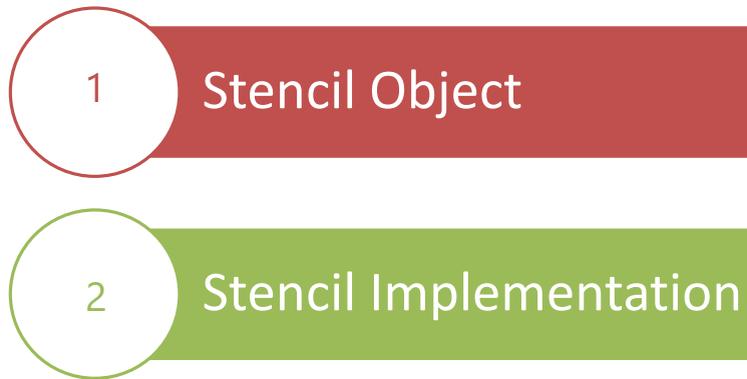


GeneXus[™]
The power of doing.

Stencils

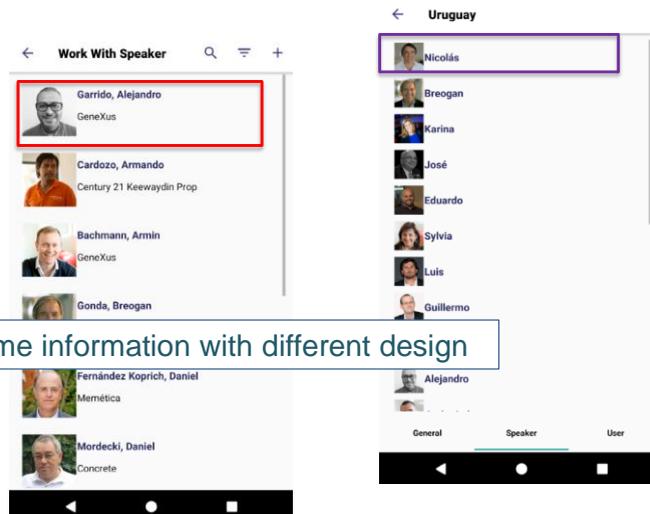
User Interface & User Experience

GeneXus™ 16



Vamos a comenzar con el estudio de los objetos Stencil para que se utilizan y luego veremos como aplicar Stencils en nuestra Base de Conocimiento.

UI Control Compositions



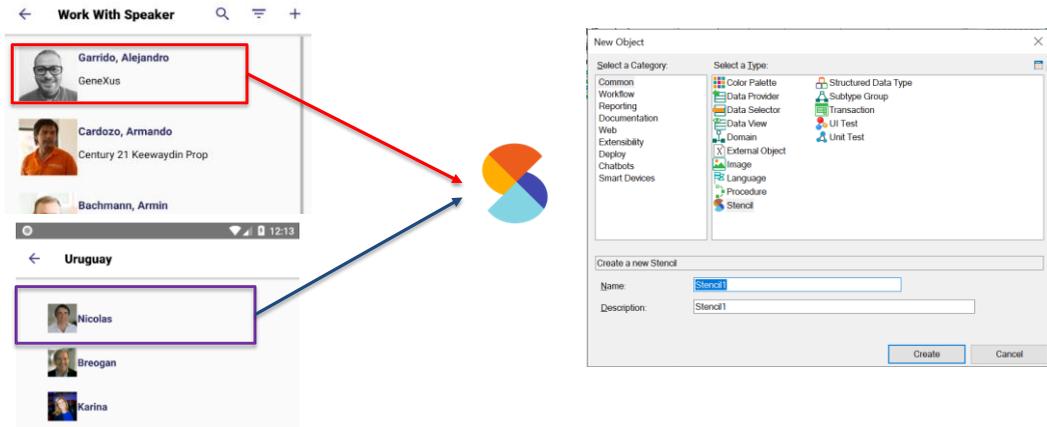
Es común a la hora de desarrollar una aplicación que existan elementos que se repiten en varios lugares, por ejemplo, en la app del evento que estamos desarrollando, vimos que en varios lugares se muestra una lista de oradores. En la imagen de la izquierda vemos el List del Work With for Smart Devices de Speakers y a la derecha tenemos la misma lista que se muestra en la sección Speakers del Detail del Work With For Smart Devices de Countries.

