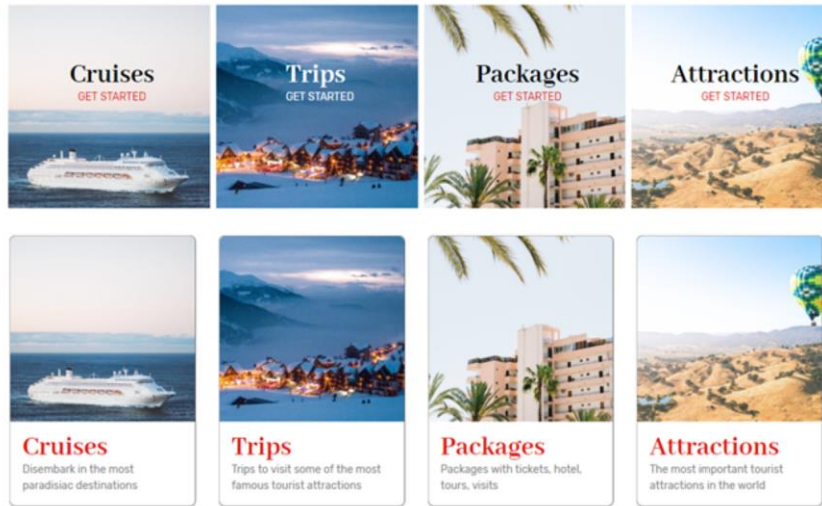


# Design System en GeneXus

Frontend: STENCILS

*GeneXus* 16

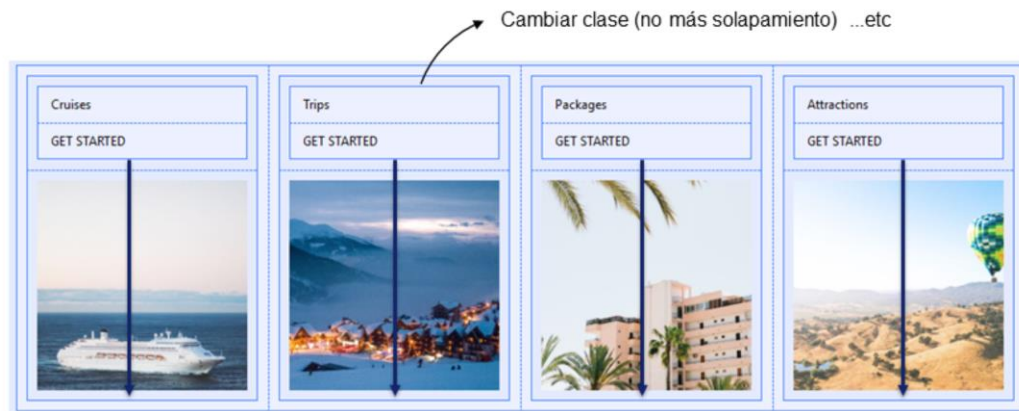
## Cambio de diseño de web panel Home



Supongamos que queremos cambiar nuestro diseño de la página home del frontend que venimos desarrollando, para que en vez de lucir de este modo, luzca de este otro modo.

Lo que vemos, fundamentalmente, son cambios en el diseño, que se repiten en cada uno de estos cuatro elementos.

## Cambio de diseño de web panel Home



¿Tenemos que repetir los cambios cuatro veces?

En principio, vemos que estas tablas tendrán que moverse para colocarse debajo de cada imagen. Además tendremos que cambiarles la clase, para que no se solapen más sobre la imagen correspondiente, sino que queden en su lugar. Y más...

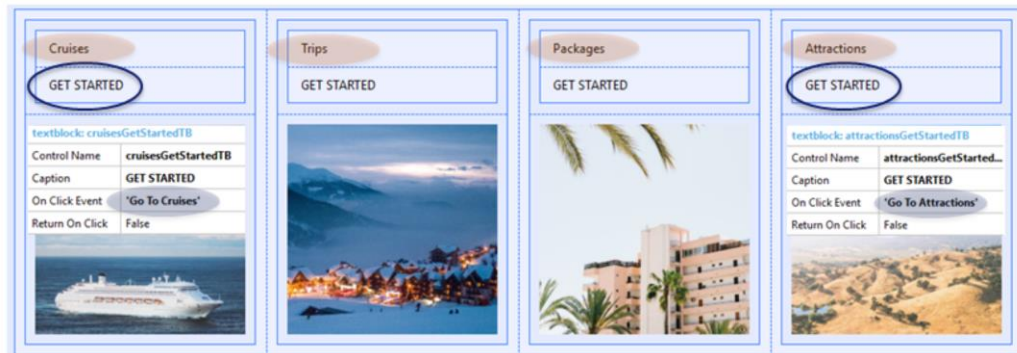
La pregunta que surge es: ¿estamos obligados a repetir los cambios cuatro veces?

