

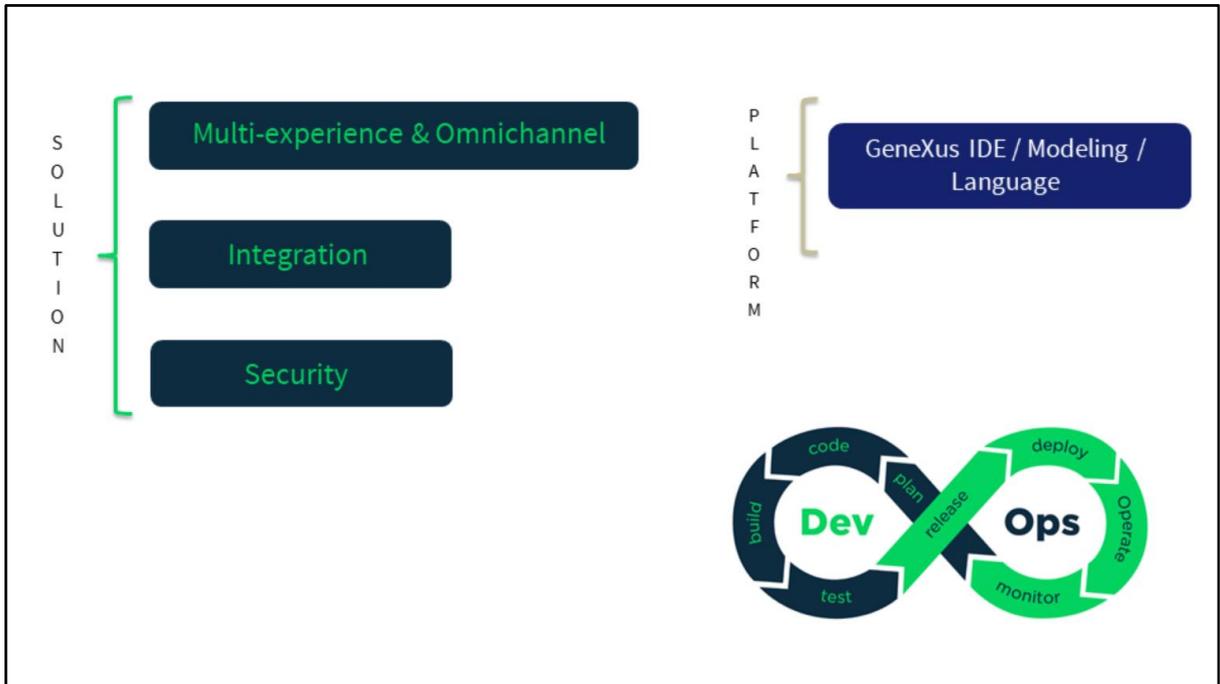
GeneXus[™]
The power of doing

Development

GeneXus™ 16

Yo soy Federico Salomon, vamos a estar viendo con Martin, ambos somos parte del equipo técnico de GeneXus.

Vamos a ver puntualmente la parte de desarrollo de la Versión V16.



Comenzando por los Design Systems...

Mobile & Web



Multi-experience & Omnichannel

- Design systems
- Streams of information



Web/ Web mobile

Native mobile

Chatbots

Wearables/TV/AR

Design Systems

... que es un concepto que se vino trabajando mucho con el GX28, hubo tracks específicos con el concepto design systems.

Y hablamos de design systems como algo mucho mas avanzado, como una apuesta mucho mas alta que solo el hecho de aplicar diseños a las aplicaciones.

El design systems levanta mucho mas la "vara" si se quiere...

Principles

Patterns

Practices

y habla de Principios (Principles), de Patrones (Patterns) y de prácticas (Practices) en base al diseño, cómo tienen que comportarse las pantallas, como deben ser las interacciones de nuestro Software con los usuarios, habla de una serie de patrones que hay que respetar y bueno... es un concepto mucho mas amplio que solo aplicar diseño y que las cosas se vena bonitas ¿verdad?

Design Systems



En Design Systems lo que vemos son grandes organizaciones, grandes empresas que comenzaron con esta puesta y empezaron a implementar su design systems para tener una consistencia en los productos que ofrecen, para que todos sus productos tengan la misma visualización, tengan las mismas interacciones con los usuarios, independientemente de la plataforma en la que ejecutan.

Ahí vemos grandes empresas, bueno, tenemos IBM, Airbnb, Mozilla, GitHub, General Electric, un montón de empresas que comenzaron con esta idea de unificar sus diseños, de unificar las interacciones de sus productos con los usuarios, por eso es que decimos y creemos que design system es muy importante y tenemos que comenzar a incorporarlo dentro del conocimiento de GeneXus, de nuestra base de conocimiento.

Integrate the designers (Design Ops)

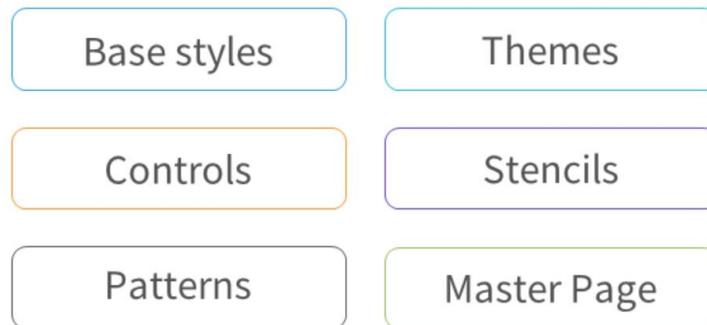


El Design System también habla de integrar a los diseñadores, como Armin les comentaba antes, un concepto que se está hablando mucho también es el tema de las DevOps de las development operation.

El Design System también habla de Design Ops, habla de integrar a los diseñadores dentro de esa cadena de DevOps, que sean parte del trabajo y que trabajen codo a codo con los desarrolladores para que los resultados sean realmente los esperados y los que ellos proponen.

Lo que vemos ahí abajo son herramientas, las principales que usan los diseñadores, justamente para hacer su trabajo y para llevar adelante todo este "esquema" digamos, de integración.

Design system



El Design System que concebimos en GeneXus es una serie de estructuras que algunas ya las conocen, como les comentó Armin, los temas, los Patterns, las Master Page, son ya estructuras conocidas por los usuarios GeneXus, por la comunidad GeneXus que ya estamos acostumbrados a utilizar en nuestros proyectos, son parte importante de ese modelado de UI, UIX que debemos tener en nuestras aplicaciones, pero, en la V16 incorporamos nuevos componentes que hacen a este conjunto de estructuras que necesitan trabajar juntas para lograr los resultados.

Hablamos de Base styles, hablamos de User Controls y hablamos de los stencils.



Otro concepto que está sobre la mesa hoy y suena mucho, son los CSS Frameworks. Los CSS Frameworks son los que nos van a dar a nosotros esa interface, ese diseño bonito que vemos en las aplicaciones. Si se quiere son parte de esos Design Systems y son aquellos que nos definen como se debe ver la aplicación.

Los Design Systems están muy relacionados con lo que son los User Controls y los Base Styles. Yo particularmente he trabajado mucho con los Semantic UI (que me gusta mucho) y también vemos que acá aparece Bootstrap que es bastante conocido ya para nosotros también, ya que Bootstrap es el motor que nos provee a nosotros el comportamiento responsivo de las aplicaciones.

El motor de Bootstrap es quien decide cómo se tiene que visualizar la aplicación cuando nosotros alteramos el tamaño de las pantallas, o sea que Bootstrap ya es un viejo conocido para GeneXus, pero tenemos hoy la posibilidad de incorporar muchos más y Bootstrap en cierta forma queda desacoplado de las soluciones, ahora no decide tanto como se van a visualizar las cosas, sino que se encarga de realizar el comportamiento responsivo y vamos a poder utilizar la interfaz, la UI que proponen otros CSS Frameworks.

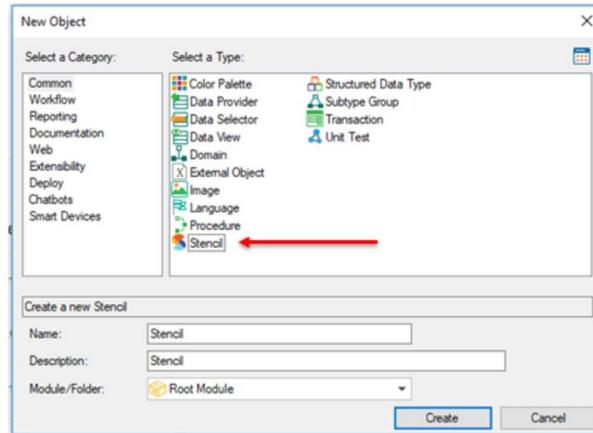
Con esto quiero decir que ahora no vamos a ver el diseño que propone Bootstrap en otros controles que nosotros proveemos, sino que vamos a poder definir muchos. Hay muchísimos CSS Frameworks.



