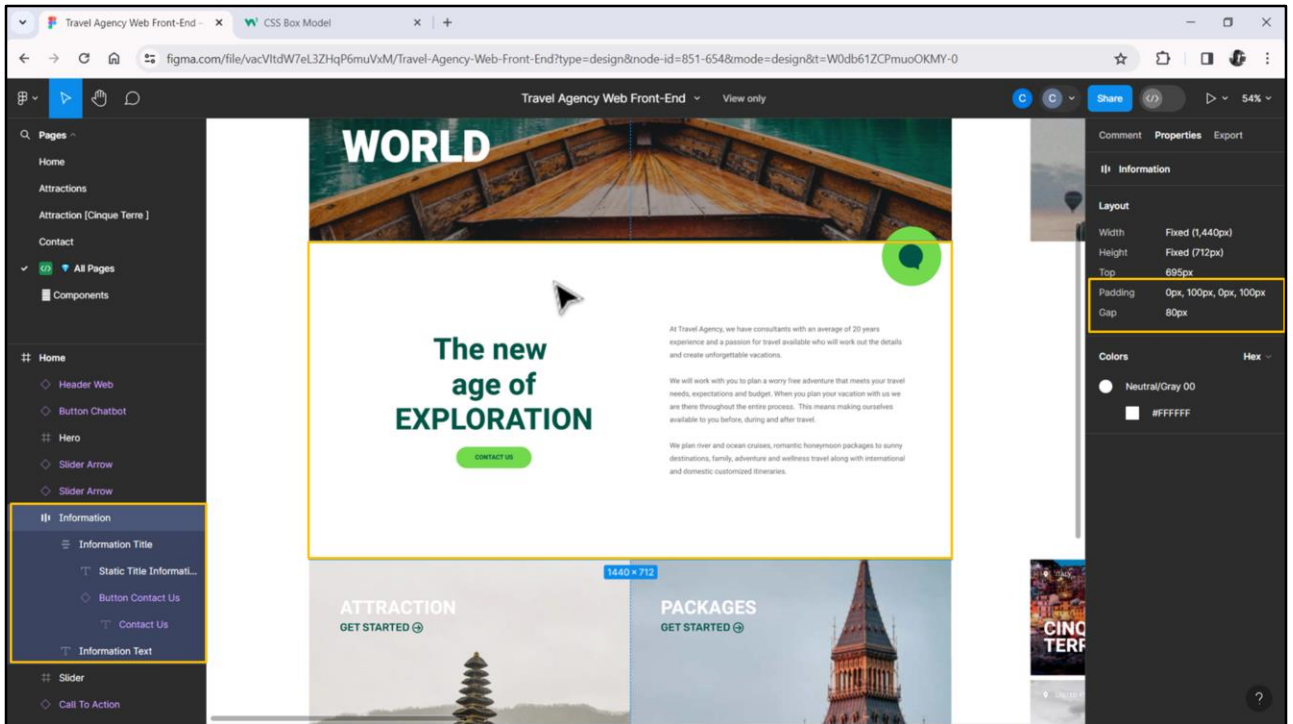


First Layout in GeneXus. Structure: spacing



Cecilia Fernández

Vamos a retomar donde habíamos quedado en el video anterior.



Allí habíamos dicho que antes de investigar los tamaños de esta tabla interior íbamos a ver algunos aspectos de la tabla exterior.

¿Qué más sabemos inspeccionando este frame? Vemos acá estas propiedades, y en particular prestemos atención a estas dos, que van a tener que ver con el espaciado, justamente, entre el contenido.

Travel Agency Web Front-End ... CSS Box Model

w3schools.com/css/css_boxmodel.asp

Tutorials Exercises Certificates Services Search... Set Goal Spaces Get Certified Sign Up Log in

HTML CSS JAVASCRIPT SQL PYTHON JAVA PHP HOW TO W3.CSS C C++ C# BOOTSTRAP REACT MYSQL JQUERY EXCEL XML DJANGO NUMP

CSS Tutorial

- CSS HOME
- CSS Introduction
- CSS Syntax
- CSS Selectors
- CSS How To
- CSS Comments
- CSS Colors
- CSS Backgrounds
- CSS Borders
- CSS Margins
- CSS Padding
- CSS Height/Width
- CSS Box Model**
- CSS Outline
- CSS Text
- CSS Fonts
- CSS Icons
- CSS Links
- CSS Lists
- CSS Tables
- CSS Display
- CSS Max-width
- CSS Position
- CSS Z-index
- CSS Overflow

The CSS Box Model

In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: content, padding, borders and margins. The image below illustrates the box model:

The diagram illustrates the CSS Box Model as a series of nested rectangles. The outermost layer is a light gray area labeled "Margin". Inside this is a thick green border labeled "Border". Within the border is a light gray area labeled "Padding". The innermost area is a white rectangle labeled "Content".

Explanation of the different parts:

Start today

W3schools

Create a Free Website

Get started!

Ya sabemos que el Padding dentro del box model de CSS es el espaciado interno del contenido respecto de los 4 bordes...

Es decir, este es el límite del elemento, que en esta representación tiene un borde de este tamaño, y un espaciado interno de este tamaño.

Travel Agency Web Front-End | CSS Padding | W3Schools Tryit Editor

w3schools.com/css/tryit.asp?filename=trycss_boxmodel

BUILD YOUR CAREER. GET FULL ACCESS. SAVE 770\$ Start today

Home CSS CSS Box Model Tryit: The box model

Run Result Size: 753 x 590 Get your own website

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
background-color: lightgrey;
width: 300px;
border: 15px solid green;
padding: 30px 60px 90px 120px;
margin: 20px;
}
</style>
</head>
<body>
<h2>Demonstrating the Box Model</h2>
<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>
<div>This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</div>
</body>
</html>
```

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Como sabemos el Padding no tiene por qué ser uniforme. Acá estamos viendo un ejemplo, una prueba, donde estoy colocando padding-top, el padding respecto al borde superior, de 30 píxeles; el padding respecto al borde de la derecha, padding-right, de 60; respecto abajo 90; y respecto a la izquierda, 120.

Esta es la notación abreviada. Podríamos utilizar cada una de las propiedades en forma separada.

Travel Agency Web Front-End | CSS Padding | W3Schools Tryit Editor

w3schools.com/css/tryit.asp?filename=trycss_boxmodel

BUILD YOUR CAREER. GET FULL ACCESS. SAVE 770\$ Start today

Home > CSS > CSS Box Model > Tryit: The box model

Run > Result Size: 753 x 590 Get your own website

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 30px 120px;
  margin: 20px;
}
</style>
</head>
<body>
<h2>Demonstrating the Box Model</h2>
<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>
<div>This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</div>
</body>
</html>
```

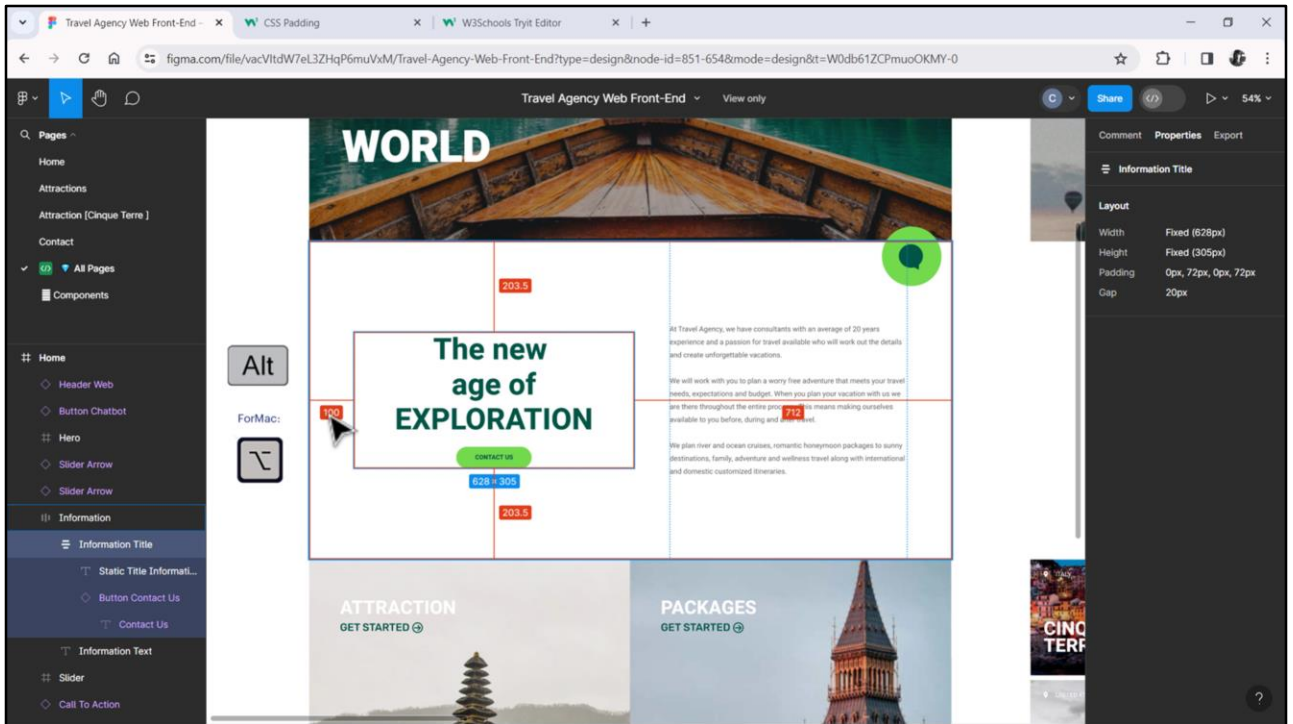
Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

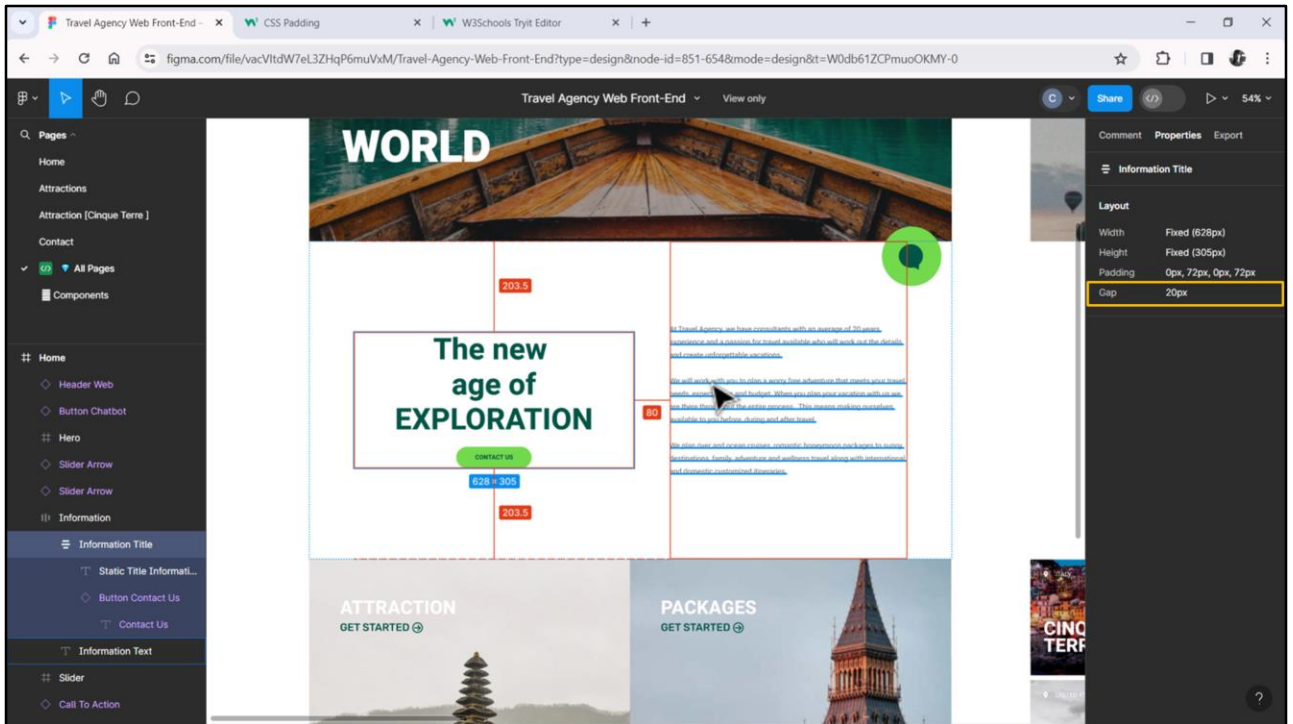
This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Y si arriba y abajo, por ejemplo, queremos que sean lo mismo y derecha e izquierda también, entonces abreviamos de esta manera, y allí vemos cómo arriba y abajo están con 30, derecha e izquierda con 120...