También es común que nosotros deseemos que cuando se muestra la misma información se haga respetando un mismo diseño.

En estos casos corremos el riesgo de que no se vean iguales, que no tengan el mismo diseño, además cada vez que requerimos cambios en la forma de presentar esta información deberemos modificar todos aquellos lugares en donde se encuentra ese diseño.

Uno de los trabajos que deberán realizar en conjunto los diseñadores y los desarrolladores es identificar estos patrones en la aplicación, nosotros los denominaremos Composición de Controles UI. Una vez que los hayamos identificados, vamos a establecer un único diseño, de modo que esta composición se vea en todos lados iguales.

Stencil Object



El nuevo objeto Stencil, nos permitirá crear estas Composiciones de Controles UI definiendo el diseño de estos sin pensar en el código, estos componentes serán reutilizables en distintos objetos, de esta forma mantendremos la consistencia de las aplicaciones con menos esfuerzo.

Cuando deseemos modificar el diseño lo haremos en el Stencil y los cambios se replicarán en todos los objetos que lo utilicen.

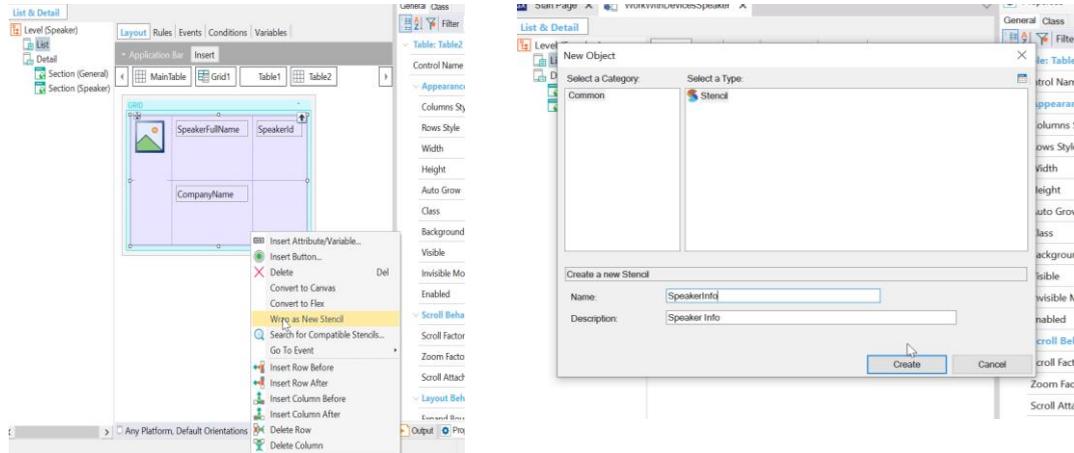
Al ser un objeto independiente contaremos con todas las ventajas de estos, por ejemplo pueden ser exportados para ser compartidos, podemos mantener la historia de cambios que va sufriendo el mismo y podremos contar con la Referencia Cruzada para saber en donde se utiliza el Stencil y cuando lo modifiquemos, cual va a ser el impacto de estos cambios.

En el Objeto Stencil va a confluir el trabajo de los desarrolladores con el de los diseñadores en un mismo objeto.

Demo: Stencil Implementation

Vamos a ver todo esto que aprendimos en GeneXus.

Demo: Stencil Implementation



Primero vamos a revisar la lista de Speakers, como habíamos visto ya en la presentación, y también en Countries, Vamos a ver los Speakers de un País, vemos que son diseños distintos.

Vamos a ir a GeneXus ahora.

Vamos a abrir primero el WorkWithDevicesSpeakers y vamos a ver el List, vemos que acá tenemos una tabla que es la que vamos a convertir en un stencil.

