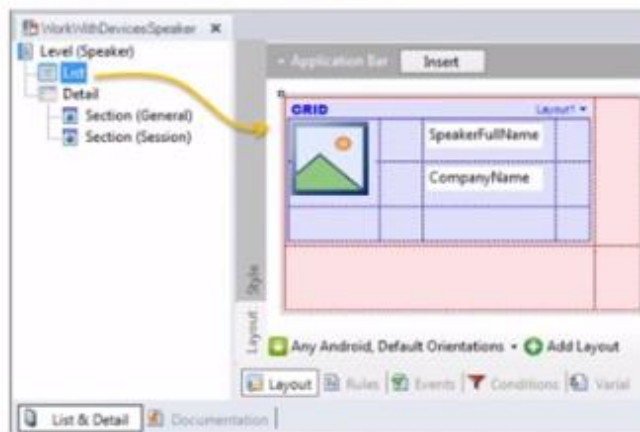
List



¿Dónde se encuentra diseñada la pantalla del List que vemos en ejecución?

Work With for Smart Devices

List



En el nodo List del “Work with”.

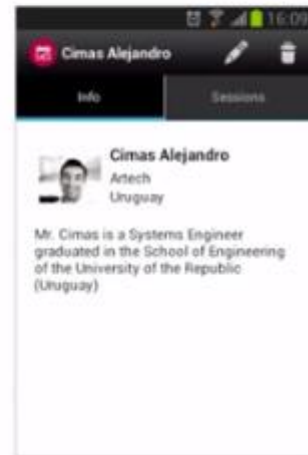
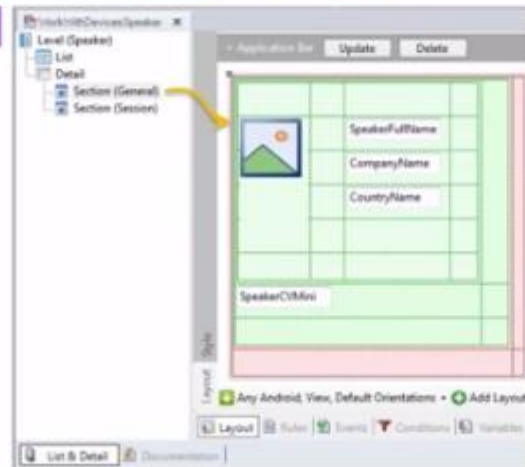
¿Y dónde la pantalla de visualización del detalle de la información general de un speaker?

Work With for Smart Devices

Detail

View

General Section



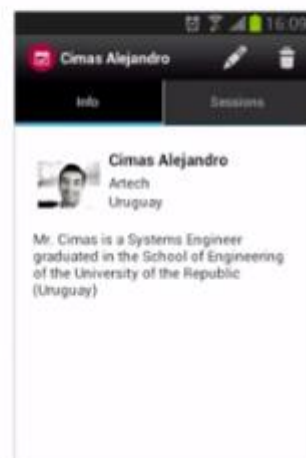
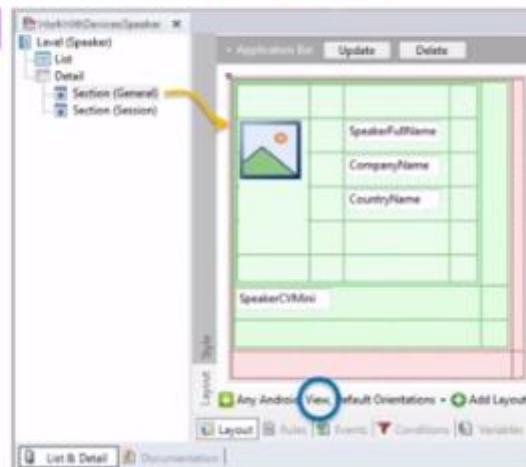
En el nodo Section (General) del nodo Detail en modo View.

Work With for Smart Devices

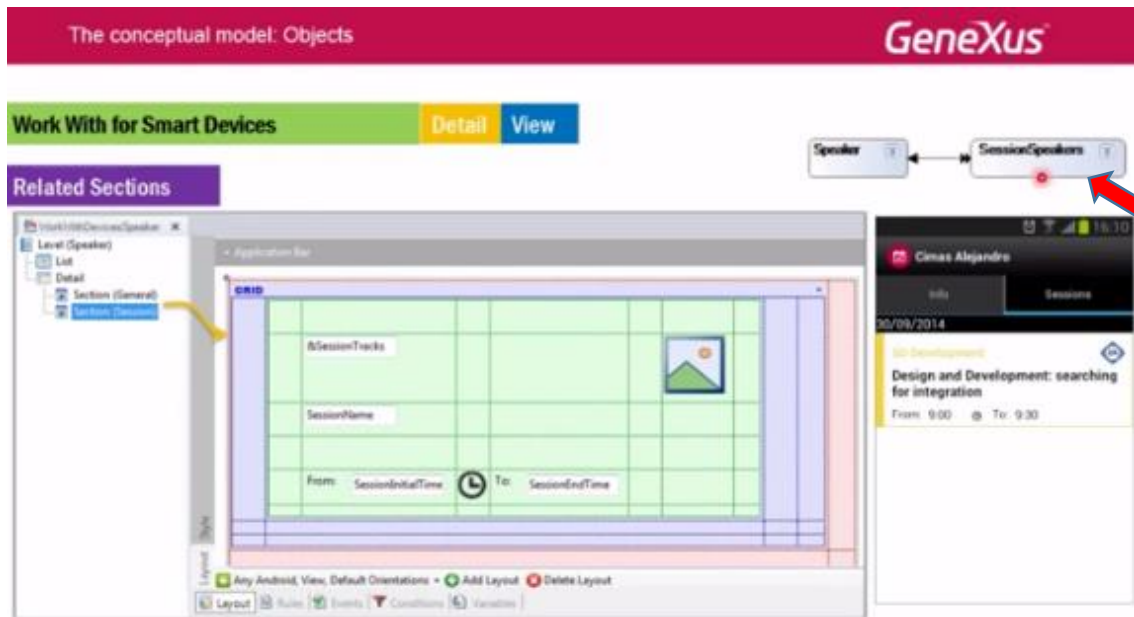
Detail

View

General Section



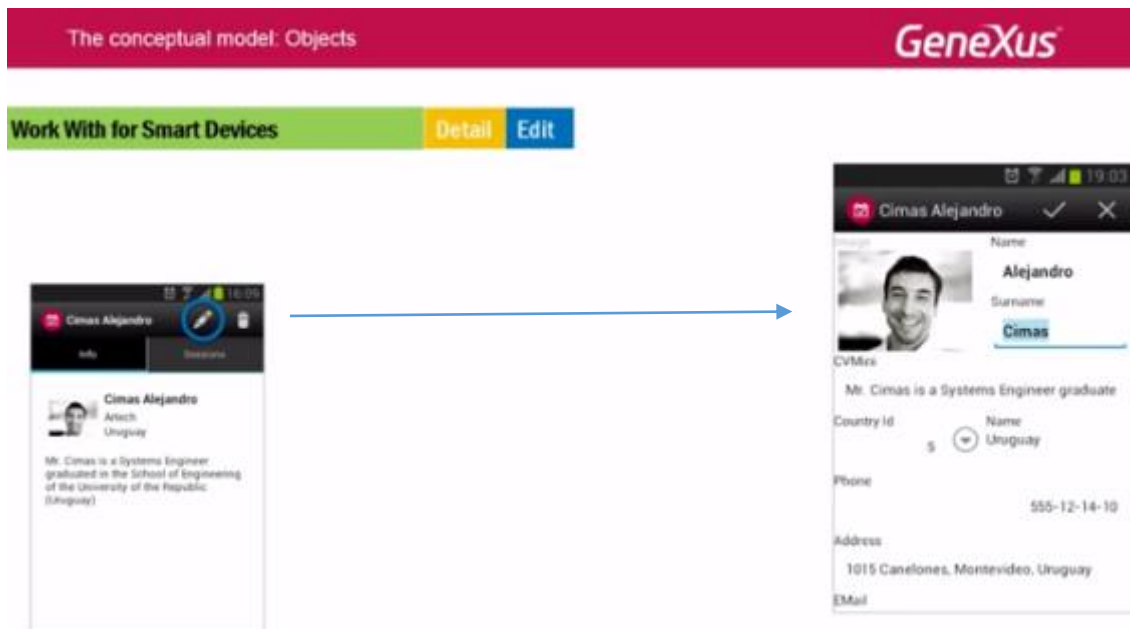
¿Y la pantalla correspondiente a las conferencias **de ese speaker**?



En el nodo Section (Session) correspondiente a datos de la tabla subordinada a speaker.

Por último, ¿dónde está implementada la pantalla que permite las operaciones de CRUD (Create/Update/Delete)?

Aquella a la que accedíamos, por ejemplo cuando hacíamos tap sobre el ícono de Update de la pantalla de view?

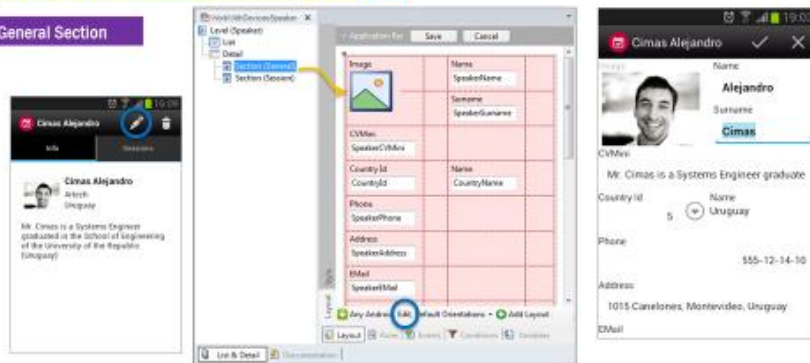


Es también en el nodo Section(General), pero eligiendo la pantalla de Edit del combo inferior.

Work With for Smart Devices

Detail Edit

General Section



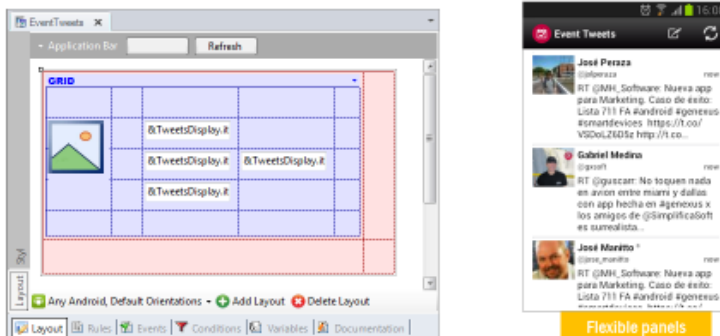
Para implementar un panel desde cero, como el de los tweets por ejemplo:

Panel for Smart Devices



tenemos un objeto que se parece **en todo** a cualquiera de los nodos del Work With.

Panel for Smart Devices



La diferencia respecto al work with aplicado a una transacción, es que, mientras las pantallas del work with ya aparecen inicializadas, el panel en cambio, estará por completo vacío. No tiene ningún tipo de lógica asociada.

En este panel, en particular, podemos ver cómo el desarrollador ha insertado en el layout, un grid con variables, que serán cargadas en el evento Load, para cada línea (en este caso, consumiendo un web service), de manera análoga a como se hace en un web panel.

GeneXus

Demo

Starting to develop the web portion of the application

Llegó el momento de comenzar a desarrollar nuestra aplicación EventDay desde cero..

Lo haremos en el siguiente video.