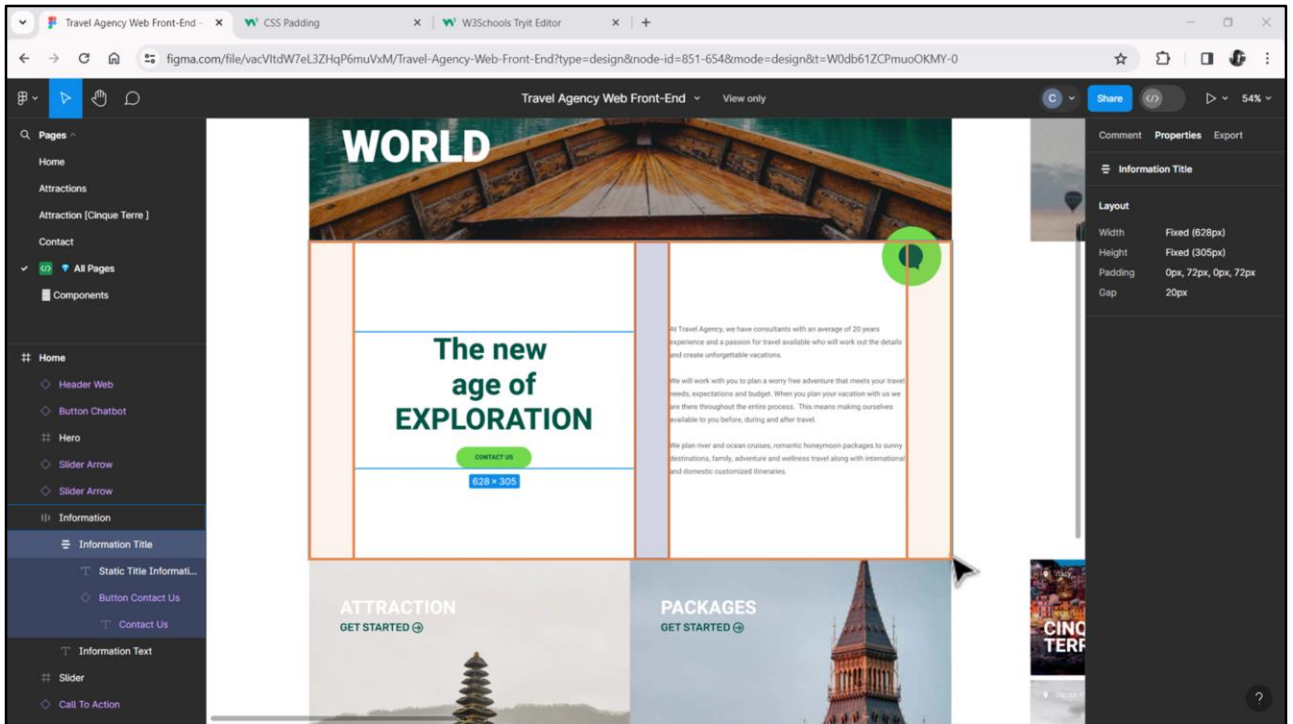
... o incluso si dejamos solamente uno, eso significa que de los cuatro costados va a aplicar el mismo padding.



Entendido esto, entonces, sabemos aquí que va a haber un padding de 100 píxeles de la izquierda y de 100 píxeles de la derecha. Y esto también lo podemos ver gráficamente eligiendo el primer elemento y con la tecla ALT si estamos en Windows vemos estos 100 píxeles de aquí... y si ahora nos posicionamos en este otro elemento vemos estos 100 píxeles de aquí.



Y por último ¿qué pasa con el gap, qué es el gap? Ni más ni menos que la distancia entre los elementos internos al contenedor. Otra vez, en este caso, si nos paramos en este, y vemos la relación con respecto a este elemento vemos esos 80 que está indicando allí.



Hay varias formas de establecer este espaciado en GeneXus. Una va a ser casi una traducción literal de estas propiedades. La veremos luego, cuando pasemos a especificar el DSO. Pero otra será más fuertemente estructural: ¿qué tal si pensamos estos espacios vacíos no como paddings, sino como columnas vacías? Es decir, la tabla Main tendrá 5 columnas en lugar de 2: la primera y la última de ancho 100 píxeles y la del medio de ancho 80.

Travel Agency Web Front-End

Pages

- Home
- Attractions
- Contact
- All Pages
- Components

Home

- Header Web
- Button Chatbot
- Hero
- Slider Arrow
- Information
- Information Title
 - Static Title Informati...
 - Button Contact Us
 - Contact Us
- Slider
- Call To Action

WORLD

The new age of EXPLORATION

CONTACT US

828 x 305

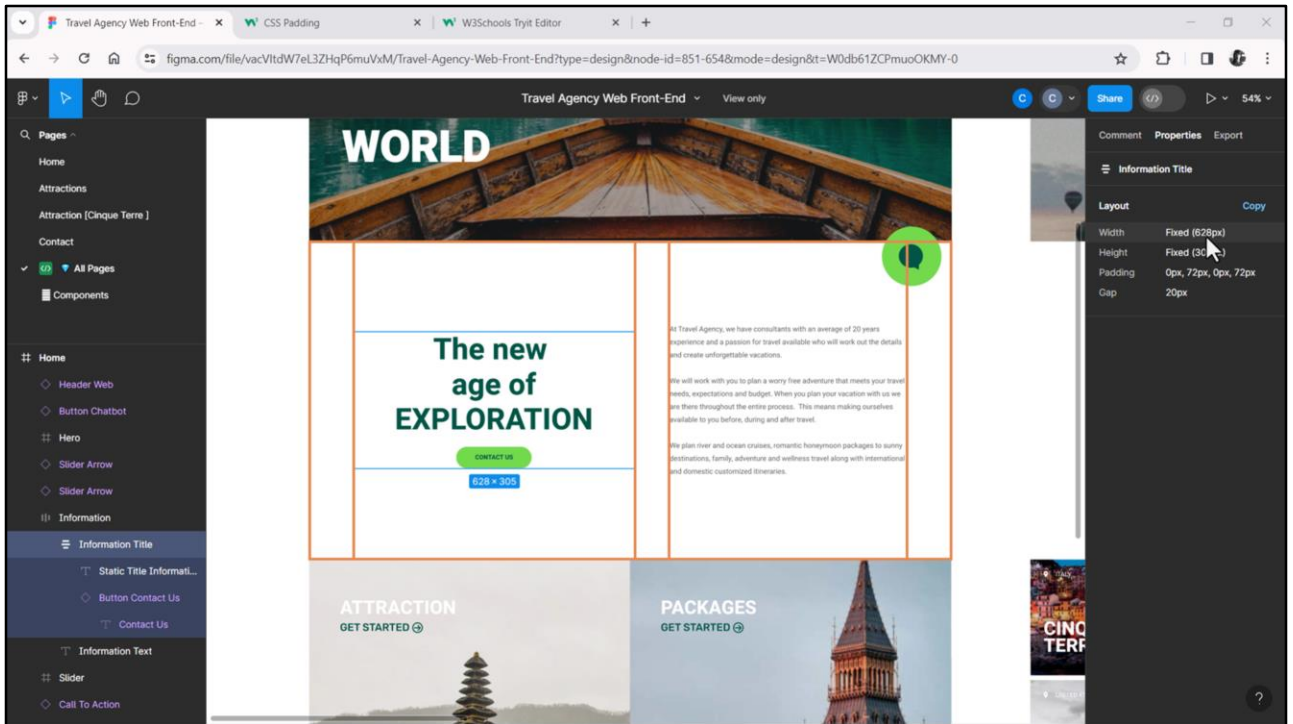
ATTRACTION GET STARTED

PACKAGES GET STARTED

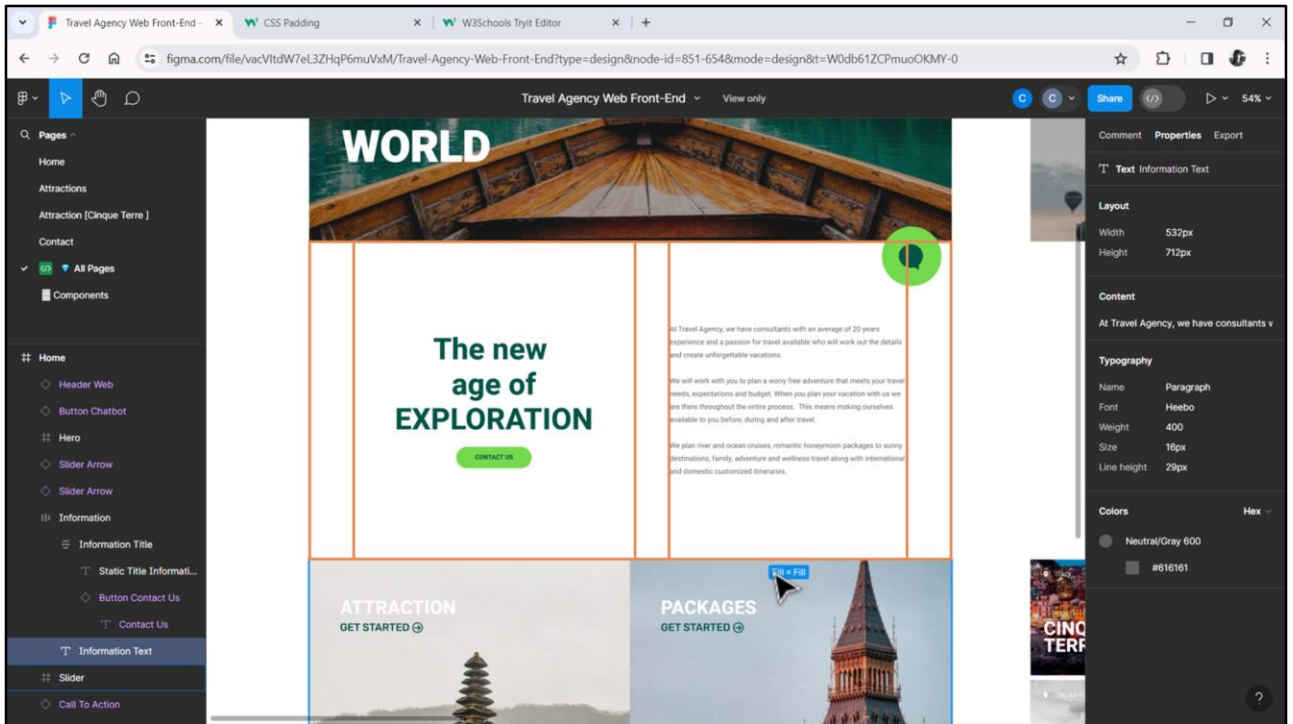
CINQUE TERRE

Table control can be seen as an abstraction of the CSS "display: grid", which allows designing a grid of rows and columns to place elements in certain places, allowing empty cells where no elements are placed, unlike what happened when the html table tag was used.

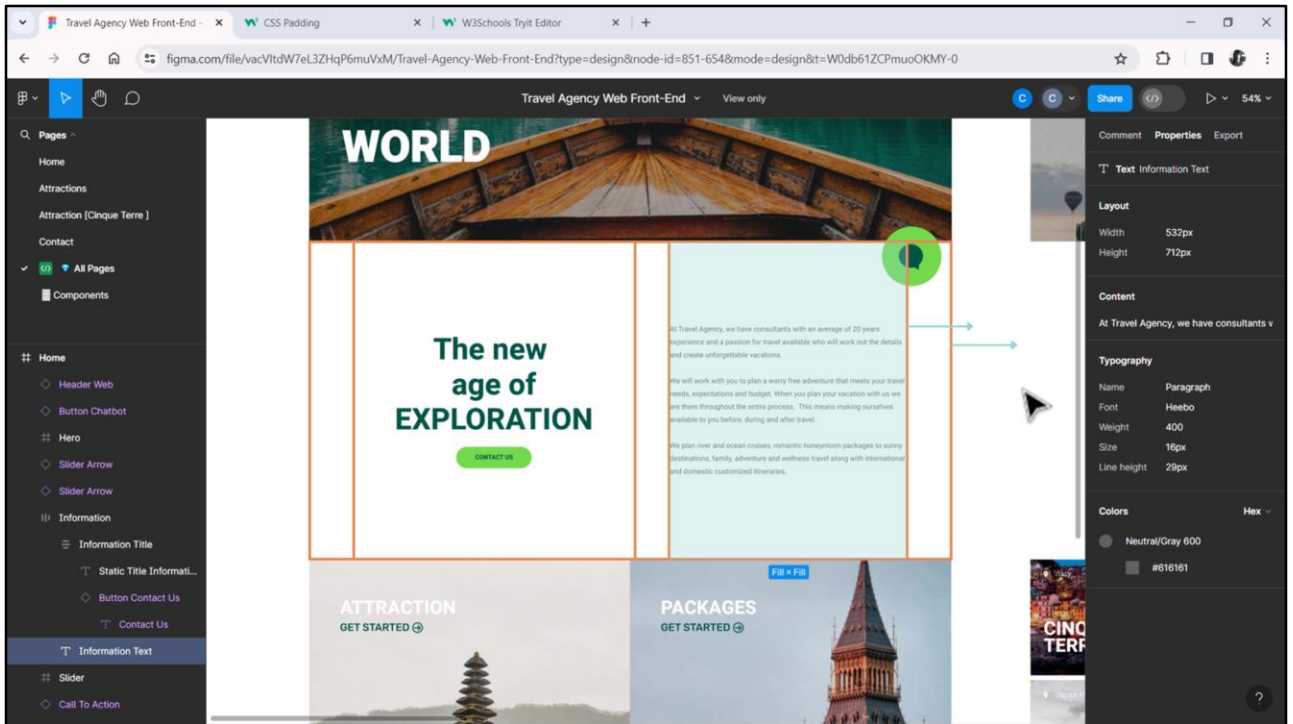
Algunos de ustedes se pueden estar preguntando si dejar columnas vacías no representará un problema de rendimiento. Y la respuesta es que no y les dejo acá una breve explicación.