Con botón derecho y vamos a usar la opción Wrap As New Stencil, esto va a convertir la tabla en un Stencil, vamos a ponerle de nombre SpeakerInfo y create.

Demo: Stencil Implementation

The screenshot shows the GeneXus IDE interface for a stencil implementation. The main window is titled 'SpeakerInfo' and has two tabs: 'Layout' and 'Variables'. The 'Layout' tab is active, showing a grid with three variables: '&SpeakerFullName', '&SpeakerId', and '&CompanyName'. The 'Variables' tab shows a table with columns for 'MainTable', 'Table2', and 'SpeakerFullName'. The 'Properties' window on the right shows the configuration for the '&SpeakerFullName' variable.

Properties	
General Class	
Filter	
Attribute/Variable: &SpeakerFullName	
Control Name	&SpeakerFullName
Attribute	&SpeakerFullName
ReadOnly	False
Appearance	
Label Position	None
Class	AttributeTitle16
Visible	True
Invisible Mode	Keep Space
Enabled	True
Format	Text
Invite Message	Speaker Full Name
Control Info	
Control Type	Edit
Auto Grow	False
Input Type	Values
Suggest	No
Auto correction	True
Auto capitalization	First word
Cell Information	
Row Span	1
Col Span	1
Horizontal Alignment	Default
Vertical Alignment	Default

Lo primero que vamos a ver acá es que en los atributos que teníamos en la grilla ahora son variables, por otro lado no tenemos todas las propiedades disponibles en el Stencil, solamente aquellas que tengan que ver con su diseño no con su comportamiento.

Nos respeta las clases, los tamaños de las tablas que habíamos definido, etc.

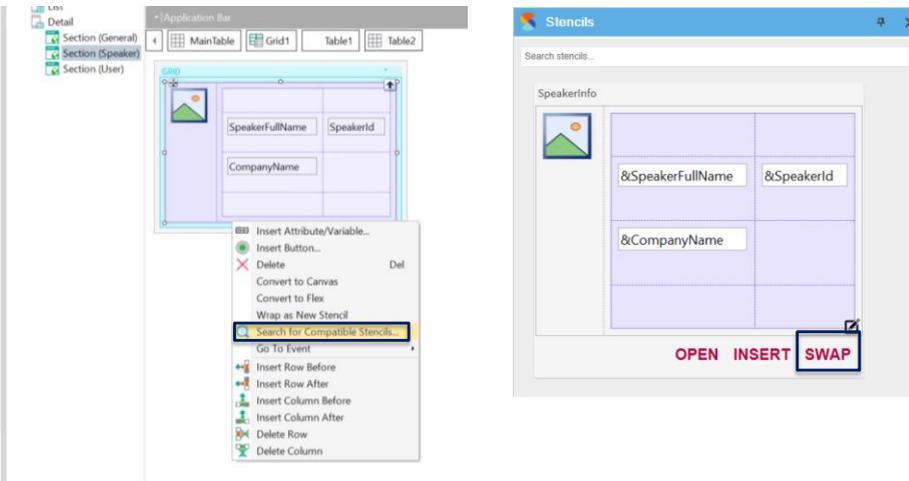
Fíjense que solamente tenemos el Layout y Variables, no tenemos Eventos ya que los Stencils no implementan ninguna lógica. Vamos a grabar el Stencil.

Demo: Stencil Implementation

The screenshot shows the GeneXus IDE interface. On the left, a project named 'WorkWithDevicesSpeaker' is open, showing a 'List & Detail' view. The 'List' view is selected, and a stencil named 'SpeakerInfo' is applied to the grid. The grid contains three rows of data: a header row with 'SpeakerFullName' and 'SpeakerId', and two data rows with 'CompanyName' and 'SpeakerId'. The 'Properties' window on the right shows the configuration for the 'SpeakerInfo' stencil. The 'Attribute/Variable' is 'SpeakerFullName', and the 'ReadOnly' property is set to 'True'. Other properties include 'Link' (Autolink: True), 'Virtual Keyboard' (Enter Event: <Platform Default>), and 'Behavior' (Input History: True, Is Password: False).