## Cambio de diseño de web panel Home



Es claro que cada una de estas tablas pueden ser pensadas como un elemento de diseño en sí mismo, que contiene un conjunto de controles (en este caso, una tabla con dos text blocks, y una imagen).

## Cambio de diseño de web panel Home

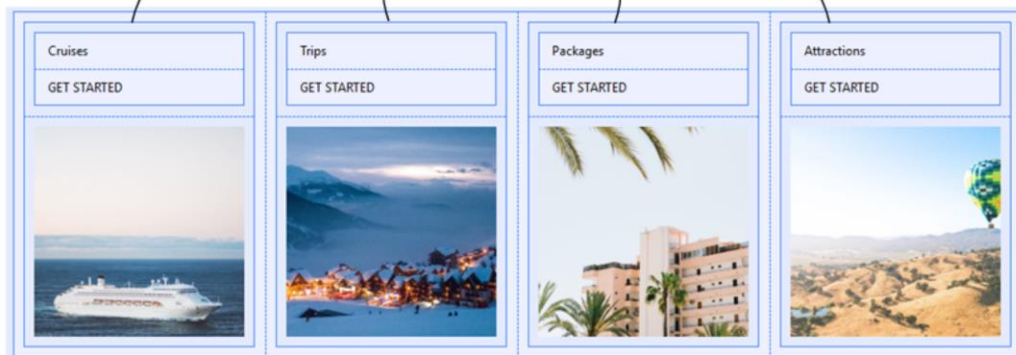


Particularidades en cada componente

Claro que cada una tiene particularidades propias (por ejemplo, cada GET STARTED tiene un evento diferente asociado, los captions e imágenes son diferentes), pero el diseño es el mismo (a excepción del de Trips, que tiene los textos en blanco. No atendamos esto ahora).

## Objeto Stencil

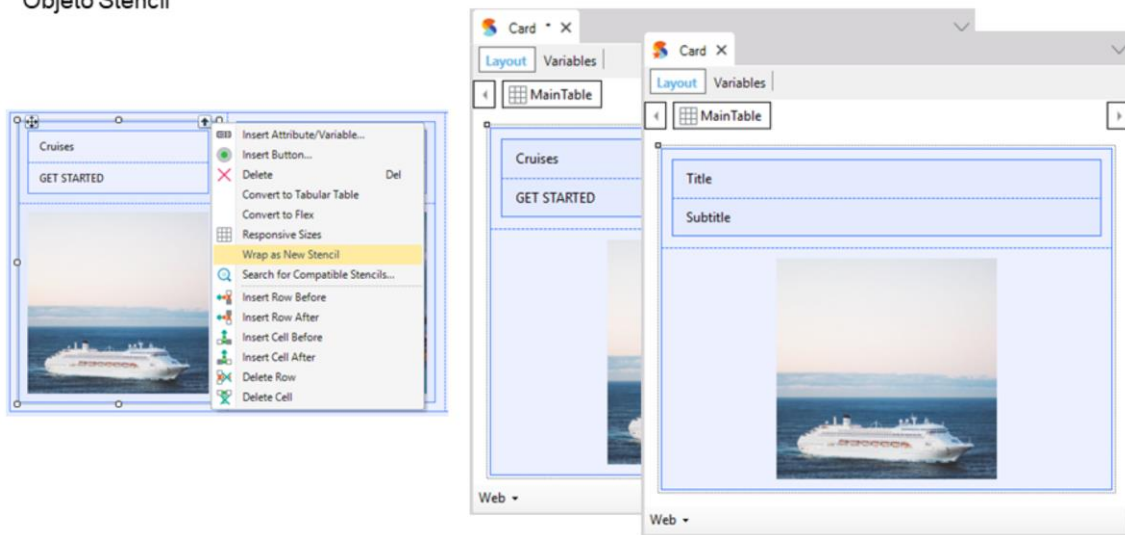
Para centralizar el diseño de cada conjunto de controles



¿Cómo hacemos para centralizar el diseño de ese conjunto de controles en un único lugar, de manera de modificarlo solo allí y que eso ya impacte en todos los lugares que lo utilicen?