¿Y las otras dos, las del contenido propiamente dicho? ¿Qué ancho deberán tener? Si analizamos el ancho del primer elemento, vemos que es fijo, de 628 píxeles.



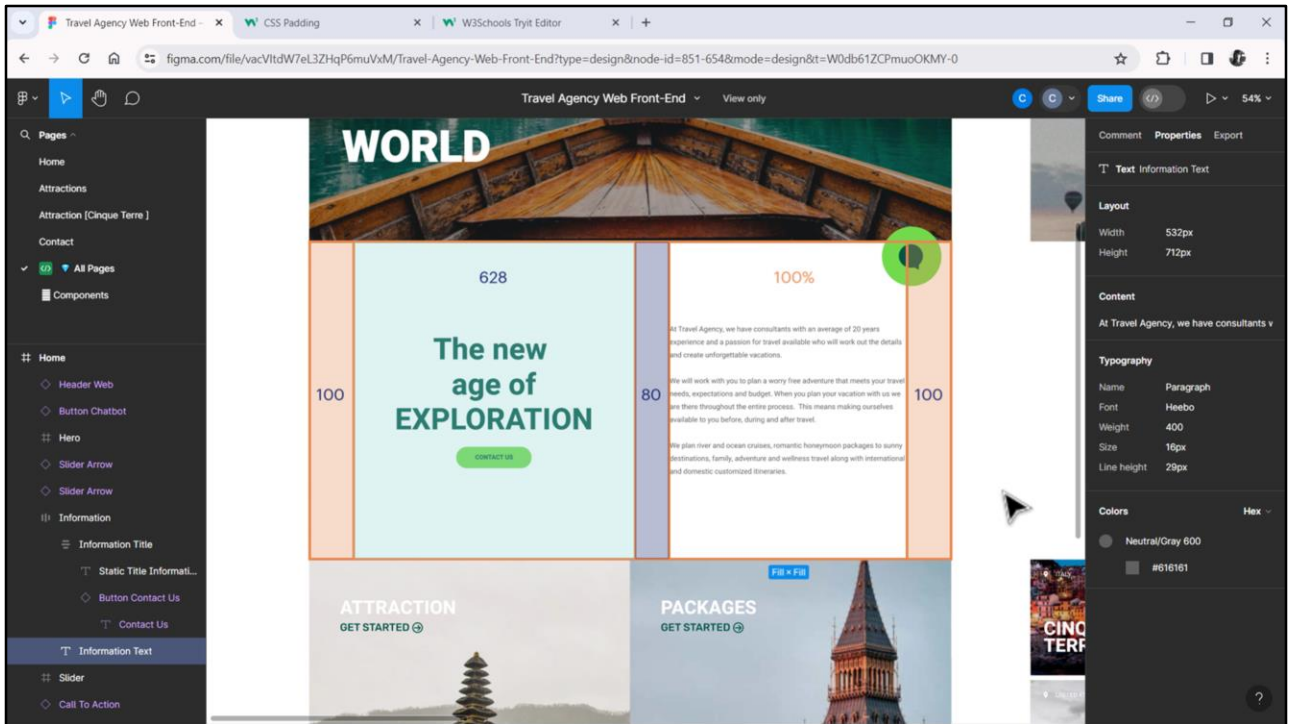
Y si analizamos el segundo elemento, ya no es fijo, dice Fill. Es decir, que va a ocupar todo el tamaño del que dispone, tanto horizontalmente como verticalmente.



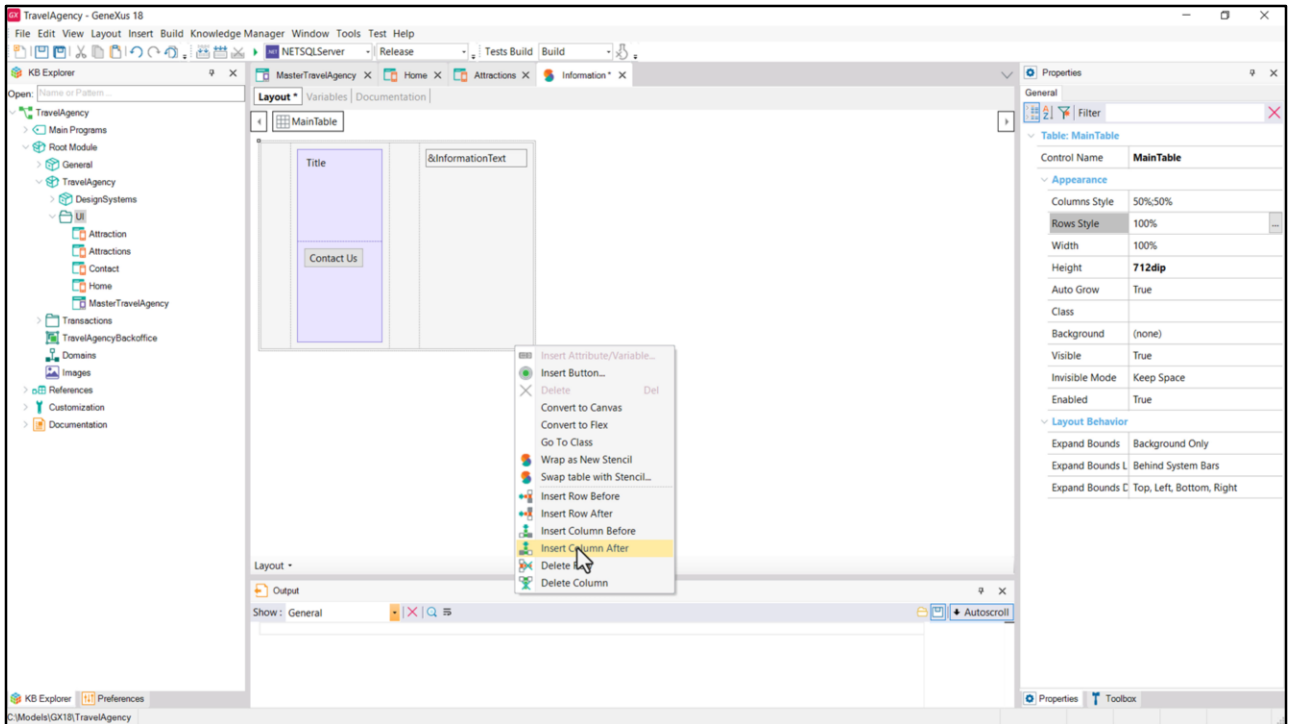
Esto significará que si yo extiendo hacia la derecha el borde del browser, mientras el elemento de la izquierda quedará tal cual, el de la derecha se expandirá también hacia la derecha.

Y si reduzco el tamaño, también se achicará. Los espacios, en cambio, permanecerán siempre fijos, de 100, 80 y 100 píxeles.

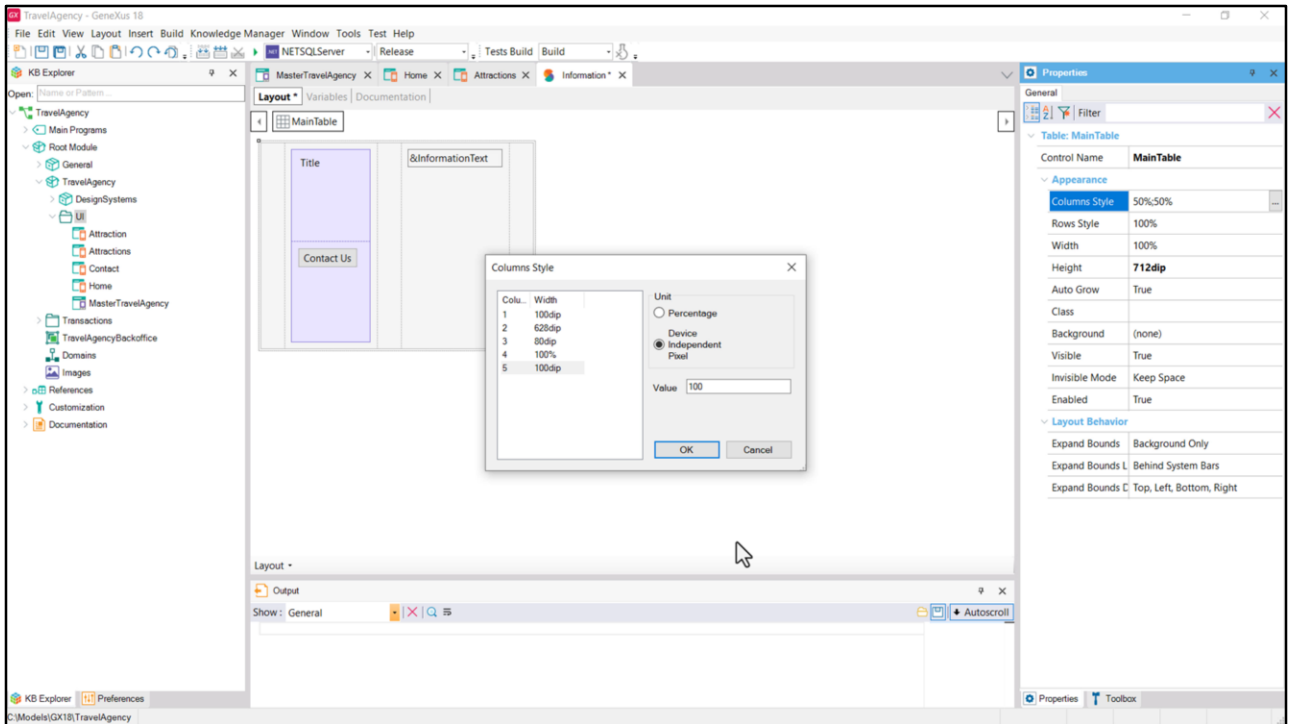
Es bueno preguntar a la diseñadora si esto que dedujimos es efectivamente el comportamiento esperado por ella al diseñar así. Si nos dice que sí, entonces ya sabemos que nuestra columna 2 será de tamaño fijo, 628 dips y, en cambio, la cuarta columna es la que tendrá que acomodarse al ancho que le quede.



Es decir, le diremos que su ancho sea del 100%. ¿De qué? Del ancho que le quede de restar del ancho total de la tabla, los anchos fijos.

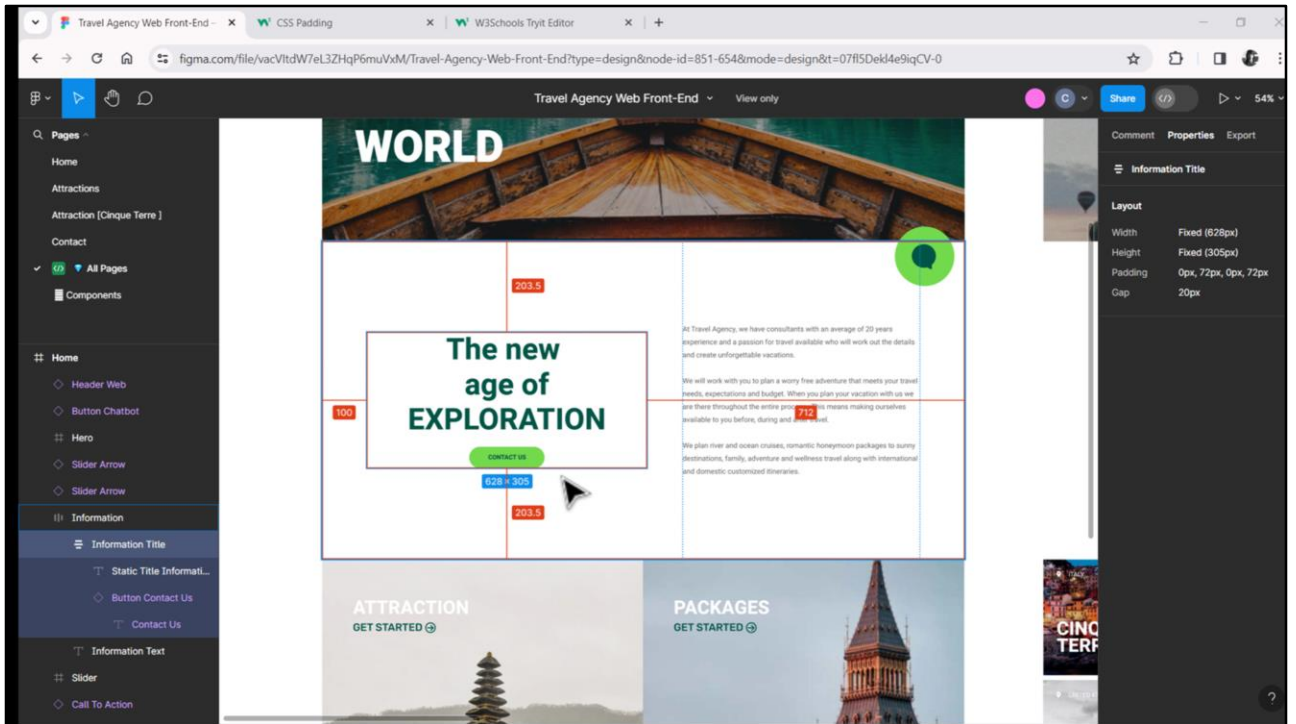


Por tanto agregamos columna a la izquierda de esta, y después, y una después de esta otra.