Fíjense que ahora en el WorkWithDevicesSpeaker, en el List la tabla pasó a ser un Stencil, SpeakerInfo, y fíjense que en los atributos, en las propiedades tenemos muchas menos propiedades para configurar, esto es porque todas las que tengan que ver con el diseño quedaron del lado del Stencil y acá solamente tendremos propiedades que tienen que ver con el comportamiento y fíjense que acá quedaron los atributos como estaban originalmente. Por ejemplo si quisiéramos mover un atributo de lugar no podemos, esta totalmente ReadOnly todo lo que esta marcado en gris dentro de la tabla o el Stencil SpeakerInfo. vamos a grabar también este objeto.

Demo: Stencil Implementation



y ahora vamos a aplicar el Stencil en nuestro WorkWithDevicesCountry, pero dentro del Section Speakers.

fíjense que acá no tenemos una tabla, tenemos directamente los atributos dentro del grid. Vamos a crear una tabla para poder convertirla en un Stencil, vamos a llevar los atributos dentro de la tabla y ahora si vamos a tener la opción de convertir esta tabla a Stencil. Vamos a usar la opción Search for Compatible Stencils, esto nos va a abrir la solapa de Stencils, acá solamente tenemos uno por ahora, pero si tuviéramos muchos podríamos buscar acá por nombre.

Y aquí tenemos tres opciones, Open, Insert y Swap, vamos a ver cada una.

La opción Open va a abrir el Stencil directamente, la opción Insert va a insertar el contenido del Stencil en donde nos encontremos posicionados en el Layout, vamos a probarlo, yo voy a hacer Insert al final, fíjense que directamente agrega la tabla completa del Stencil con las variables en donde estábamos posicionados. vamos a eliminar.

Y la opción Swap que es la que nos interesa explicar en este momento va a tener cierta inteligencia y va a tratar de buscar para cada variable en el Layout una coincidencia por tipo de dato en el Layout del Panel donde estemos aplicando ese Stencil.

En este caso la variable &SpeakerImage del Stencil va a encontrar una correspondencia con el atributo SpeakerImage, entonces va a dejar el atributo en lugar de usar la variable, obviamente le va a aplicar todo el estilo que tiene el Stencil. En el caso de &SpeakerFullName va a buscar un atributo o variable de tipo Char, encuentra

SpeakerName, lo va a usar y va a dejar SpeakerName, no lo va a convertir a SpeakerFullName.

En el caso de &SpeakerId no encuentra un atributo o variable numérico actualmente en el Layout por lo cual va a agregar &SpeakerId al Layout, lo mismo va a pasar con &CompanyName, no tenemos otro atributo, porque este ya fue mapeado (SpeakerName), no tenemos un segundo atributo de tipo Char, por lo cual va a agregar &CompanyName al Layout. Vamos a verlo.

Presionamos Swap y tal cual habíamos adelantado tenemos SpakerImage (Atributo), SpeakerName (Atributo) en lugar de &SpeakerFullName, &SpeakerId (como variable) y &CompanyName (como variable).

Nosotros podríamos dejar las variables y cargarlas en el evento Load, en este caso no nos interesa, por lo cual vamos a cambiar para que no use variables y use directamente el atributo.