La respuesta es: a través del objeto Stencil.

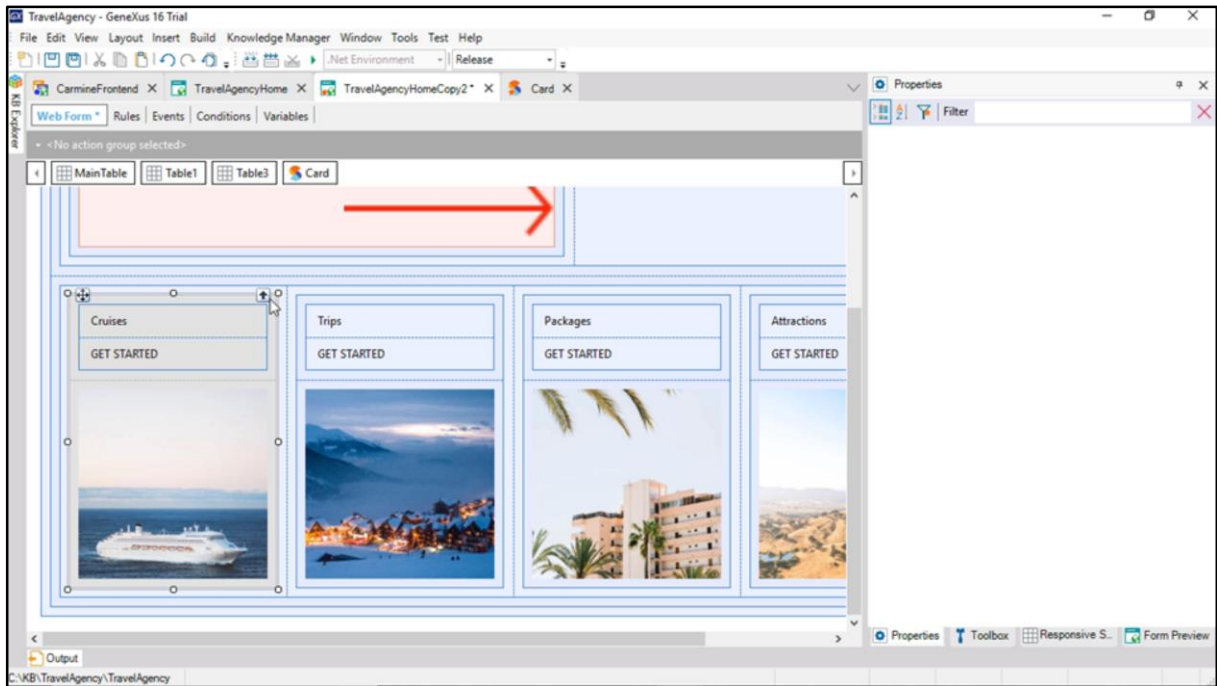
## Objeto Stencil



Grabemos el web panel home con otro nombre.

Nos posicionamos sobre la tabla y vemos que presionando botón derecho, entre las opciones tenemos esta, Wrap as new stencil (que sería como “envolver como un nuevo stencil”). Con esto crearemos ese objeto que centralizará el diseño. Tenemos que elegir un nombre que abstraiga su cometido, para poder luego utilizarlo en otras situaciones análogas. Llamémosle Card, porque es como una tarjeta de presentación de cada opción.

Vemos que se copió esa tabla a este layout. Si bien estamos viendo la imagen del crucero, y los captions Cruises y GET STARTED, están solo a modo de ejemplo. Vamos a modificarlos para que queden más generales, y poder reutilizar luego este stencil en diferentes situaciones.. Si los dejamos como estaban, con Cruises y Get Started, igual va a funcionar, pero los cambiamos por prolijidad, porque la idea, es justamente, conseguir una abstracción. Grabemos.



[ DEMO: [https://youtu.be/Cg\\_xnaULcgU](https://youtu.be/Cg_xnaULcgU) ]

Vemos que en el web panel la tabla que envolvimos como stencil ya es distinta. No tiene propiedades de diseño (por ejemplo, comparemos con la de al lado: no tiene la propiedad class, que es una propiedad de diseño).