Ahora los dejo con Martin que les va a hablar de Stencils para después seguir continuando con los otros controles que les comenté.

Como decía Fede, yo voy a estar hablando un poco de este objeto nuevo de GeneXus 16, que en realidad ya está disponible desde GeneXus 15 U12 que son los Stencils.

What is a Stencil?



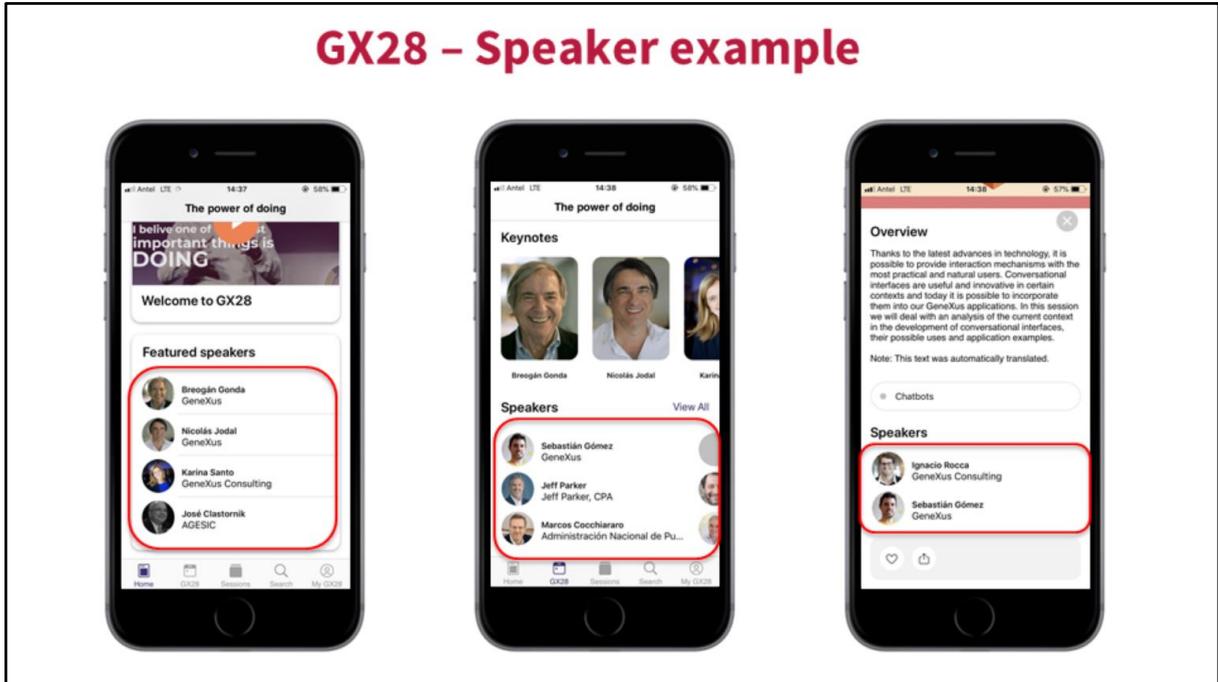
Entonces comencemos por ¿qué es un Stencil?

Un Stencil es un objeto nuevo que nos ayuda a todo esto que hablaba Fede, a crear de manera más fácil estos Design Systems.

Nos ayuda a crear, son objetos de diseño que nos ayudan a crear interfaces de usuario desacoplado el código y favorece muchísimo todo lo que es el reúso y todo lo que es la modificabilidad.

Entonces como pueden ver, es simplemente Ctrl-N(ew) y tenemos el objeto nuevo ahí para crearlo desde cero.

GX28 – Speaker example



Entonces ¿en que caso yo utilizaría un Stencil?

Bueno, a mi me pareció que un ejemplo bien claro es la aplicación del evento en la cual nosotros estuvimos utilizando este nuevo objeto.

Y como pueden ver, les traje tres pantallas: la primera del lado izquierdo muestra los oradores (los oradores favoritos digamos).

La segunda muestra una lista completa de oradores y la tercera es el detalle de una conferencia que muestra cuales son los oradores responsables de esa conferencia.

Lo que tienen en común estas pantallas y algunas otras que no puse por tema de espacio, es que nosotros mantenemos un lenguaje visual de cómo mostramos a los oradores.

Entonces, los mostramos con una foto redondeada, los mostramos con el nombre del orador en negrita con determinada fuente, con determinado margen que los separa del nombre de la compañía en la cual trabajan... ciertas cosas que mantenemos a lo largo y ancho de la aplicación y que siempre son iguales, entonces ¿cómo lo hacíamos hasta hoy? Bueno, lo hacíamos una vez en alguno de los paneles y después ese diseño lo copiábamos y lo llevábamos a las demás pantallas.

Esto trae un problema que es ¿Qué pasa si el día de mañana yo quiero cambiar la forma en la cual se ven los oradores? Quiero que la foto ahora sea cuadrada y no redonda, bueno, entonces tendría que ir panel por panel para modificar esa imagen.

El poder que hoy nos dan los stencils es: yo hoy creo un Stencil con ese diseño, lo aplico en todas las pantallas que yo quiera y luego esas pantallas quedan "digamos como que heredando" ese stencil, yo modifico el Stencil y se modifican todas las pantallas.

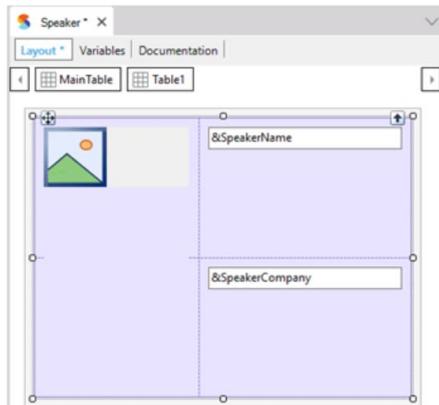
Entonces la modificabilidad se ve muy beneficiada en estos casos, porque aparte, en el otro caso si yo modifico pantalla por pantalla, me puedo olvidar de pantallas, me pueden no quedar exactamente igual porque le puse un margen un poco mayor, entonces esto nos asegura que se va a ver siempre igual, el lenguaje visual lo vamos a mantener, (encapsulamos ese conocimiento de diseño en un stencil) nos es mucho más fácil de compartirlo también entre distintos objetos de mi KB, entre otras KBs, intra empresa y con la comunidad, porque como les decía, esto está desacoplado del código, entonces es mucho más fácil también de compartir .

- Una pregunta... ¿esto vendría a ser el equivalente a un Web component en web?
- Si, la diferencia expresivamente... una de las grandes diferencias es eso que

comentaba a lo último, esto está completamente desacoplado del código.

Stencil Object

Controls + design properties



Sebastián Gómez
GeneXus

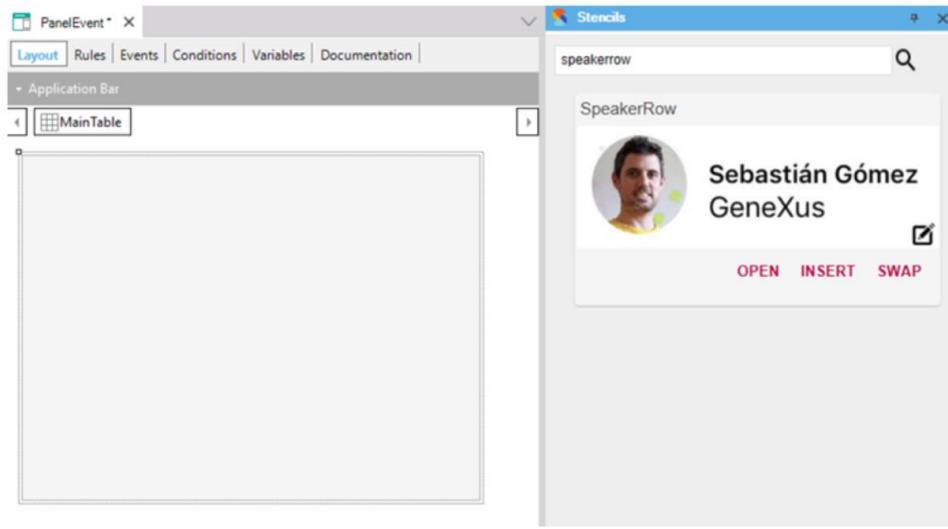
...Yo ahora iba a comentar, por ejemplo, que acá tenemos un Stencil Object, es un layout igual a un SD panel o un Web panel que nosotros estamos acostumbrados a trabajar, con la diferencia que es mas sencillo ¿Por qué? Porque tiene solamente lo relacionado al diseño, esto no tiene código, es solamente propiedades de diseño.

Entonces, nosotros acá definimos el layout de cómo queremos que se vea un orador en ejecución y nos va a quedar de esta forma, después ¿cómo lo cargamos, como cargamos esas variables, como se carga de datos y todo lo demás? Lo referente a código queda separado, entonces esa es una de las grandes diferencias y por esa misma cualidad es mucho más fácil de compartir.

- Ok ¿y es valido para Web tanto como para Smart Devices?
- Es válido como para Web como para Smart Devices...
- Gracias!

Muchas veces pasa (como anécdota) a veces estas con un colega y te dice "ah ¿me podrías pasar estos paneles que están en tu aplicación? Y en realidad lo que quiere es el diseño y vos le tenés que pasar un XPZ con el código y con todo, entonces, esto facilita mucho más el tema de compartir ese tipo de cosas.

Inserting an Stencil on my layout



Entonces, acá tenemos el objeto stencil nosotros creamos el diseño de como queremos que se vea (se va a ver de esta manera) y luego para insertarlo en mis lay out lo que tengo que hacer es ir a mi SD Panel o a mi Web Panel (lo que sea) y buscar en la tool box de stencils buscar el stencil que quiero insertar y simplemente con el botón de INSERT lo agrego al Layout. Y ahí sí, es lo que tu decías, ahí sí le puedo empezar a codificar como manejo ese comportamiento.

- ¿Ese stencil se puede crear a partir de un diseño sketch?
- Sí, ahora mas adelante vamos a verlo

Entonces, esta sería la "Tool Box" digamos, para instertar stencils, como pueden ver, contamos con una ScreenShot para que nos sea mas fácil identificar que es lo que estamos agregando a nuestro layout, entonces con el botón Insert lo insertamos. Lo insertamos en este panel como lo insertamos en "n" paneles que lo queramos usar y como les decía... después modifico simplemente el stencil y queda el cambio impactado en todo.

Stencil versions

V.1



Sebastián Gómez
GeneXus

V.2

Sebastián Gómez
GeneXus



V.3



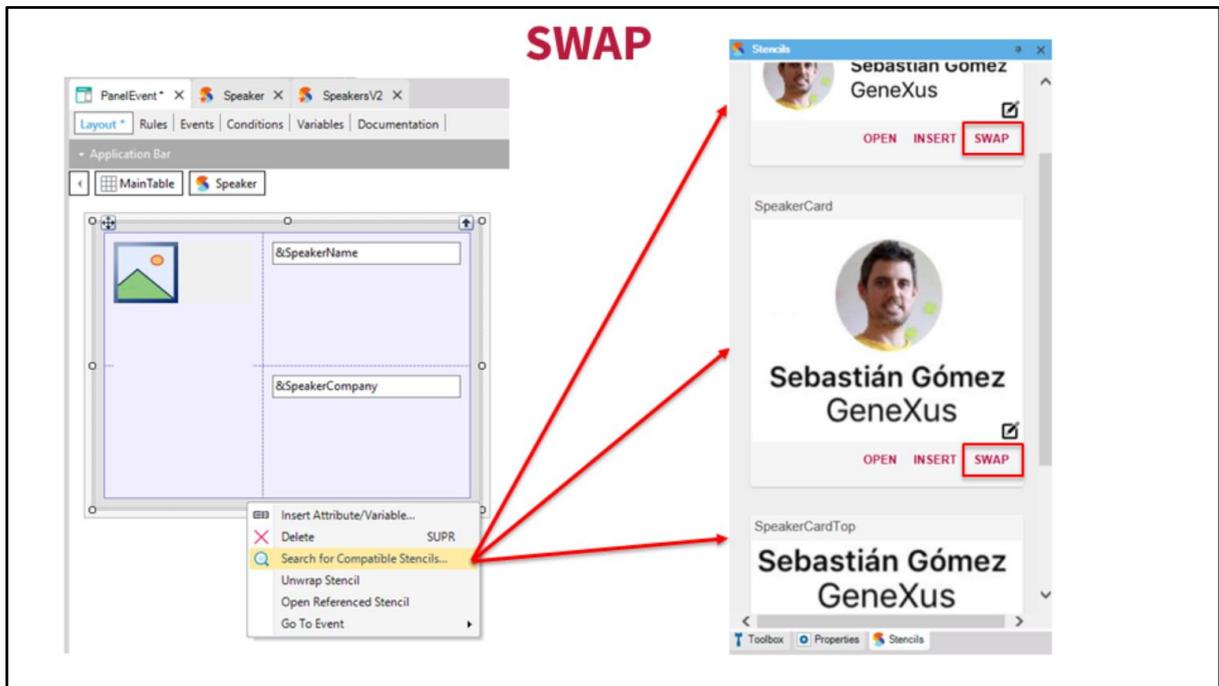
Sebastián Gómez
GeneXus

V.n...



Lorem Ipsum

Los stencils también nos ayudan al tema del versionado ¿Qué pasa si nosotros estamos trabajando con gente de diseño y nos dan nuevas ideas o incluso el cliente quizás dice, me gustaría mas que los oradores se vean de esta otra forma? Entonces nosotros podemos tener distintas versiones de como mostrar ese lenguaje visual que yo les decía, de como mostrar un orador. (ahí lo tenemos, con texto a la derecha, arriba, abajo) y “n” cantidades de formas en la cuales nosotros podemos mostrar un orador. Entonces con los stencils podemos crear distintos stencils con cada una de estas representaciones...



... y para hacer el intercambio en nuestros paneles es bien fácil, simplemente tenemos que volver al panel que yo les había mostrado recién que ya tiene el stencil insertado, le damos "botón secundario", "Search for Compatible Stencils" y me va a mostrar de vuelta la Toolbox con los Stencils compatibles con ese esquema.

Y acá vemos todo lo que les mostraba anteriormente (todos estos de aca)

Entonces yo lo único que hago es elegir cual quiero y presionar sobre el botón SWAP, y eso lo que hace es el intercambio del stencil.

- ¿Entonces para cuando vos haces una actualización tenés que ir panel por panel para actualizarlo? Ahí mostraste tres versiones diferentes del mismo Stencil, la pregunta es: si vos quieres que aplique... tenés aplicado todo sobre el Web Panel con el Stencil 1 (con la versión 1), cuando quieres pasar todo a la versión 3 ¿tenés que ir panel por panel para actualizarlo? ¿No hay una forma de que te actualice todos?
- Si, que tenga entendido... capaz que Juan nos puede ayudar...
- Si estamos usando el 1, actualizamos el 1 y se actualizan todos, si queremos cambiar al 3 hay que ir uno por uno, pero la fichier de cambiar todas las versiones del 1 por la 3 me encanta y la vamos a agregar en cuanto podamos.

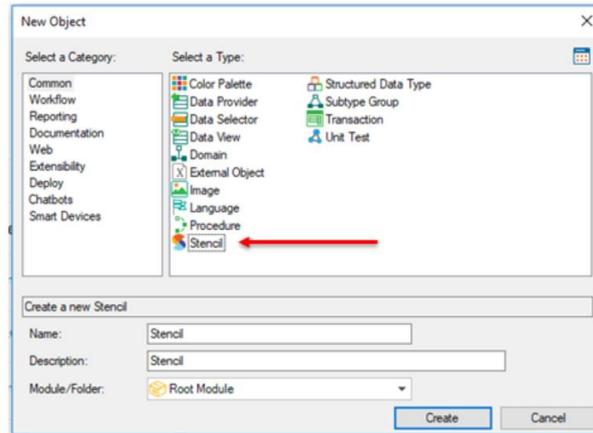
How do I create a new Stencil?

1. Create new stencil object
2. Wrap as stencil an existing design
3. Import from Sketch 

Bien, entonces seguimos.

Las formas de crear Stencils, nosotros estábamos viendo (ahora con el ejemplo de la charla del evento), el punto número 1: Crear un Stencil desde cero

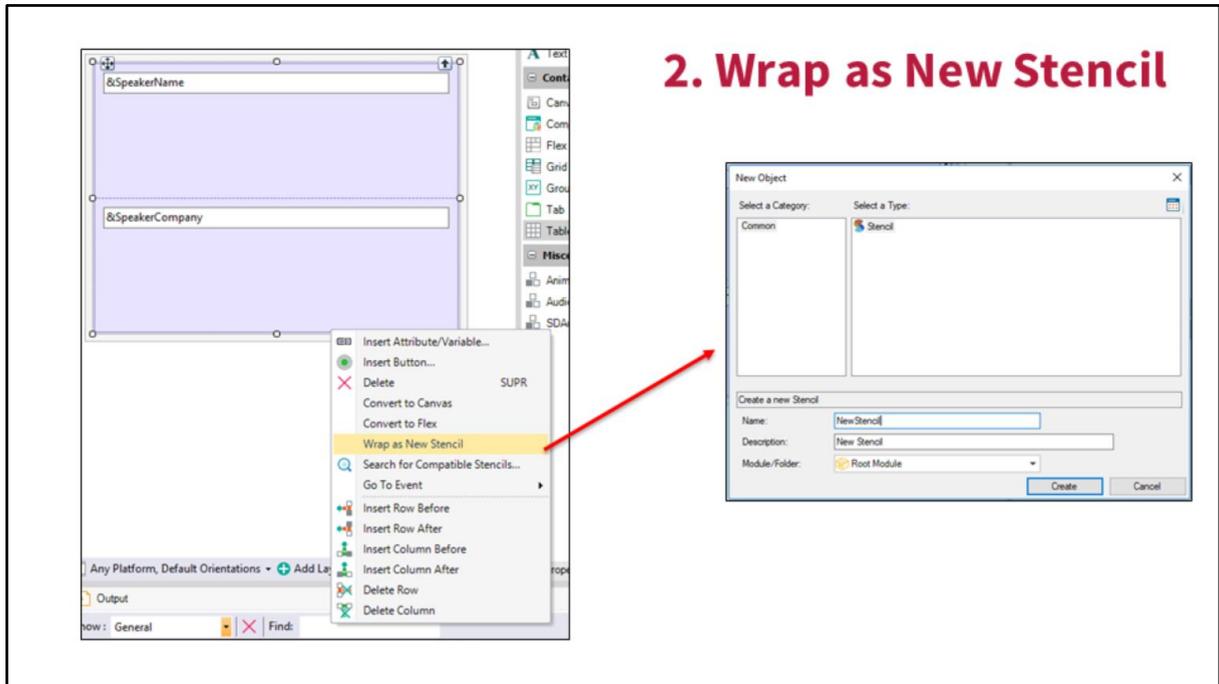
1. Create new Stencil



Ctrl+N(ew) y creo un Stencil vacío y a partir de ahí empiezo a crear el diseño con un layout vacío.

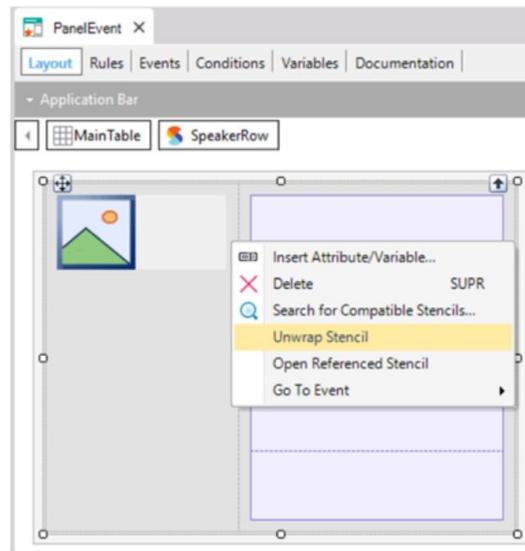
Hay otras formas de crear un Stencil por ejemplo, con una base de conocimiento que nosotros ya tenemos, que ya tenemos diseño aplicado y queremos ese diseño encapsularlo y poder reutilizarlo.

2. Wrap as New Stencil



Entonces, lo que deberíamos hacer es ir a las secciones de layout que queremos encapsular, como botón secundario ponemos “envolver como Stencil” (Wrap as New Stencil) y automáticamente nos sale este diálogo para customizar el nombre y guardarlo. Y eso lo que hace es desacoplar el diseño del código y lo guarda como un Stencil para que nosotros lo podamos reutilizar en cualquier pantalla como les comentaba antes.

Unwrap Stencil



Lo mismo para el otro lado, es decir, ¿Qué pasa si yo estoy trabajando con un Stencil y quiero sacárselo, cortarle el cordón umbilical del Stencil a mi Layout?
Lo que tengo que hacer es, simplemente al Stencil aplicado hacerle un “desenvolver Stencil” (Unwrap Stencil), lo que pasa ahí automáticamente es que ya pierde esa conexión con el Stencil (si yo modifico el Stencil, no se me va a modificar esta pantalla) pero me va a mantener todas las propiedades de diseño que ya tenía aplicadas, pero yo ahora voy a poder trabajar con esos controles libremente, totalmente desacoplado del Stencil.

3. Import from Sketch (Coming soon)



GeneXus[®] 16
The power of doing

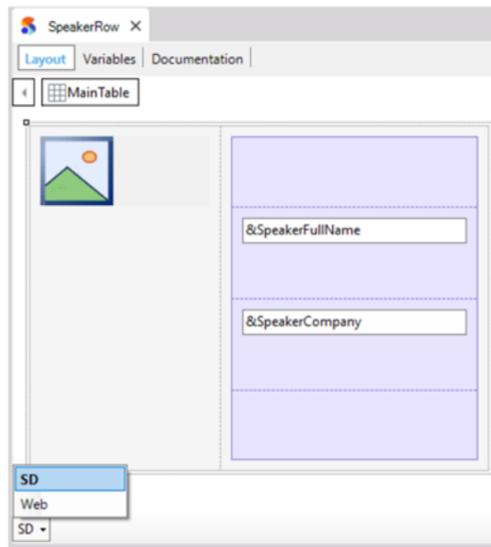


Y un poco lo que preguntabas tu hoy, la Tercera forma que próximamente se va a venir para poder crear Stencils es a partir de las extensiones que nos pasan los diseñadores de por ejemplo "Sketch".

Entonces, aquí a la izquierda pueden ver el archivo Sketch con las pantallas y su diseño, también con sus navegaciones y lo que voy a hacer desde GeneXus es un IMPORT de ese archivo.

Entonces, lo que va a hacer GeneXus es: me va a crear los SD Panels, me va a crear los Stencils y me va a crear las navegaciones entre esos SD Panels, entonces yo voy a poder ejecutar con el diseño "al desnudo" sin código por atrás. Lo único que me quedaría es codificar el comportamiento, pero el diseño lo estaría trayendo y estaría creando los Stencils.

SD & WEB



También se preguntó por ahí si esto es para SD y para Web.

Los Stencils tienen Layout, porque generalmente nosotros si bien la presentación de una página web o de una aplicación SD no son las mismas, el concepto que hay por detrás, de lo que se quiere mostrar sí es el mismo, yo los oradores los voy a querer mostrar capaz de una forma web y en otra forma SD pero el concepto es el mismo, entonces tenemos un layout para mantener esa jerarquía de Stencils (tenemos un layout SD y un layout Web) que los podemos modificar por separado.



+ Modifiability
+ Reuse
+ Flexibility

“Stencils help to define standards and consistency, simplifying the implementation of Design Systems.”

En resumen, los Stencils definen un estándar, nos dan consistencia, nos ayudan muchísimo a crear Design Systems, aumentan muchísimo la modificabilidad por eso que hablábamos, el re- uso sin duda, facilita muchísimo el compartir diseño y nos da una flexibilidad enorme. Entonces los invito a probarlo y ahora Fede les va a hablar un poco sobre User Control.



User Control Object

Bueno, respecto al User control Object

para los que tengan experiencia desarrollando User controls sabrán que es un concepto bastante conocido ya y ya lo venimos trayendo desde varias versiones anteriores.

El User Control Object si se quiere es una nueva forma de modelar esos controles que no suplanta a la forma anterior sino que, los User Controls como los conocemos van a seguir existiendo, pero User control Object presenta una serie de ventajas que facilitan esa integración con el código nativo, con el modelado de UI y UIX.

No se si tuvieron la oportunidad de ver (yo vi una presentación ayer sobre el tema) los invito a verla, van a tener muchos detalles sobre el objeto.

Pero básicamente lo que permite esto es facilitar la forma de modelar el User control, tiene una serie de ventajas, que por ejemplo al ser un objeto de la KB GeneXus es Built-in, por ende se resuelve un montón el hecho del modelado del control, su mantenimiento.

Para los User Controls anteriores teníamos los User control Editor como parte de la suite GeneXus, que permitía el modelado del control pero era una herramienta satélite, el control no era parte de ese conocimiento de GeneXus, sino que había que modelarlo por fuera, había que incorporarlo a la KB, después instalándolo en nuestra instancia de GeneXus y eso era un trabajo bastante tedioso y había que ser muy meticuloso porque podía fallar. Entonces el hecho de que sea Built-in ya facilita bastante las cosas.

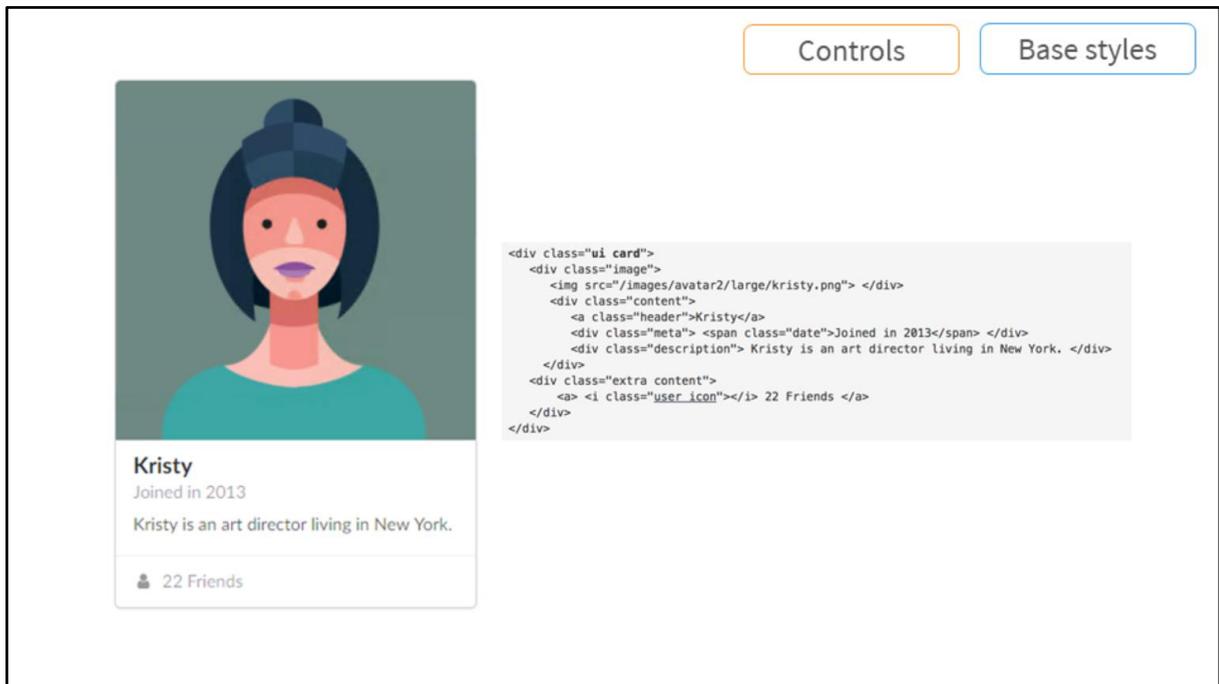
En Segundo lugar, la portabilidad aumenta muchísimo, es un objeto mas de la KB, por lo tanto si estamos trabajando con un esquema de SCM con GeneXus Server va a ser mucho mas fácil también.

En el caso anterior de los User Control, si estábamos trabajando con un User control en la KB en la instancia local de GeneXus y hacíamos un COMMIT al Server y ese Server no tenía instalado el control, nos iba a salir un mensaje del estilo "no conozco esta estructura, por favor revisa que es lo que pasa" "instálalo en su caso y después podemos seguir trabajando".

Nos vamos a olvidar de este problema, ya que es un XPZ mas, el esquema de consolidado de GeneXus Server funciona de la misma manera que con cualquier otro objeto, por lo tanto la portabilidad es muchísimo mejor, la distribución también, es solamente un XPZ, hacemos un Export y lo podemos distribuir, no tenemos que Zippear, no tenemos que mantener versiones por fuera, es realmente mucho mas fácil, es mucho mas potente, tenemos histórico, tenemos

el control del versionado, tenemos cross reference para saber quien hace uso del control y a quien referencia ese control, por lo tanto es muchísimo mas simple de usar.
Por otro lado también puede utilizar el Base Style (el Base Style que comentamos antes) que le va a dar esa presentación que el User Control necesita, basándonos en CSS frameworks que vamos a incorporar en la KB.

Por otro lado, estos User Controls presentan la particularidad de que generan server side, el HTML que nos va a servir el control va a ser generado por el server, por lo tanto le delegamos la responsabilidad al servidor de presentar ese control de forma óptima y de forma rápida, es mucho mas fácil escalar el servidor que escalar el browser del cliente, por lo tanto le delegamos la responsabilidad del dibujado del HTML y ese dibujado se traduce en "Load performance", va a ser mucho mas rápido, va a ser mucho mas optimo.
Y esto para objetos que tienen una interfaz muy rica, que hacen mucho uso de controles, realmente hace la diferencia.



Ahí lo que vemos es un Control Card (Card control) de SemanticUI que lo que hace básicamente es mostrar información de usuarios, de clientes o de lo que sea, de una forma mucho mas linda de lo que estamos acostumbrados a ver.

Muestra la foto da la persona, su nombre, una fecha (en este caso de cuando se unió a una red social, por ejemplo) y una breve Bio de la persona.

Esto que vemos aquí, se modela con estas 10 líneas de código HTML, que nos provee CSS framework de Semantic Provider.

Entonces vamos a ir a la pagina de Semantic UI, vamos a buscar el control card, lo que vamos a hacer simplemente es copiar este código HTML, lo vamos a pegar dentro de la estructura de nuestro User Control (que ahora les voy a contar como es la estructura) y automáticamente al utilizar este user control en un web panel o en un objeto que tenga interfaz gráfica lo vamos a ver de esta misma forma.

Lo que vemos ahí es que este control tiene clases definidas que son propias de SemanticUI, y esas clases van a estar disponibles dentro de la KB GeneXus cuando utilicemos el Base Styles, como vemos acá arriba, el Base Styles esta directamente relacionado con el user control.

El Base Styles lo podemos utilizar tanto en user controls como en el objeto tema (Theme Object)

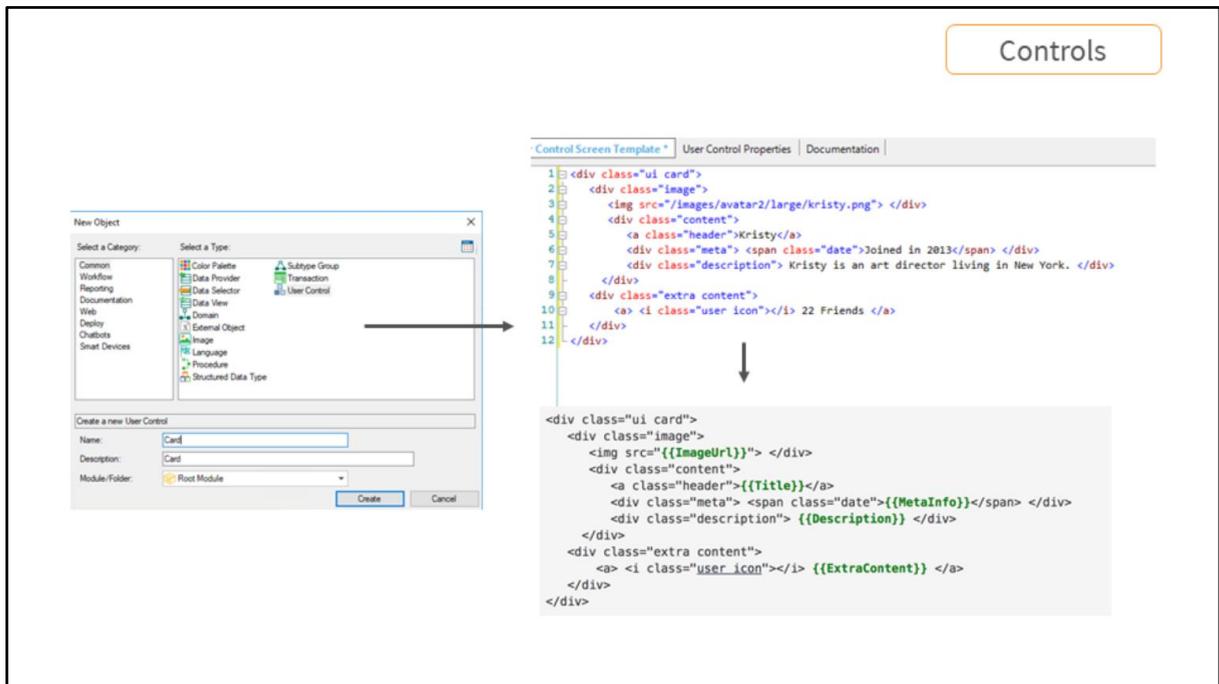
Agregamos una nueva propiedad que permite referenciar a esos CSS frameworks dentro de la KB.

La forma que tenemos de referenciarlos es utilizando el file de la KB (que seguramente ya lo conozcan también) y lo que vamos a hacer es un ZIP de ese Set de archivos que nos brinde el CSS framework o nuestros diseñadores, porque el resultado final que generan nuestros diseñadores es básicamente el mismo que tiene el CSS framework, es una serie de CSS y tiene algunas otras estructuras estáticas que son las que vamos a utilizar , entonces lo que hacemos ahí es: le ponemos una extensión que es GXLibrary (esto lo vamos a ver en el practico después para que tengan mas detalle de como funciona y van a ver que es realmente muy fácil) pero vamos a poner una extensión que es gx.library, y en ese momento al utilizarlo como file de la KB GeneXus se va a dar cuenta de que tiene que proveer esos estilos para poder utilizar mediante una propiedad en nuestros controles.

- Mi consulta es: nosotros hacemos referencia a clases ¿cierto? Están en una librería, esa

librería va a estar empaquetada en una nube... el tema es... ¿crece demasiado el tamaño de la KB (imagina cuando tu la construyes, para poder correrla en forma local desconectada de la red) esas clases son incorporadas a la KB?

- Correcto! viste que CSS trabaja por referencia de nombre y tu cuando defines el HTML y la clase que estas utilizando ahí la tiene definida dentro del CSS framework que estás utilizando, en este caso Semantic UI, ya automáticamente la va a tomar a la clase.
- ¿Va a tomar esa clase?
- Va a tomar esa clase! Pero tu estas importando todo el set de clases de Semantic UI, estas importando el CSS completo, no debería ser muy pesado ese archivo estático, pero sí, puedo aumentar un par de kb (kilo bites) el tamaño.
- ¿Esto es solo para web?
- Sí, es web
- ¿Y se podrá hacer mas adelante para SD?
- Lo que pasa que tenemos otras estructuras que pueden modelar una serie de cosas en SD y en realidad la necesidad inicial fue Web, no solo por el modelado UI, sino también porque tiene mucha potencia de Script-in esto (después les voy a comentar) y permite una integración con muchas apis red que en web son muy utilizadas, por eso la primer intención es desarrollar lo web y después veremos, pero no me imagino ahora un caso muy puntual en SD del user control, lo podemos evaluar, pero en primera instancia Web para dar potencia de modelado y potencia de script-in para integración de apis y veremos como madura y las necesidades que surjan.



Lo que les comentaba aquí es cómo se hace la creación del user control.

Es un nuevo objeto de la KB, vamos a hacer Ctrl+N (el diálogo nuevo objeto) le vamos a definir un nombre y lo que vemos aquí a la derecha es la estructura que yo les comentaba.

La anatomía de este nuevo objeto básicamente consta de dos partes: el Screen Template y las Properties

El Screen Template es el que va a permitir ese modelado nativo, nosotros decimos "modelado nativo de UI y UIX" en GeneXus y esta sección de user controls es justamente la que nos permite eso.

Si se fijan aquí, este código HTML es exactamente el mismo que estábamos utilizando en las card de Semantic UI dentro de GeneXus, ya está incorporado en la base de conocimiento y ahí lo que vemos es que tiene datos que son estáticos, vemos una ruta a una imagen, vemos el nombre "Header Code" ahí adentro del Tag "a", vemos los datos que les comenté antes y lo que vemos aquí abajo es cómo permite el user control el dinamismo de esos datos, porque lo que queremos justamente es utilizar datos de atributos de nuestras transacciones, de variables que tengamos definidas en la KB, etc

Esta sintaxis de aquí que se llama "moustache" por la llave que es similar a un bigote (por eso se llama moustache) es la que nos va a permitir ese dinamismo en los datos del control.

Nosotros al definir esta sintaxis aquí, automáticamente estamos diciéndole a GeneXus que puede ser un dato dinámico, que podemos inicializar después en un evento GeneXus como cualquiera, por ejemplo en el Load del web panel, yo le voy a decir:

"ControlCard.propiedadImageUrl", la Url de la imagen.

SemanticUI

- BSBreadcrumb
- BSPanel
- ButtonLike
- Card



Facundo
Nacio en 2011
Minecraft fan
 No tiene Facebook

```
Event Start
  card1.Title = "Facundo"
  card1.MetaInfo = "Nacio en 2011"
  card1.Description = "Minecraft fan"
  card1.ExtraContent = "No tiene Facebook"
  card1.ImageUrl = "https://genexususa.com/wp-content/uploads/2017/08/Gaston-Milano-GeneXus-CT0-cropped.jpg"
EndEvent
```

Acá lo que vemos son una serie de controles dentro de la ToolBox, al crear el user control automáticamente GeneXus lo pone como parte de la ToolBox, como parte de controles Built-in.

Acá tenemos la inicialización que les comenté antes, en el evento style inicializa el título que quiero que se muestre para la persona y los datos que vimos en el ejemplo anterior. Y esto es como se genera el resultado en ejecución.



+ Portability
+ Load Performance
+ Easy to extend & share

¿Qué podemos concluir de este objeto user control?

En primer lugar, la portabilidad (Portability) que ofrece, por lo que les comenté antes, por el tema de la facilidad de trabajo con GeneXus Server, porque es Built-in, porque es muy fácil de distribuir y de mantener, porque es una estructura más de GeneXus.

Una mejor performance en la carga (Load Performance) porque las aplicaciones no solamente tienen que lucir bien sino que tienen que responder rápidamente porque el usuario tarda menos de un segundo en decidir si quiere seguir navegando ahí o no.

Y en tercer lugar, muy fácil de extender y de compartir (Easy to extend & Share) al igual que el objeto stencil.

- ¿Se podría meter uno de estos controles dentro de una grilla?
- Se podría meter...
- ¿Se podría meter varios controles en la grilla?
- Sí, puedes iterar dentro de un objeto GeneXus o puedes iterar dentro del mismo User control usando un SDT, las propiedades que yo les comenté del user controls, son las que estamos acostumbrados a ver en GeneXus, puedes definir una propiedad como SDT, puedes definir una propiedad como numeric, como stream, puedes incluir el control dentro de un grill o puedes iterar dentro del mismo control usando un SDT, o sea que tenés un ciclo ahí... iterativo.
- Hola! Una pregunta, no se si entendí bien pero, ¿se pueden anexar archivos Java Script al User control?
- Sí sí, se pueden... esto es un OverView básico de lo que es el objeto, pero sí, tiene mucha potencia de Script in, dentro de la sección property que yo les comenté, lo que hace GeneXus es inicializar la propiedad que tu defines dentro del Screen Template, le pone la entrada en el archivo, le define el tipo y a su vez dentro de esa sección tu puedes definir Script, utilizas el Tag Script como estas acostumbrado y ahí es donde tu defines tu Java Script para poder integrarte con algunas Apis.
- Perfecto! ¿y se va a poder utilizar lo mismo que se utilizaba en los user controls anteriormente que eran los archivos compartidos, los Shared? Por ejemplo para Jay Query que lo utilizan...
- Sí, podrías hacer una referencia a archivos de Jay Query o lo que sea que estés utilizando.

- Mi pregunta era la misma, si van a poder migrar los controles que hoy ya tengo y uso de Marketplaces.
- Sí, correcto, se van a poder migrar, la idea es comenzar en algún momento a convertir ciertos user control a este nuevo esquema justamente para facilitar y tener la potencia que el objeto provee.
- Hola, a mi me quedó una consulta sobre lo de Sketch...si va a estar disponible también para web la importación desde Sketch
- La importación desde Sketch para web... juan a ver si nos puede contestar...
- En principio estamos para SD (que todavía no salió pero va a salir en venta dentro de poco) no hay nada que nos impida hacerlo para web pero para vos tenemos justamente User Controls, entonces no está en prioridad por ahora, como querer queremos, pero primero en prioridad para SD que de ultima si tenés diseñadores pueden hacer HTML Java Script y lo inscribis como user control.
- Yo quería hacerte una pregunta sobre el Deploy de user control, o sea.. Suponete, vos referencias alguna librería y ¿cuando hagamos el Deploy de esa aplicación, (si vamos a llevar esa librería), si hay algún built-in...?
- Sí, debería llevarse. Respecto a las referencias hay varias opciones, podes tener referencias a cosas que tengas dentro de tu modelo, a archivos estatic como por ejemplo JQuery o podes usar CDN también, podes usar referencia a URL de esos archivos que estén en algún lugar y tu los podes referenciar, en ese caso la referencia al CDN no debería llevar nada adicional en el Deploy, en el caso de que tu utilices ciertos archivos estáticos en tu modelo o si lo va a llevar.
- ¿Pero cómo se entera GeneXus?
- Bueno, el motor de Deploy ahora lo estamos trabajando justamente para que calcule las referencias y pueda llevar esas dependencias, como en el caso del Deploy a tomcat que hacemos cuando prototipamos local, la idea es que lleve esa Base Library y las dependencias que tenga el control.

**Streams of information
and more about grids**

Flex layouts



Les vamos a contar un poquito sobre los Flex Layouts que hemos estado trabajando también y que los Flex Layouts básicamente lo que permiten es mayor flexibilidad. Como les comentó Armin en el OverView, la idea es no solamente trabajar con la potencia que nos ofrece Bootstrap, en el concepto de la división de la pantalla en doce columnas y cómo tienen que organizarse los datos cuando yo juego con los tamaños de pantalla. Esto es complementario y da una flexibilidad mayor ya que permite (como decía Armin antes) este tipo de interfaces.

Web/SD

GeneXus



Flex layout container, Flex Grid

... donde no sabemos exactamente cual es el ancho que tiene cada uno de los elementos, pueden tener ancho variable y también la posibilidad de calcular y de mover esos elementos en base al tamaño de la pantalla.

También no solamente mover el elemento, todo el elemento en sí, sino también de poder ajustar particularmente las tablas y el contenido de cada elemento en base al tamaño. En este caso, en una ejecución web vemos que tenemos cada una de las fotos con sus acciones (a la derecha) y aquí en esta pantalla mas chica las acciones se bajan y se ponen debajo.

Con eso hablamos de Flex, con ciertas funcionalidades que tiene CCS (en el caso de Web) que nos permite modelar este tipo de interfaces de una forma mucho mas potente a la que hizo Bootstrap inicialmente con la división de 12 columnas.

Web
Flex layout container, Flex Grid
GeneXus

Flex Direction

Flex Wrap

Justify Content

Para el caso de web, lo que tenemos es:

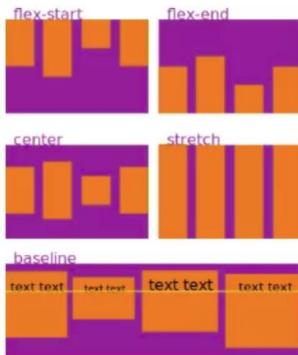
Una nueva forma de visualizar los grids (un Custom Render de un grid) que ahí permite definirlo como FlexGrid y al momento de habilitar esa propiedad lo que hace el ID es disponibilizar estas otras propiedades que dicen justamente cómo se tiene que comportar ese Flex Layout, la dirección que va a tener, si es por fila, si es por columna, si es fila invertida o columna invertida (que son estas cuatro)
 Cómo se hace el wrap-in de los elementos.

¿Qué pasa cuando yo tengo una cierta fila con una cantidad de elementos y no es suficiente el tamaño para mostrarlos todos?
 Por ejemplo aquí, lo que está haciendo es “tirarlo a la próxima”

¿Cómo se va a justificar el contenido? si comienza de izquierda a derecha, de derecha a izquierda, si va a estar centrado, si vamos a dejar espacio entre los elementos.
 Acá hay un montón de opciones que nos permiten a nosotros justamente esa flexibilidad.

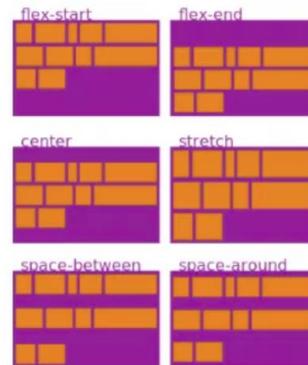
Align Items

It defines the alignment of the items in the rows.



Align Content

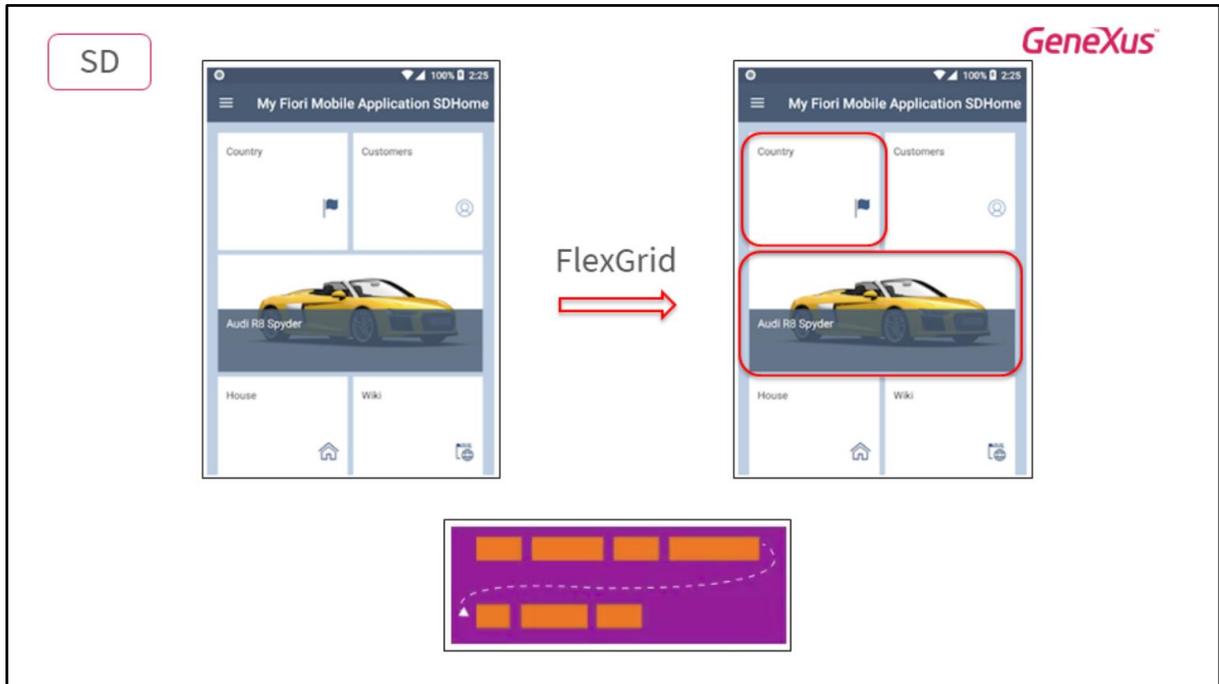
Aligns the row when there is extra space in the container.



Más opciones dentro de Flex Layout...

Cómo queremos la alineación de los ítems, esto es básicamente cómo se alinean los ítems en las filas

Cómo vamos a alinear las filas cuando hay espacio extra en los contenedores, O sea, si nos sobra espacio podemos visualizar ese contenido de distintas maneras.



En lo que respecta a SD, aplica el mismo concepto de lo que estaba hablando Fede para Web, exactamente igual.

El caso canónico en SD es cuando queremos mostrar componentes con ancho variable.

Este es un ejemplo en el que tenemos este tipo de componentes que son de un determinado ancho, y después esto otro, que en este caso ocupa todo el ancho de la pantalla, entonces, se van a ir acomodando con el FlexGrid (se van a ir acomodando según el espacio del contenedor se lo permita).

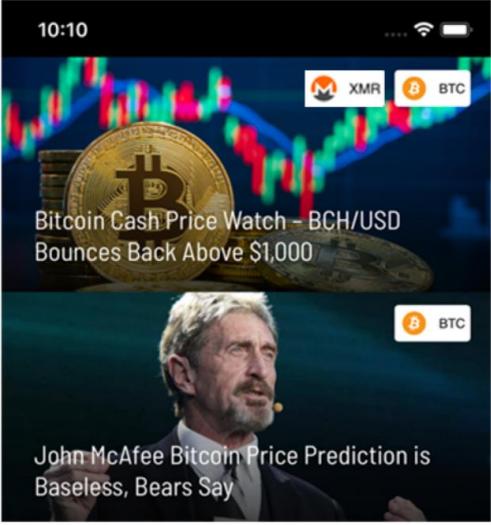
Si esto lo vemos en una tableta, se acomodarían en una sola fila y cuando se les termina el espacio, siguen para abajo.

Por ejemplo con la propiedad FlexWrap (en el dibujo se ve el comportamiento de cómo va cargando los items.)

SD

BitBitNews

GeneXus
Flex + Nested Grids
Justify Content



The screenshot shows a mobile app interface with a dark theme. At the top, there's a status bar with the time 10:10 and icons for XMR and BTC. Below that is a news article with a colorful candlestick chart background. The headline reads "Bitcoin Cash Price Watch - BCH/USD Bounces Back Above \$1,000". Below the chart is a photo of John McAfee with the text "John McAfee Bitcoin Price Prediction is Baseless, Bears Say". There are also small icons for XMR and BTC in the top right corner of the article area.



&Categories.item(0).CoinCode

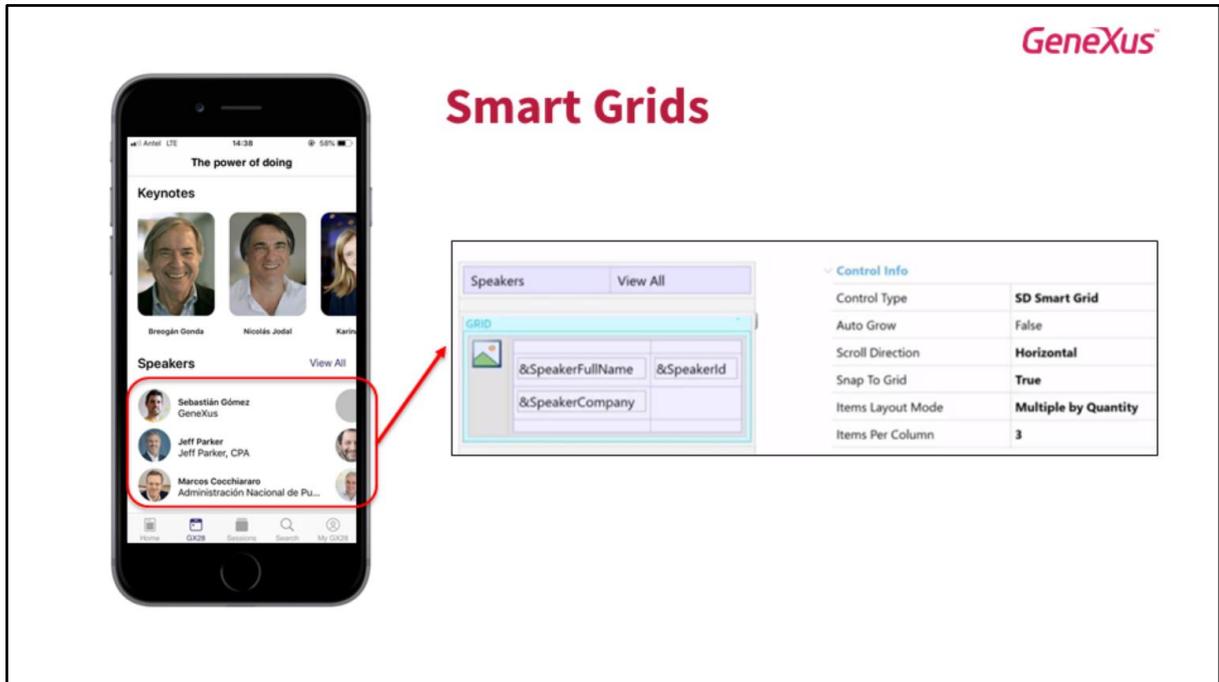


&ArticleTitle

Entonces, como para aportar algo mas... en el Wiki podemos ver ejemplos de la aplicación "BitBitNews" (es una aplicación de noticias sobre criptomonedas)

Tenemos un Grid con las noticias de las criptomonedas y anidado tenemos otro Grid que muestra la criptomoneda asociada a esa noticia
 Entonces como nosotros no sabemos la cantidad ni el tamaño, decimos que con "Justify Content" se van a ir agrupando esas criptomonedas de derecha a izquierda en la parte superior de la noticia.

Smart Grids



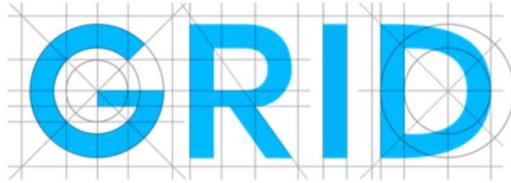
Un poco más sobre Grids...

También en la V16 tenemos los Smart Grids, que lo que hacen es darle mucho más potencia a los grids que nosotros veníamos trabajando.

Este es un ejemplo de la KB del evento con los oradores, el comportamiento de ese grid de oradores, es paginado horizontal hacia la derecha, en el cual se puede ver como un hint, (una pista de que hay navegación hacia la derecha, porque se ve un pedacito del ítem que está oculto) se cuentan como pueden ver, tres ítems (tres oradores) por columna y ese paginado.

Antes, si lo queríamos hacer de esa forma (se podía hacer) teníamos que codificar mucho más, ahora es simplemente como la imagen que ven aquí a la derecha, aplicando SD Smart Grids con el "Scroll Direction" Horizontal y con "Items Per Column" (ítems por columnas) 3, entonces, solo con esta programación nosotros ya tenemos los resultados que ustedes ven ahí en la pantalla.

More about...



Algunas cosas mas...

More about Grids

SD

- Unique property for grids (Web & SD)
- Inverse Loading property
- Grid Refresh method is available on SDPanels and Work With for Smart Devices
- Improvements to Grids with Multiple Selection for Smart Devices
- ShowPageController property in runtime

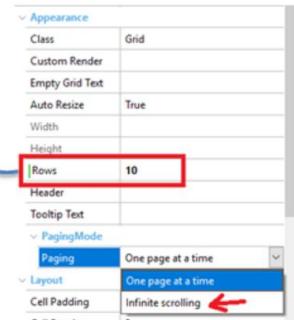
GeneXus

- La "Unique property para Grids (tanto en Web como para Smart Devices) se agregó.
- "Inverse Loading en los grids, que es la carga inversa, un caso de uso podría ser en una aplicación de chats en el cual los mensajes se cargan al revés.
- "Refresh method" a grids en SDPanels, tanto como en Work With for Smart Devices.
- Arreglos y mejoras en lo que es "Multiple Selection for Smart Devices"
- El cambio de "runtime" de lo que es la propiedad ShowPageController.

More about Grids

Web

- Unique property for grids
- Inverse Loading property (when **Rows \neq 0**) \rightarrow use case: chat
- It's possible to set up a Fixed Grid header with vertical scroll
- Header property and Header Text Class property (for grids and tables)
- Select method and Deselect method



GeneXus

Algunas cosas mas sobre Grids en Web.

Sobre los Grids en web, también mejoras que hemos hecho en esta última versión:

- Propiedad Unique
 - Inverse Loading, teniendo esta consideración, cuando las Rows son distintas de 0 (también el caso de los chats.)
 - Es posible para setear un Header Fixed en los grid. (El tema de la dirección de los grids y el modelado de los estilos va a ser bastante mas simple ahora)
 - Header property and Header Text (que son dos nuevas propiedades)
 - Y Select y Deselect que también hemos hecho muchas mejoras ahí.
-
- Respecto al tema de los CSS frameworks, estabas hablando de que se subía el zip como archivo y de ahí se tomaban las referencias de las clases, etc. ¿va a haber algunos que ya estén incluidos en GeneXus? ¿que ya podamos meter el HTML y ya GeneXus lo tenga dentro de sus clases o referencias?
 - En principio no, porque no vemos necesidad de ya proveer cosas que sobrecarguen las KB partiendo de cero y eso lo dejamos a decisión del usuario...
 - Pero por ejemplo, BootStrap que ya está incluido en el empaquetado de GeneXus...
 - Sí, pero BootStrap la particularidad que tiene es que nosotros lo utilizamos justamente para resolver el tema de las columnas responsivas, por eso es que nosotros utilizamos BootStrap inicialmente con todo su set de cosas, no solamente los Script que resuelven (como tiene que comportarse la aplicación en base al tamaño) sino también, alguno estilos que ya estaban aplicados a controles que nosotros les distribuíamos. Ahora lo que hicimos fue desacoplar eso, dejar la potencia de BootStrap para calcular los tamaños de los diferentes controles en las interfaces y permitir integrar otros CSS FrameWorks que pueden coexistir con esa lógica pero no con el estilo de Bootstrap.
 - Ok, ¿entonces nosotros nos haríamos responsables de mantener la versión del FrameWork que queremos?
 - Correcto! Dependiendo de que controles quieras utilizar y de cual sea tu proveedor (en este caso Semantic UI) tu vas a importar ese CSS Framwork y vas a utilizarlo dentro de los controles.
 - Hoy decían de que cuando vas a importar un objeto, podrías hacer una referencia a un

CSS Framework o incluso a un CSS hecho por un diseñador sobre ese CSS Framework ¿Qué pasa si el diseñador utiliza otras clases que no son de CSS Framework (o sea, son clases definidas por él) al momento de la importación? ¿Eso reemplaza las clases que ya tengo en el tema que tengo adentro de la KB de GeneXus? ¿o como funciona en ese caso?. En nuestro caso en particular, trabajamos con diseñadores que arman un CSS y después esa consolidación es un proceso manual que tenemos que hacer...

- Claro! Ahí dependen muchas cosas del CSS mismo, si tu utilizas los mismo nombres de clase (si tu diseñador utiliza los mismo nombres de clase) que tu ya tenías definida, ahí probablemente va a haber un conflicto y tu vas a tener que definir prioridades, la propiedad de la clase para que una "le gane" a la otra, pero las clases las puedes complementar, puedes utilizar clases de tu diseñador o del CSS Framework y adicionarle clases propias tuyas del tema, con la concatenación de clases que permite CSS (un espacio, el nombre de la otra clase) y creo que está bastante bien resuelto ya que con esta potencia tu vas a poder tener un modelado bastante más simple de lo que es ahora.
- ¿No hay un proceso de importación de un archivo externo con una consolidación?
- No!, El CSS que genera tu tema es independiente al que va a utilizar el CSS Framework, el CSS Framework al hacer el unzip GeneXus, va a definir ese archivo estático que se va a llamar, por ejemplo: SemanticUI.CSS y va a ser independiente del que genera tu objeto tema.
- Gracias!

Mobile & Web



Multi-experience & Omnichannel

- Design systems
- Streams of information
- **Conversational UI: Chats/ Chatbots**



Web/ Web mobile

Native mobile

Chatbots

Wearables/TV/AR

GeneXus[™]
The power of doing