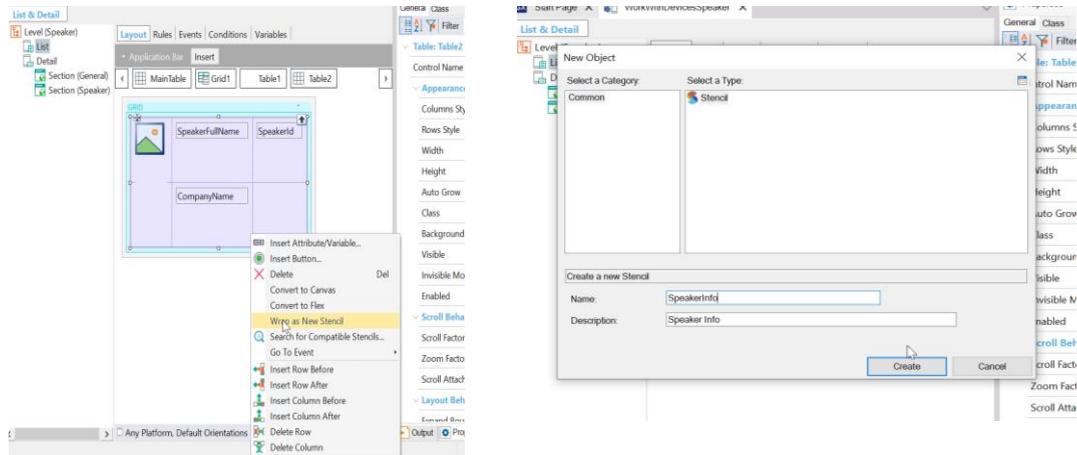
Bien vamos a grabar.

Me olvide de cambiar SpeakerName por SpeakerFullName, ahora si. grabamos nuevamente y vamos a correr la aplicación.

Vamos a revisar como quedo el sistema, vemos la lista de Speakers con el diseño que ya tenia anteriormente, solamente que ahora esta usando y hace referencia al Stencil.

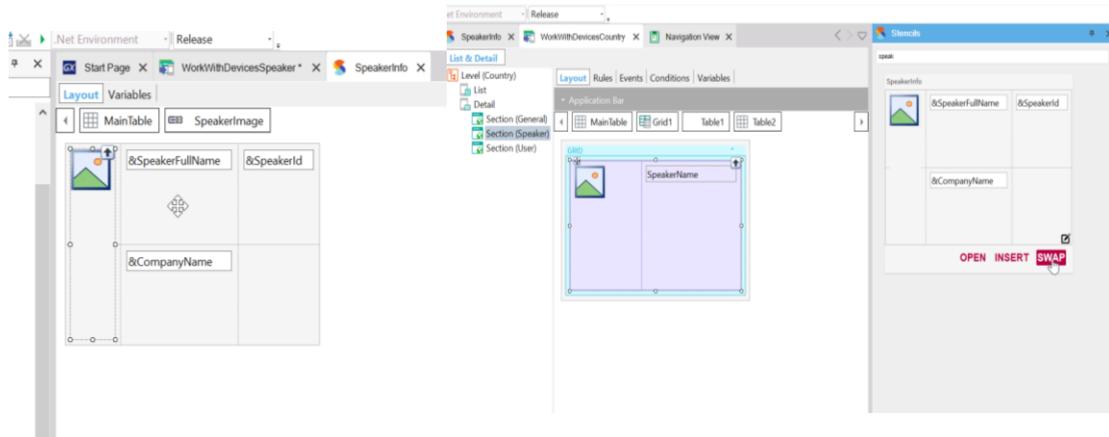
Vamos a ver , por ejemplo el que ya habíamos visto, Uruguay, en Speakers, vemos que ahora si tenemos exactamente el mismo diseño en las dos listas.

Stencil Creation



Vamos a repasar entonces lo que vimos, comenzamos convirtiendo una Tabla en un Stencil. Esto lo hicimos para simplificar, pero podríamos haberlo creado desde cero como cualquier otro objeto GeneXus.

Stencil Implementation



Vimos también que solo podremos modificar el diseño dentro del Stencil, en los Objetos que lo implementan los controles son ReadOnly, vemos el panel donde se muestran los Stencils de la base de conocimiento, acá solo vimos la opción SWAP, la Opción INSERT no va a buscar el mapeo de variables en atributos, directamente se va a insertar el Stencil en donde estemos posicionados, pero no va a reemplazar ningún contenido, lo va a agregar y la opción OPEN simplemente permite abrir el Stencil.

Los invitamos a buscar más información relacionada al uso de los Stencils en Wiki de GeneXus.
<https://wiki.genexus.com/commwiki/servlet/wiki?38418>

En el próximo video empezaremos a estudiar mas en detalle el uso de distintos controles para mejorar el diseño de la UI y UX en nuestro sistema.

GeneXus™

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications