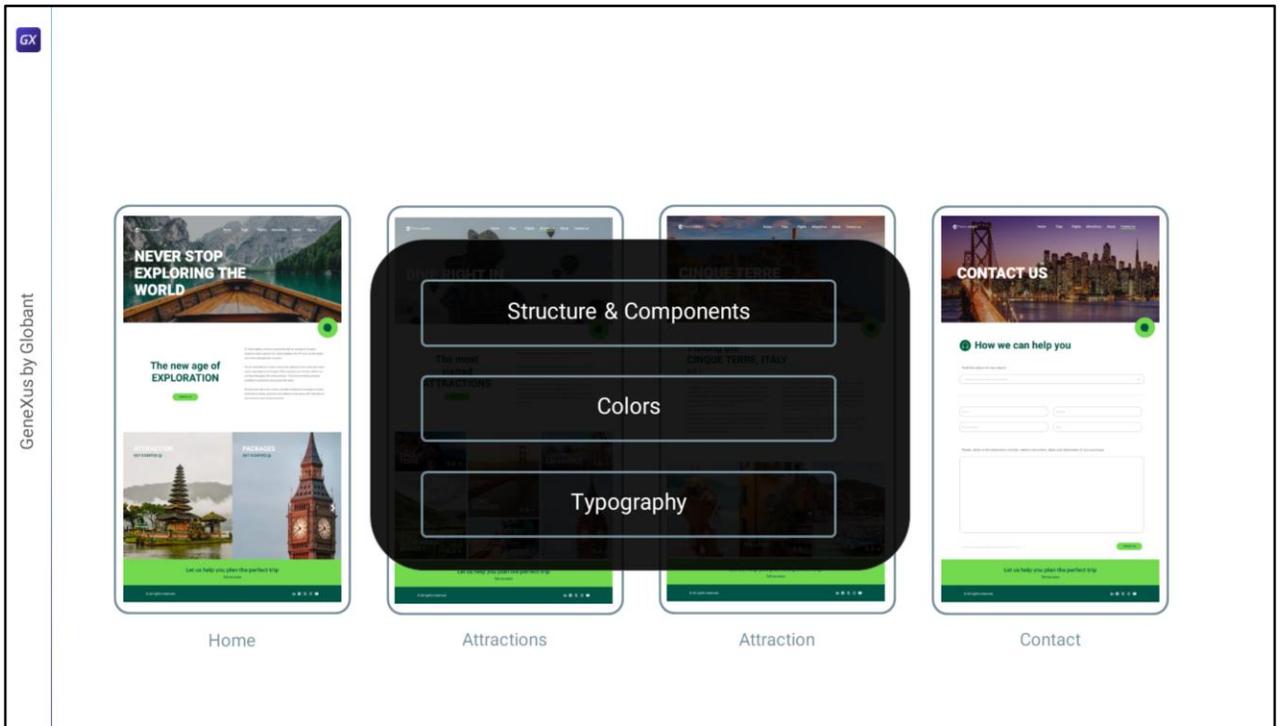


Design in GeneXus. Players

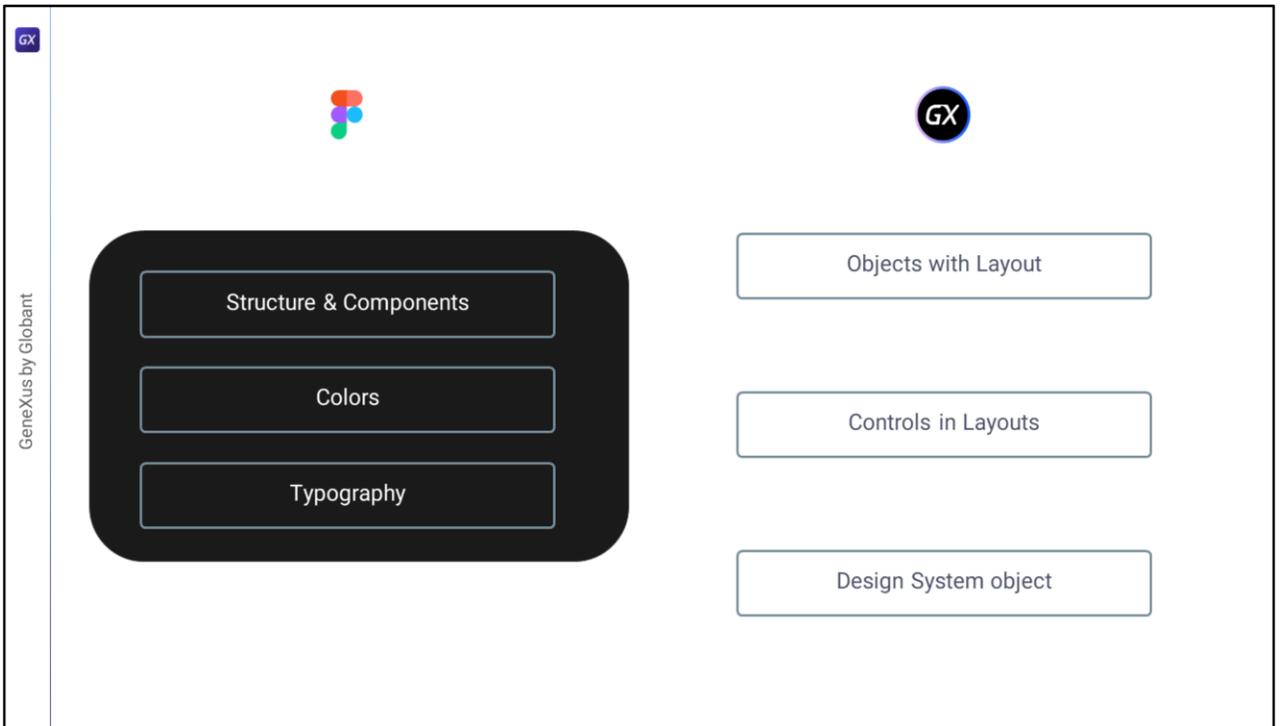


Cecilia Fernández



En la clase anterior Chechu, nuestra diseñadora estrella, había analizado estas dimensiones del sistema de diseño que armó para implementar el diseño justamente de la aplicación Travel Agency que queríamos desarrollar.

La pregunta que va a comandar no solamente esta clase, que es una suerte de repaso, sino en general todo el curso va a ser cómo reflejar todo esto, de lo que ella nos estuvo hablando que tiene que ver con el armado del sistema de diseño, cómo llevarlo a GeneXus.



Entonces vamos a empezar por analizar estos puntos a modo de repaso para introducirnos en el curso.

El primer punto, la estructura y los componentes, que tienen que ver con cómo se estructuran las pantallas de la aplicación y se componentizan, va a tener que ver en GeneXus con, en una primera dimensión, con la identificación de objetos con layout, que van a representar cada parte de esas pantallas.

Y luego en una segunda instancia, ya más micro porque es a la interna de cada objeto, vamos a tener que ver cómo se organizan los controles dentro de cada layout, para reflejar esa estructura.

Y luego vamos a tener que meternos ya de lleno en el tema que tiene que ver con el estilo propiamente dicho, donde van a intervenir los colores, la tipografía, y otras cuestiones más que veremos y que van a venir dadas en GeneXus por el objeto Design System.

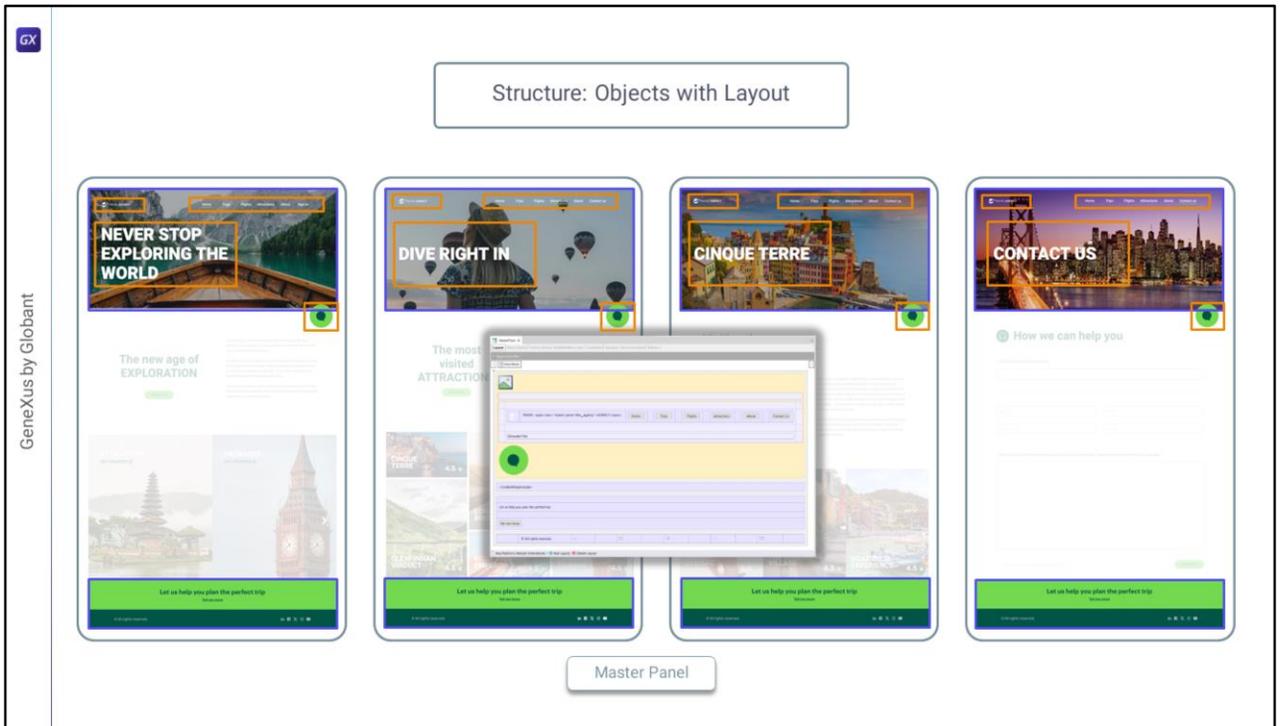
Entonces vamos a repasar en esta clase todo esto antes de pasar a profundizar en cada cosa.

Entonces empezamos por visualizar un poco a qué nos referimos con la división de todas las pantallas en objetos con Layout.

Structure: Objects with Layout



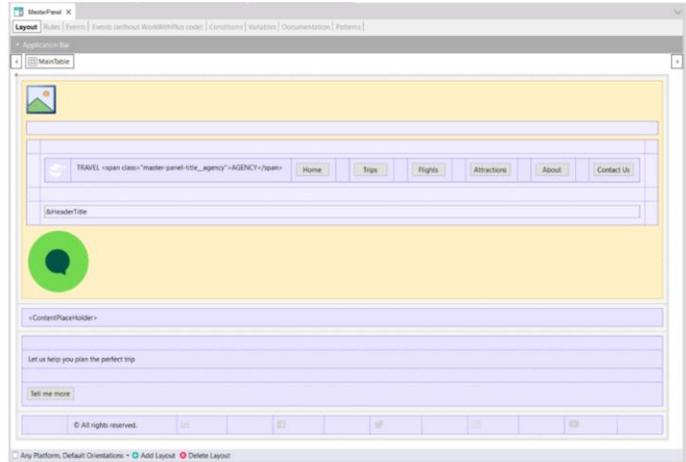
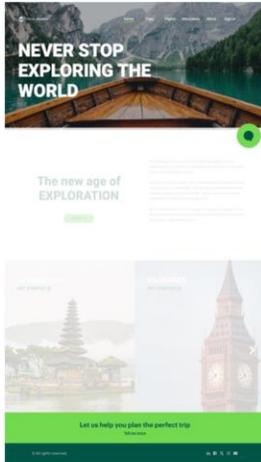
Si observamos las 4 pantallas que Chechu definió...



...ya vemos que todo esto se repite en estructura y estilo. Así, el encabezado de cada página va a tener una imagen de fondo que ocupa el mismo espacio en cada pantalla (más allá de que la imagen en sí varíe, el tamaño y el tipo de imagen es igual); sobre la imagen se va a superponer el logo de la compañía y el menú, un texto (que también variará de pantalla a pantalla pero no en estilo ni en estructura) y una imagen para chatear. Y por otro lado también vamos a encontrar repetido entre las pantallas el pie de página, que en este caso sí se va a repetir igual no solo en estilo sino también en contenido.

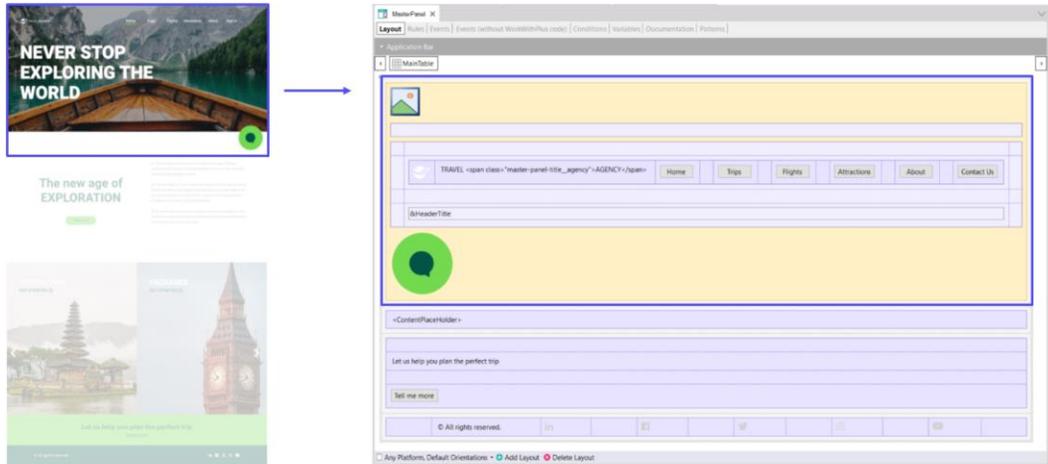
Sabemos que en GeneXus tenemos el objeto **Master Panel** que permite diseñar aspecto y comportamiento de las partes que van a ser comunes a todas las pantallas (o por lo menos a un conjunto de ellas: en este caso a estas primeras 4 pantallas-).

Structure: Objects with Layout | Master Panel



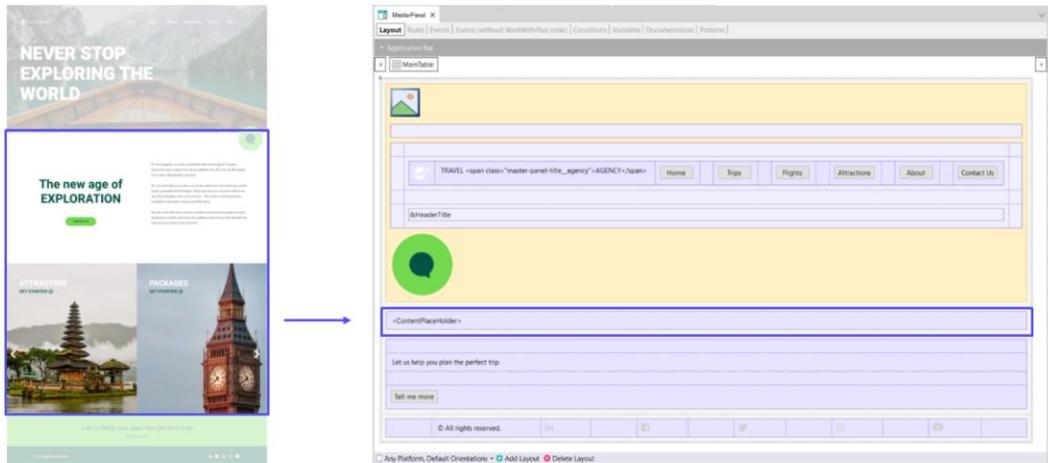
Así que todo esto que será repetido lo vamos a implementar en un objeto Master Panel.

Structure: Objects with Layout | Master Panel



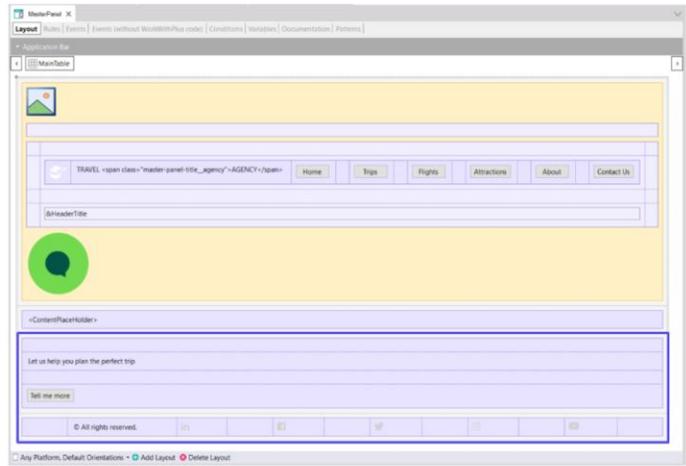
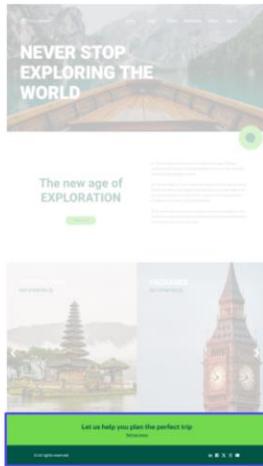
Donde en la primera fila de la tabla del Master Panel vamos a modelar el encabezado...

Structure: Objects with Layout | Master Panel

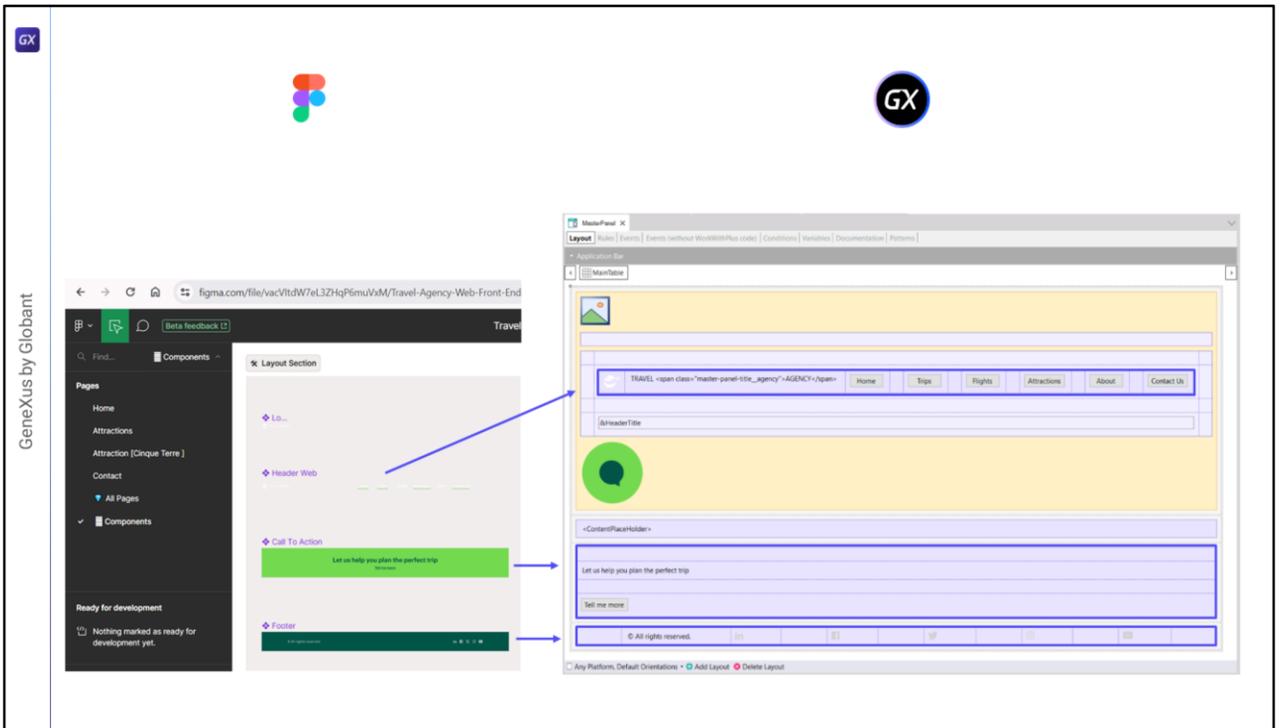


En la segunda fila vamos a tener este control ContentPlaceholder que es propio del objeto Master Panel que es justamente el que indica que allí dentro es donde se va a cargar la información particular de cada página que tenga a éste como su Master Panel...

Structure: Objects with Layout | Master Panel

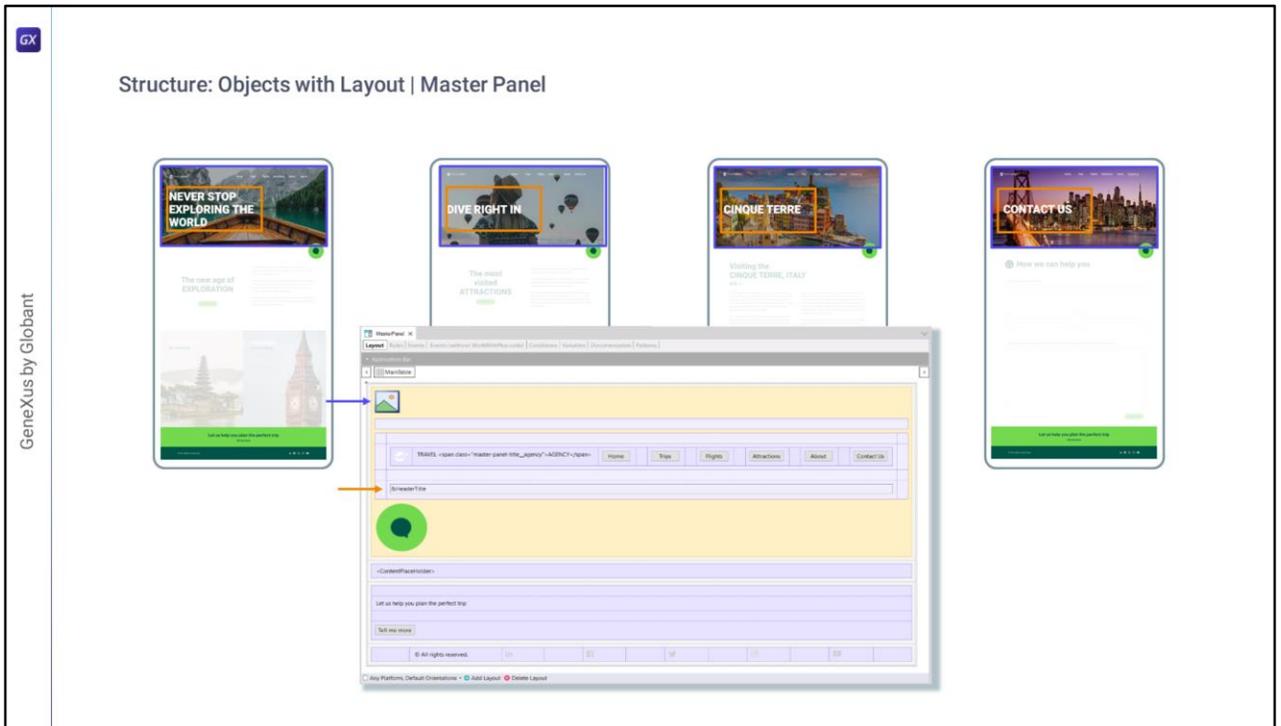


... y por último, propio del Master Panel también, en las filas 3 y 4 vamos a modelar este pie de página.



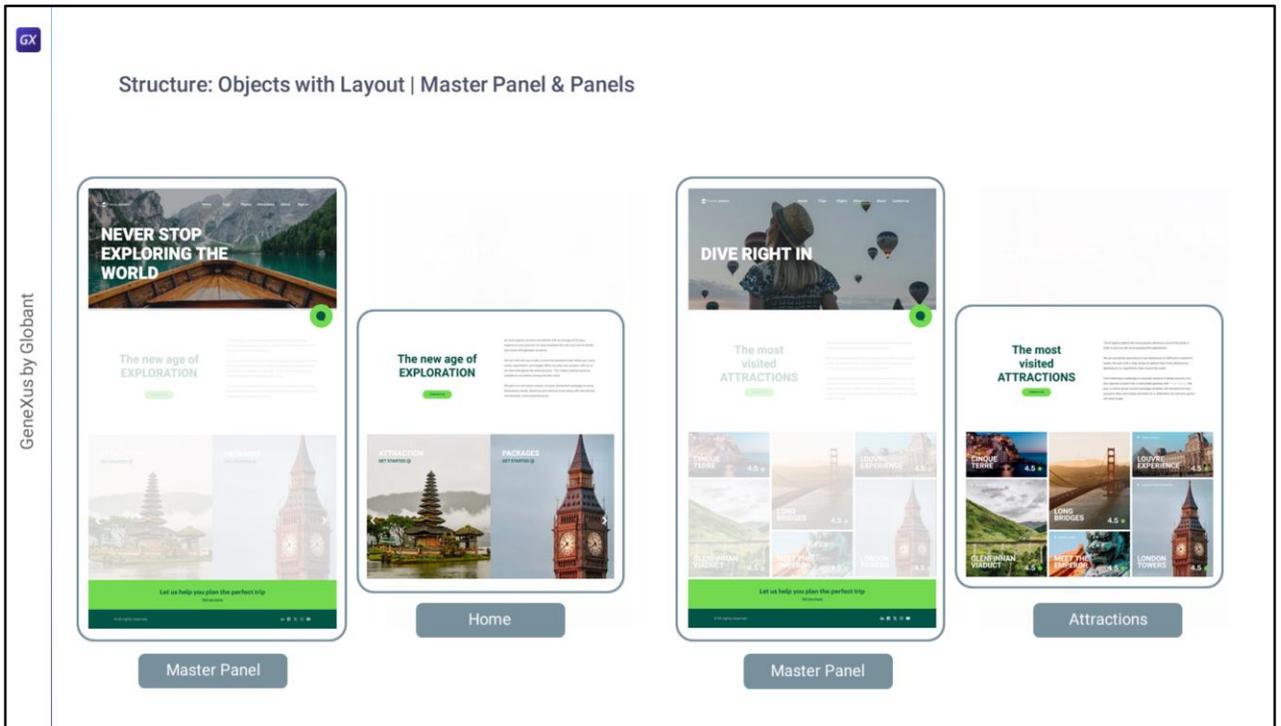
Bien. Si atendemos a lo que modeló Chechu en Figma, ella no tenía forma de expresar esto. Lo máximo que pudo hacer fue modelar estas partes (el menú y estas partes del pie de página) como componentes, pero no logra establecer una relación entre ellos porque no cuenta con una herramienta que le permita expresar esto mismo que nosotros sí podemos expresar en GeneXus a través de un objeto Master Panel.

Acá es donde empezamos a ver que las herramientas de diseño y GeneXus no tienen exactamente la misma capacidad de expresión, entonces hay que entender un poco de ambas herramientas, en particular de la herramienta de diseño que se esté utilizando, para poder hacer el mejor uso combinado de las dos.



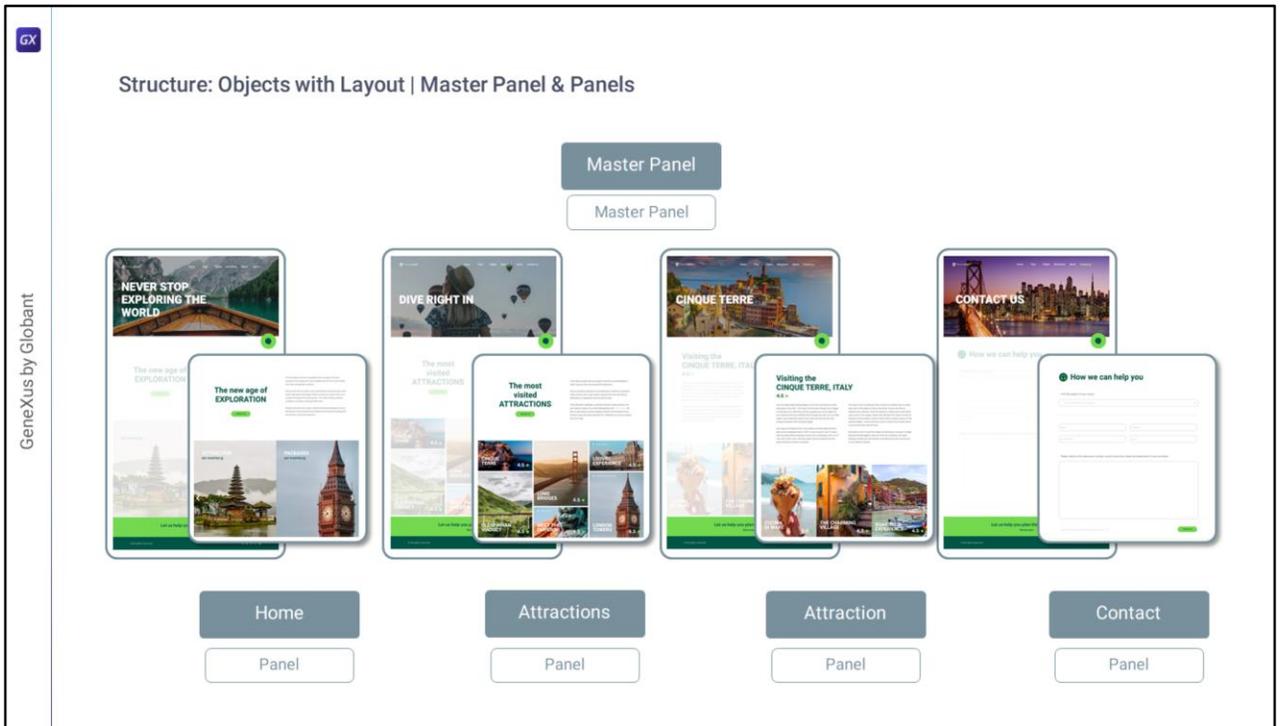
Bueno, antes de pasar a otra cosa. Tal vez les haya quedado una duda rondando en la cabeza respecto a la parte que varía en contenido entre una página y otra, pero que dijimos que igual se iba a implementar en un Master panel, que es justamente la imagen del fondo, que vemos que cambia de pantalla a pantalla, y el texto este que se le superpone.

Bueno, si en el Master Panel los implementamos, estos dos elementos, con esta variable imagen y este control variable, entonces es fácil la respuesta, va a venir por el lado de hacer variar el contenido de esos controles de acuerdo al objeto que se esté cargando en cada oportunidad en el ContentPlaceholder. Y esto ya se imaginarán que va a ser por código en la sección de eventos. Lo vamos a ver más adelante en el curso.



Volviendo... con el Master Panel podemos empezar a realizar una primera estructuración: separando cada pantalla en dos: una con todo lo repetido, en este caso header y footer, y otra con el contenido específico de cada página, que se modelará en un objeto Panel, que para esta pantalla he llamado "Home".

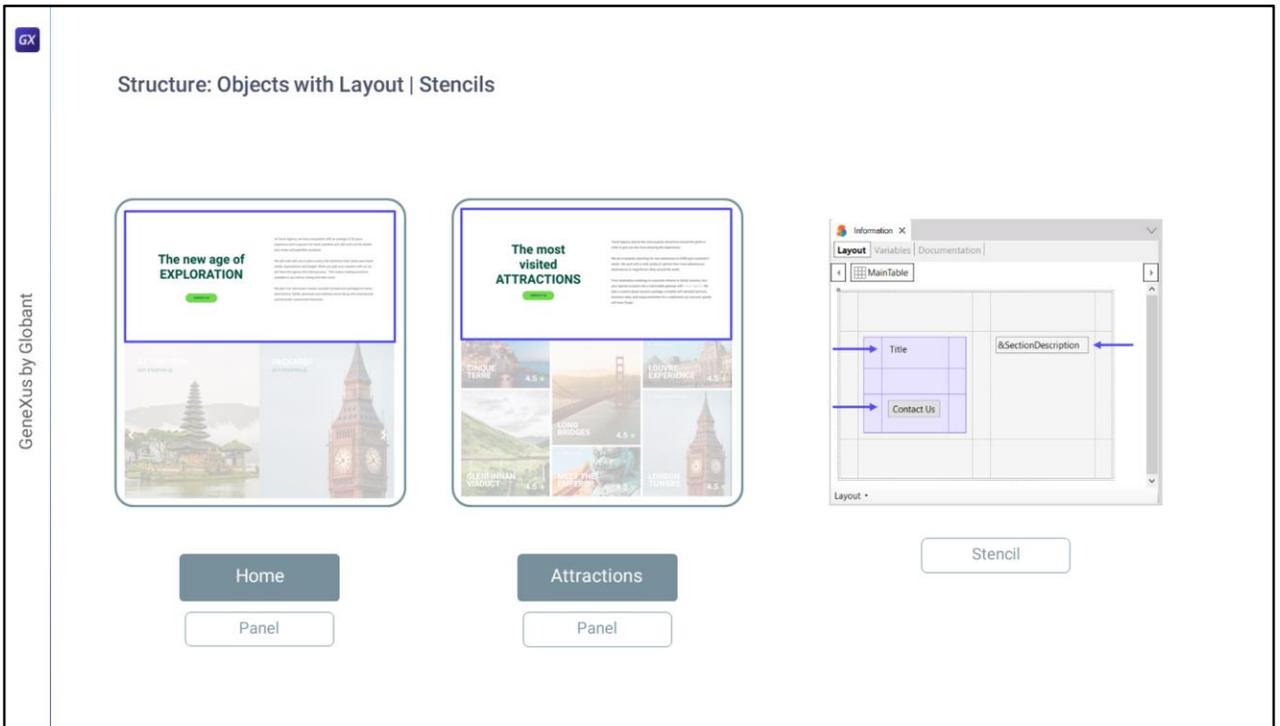
Para cada pantalla se repetirá lo mismo: la separación en el Master Panel y el Panel.



Así, entonces, nos quedará un objeto Master Panel, uno solo (aquí le llamé Master Panel al objeto, que es de tipo Master Panel; le podría haber llamado TravelAgency, o de cualquier otra manera, no estuve muy creativa), y 4 objetos panels, que se cargarán, cada uno, en el contentplaceholder del Master Panel.

Así que hicimos esta primera división de las pantallas en Master Panel y Panels.

Pero no nos vamos a detener acá, porque si seguimos analizando las pantallas podemos notar que hay más estructuras que se repiten, aunque el contenido varíe, igual que como vimos antes, y que, por lo tanto...



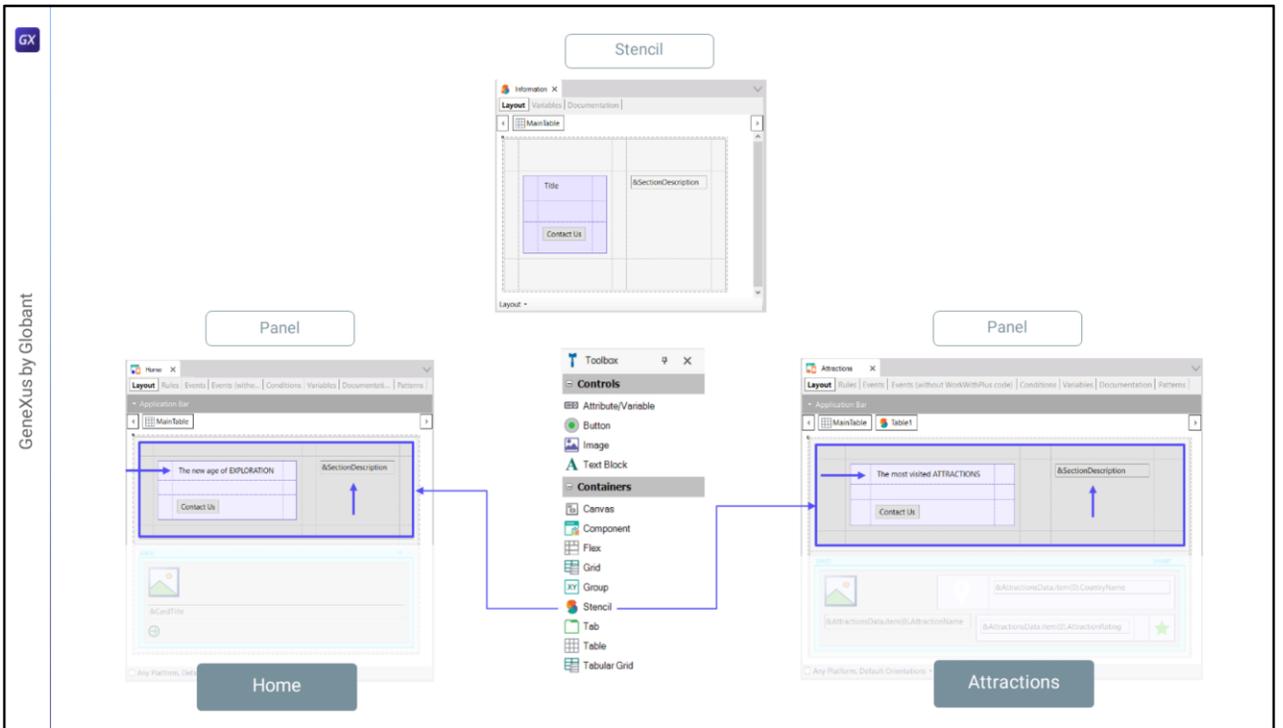
... pueden ser aisladas esas partes para definir las una vez y reutilizarlas todas las veces que se necesiten. Es el caso de lo que ocurre entre esta parte del panel Home y esta del de Attractions.

Son iguales en estructura visual y de comportamiento (y digo de comportamiento porque vemos ahí un botón que va a llamar en ambos casos a la misma pantalla exactamente), por lo que puedo implementar todas estas secciones que se repiten con un objeto Stencil, que se utiliza justamente para eso.

Es decir, es un objeto GeneXus con Layout, pero a diferencia de los otros, de estos que veíamos, este no es ejecutable por sí solo. Para utilizarse y ejecutarse tiene que instanciarse.

De esta manera lo que hago es, en el objeto stencil, definir la estructura y estética sin preocuparme por el contenido, que será propio de cada instancia.

Por eso vemos ahí un textblock, un botón y una variable de texto. El textblock, vemos, tiene un caption genérico, y la variable no tiene nada cargado, es una variable de texto nada más.

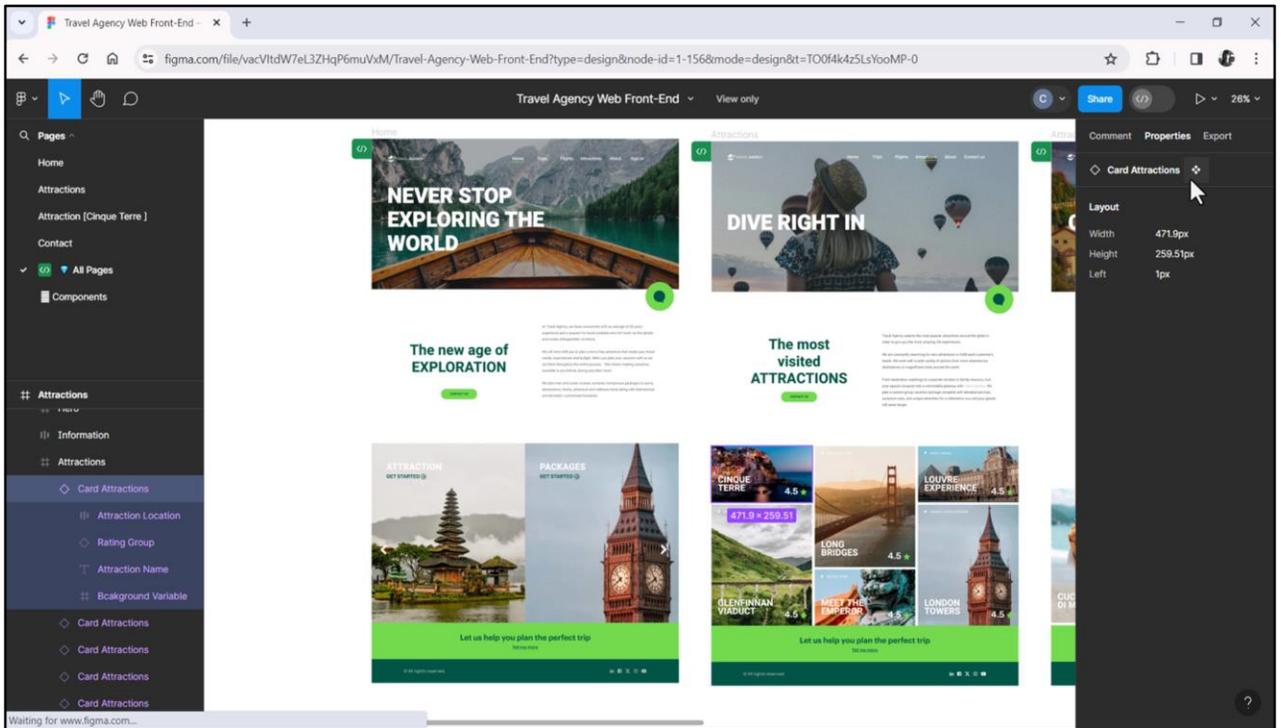


Y luego lo que hago es **instanciar** el stencil en los paneles en los que lo necesite.

Así, insertamos en la primera fila de ambos paneles un **control** stencil, en el que se cargará el contenido del **objeto** stencil, pero en el que podremos modificar aquellas cuestiones que tienen que ver con el contenido. Por ejemplo, el caption del textblock lo modifico en cada uno de esos paneles para que adopte la particularidad que necesito para cada uno. Y por ejemplo también voy a tener la variable esta de aquí que la voy a cargar en ejecución en los eventos con el contenido particular y específico.

Así entonces conseguimos otra forma de reutilización. Si debemos modificar algo de la estructura o la estética lo hacemos en el objeto stencil, una vez, y eso aplica a los paneles donde ese stencil esté instanciado.

Chechu, nuestra diseñadora, no identificó esta repetición...

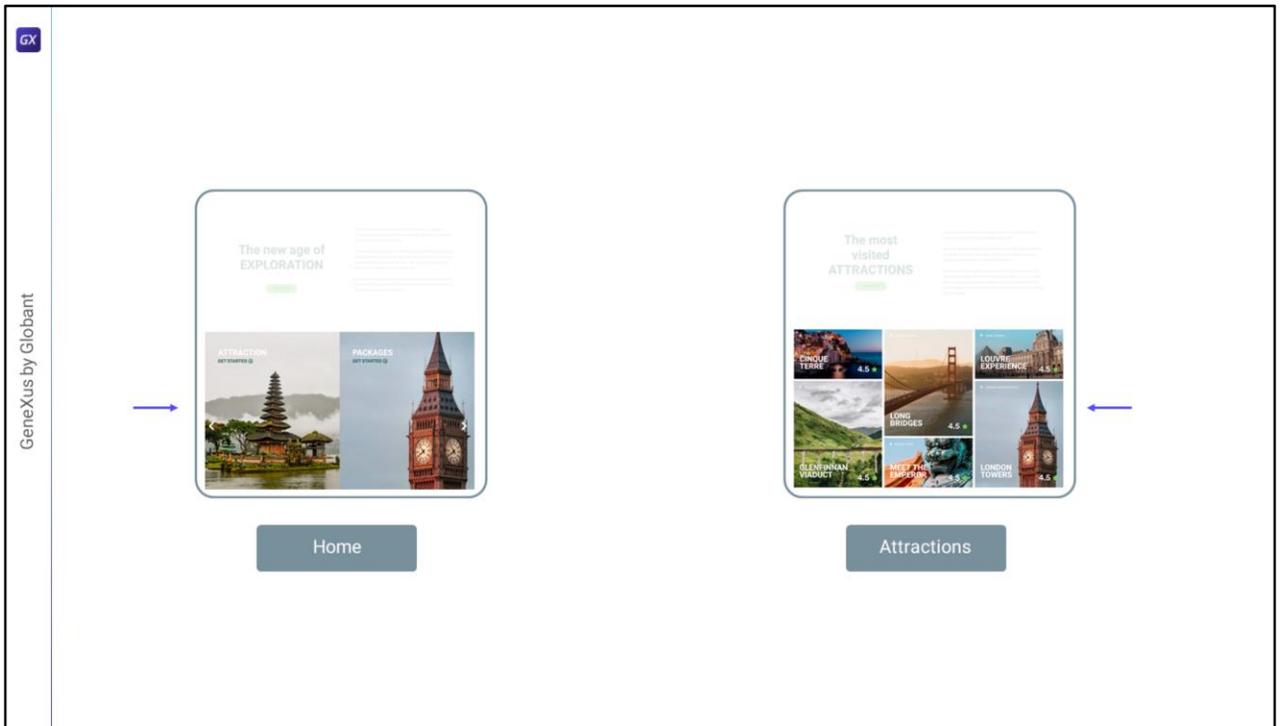


Si vamos a inspeccionar las dos pantallas, vemos que tienen un grupo al que llamé Information, lo llamé igual en ambos casos, pero no lo componentizó... y por eso no aparece en la página de componentes, como vamos a ver, como sí sucederá por ejemplo con estas cards... Si vamos a ver acá vemos que está indicando que es una instancia de un componente y ahí accedo al componente, de hecho me lleva a la página de componentes, y hasta podría volver a la instancia de la que partí. Y aquí vuelvo a la card esta dentro de la página Home.

Y lo mismo, si me posiciono en este otro elemento vemos que es una instancia de un componente... que es este de aquí, que está en la página de componentes.

Chechu seguramente no componentizó el grupo Information del que hablábamos porque no le resultó tan evidente. Nosotros podemos hacerlo. A ella no le resultó evidente, para nosotros sí lo fue.

Pero por otro lado sí le resultó evidente que tenía que componentizar estas cards, entonces, al haber identificado esas cards como componentes nos da esa pista de que posiblemente nosotros podamos componentizarlos también, en nuestro caso creando stencils para poder reutilizarlos. Entonces...

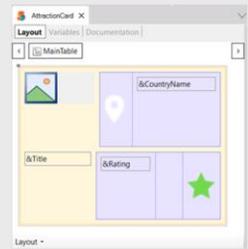
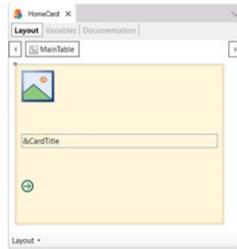


Si vamos a observar estos dos paneles, vemos que tenemos estos dos carruseles de aquí donde se repiten esas cards.

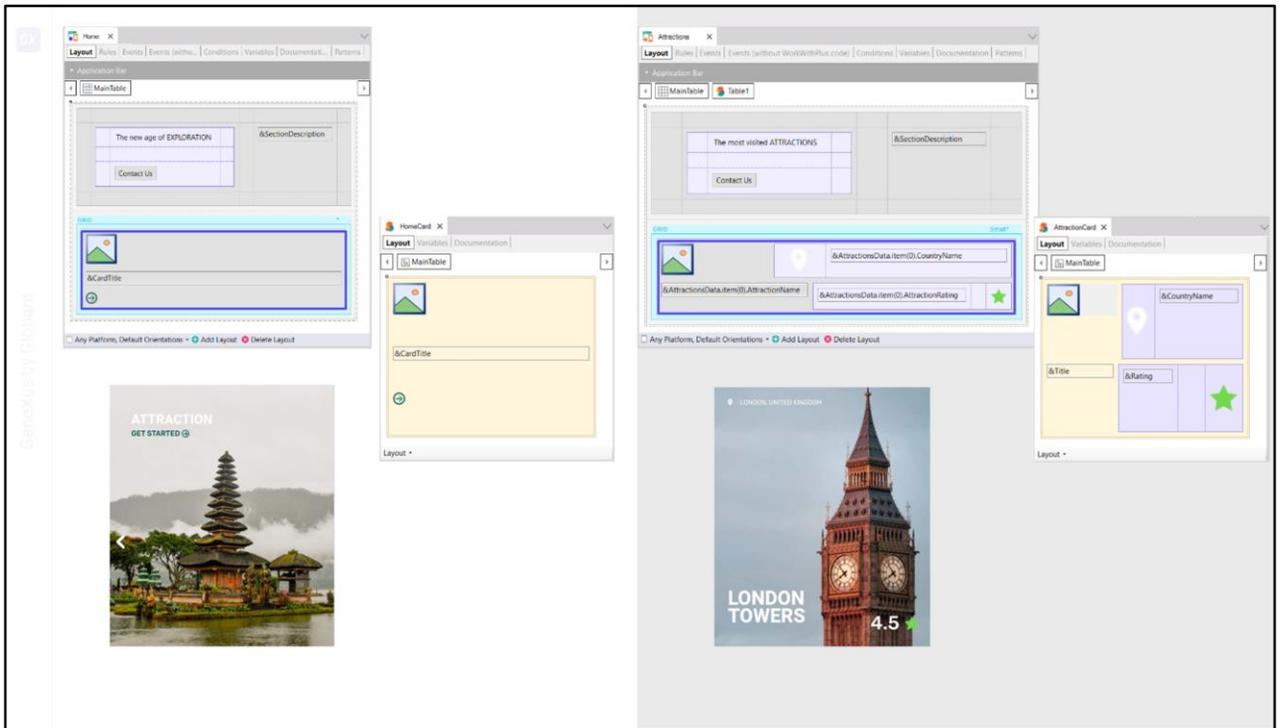
O sea, vamos a necesitar una estructura repetitiva para modelar los carruseles, que ya podemos adelantar que van a ser grids de distinto tipo: en un caso va a ser horizontal, en otro caso va a ser un grid de tipo flex...



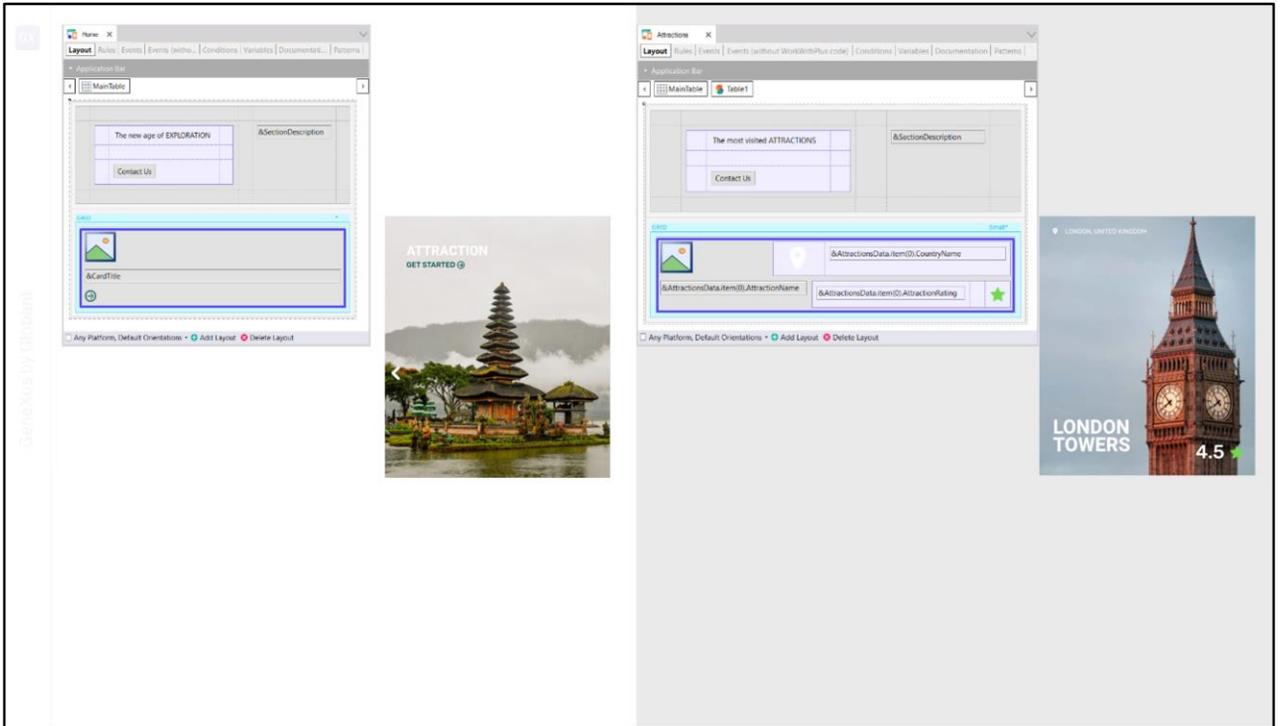
...modelarlos como stencils.



Entonces lo que vamos a hacer es crear luego un stencil para cada una, allí vamos a modelar su layout, y luego...



...simplemente tal como vimos antes vamos a instanciar cada stencil en el panel, como elementos de cada uno de esos grids.



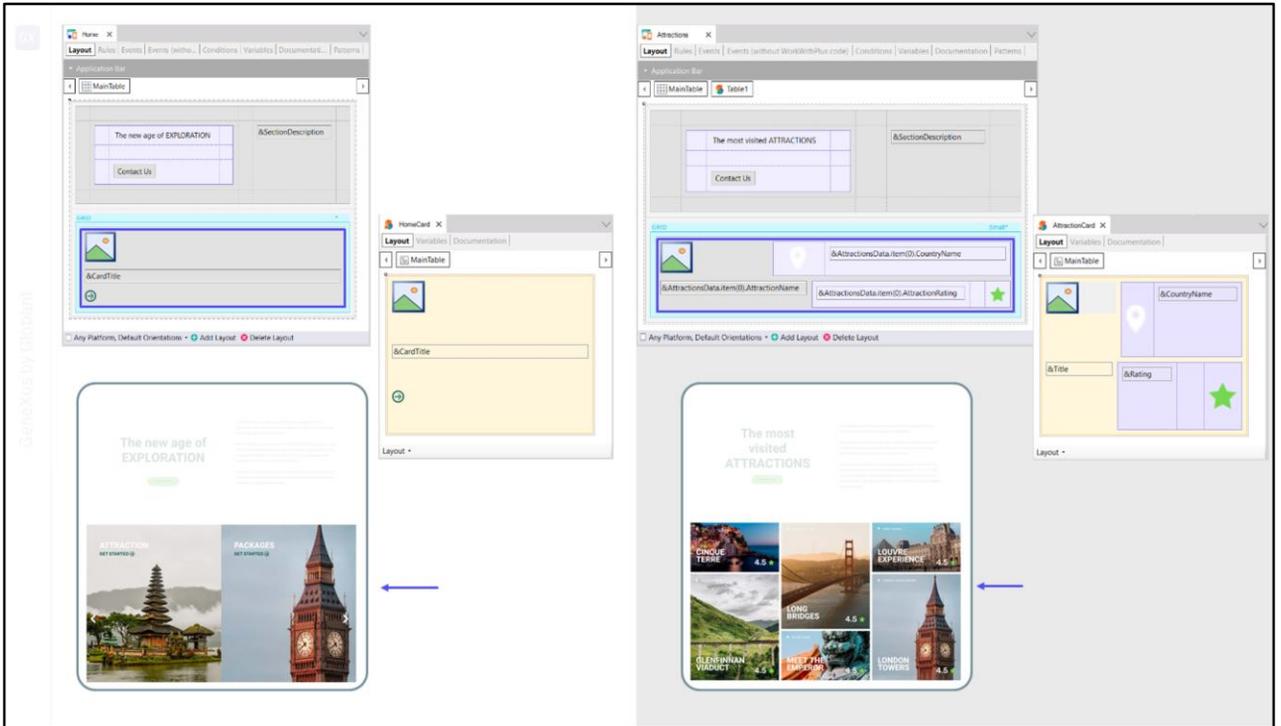
Entonces, al hacer esto...

Home X
Layout Rules Events Events (with...) Conditions Variables Documentation Patterns
Application Bar
MainTable
The new age of EXPLORATION &SectionDescription
Contact Us
Title
&CardTitle
Any Platform, Default Orientations - Add Layout Delete Layout

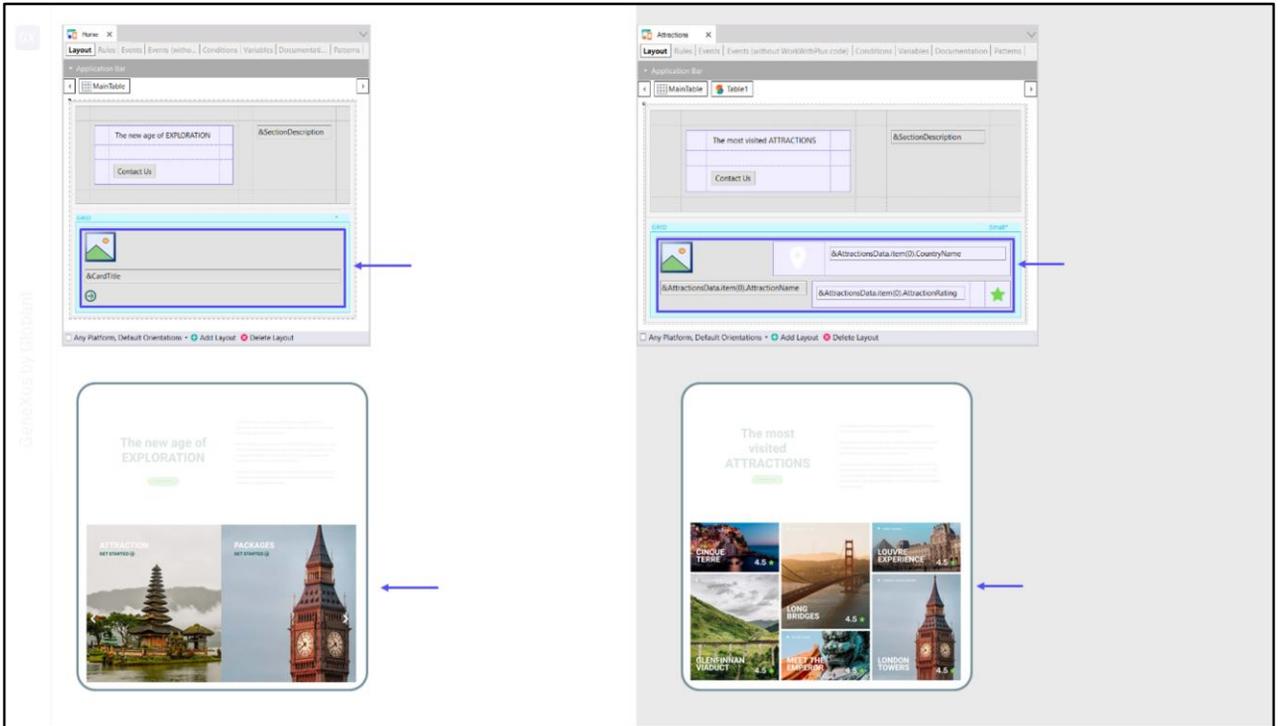
HomeCard X
Layout Variables Documentation
MainTable
&CardTitle
Layout

Attractions X
Layout Rules Events Events (without WorkWithPlus code) Conditions Variables Documentation Patterns
Application Bar
MainTable Table1
The most visited ATTRACTIONS &SectionDescription
Contact Us
Title
&AttractionsData.item(0).CountryName
&AttractionsData.item(0).AttractionName &AttractionsData.item(0).AttractionRating
Any Platform, Default Orientations - Add Layout Delete Layout

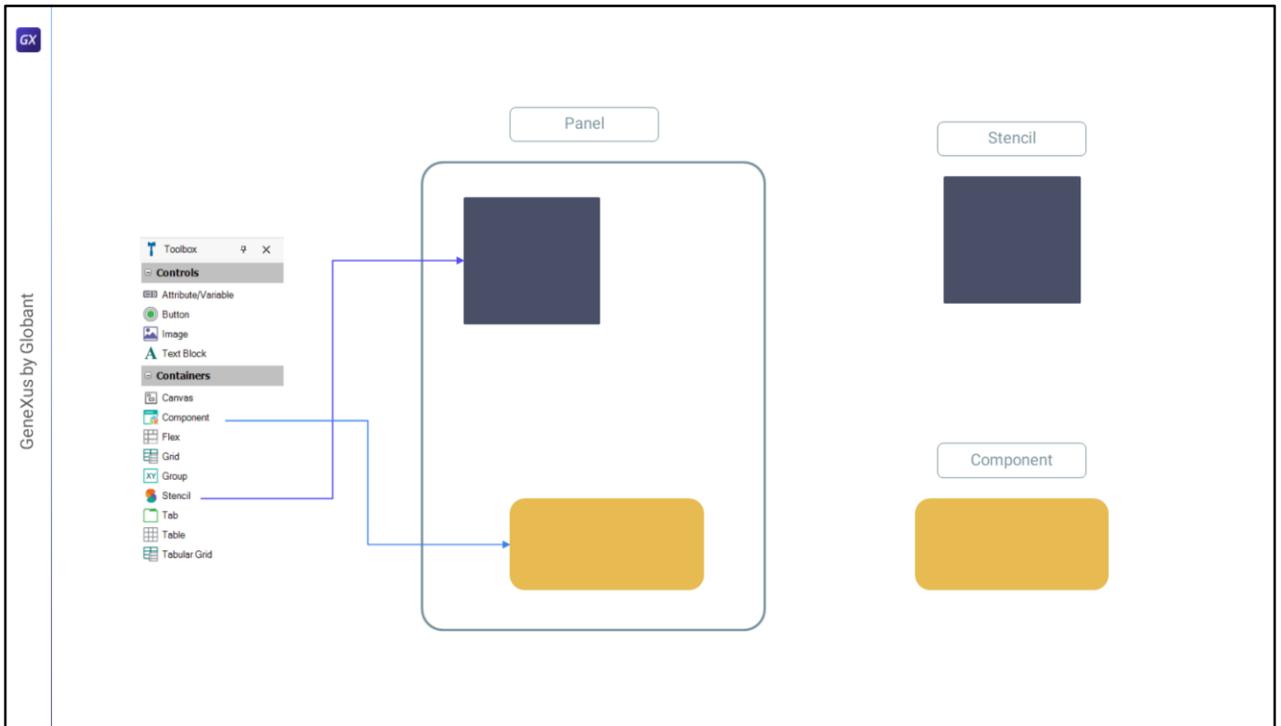
AttractionCard X
Layout Variables Documentation
MainTable
&CountryName
&Title &Rating
Layout



Tengo modelado el grid como una repetición de esos stencils...



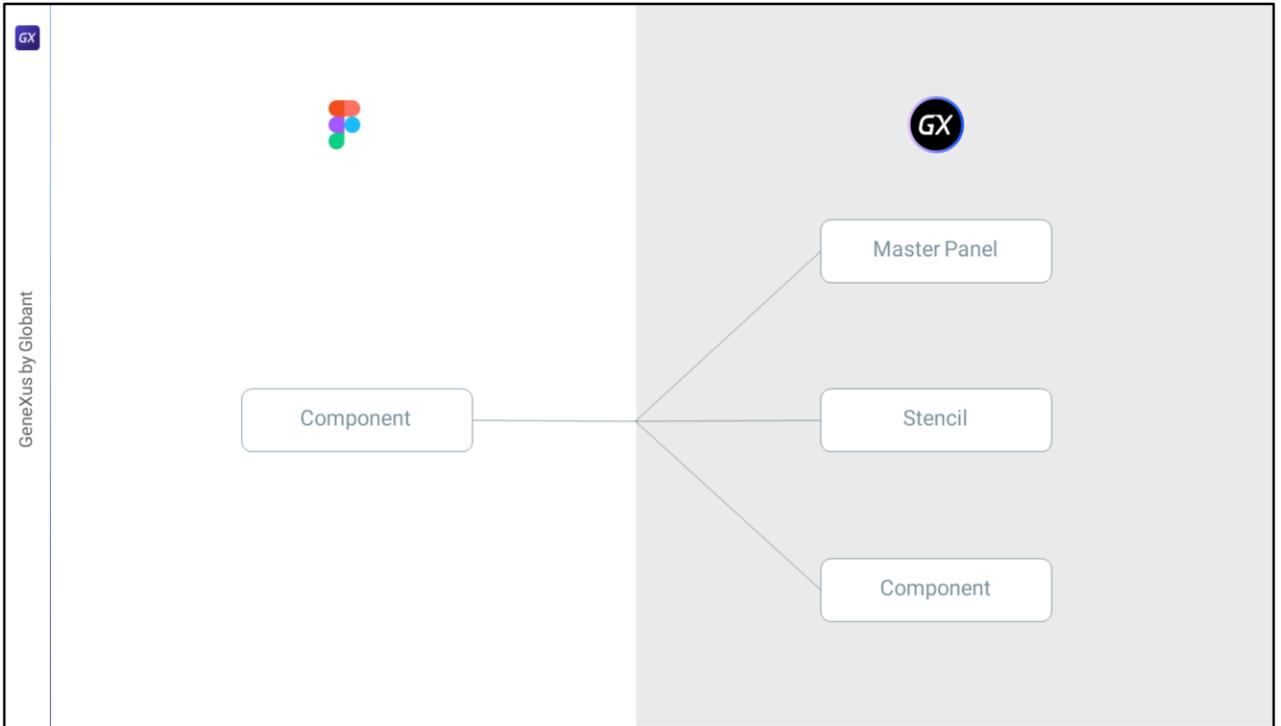
Y luego lo que voy a necesitar simplemente es, en los eventos, cargar cada ítem, cada contenido, con la información que corresponda. Que en el segundo caso tendré que tomarlo de la base de datos.



Bien, en GeneXus además de poder componentizar una sección de layout con estética a partir de definirla en un objeto stencil, como hicimos hasta ahora, al que luego instanciamos en el layout del objeto que lo necesita, tenemos otra forma, que se utiliza sobre todo cuando implica también comportamiento (aunque los stencils hoy permiten cierto grado de comportamiento)...

Si estamos desarrollando en el mundo de los Web Panels, será con el objeto de tipo Web Component, y si estamos desarrollando en el mundo de los Panels será un panel común y corriente. Así que ahí identifiqué esto como Component pero se va a implementar, dependiendo del mundo: si es el mundo Web Panels con un Web Component, si es el mundo de los Panels con otro Panel. Que se insertará en cualquiera de los dos casos, se va a insertar en el layout donde se quiera utilizar a través de un control de tipo Component.

En definitiva, ¿qué quiero mostrar con esto? Que en GeneXus tenemos varias opciones para hacer que una porción de pantalla se aíse y se modele separadamente. De hecho tenemos otra más que es el uso de User Controls.

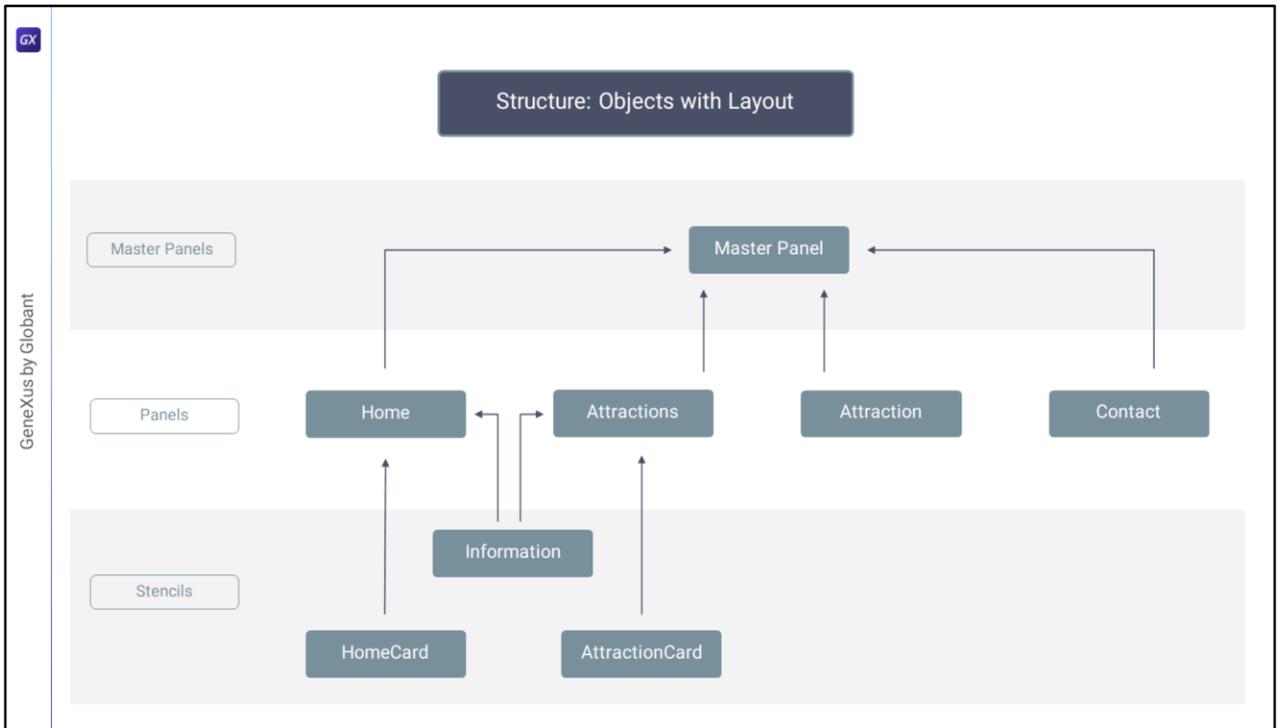


Así que entonces un componente que Chechu haya aislado en Figma puede representarse en GeneXus de varias maneras: tanto dentro de un Master Panel como de un stencil como de un component. Y también un User Control.

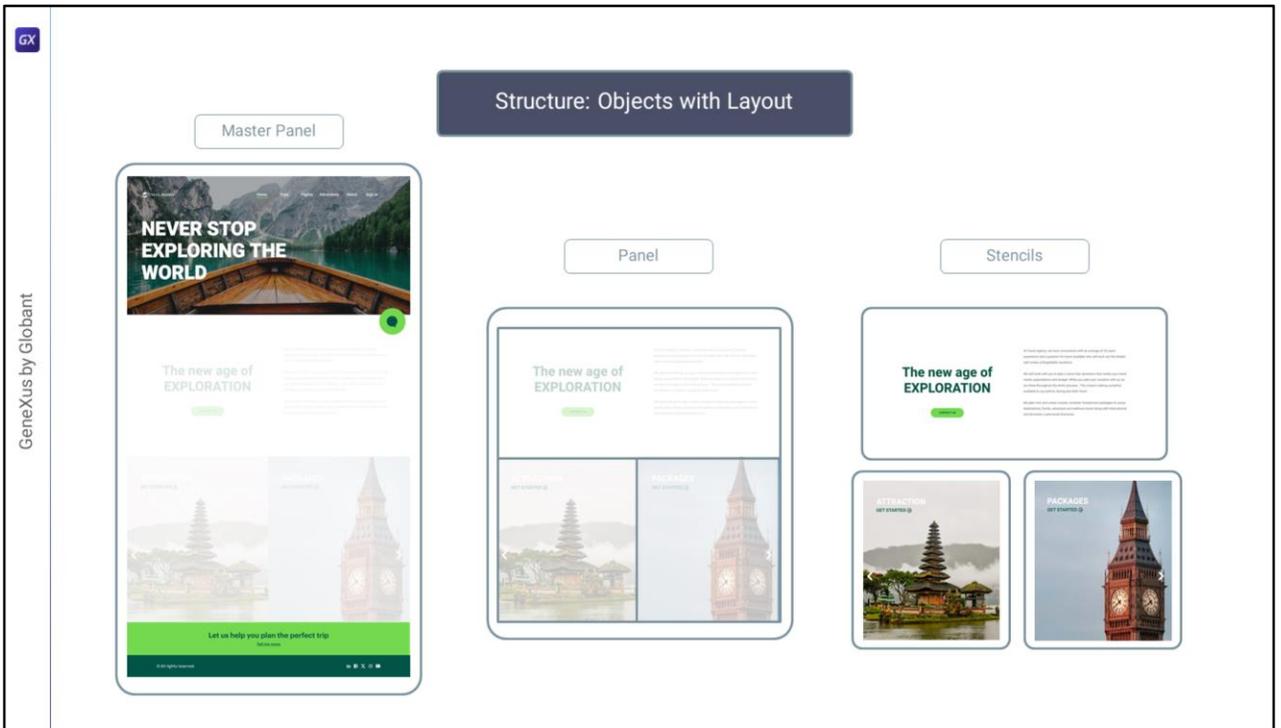
Structure: Objects with Layout



En cualquier caso, con esto nos queda clara esta primera división estructural de la aplicación en objetos con Layout, que están interrelacionados...

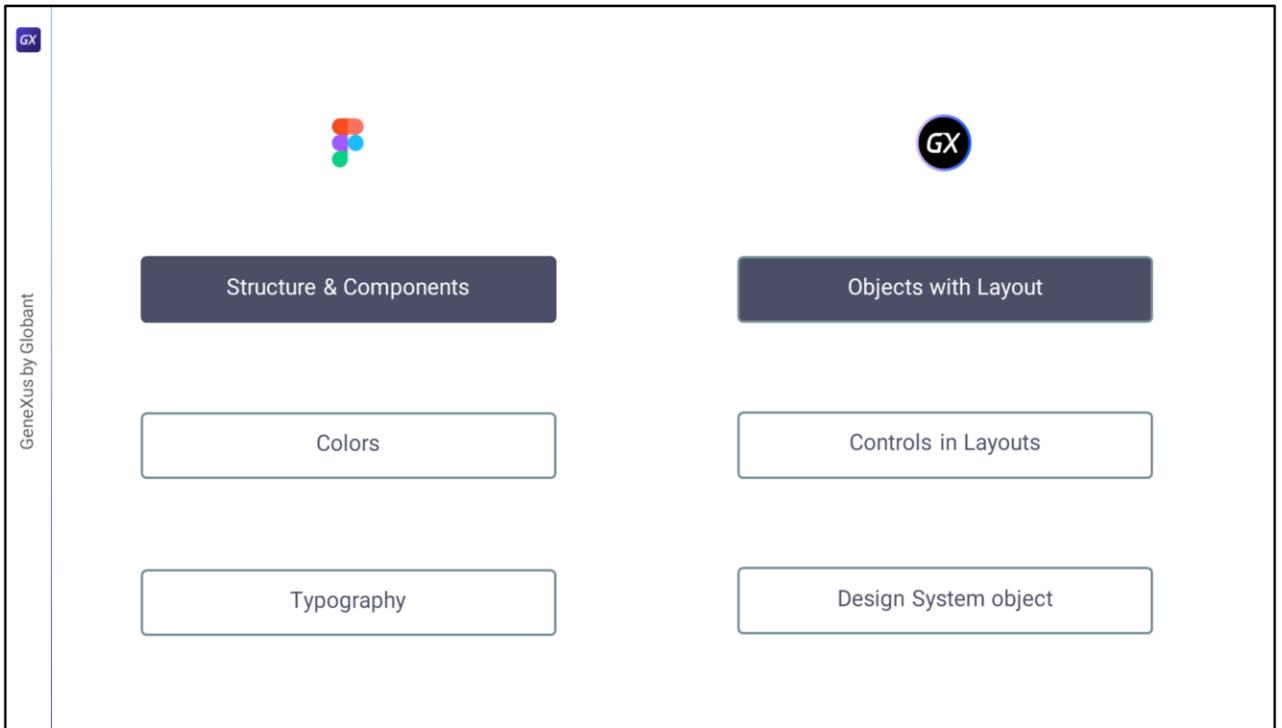


...como vemos en este esquema que hice aquí, donde vemos el Master Panel, los cuatro paneles que identificamos hasta el momento, y esos tres stencils que vimos.

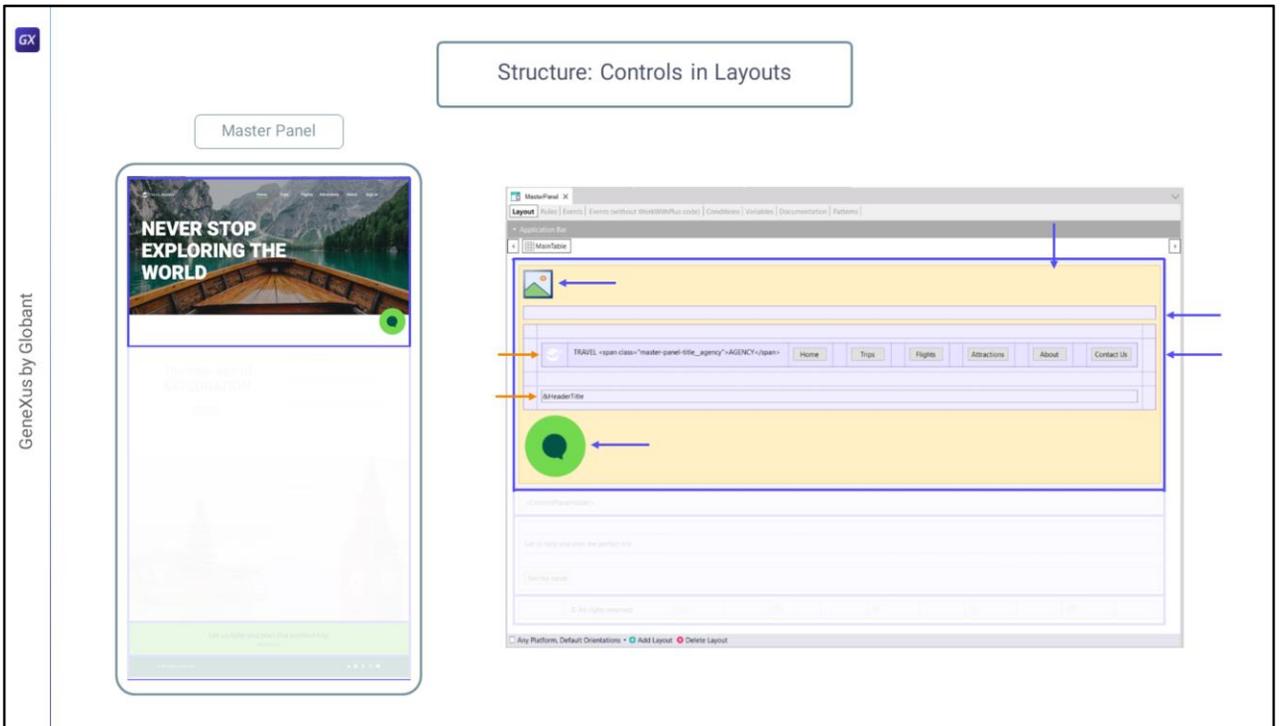


Así, entonces, si tomamos la primera pantalla... nos queda dividida primero en Master Panel y Panel; pero el Panel a su vez nos queda también dividido en sus controles, como el grid por ejemplo, y Stencils.

El problema, entonces, se nos redujo ahora a uno más pequeño pero de la misma complejidad, que es: ¿cómo implemento el layout de cada uno de esos objetos?



Y entonces ahora es que vamos a pasar al segundo punto, al de los controles en los layouts. ¿Cómo organizamos los layouts para que la estructura refleje lo que va a ser la estructura de las pantallas?



La pregunta entonces es cómo estructuramos el layout de cada objeto. Del Master Panel, del Panel, y del Stencil del que se trate.

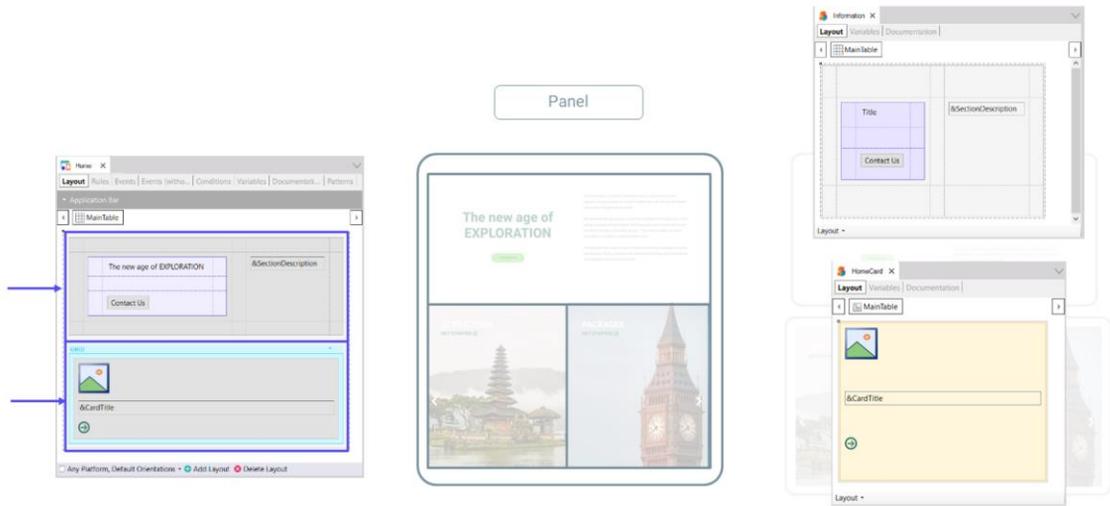
Por ejemplo, si analizamos el Master Panel: aquí vamos a tener una tabla main (principal) con 4 filas...

Dentro de cada una se ubican controles que a su vez pueden contener otros controles... como en esta primera fila donde tenemos una tabla especial, canvas, (la tabla canvas es la que permite superponer controles), que contiene a su vez un control de tipo variable imagen, dos tablas, y un control de tipo imagen.

La segunda de estas tablas a la vez contiene otra y un control variable.

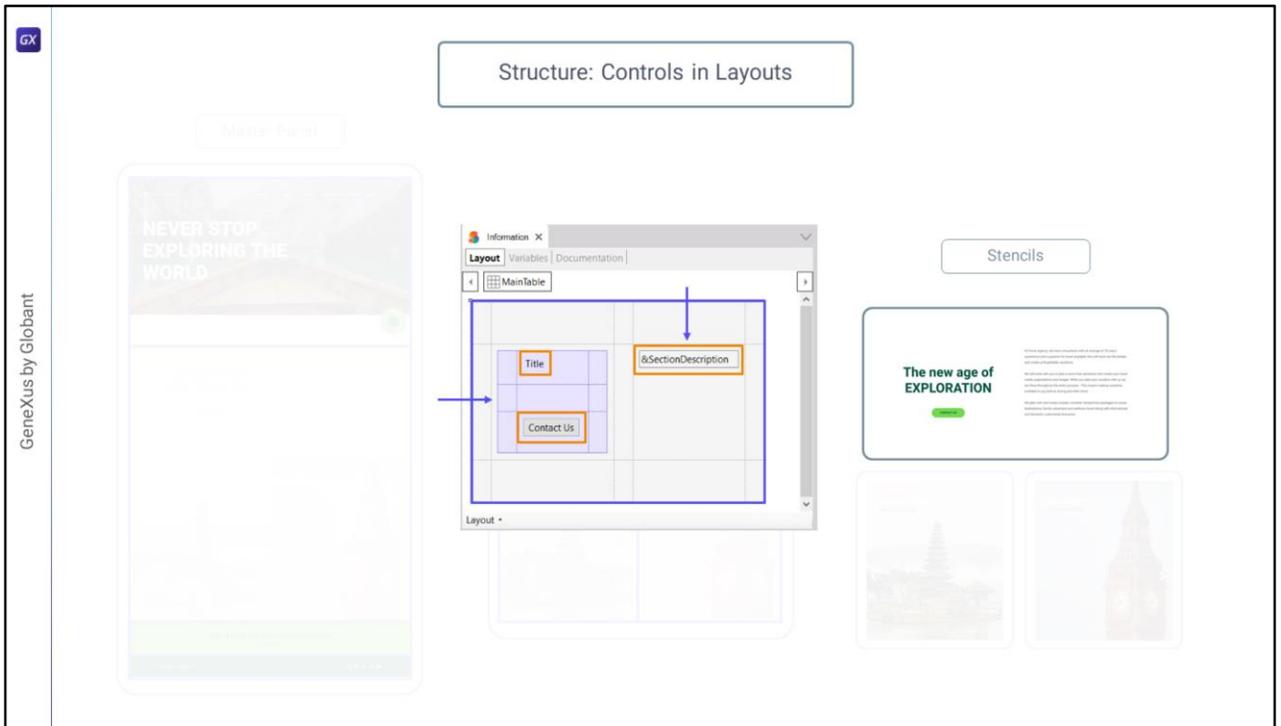
Y así sucesivamente... Es decir, tenemos una jerarquía de controles, donde el padre de todos es la tabla main.

Structure: Controls in Layouts



Análogamente si analizamos el panel Home también tendrá una tabla main... esta vez con 2 filas...

La primera contendrá un control stencil y la segunda un grid que contendrá otro control stencil.



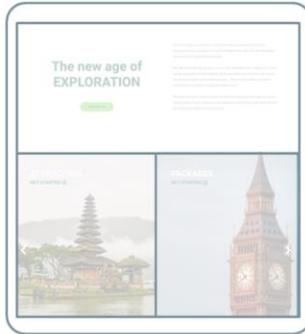
Y por otro lado, si analizamos por ejemplo el primero de los stencils... otra vez... tendremos una tabla main con 3 filas y 1, 2, 3, 4, 5 columnas. La celda de la fila 2 columna 2 tendrá a su vez otra tabla, que tendrá adentro un control text block y un botón. Y la celda de la fila 2 columna 4 tendrá un control variable.

Structure: Controls in Layouts

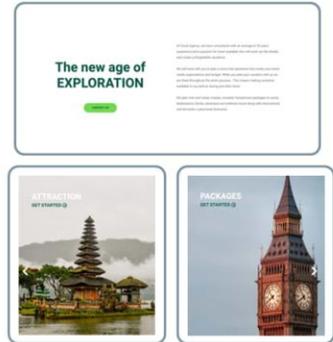
Master Panel

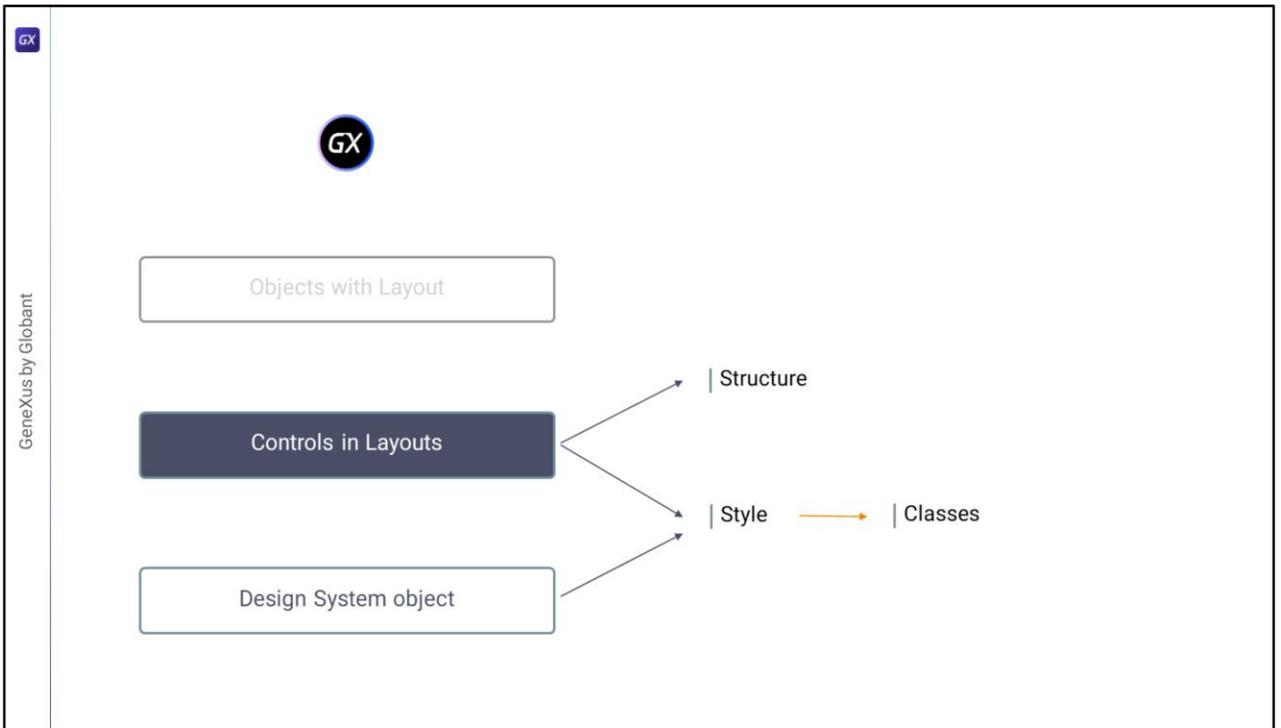


Panel

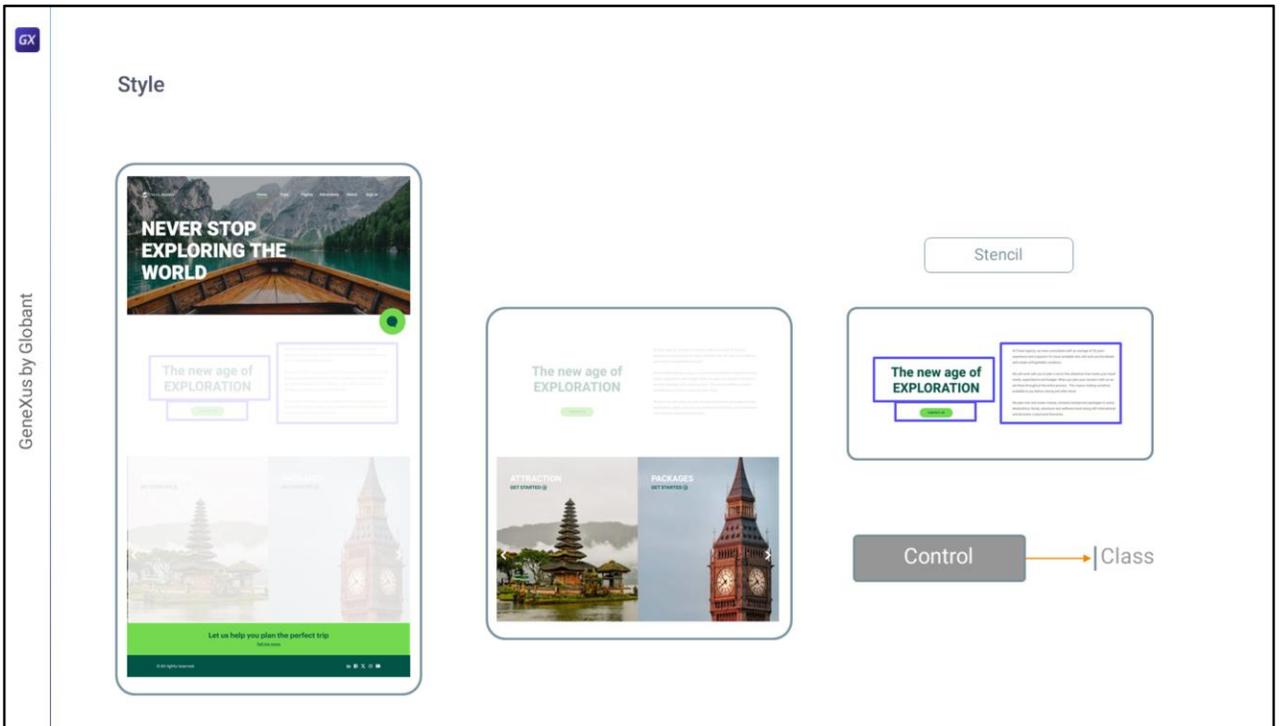


Stencils



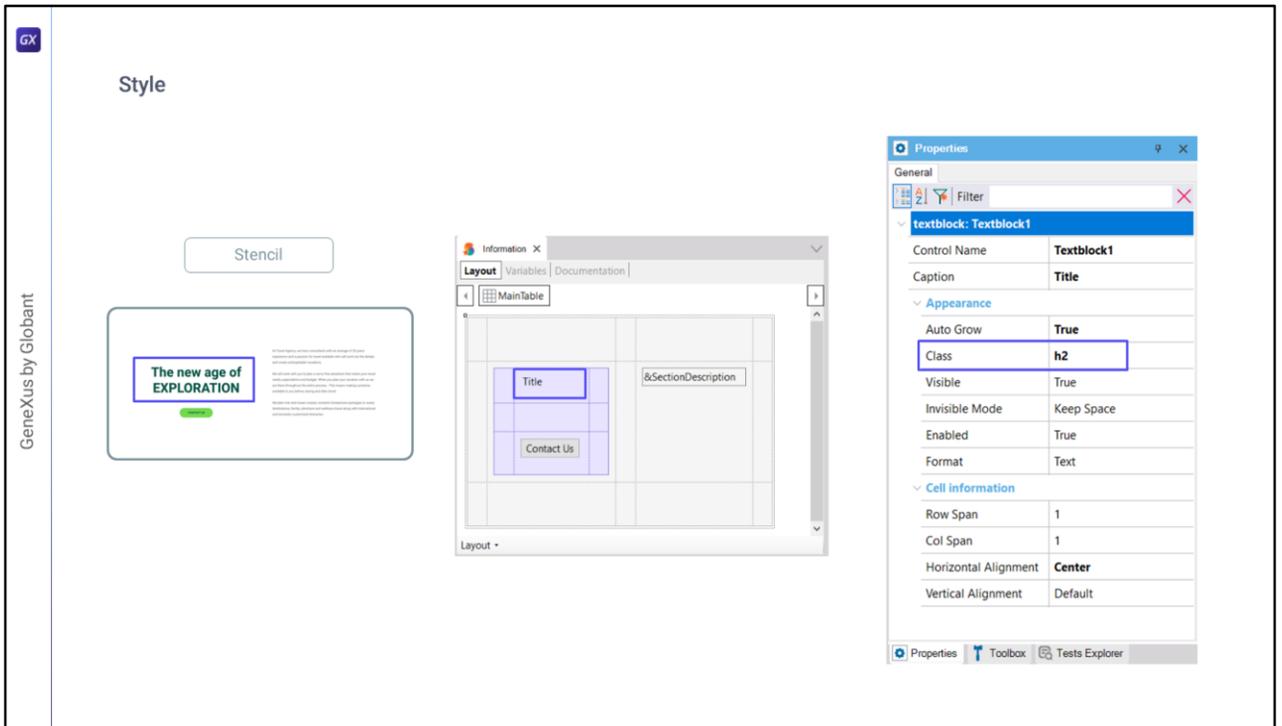


Bien, pero si dentro de lo estructural tenemos a los controles en su organización jerárquica, para el estilo, es decir, para que los elementos luzcan de una manera determinada, tenemos sus propiedades y, fundamentalmente, las de sus clases. Y allí es donde interviene el Design System object.



Así, por ejemplo, para hacer que este texto se vea con esta tipografía y este color verde, este párrafo se vea con esta tipografía y este color, y este espaciado y demás, y que el botón se vea de esta manera...

...en el stencil donde esto está implementado alcanza con asociarle a cada control una clase...



Así, al primer control, a este text block, le voy a asociar una clase a la que voy a llamar h2 (veremos que no lo inventé de la nada, sino que Chechu ya me allanó el camino porque definió ella en Figma un estilo h2 que se va a reflejar como clase en GeneXus)...

GeneXus by Globant

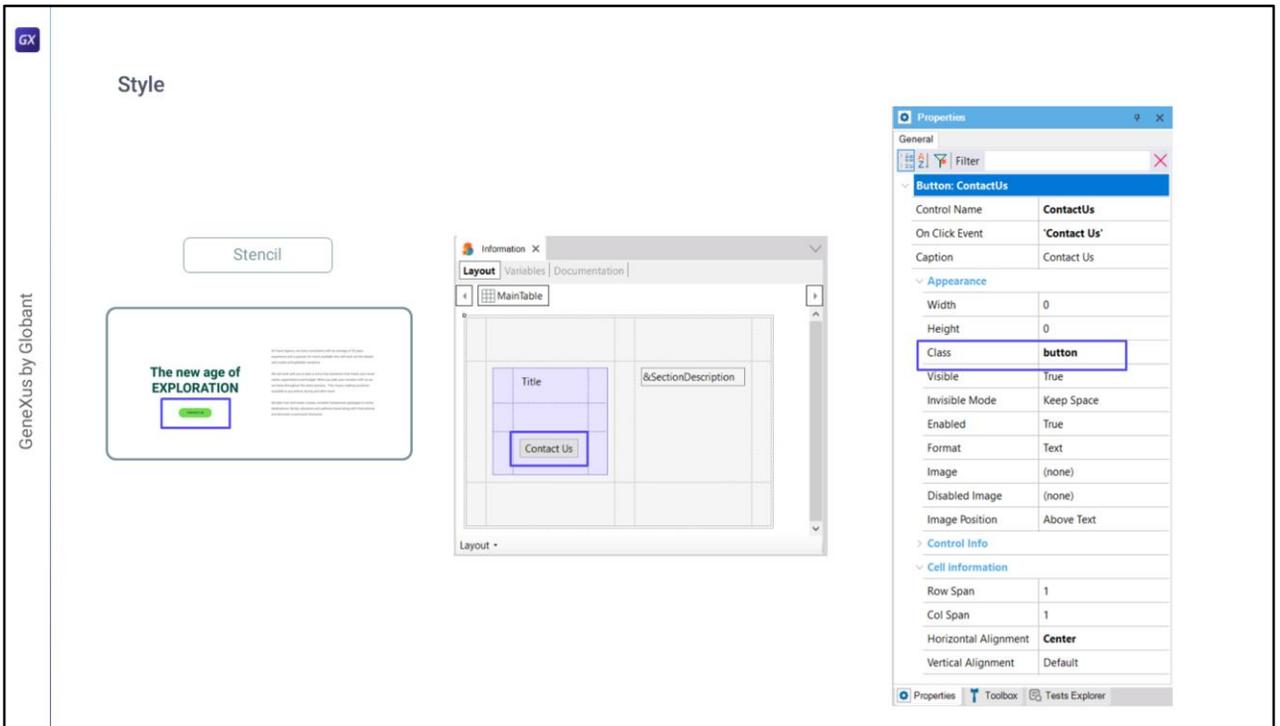
Style

Stencil

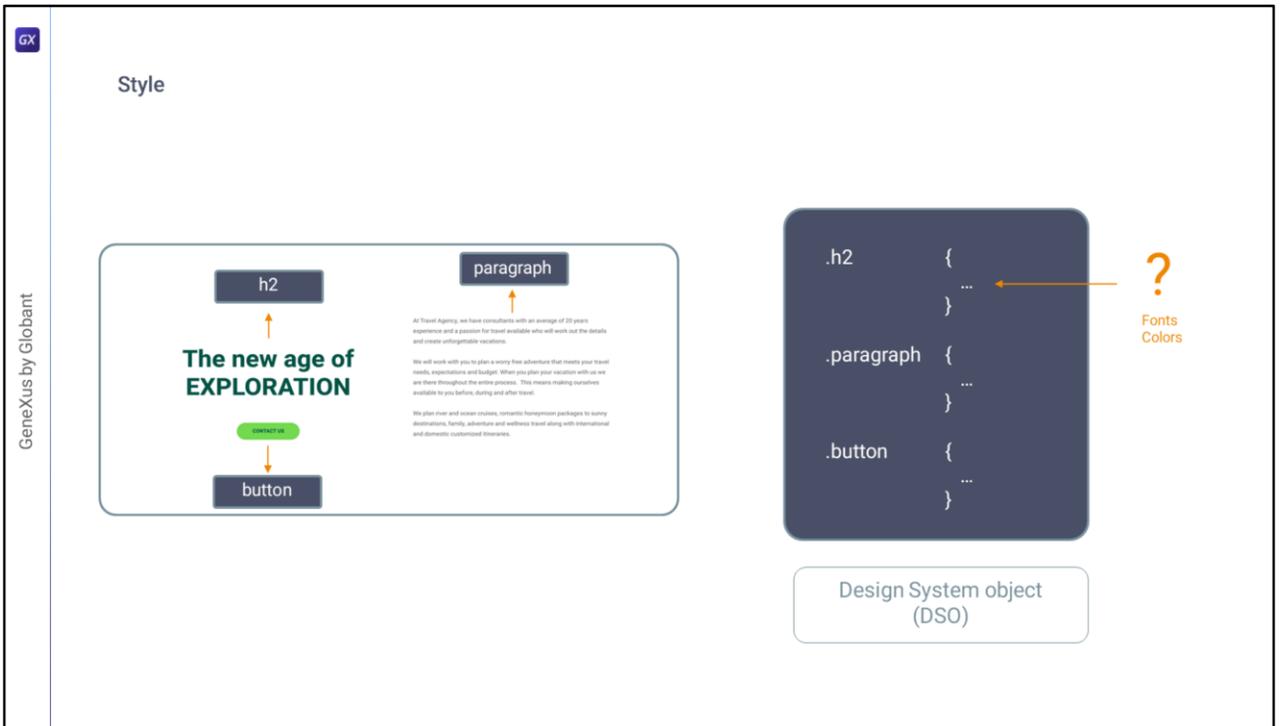
The screenshot shows the GeneXus Style editor interface. On the left, there is a 'Stencil' area containing a design template for a page header. The template includes the text 'The new age of EXPLORATION' and a 'Contact Us' button. On the right, the 'Layout' editor shows a grid with a 'MainTable' containing a 'Title' section and an '&SectionDescription' section. A 'Properties' window is open on the right side of the screen, showing the configuration for the '&SectionDescription' variable.

Attribute/Variable: &SectionDescription	
Control Name	&SectionDescription
Attribute	&SectionDescription
ReadOnly	True
Appearance	
Label Position	None
Class	paragraph
Visible	True
Invisible Mode	Keep Space
Enabled	True
Format	Text
Invite Message	
Control Info	
Cell information	
Row Span	1
Col Span	1
Horizontal Alignment	Default
Vertical Alignment	Middle

... para esta variable vamos a definir una clase paragraph...

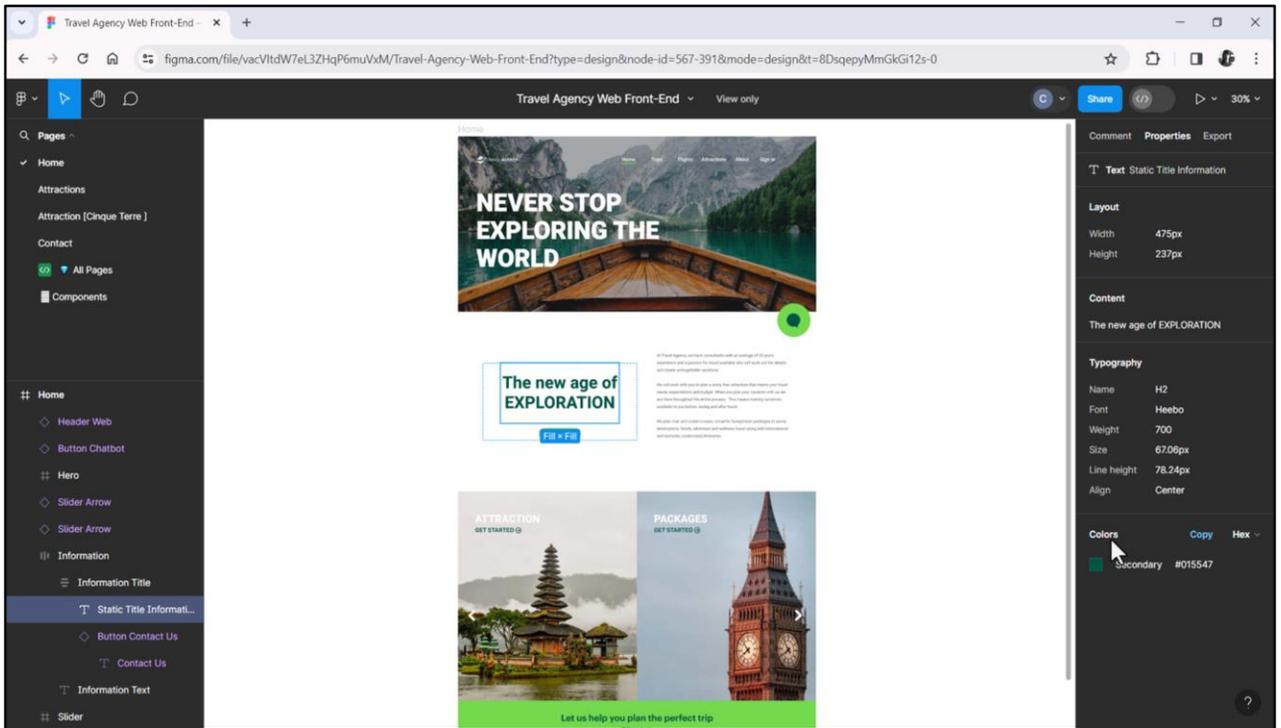


y a este botón le asocié como clase la clase de este nombre.



Y luego, entonces, una vez que asocié a cada uno de mis controles clases, va a alcanzar con, en el Design System object que se va a utilizar para dar estilo a todos los objetos, definir las características de cada una de esas clases (como la fuente, el color, el interlineado, etcétera).

Como ya sabemos estas son clases en algunos casos idénticas a las de CSS. Y como ya podemos imaginarnos, las definiremos a partir del archivo de diseño en Figma que Chechu me compartió y que ya nos había mostrado.



Así, si vamos a Figma en modo View... y nos posicionamos sobre este texto, con la tecla de Control, vemos que en la ventana de propiedades están apareciendo todas estas propiedades que tienen que ver con la tipografía del texto... propiedades a las que Chechu dio un nombre, las llamó h2, y esto va a ser un estilo en Figma que Chechu creó, que en nuestro caso se va a transformar en la clase h2. Y por otro lado tenemos este estilo Secondary, con este valor, que va a ser un estilo de color. Y ese es el color que está sumiendo, si vemos, el texto.

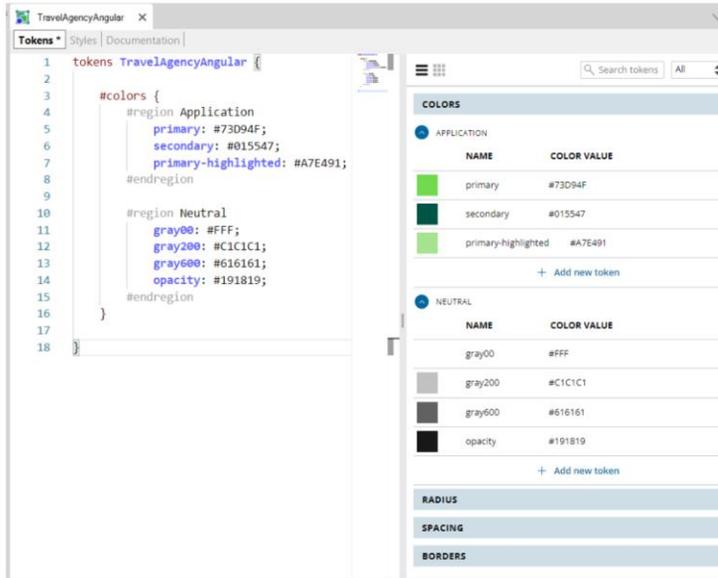
Si hacemos botón derecho sobre el texto / copy as / code / CSS, quedan copiados en el clipboard las propiedades CSS que corresponden a ese texto. Así las voy a pegar para que las veamos en este Notepad... y vemos aquí el H2 con sus propiedades CSS, la del color, la de Secondary y otras más que después veremos si vamos a utilizar o no y para qué sería...

Y lo mismo si nos posicionamos sobre el párrafo, tenemos el estilo al que Chechu nombró Paragraph, con estas características, este color y así.

Lo mismo para el botón, puedo obtener la tipografía a la que llamó de esta manera, con las características, y así sucesivamente.

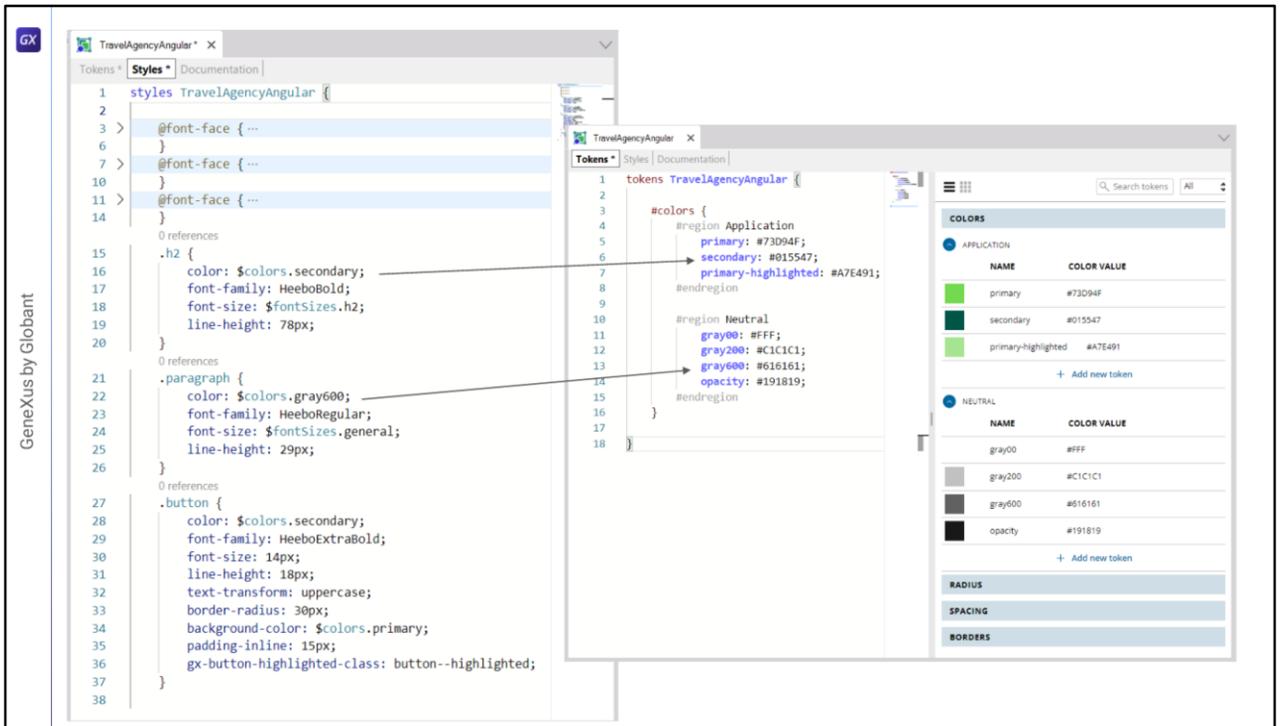
Por otro lado, por ejemplo vayamos a la página All Pages, donde posicionados fuera de los frames, vemos que en la pestaña de propiedades nos están apareciendo listados todos los estilos de texto que creó Chechu (acá tenemos el H2 por ejemplo, acá el Paragraph, y otros, los demás), los que están creados para todas estas pantallas. Y lo mismo los colores que se están utilizando, con estos nombres que son nombres que Chechu creó en Figma y a los que les dio valores.

Bien, entonces con esto...

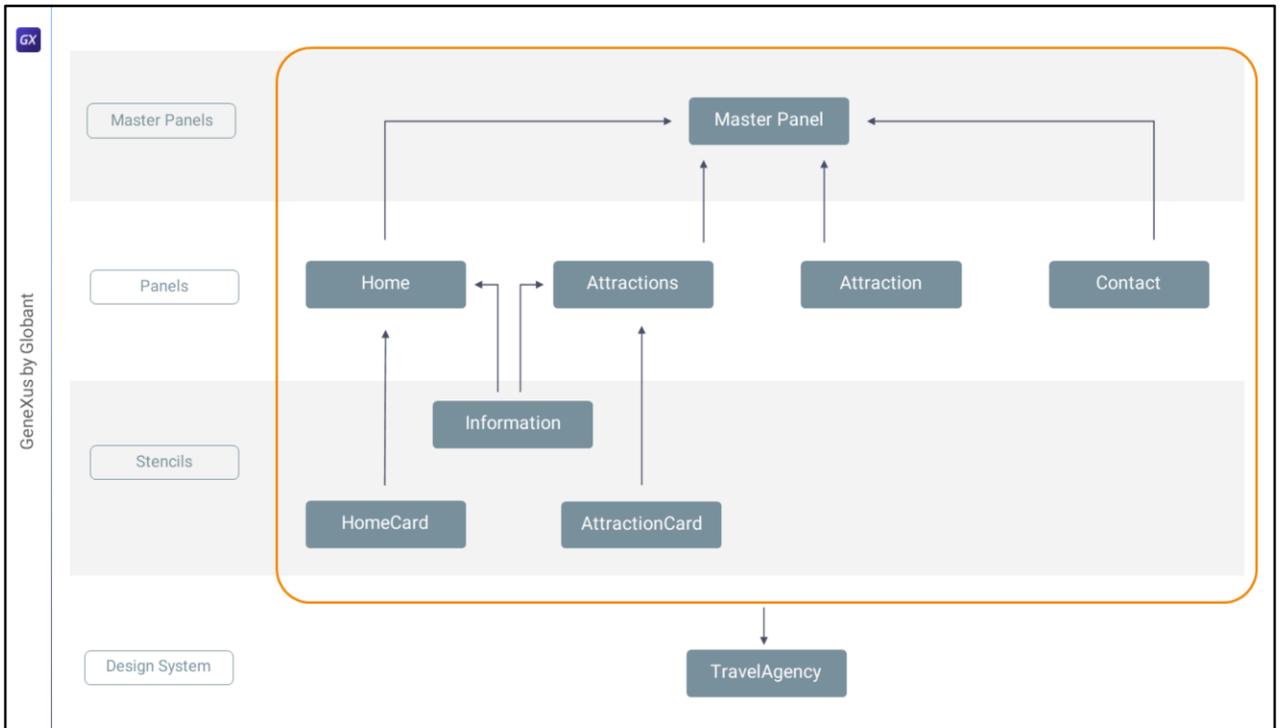


... voy a GeneXus y ya puedo empezar a construir mi objeto Design System.

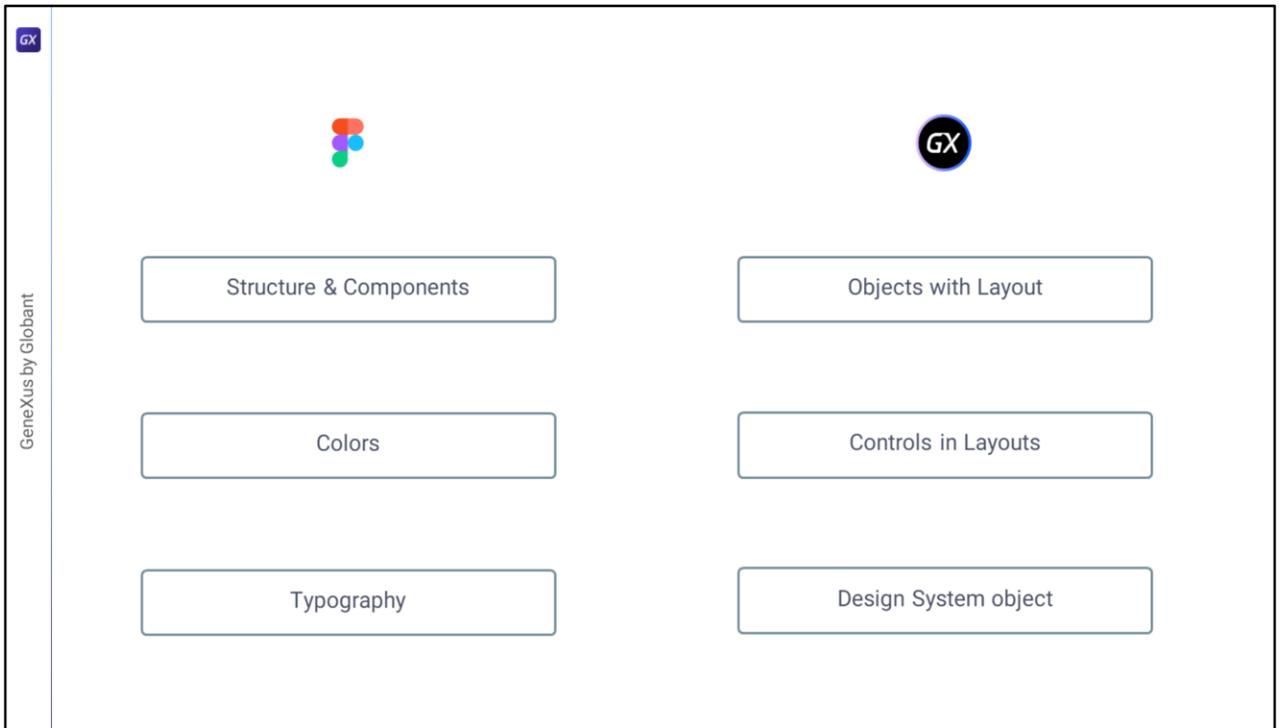
Aquí lo que hice fue, en la pestaña de Tokens, que son las constantes, definir las constantes de color, esas que vimos recién, que Chechu había identificado y nombrado de esta manera, y luego estas constantes de color se van a utilizar...



...dentro de la pestaña de estilos, dentro de las clases. Entonces, por ejemplo a la clase h2 le estoy definiendo la propiedad de color que está utilizando el token que estoy viendo allí. Y así entonces se como empezamos a construir todo lo que tiene que ver con el estilo de la aplicación, a través de la definición de las clases.



Bueno, entonces acá vemos el esquema de lo que vimos hasta el momento, separando toda la aplicación en los objetos: Master Panel, Panels, Stencils y el Design System comandando el diseño y la estética de todos los objetos...



Y con esto ya terminamos un primera repaso de todos estos jugadores que van a intervenir en GeneXus para reflejar todas las decisiones que Chechu toma en su Design System. Así que ahora lo que nos queda es poner manos a la obra.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com