Comparemos las propiedades del text block. Vemos que todas las propiedades que gobiernan el diseño del text block han desaparecido. Solamente quedan aquellas que gobiernan su comportamiento.

Por ejemplo, el caption puede ser variado respecto al del Stencil, y el evento asociado. Es que a partir de ahora, el diseño será gobernado desde el stencil. Solamente podemos manipular aquí su comportamiento. Esa es la gracia.

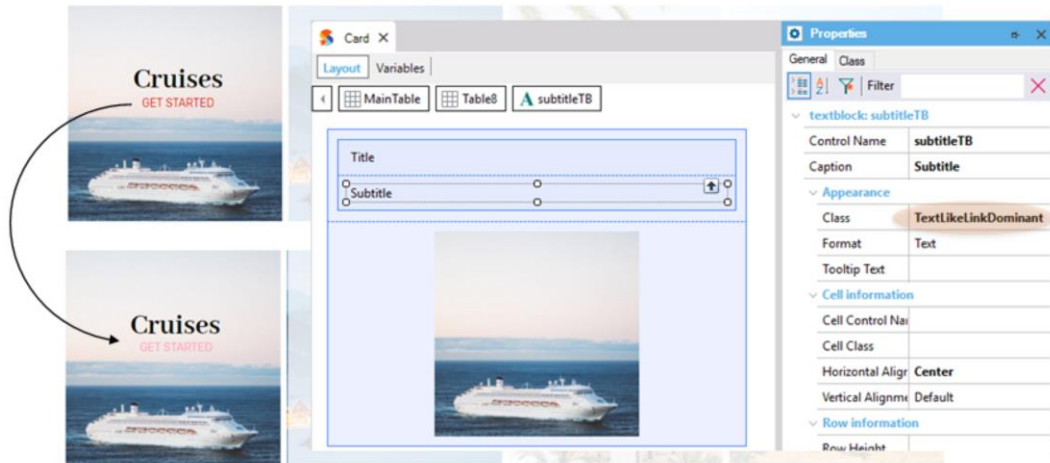
Entonces, sustituyamos, ahora, cada una de estas otras tablas por el stencil recién creado. Nos posicionamos sobre la tabla, y elegimos buscar stencils compatibles. ¿Que significa? Que tengan el mismo conjunto de controles. Por ahora solamente tenemos uno en la KB, el que acabamos de crear. Presionamos Swap, porque queremos cambiar esta tabla por el stencil. Vemos que mantuvo la información concreta (la foto, el caption), no la sobrescribió con lo que venía del Stencil. Era lo que esperábamos.

Hacemos lo mismo con las otras tablas.

En ejecución veremos exactamente lo mismo que antes (a excepción de Trips que tenía otras clases; no nos vamos a preocupar por eso ahora). Pero nuestra KB logró un grado de abstracción mayor, que mejorará la consistencia y ayudará a mantener el Design System.



## Cambio en el diseño

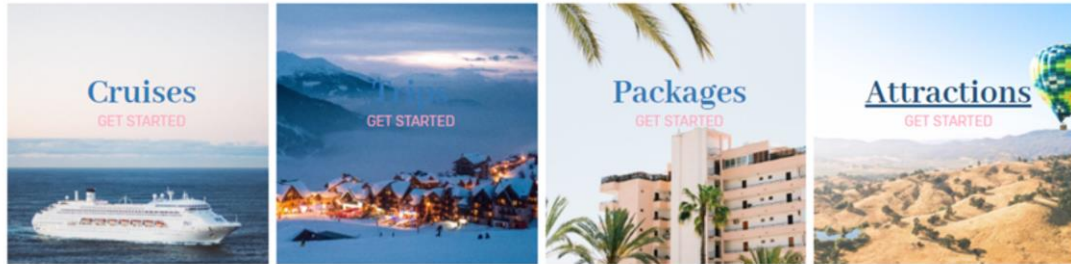


Por ejemplo, supongamos que queremos modificar el color del GET STARTED por otro. Entonces lo hacemos en el stencil. Cambiamos su clase por esta otra. Esto repercutirá inmediatamente en todas las pantallas que estén utilizándolo.

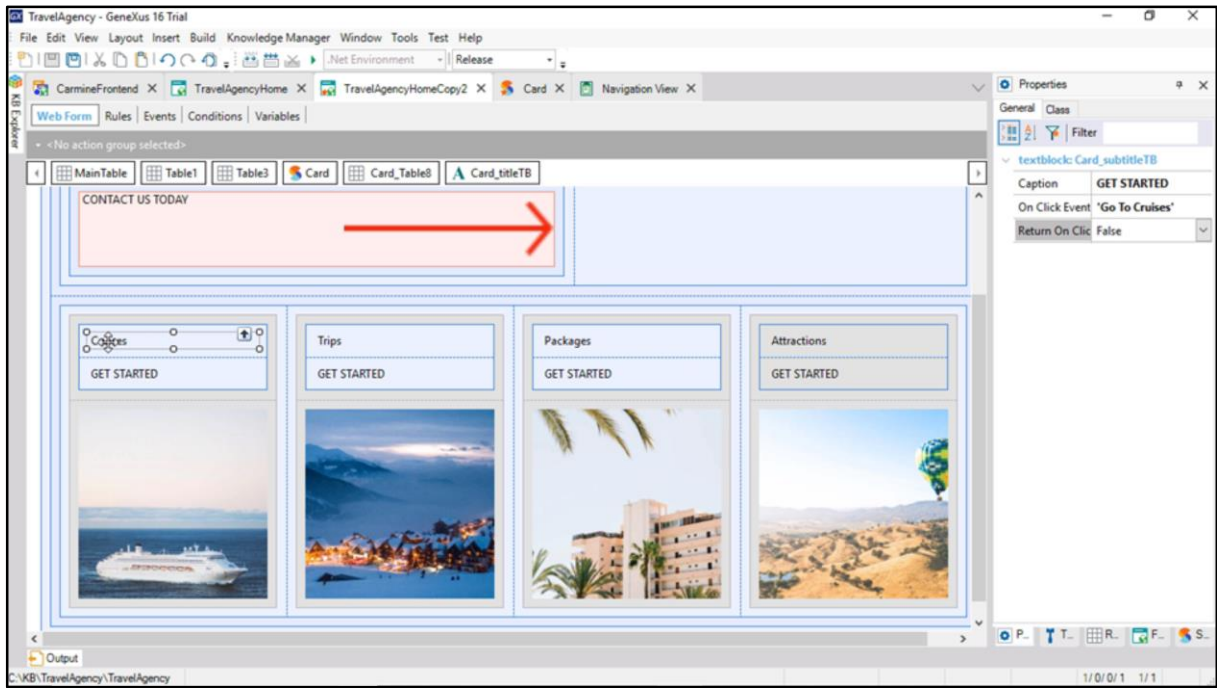
Así que volvamos a generar nuestro panel y a ejecutarlo... Vemos el cambio en las cuatro tablas, que conservan su comportamiento. Por ejemplo, esta sigue invocando al web panel de Attractions. Pero el cambio de diseño lo hicimos en un único lugar: en el stencil.

Si queremos modificar otra cosa, que tenga que ver con el comportamiento, eso sí lo tendremos que hacer uno por uno (a menos que queramos centralizar también el comportamiento... Para ello tenemos los web components, pero no entraremos en eso aquí)

Cambio de comportamiento (no de diseño)

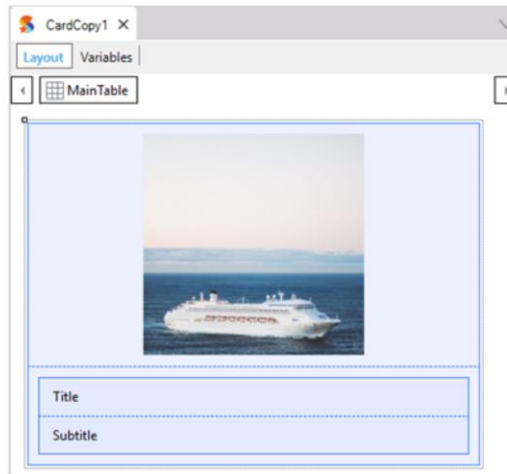


Por ejemplo, si quisiéramos que el link esté en el título y no en el GET STARTED...

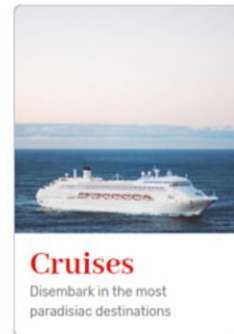


[ DEMO: <https://youtu.be/H-b2xbgk-9g> ]

Creemos otro stencil para otra variedad



Cambiamos las clases por otras,  
para implementar todos los cambios  
requeridos

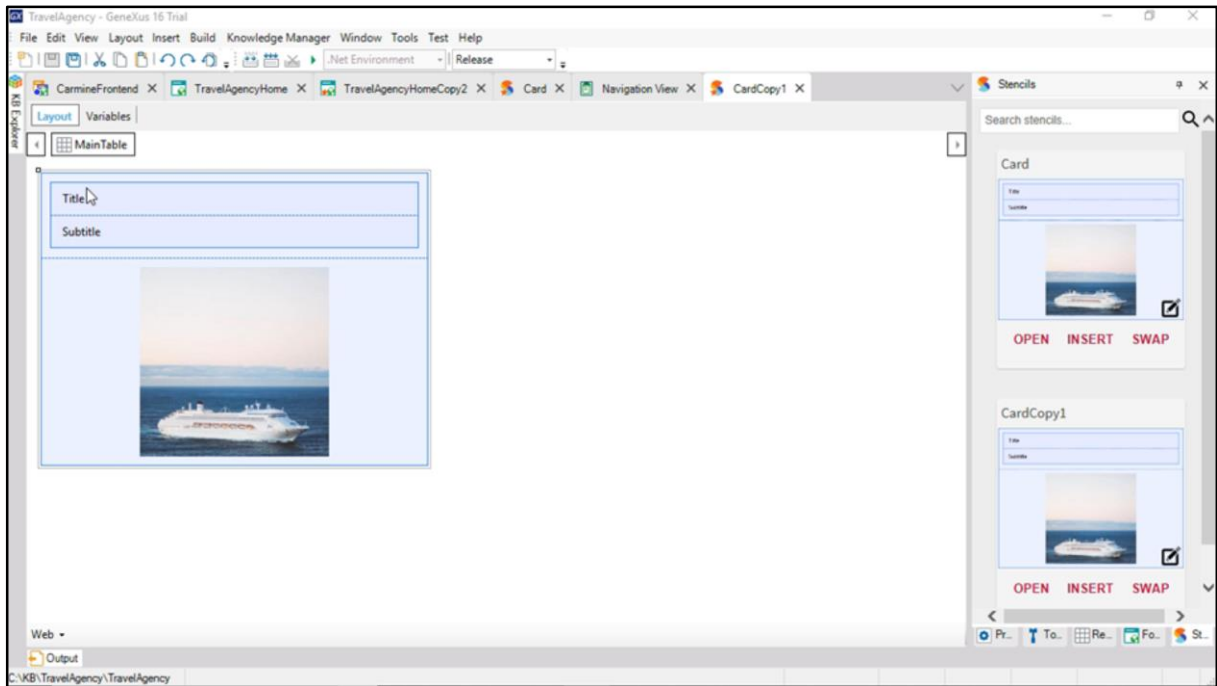


En todo momento podemos decidir dejar de utilizar el stencil, haciendo Unwrap.

También podemos insertar stencils directamente en el form...

Y crear stencils de cero...

O trabajando a partir de otro. Grabemos el que teníamos con otro nombre. Vamos a querer implementar el diseño que veíamos al principio.



[ DEMO: <https://youtu.be/hhiWa8qQSik> ]

Primero movamos la tabla con los textos para abajo.

Cambiamos su alineación (no la queremos centrada en relación a la celda).  
Y cambiamos su clase, que era la que comandaba el solapamiento sobre el otro control, por esta otra, que lo único que hace es dar paddings para los cuatro costados, para que el contenido tenga un espacio de 16 pixels respecto a la tabla contenedora.

Luego vemos el primer text block...

También le cambiamos la alineación...

Y su clase... que lo mostraba en el color default... por esta otra, que lo mostrará en el color base de la aplicación, que era este rojo.

Análogamente para este otro text block...

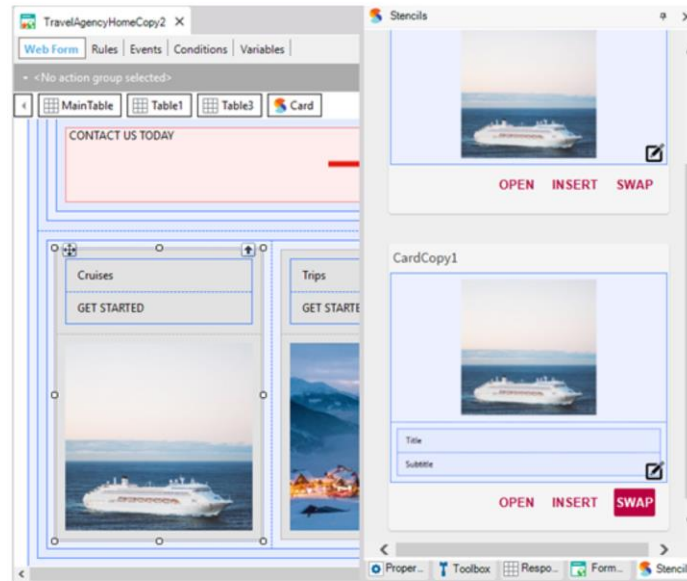
Modificamos su alineación y si vemos su clase... la habíamos cambiado por esta que tenía este color... Ahora la modificamos por esta otra...

Por otro lado, queremos que la tabla tenga un radio para que se muestren los bordes redondeados...Entonces cambiamos su clase por esta otra...

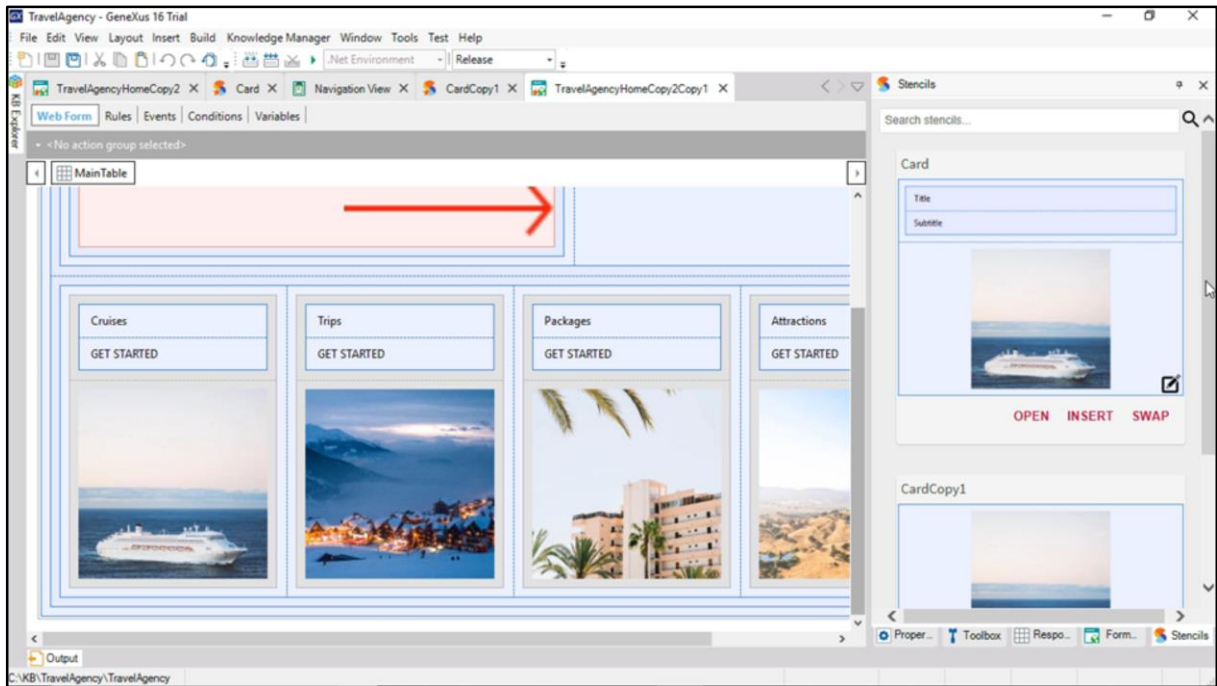
Y por último, modificamos la imagen y su celda... Teníamos esta clase para la celda de esa imagen, que quitaba el padding. No nos interesa más. La quitamos. Y la clase de la imagen querremos modificarla, para que ocupe el 100% del ancho de la tabla, y tenga sus bordes superiores redondeados...

Grabamos.

## Variación del stencil



Tenemos esta variación, entonces, del Stencil que habíamos creado. Los controles son los mismos. Solo cambia su diseño.



[ DEMO: <https://youtu.be/wfi38sjVqg0>]

Grabemos con otro nombre nuestro web panel home, y cambiemos en las cuatro celdas el stencil anterior por este otro.

Generemos y ejecutemos el nuevo web panel. Vemos perfectamente los cambios. Nos evitamos hacerlos cuatro veces.

Si ahora observamos el panel al que queremos llegar... Vemos que este texto fue sustituido por estas descripciones... Es decir, debemos modificar el Caption del text block.

¿Dónde lo hacemos? ¿En el stencil? La respuesta es claramente NO. El contenido del text block, lo que dice, no es parte del diseño. Y es particular del lugar donde se esté utilizando. Por tanto...

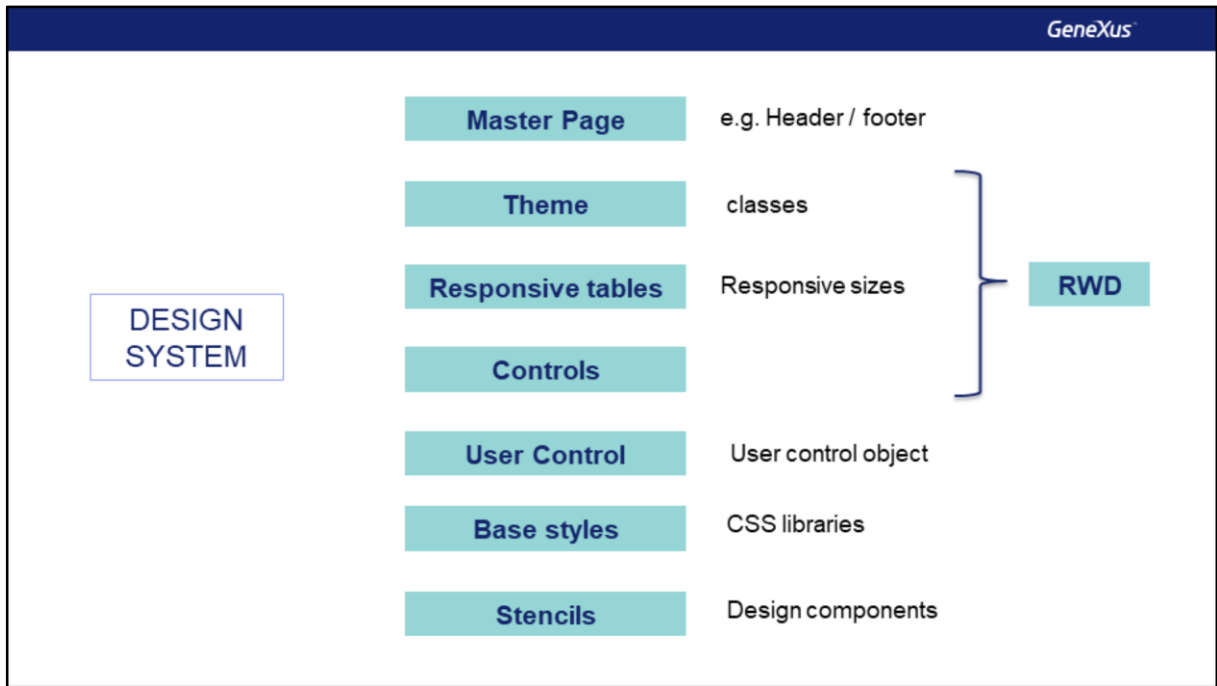
## Stencils

- Se pueden reutilizar en distintos paneles
- Si controles unidades mínimas de UI y sus clases centralizadoras del diseño → stencils constituyen unidades más grandes, compuestas de controles, para definir su diseño
- Diseño sin código
- Reutilización de diseño

Aquí vimos un ejemplo donde el conjunto de controles que se repetía lo hacía dentro del mismo form, pero no tiene por qué ser solo así. Podría repetirse en diferentes paneles.

En definitiva, si los controles pueden pensarse como las unidades mínimas que conforman la UI, y sus clases permitían centralizar su diseño, los stencils crean “unidades” de diseño más grandes, compuestas de controles, para definir su diseño (sin código asociado), de manera de ser reutilizado en diferentes objetos.





Con esto vimos una muestra de la importancia de los stencils para colaborar en la definición de estándares de diseño que nos permiten conseguir mayor consistencia, simplificando la implementación del Design System.



Videos

[training.genexus.com](https://training.genexus.com)

Documentation

[wiki.genexus.com](https://wiki.genexus.com)

Certifications

[training.genexus.com/certifications](https://training.genexus.com/certifications)