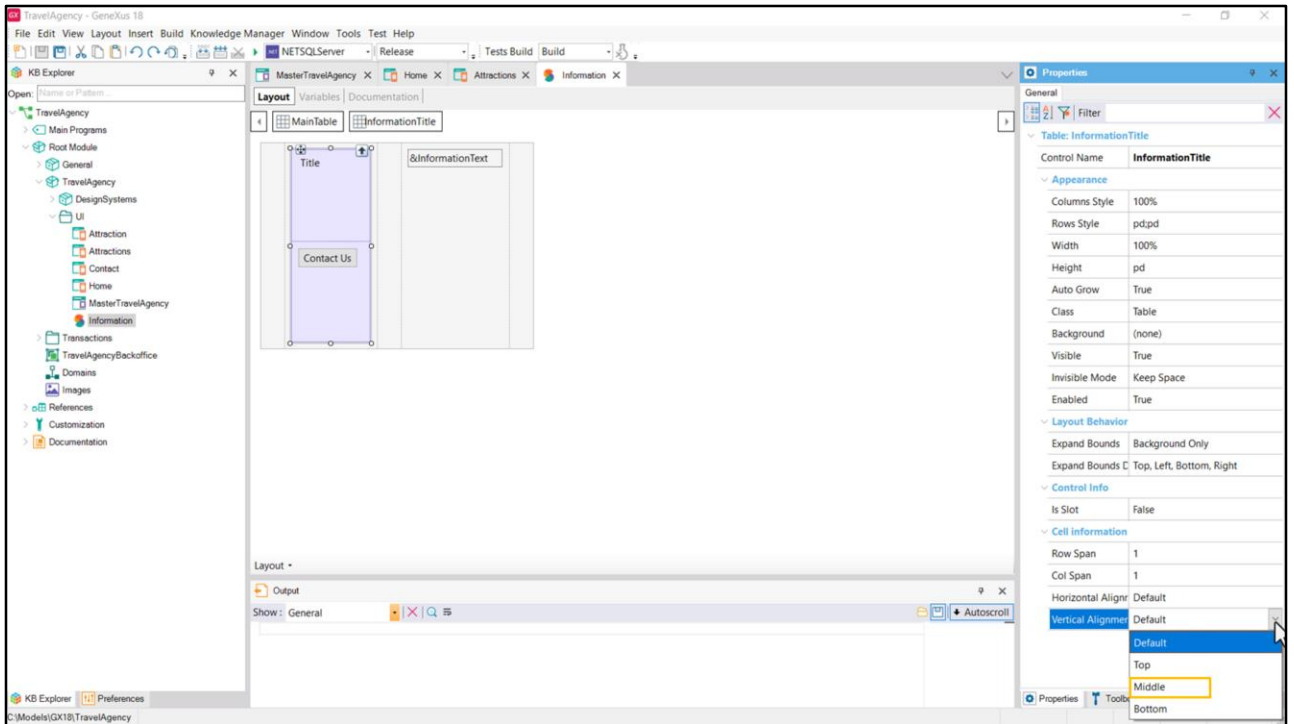
Y ahora establecemos sus anchos.

Podemos hacerlo desde aquí... 100, para la segunda: 628, para la tercera 80, dips, para la cuarta 100% y para la quinta 100 dips. Grabemos.

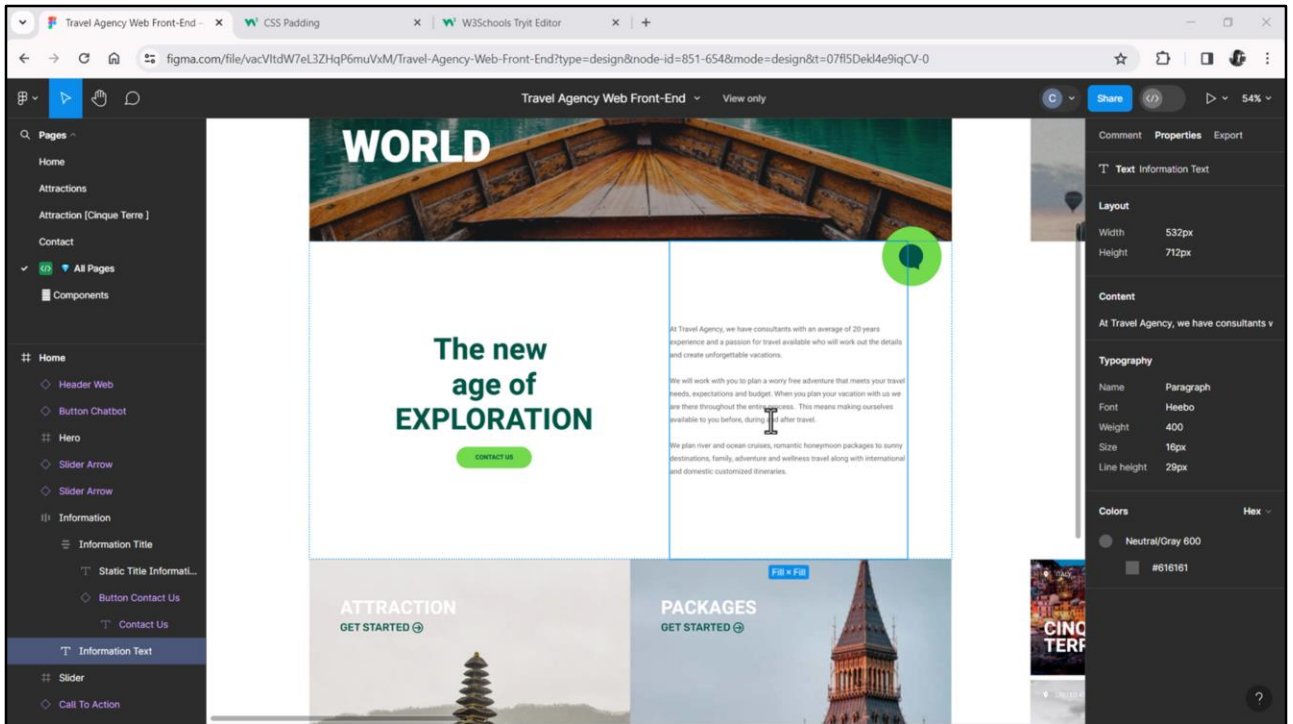


¿Ahora, qué pasa con estos espacios, de arriba y abajo? ¿Tenemos que hacer lo mismo y colocar esta vez filas vacías? No necesariamente.

Nuestra diseñadora dejó indicado que este contenedor tendrá un alto fijo, de 305 píxeles. Y claramente está centrado verticalmente. Podemos verlo con esos 203.5 píxeles de cada lado del borde. Entonces alcanzaría con hacer esto en GeneXus...

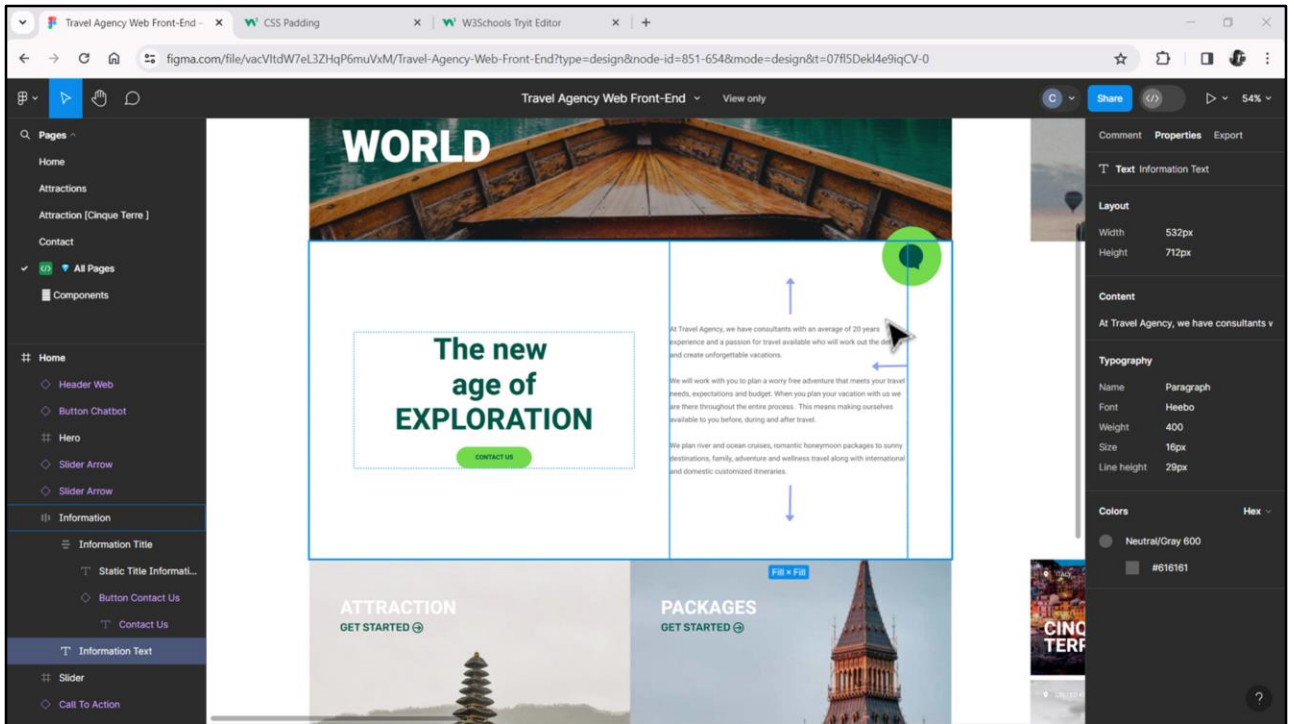


Editar la tabla, y colocar como alineación vertical Middle.

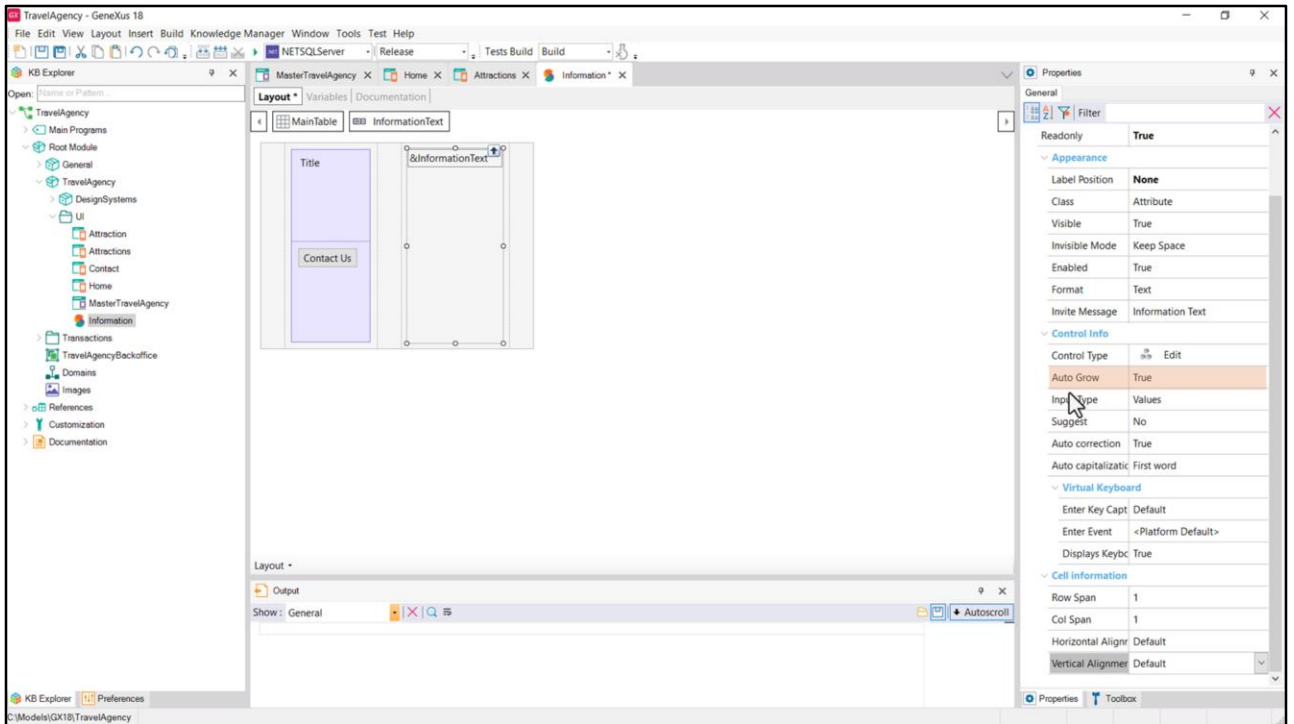


Por otro lado este otro elemento, el texto, que también está centrado verticalmente, no tiene un alto fijo, sino que dice Fill.

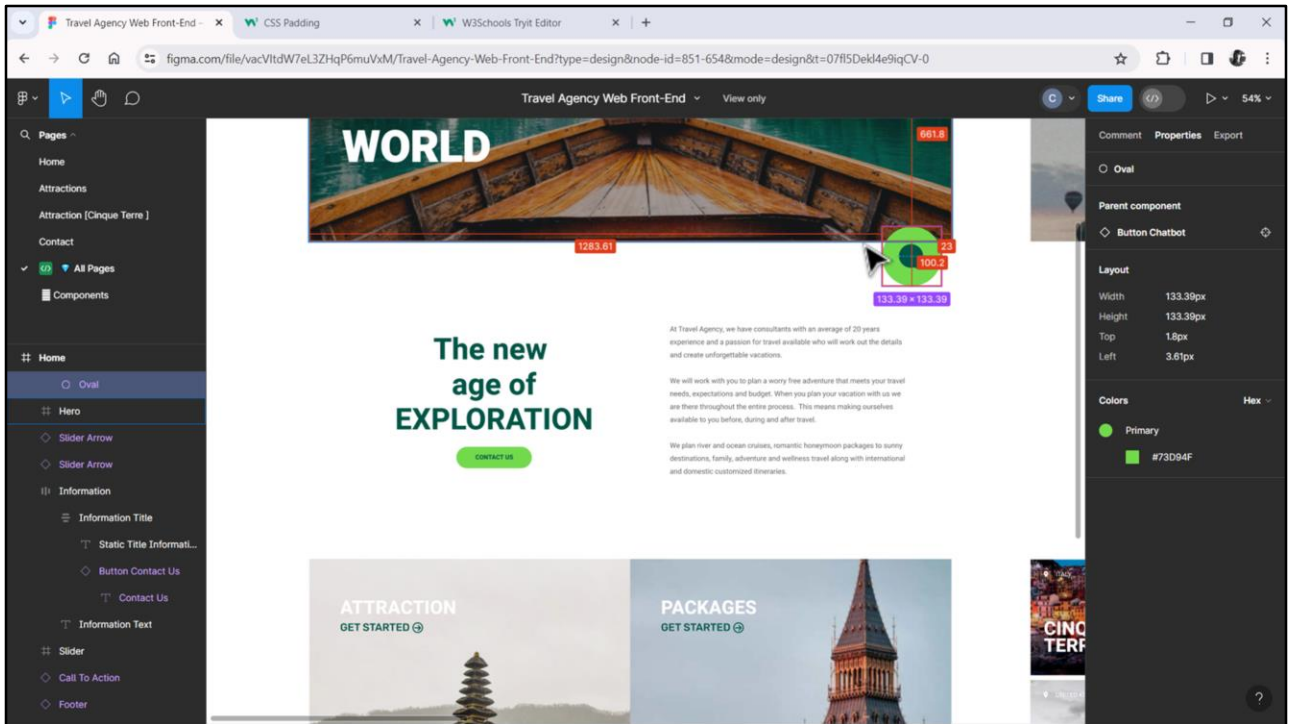
Va a depender de la cantidad de texto cuánto termine ocupando. Sabemos que si el texto es este, exactamente, va a verse así, pero si hubiera más texto, habrá menos espacio de arriba y de abajo. Eventualmente podría llegar a no haber ningún espacio.



De igual manera, si el borde del navegador se achicara, como el ancho de la columna se reduciría, el largo del texto se alargaría también, y podría suceder lo mismo. Incluso podría llegar a ser más largo que el alto de la tabla. En este último caso podría tomar la acción que es el default en el mundo web, y que es permitir que crezca elemento y contenedor para que entre todo el texto.

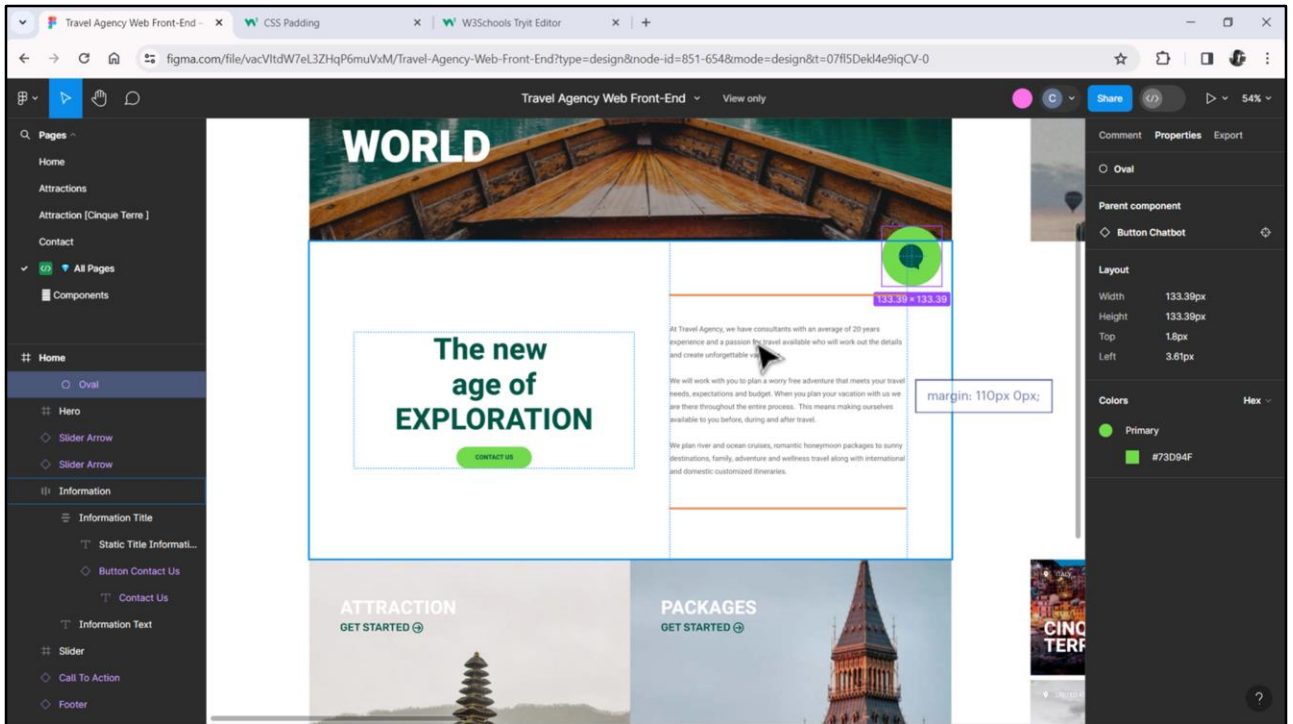


Es por eso que dejé la propiedad Auto Grow en True, tanto para la variable, como para la tabla que la contiene (es que si la tabla no tiene Auto Grow en True, no va a tomarse en cuenta el valor de la propiedad Auto Grow para los elementos contenidos en esa tabla). Ya que estamos acá vamos a alinear verticalmente en el medio la variable.

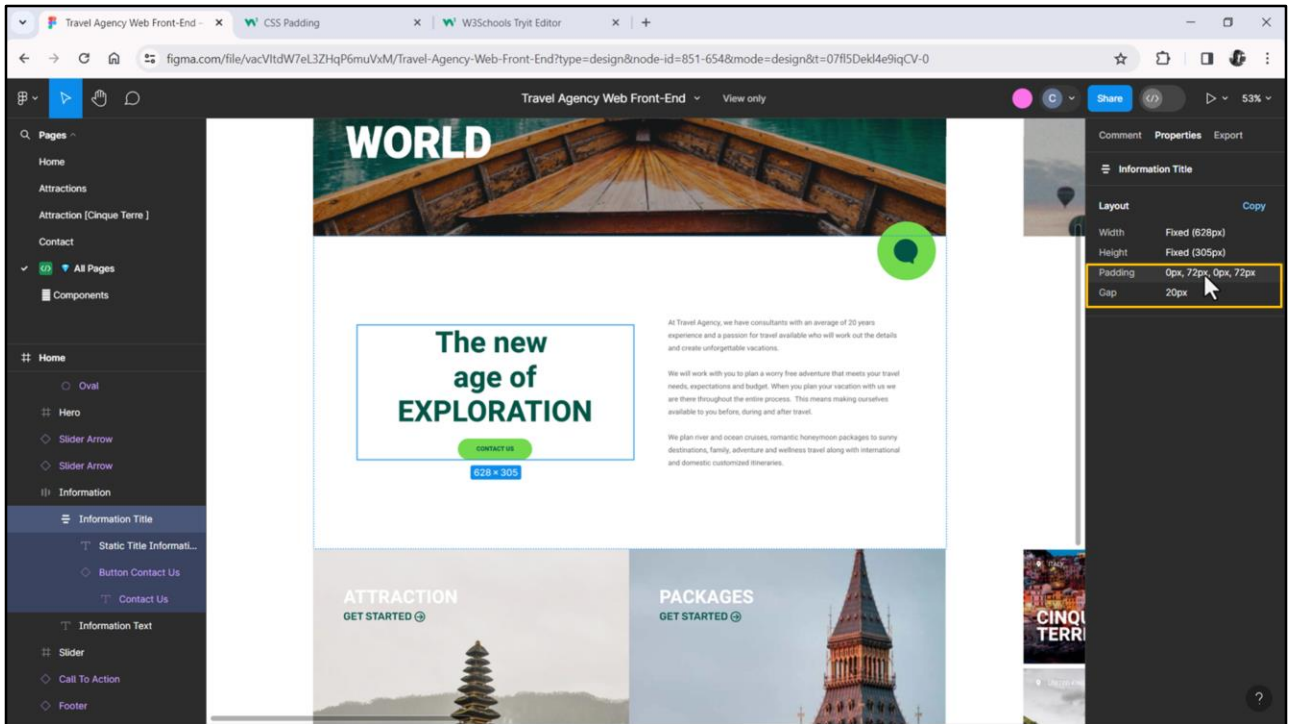


Ahora bien, si dejamos esto así, no tenemos ninguna seguridad de que en cualquier situación quede un espacio arriba y abajo.

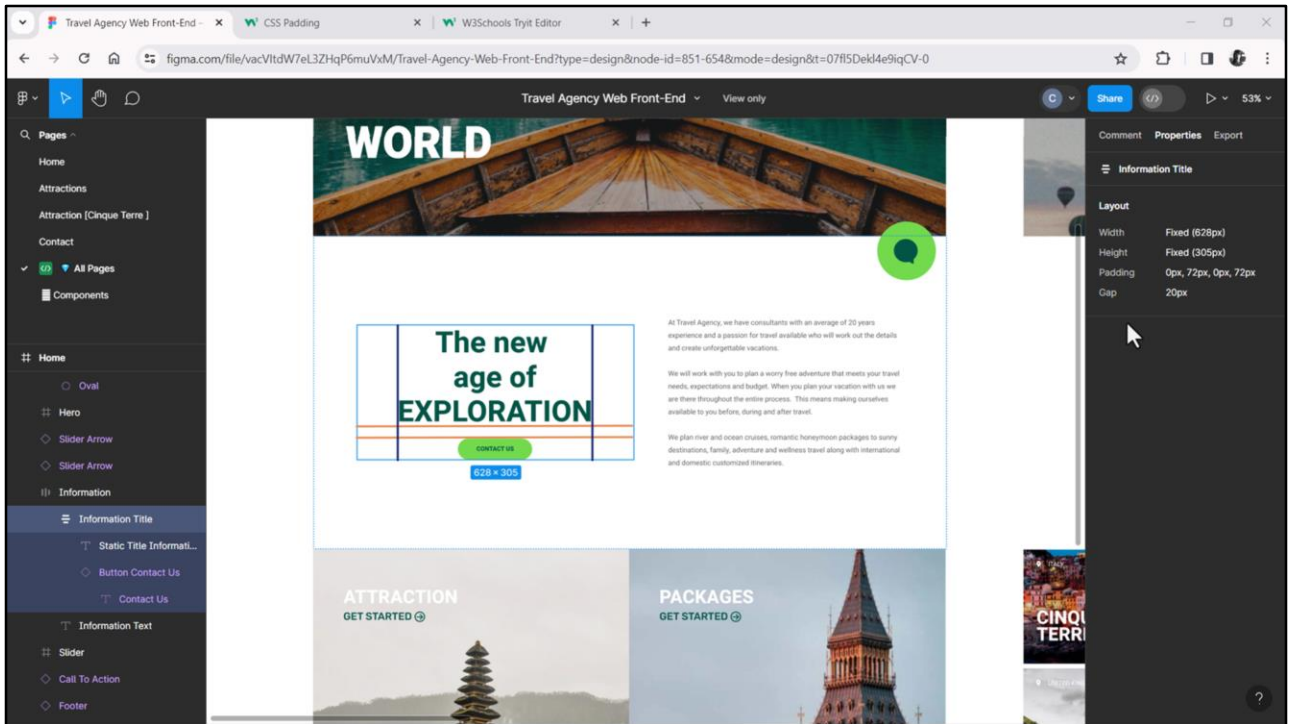
Lo que haría sería hablar con Chechu y preguntarle qué piensa al respecto, dado que eso no está modelado eso en su diseño. De hecho, si observamos el contenedor del que partimos tiene 0 de padding arriba y abajo, y podemos ver acá claramente que este elemento se está expandiendo hasta los bordes. Seguramente no vamos a querer que el texto pueda expandirse más allá del borde inferior de esta imagen del chatbot. Vemos su largo de 133.39 y si comparamos la distancia con el borde del Header vemos esos 100.2 píxeles, por lo que no sería descabellado plantear un espaciado de 110 de arriba y 110 de abajo para que sean simétricos.



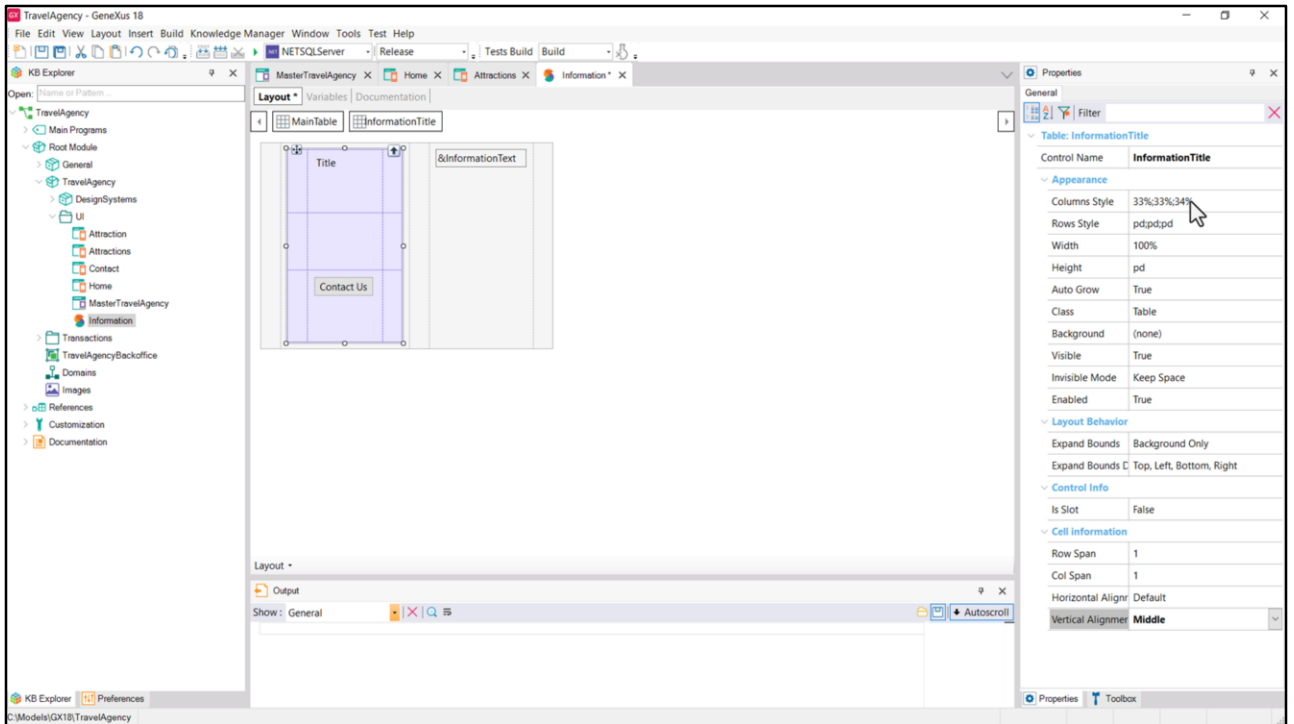
Esto, por supuesto, podemos hacerlo de la misma manera que lo hicimos para las columnas, es decir, agregando dos filas vacías, una arriba y otra abajo. Pero tenemos otras alternativa, entre varias, que va a ser a través de darle un margen superior e inferior a este elemento, y lo vamos a dejar para verlo después, cuando ya pasemos por el Design System Object.



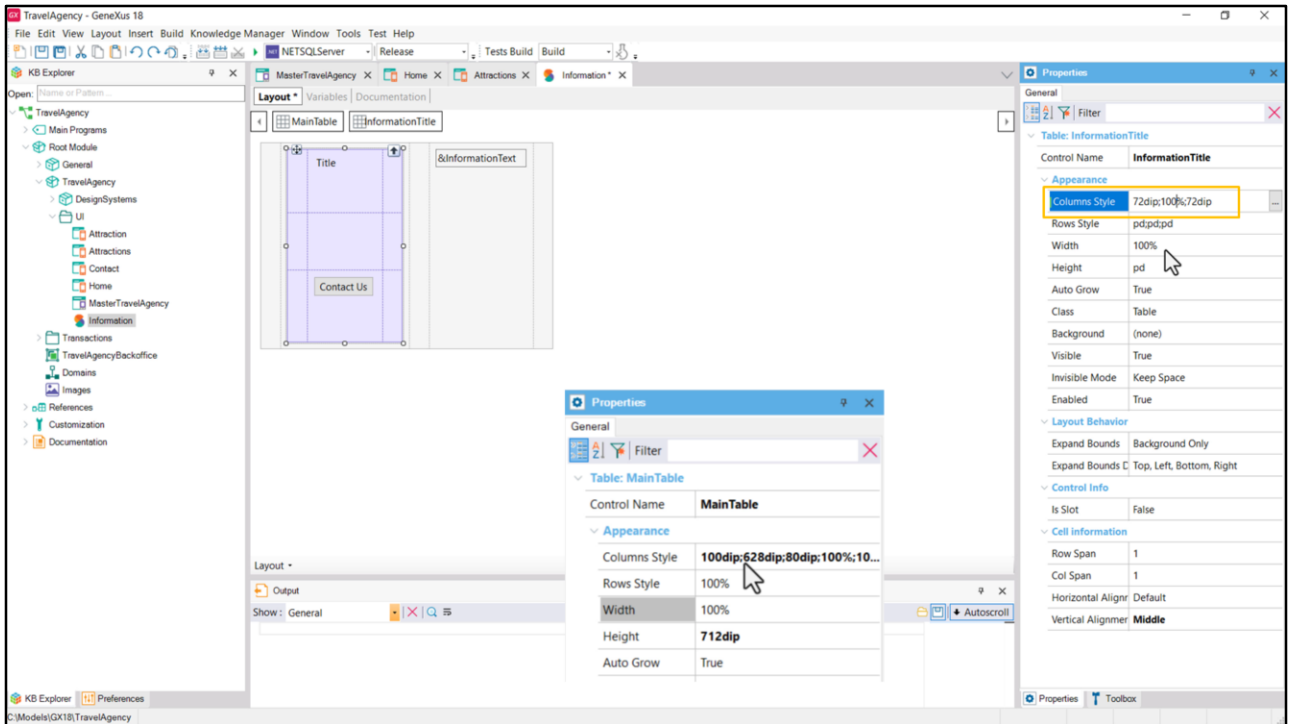
Ahora sí, analicemos el contenedor de la izquierda. Este se parece mucho al anterior, solo que acá, en lugar de estar los elementos ordenados en esta línea horizontal están verticalmente: arriba el texto y abajo el botón. Y aquí vemos el padding de esta manera: 72 píxeles de la derecha, y 72 píxeles de la izquierda. Y 0 de arriba y de abajo. ¿Y cuál es el espacio que separa un elemento del otro? Estos 20 píxeles indicados aquí.



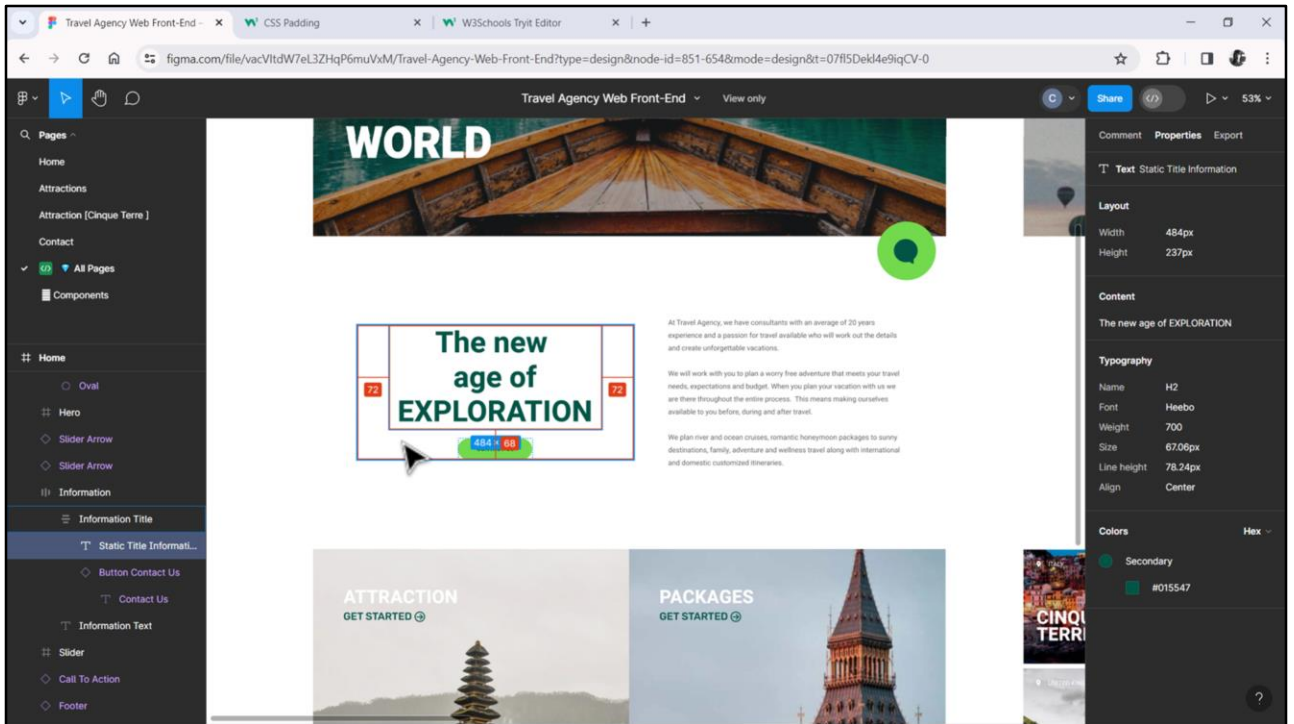
Entonces con esta perspectiva podría pensar en agregar una fila vacía entre medio, y dos columnas vacías, en los costados.



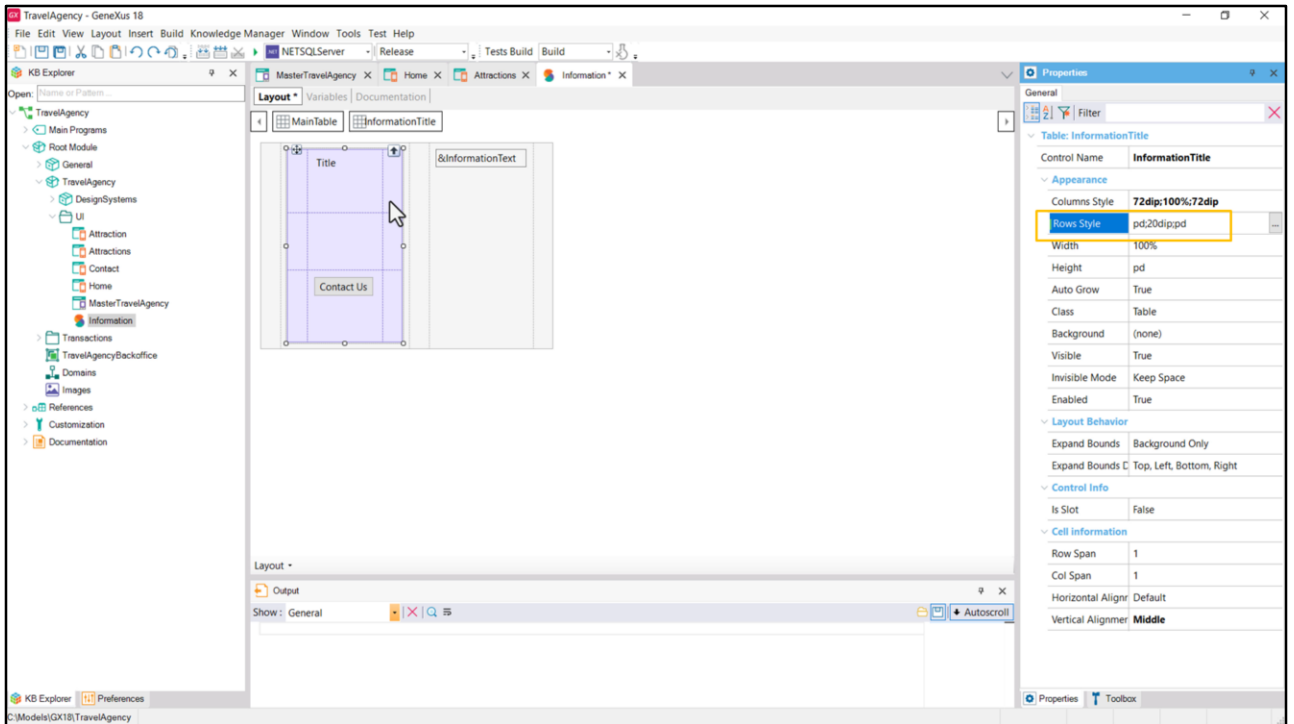
Al hacer esto, vemos que las propiedades se actualizaron, y ahora las columnas adoptaron estos valores en porcentajes, para equidistribuirse, que es como el default.



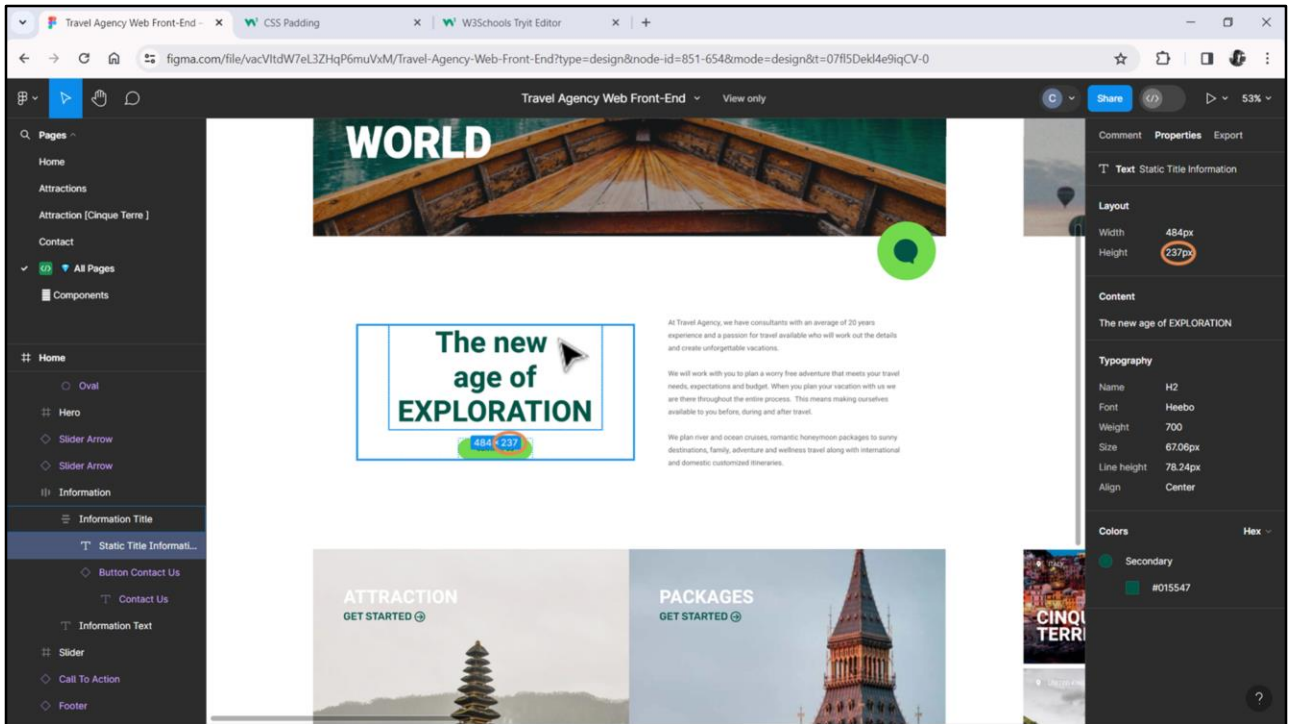
Entonces lo que vamos a colocar acá son 72 dips en la de la izquierda y 72 dips en la de la derecha. ¿Y qué pasa con la del medio? Le voy a colocar 100%... del Width de la tabla que va a corresponder entonces al ancho de la celda, de la columna en este caso, en la que se encuentra esa tabla. Es la segunda columna, por tanto va a corresponder a estos 628 dips. Por tanto, el ancho que va a asumir la segunda columna va a ser la que resulte de restar de esos 628 dips los valores de las dos columnas de izquierda y derecha.



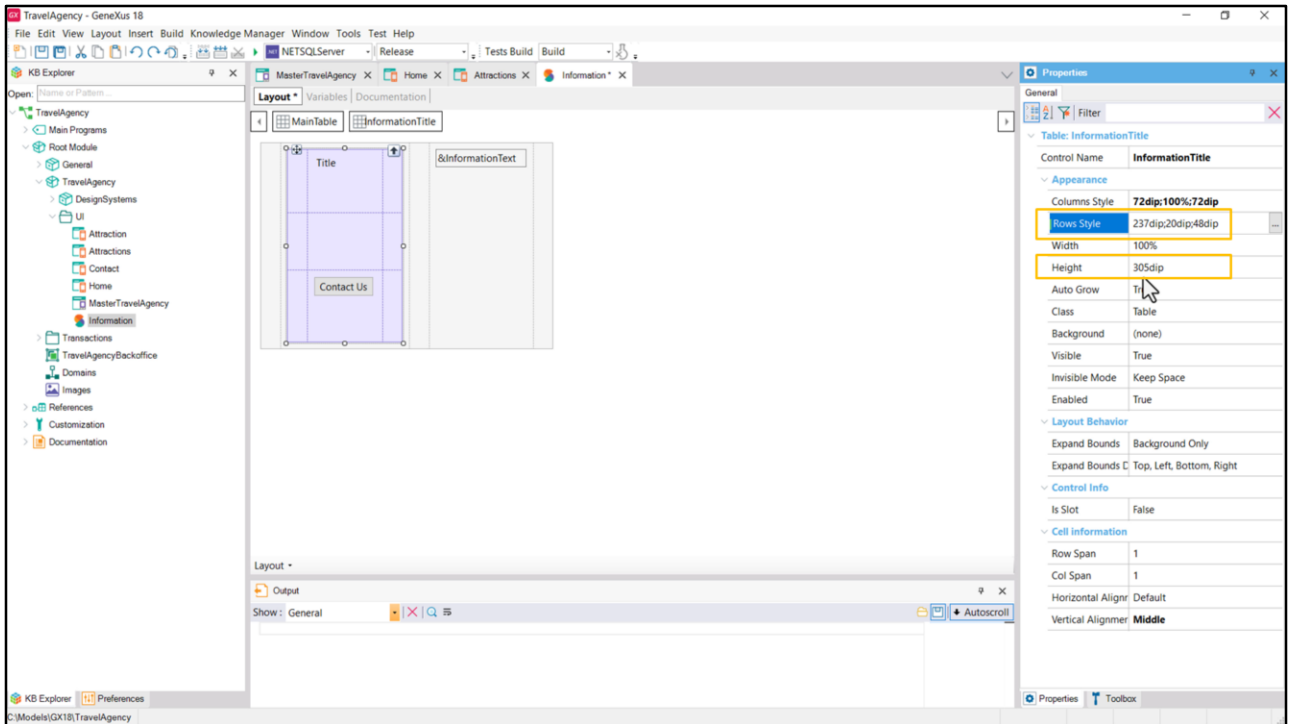
Y esto lo vemos claramente acá en Figma... 628... 72, 72, y lo que va a quedar entonces del ancho de esa columna va a ser 484.



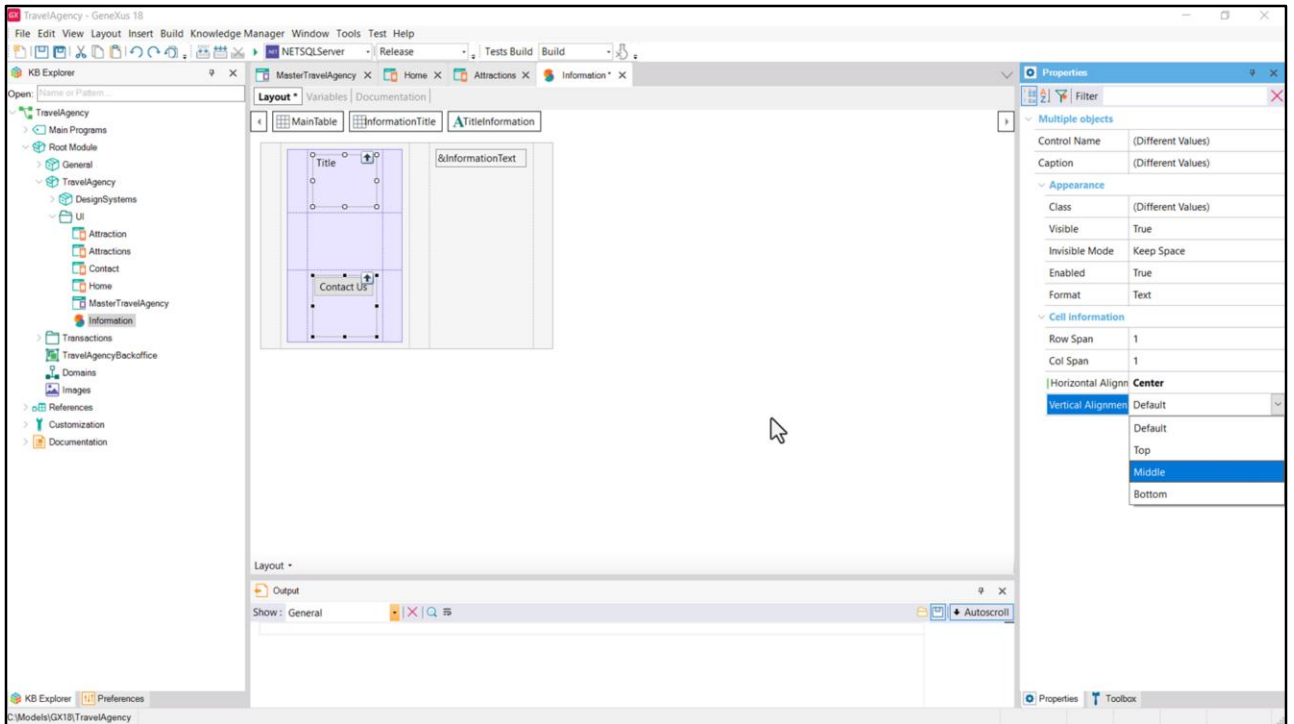
Bueno, ¿y qué pasa con el alto de las filas? Sabemos que la del medio es una fila de espaciado, que va a ser de 20 dips. Ahora tenemos que ver cuánto sería el alto de la fila... de la primera... y cuánto sería el alto de la fila que va a contener al botón.



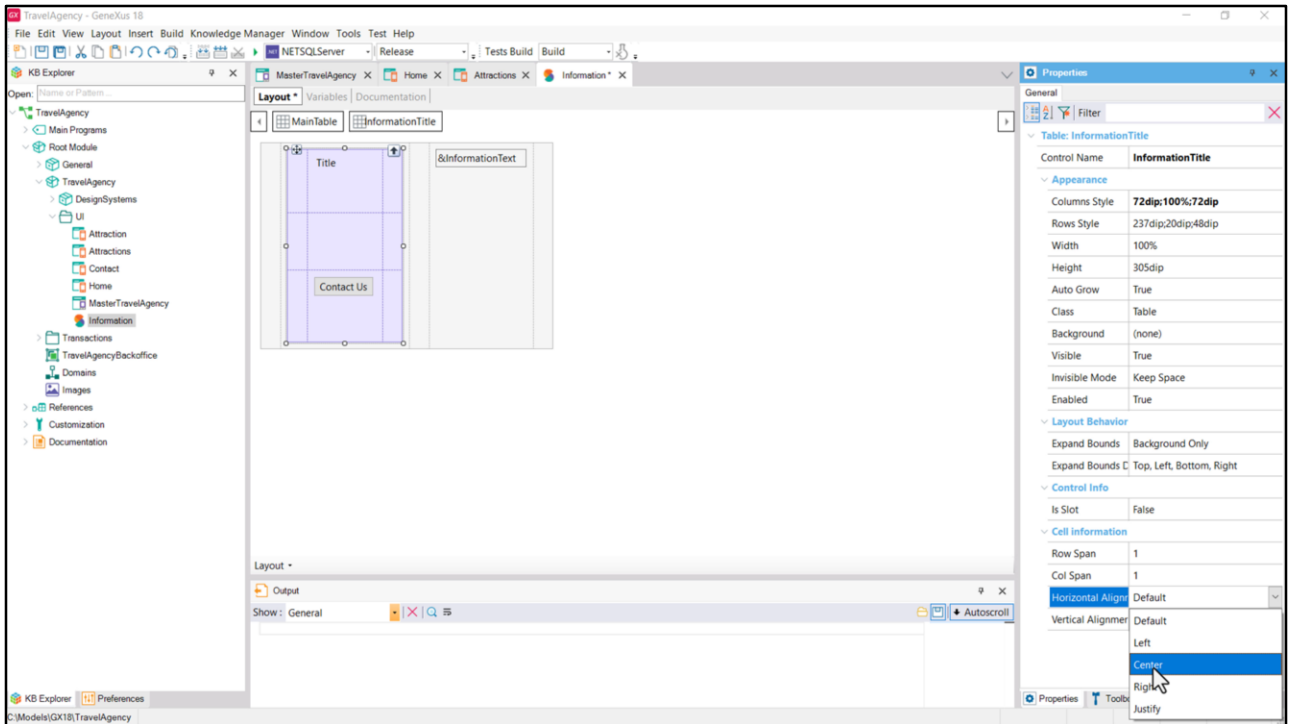
Bueno, la del texto, acá vemos, va a ser de 234 píxeles, y la del botón de 48.



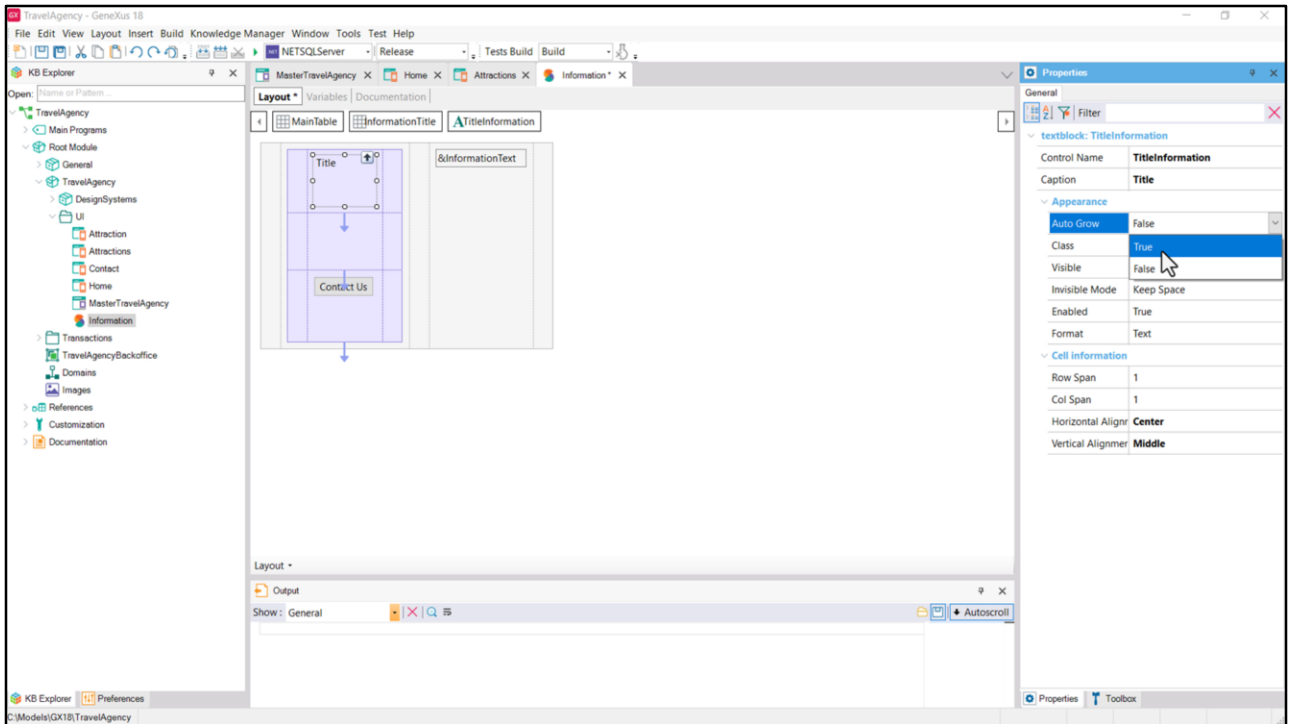
Entonces... 237... y observemos cómo al haber hecho esto, como todas las filas son de ancho fijo, ya automáticamente modificó la propiedad Height sumando esos valores.



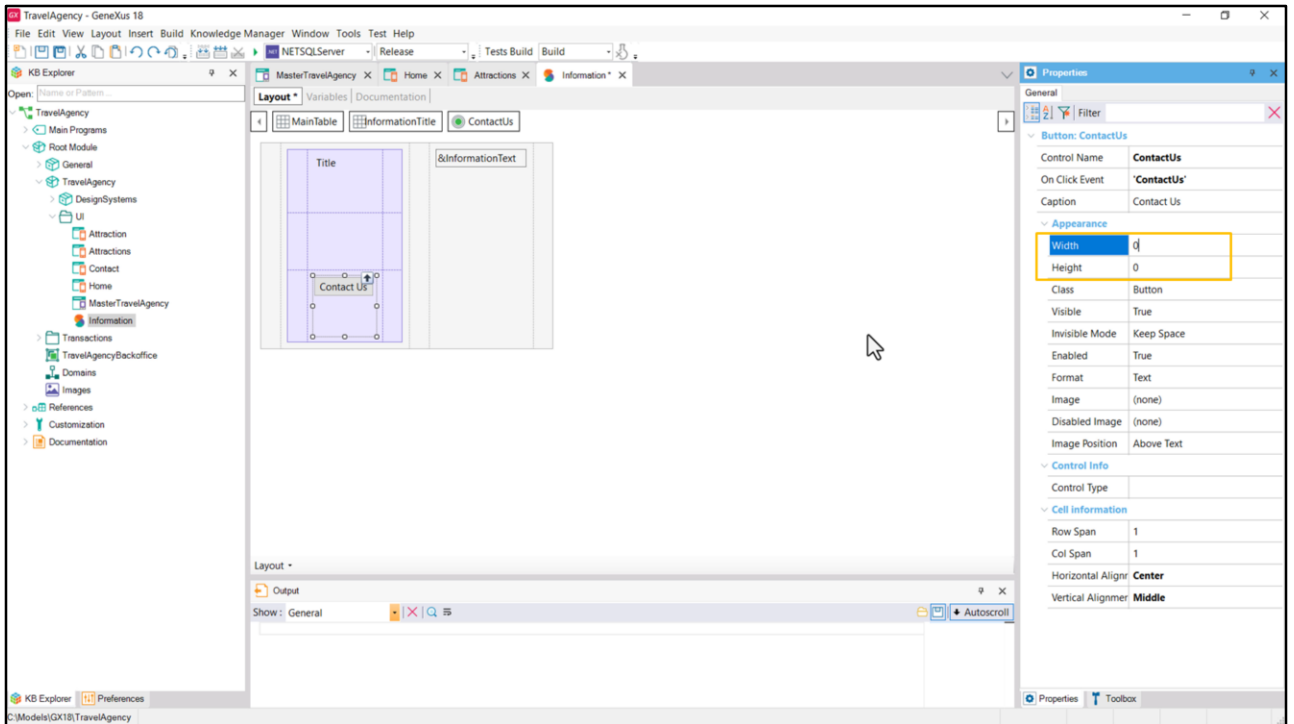
Ahora antes de seguir voy a centrar estos dos elementos... los voy a centrar horizontalmente y también los voy a centrar verticalmente.



Y lo mismo voy a hacer para la tabla, aunque no es necesario en este caso... la voy a centrar horizontalmente. ¿Por qué no es necesario? Porque va a ocupar el tamaño completo de la celda y estoy centrando todo lo de adentro.

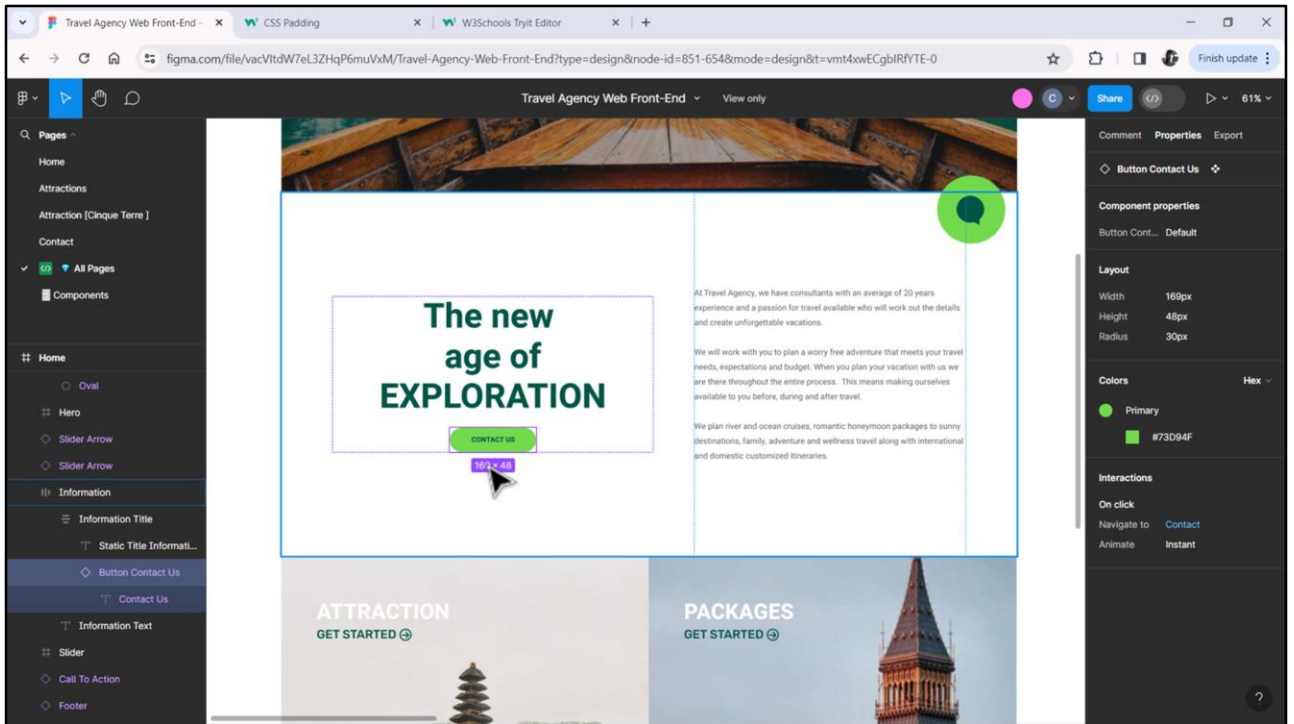


Bien, con lo que hicimos, sabemos que dependiendo del contenido del Caption del TextBlock hasta dónde llegará. Si la cantidad de caracteres hace que su ancho sea menor que el de la columna, le quedará aire equidistante de los costados, dado que lo centramos (y también de arriba y abajo). Si tiene más caracteres, entonces se hará wrap para no desbordar y el texto tendrá más de una línea. ¿Y qué pasaría si no le alcanzara el alto de la celda? Como vemos, en este caso el TextBlock tiene como default False para Auto Grow, debido a que es el default para las plataformas nativas, así que se truncará. Si lo que queremos es que si desborda el alto entonces la fila se empuje hacia abajo y por tanto también las otras dos, entonces lo cambiamos a True, que suele ser el comportamiento esperado en Web.

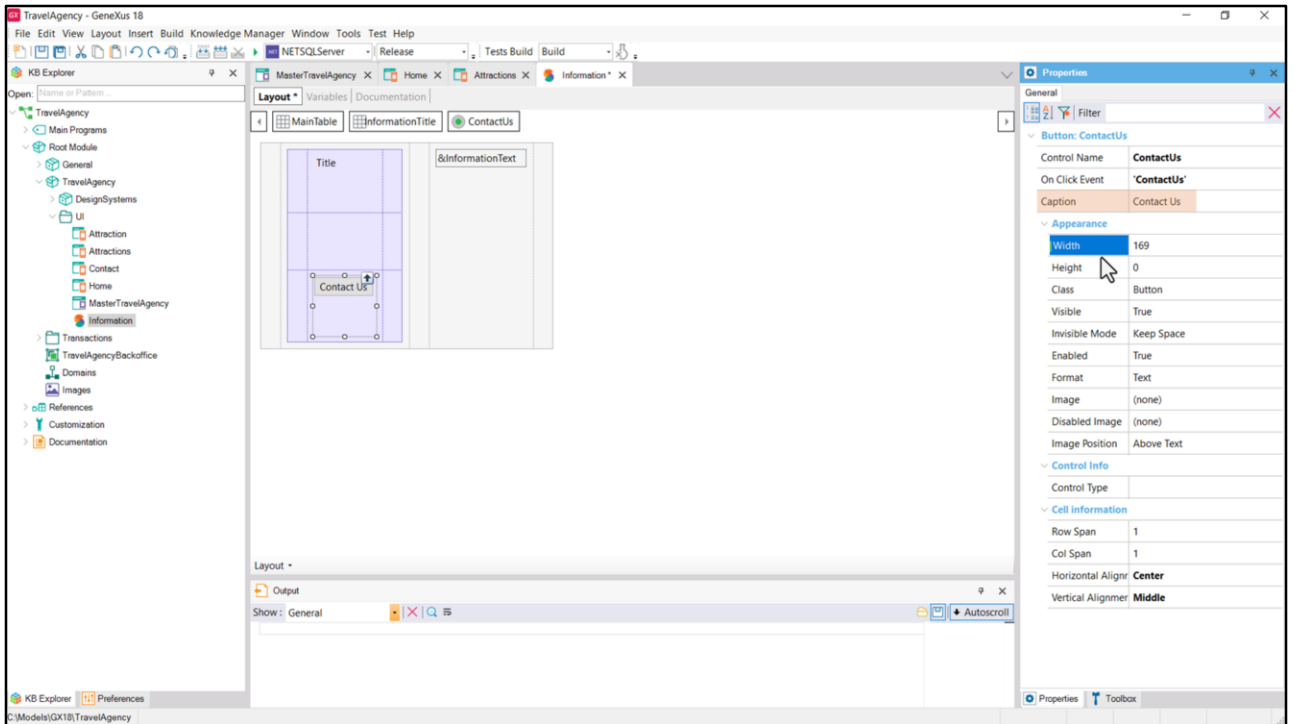


¿Y qué pasa con el botón? Por default se expandirá hasta ocupar todo el espacio del contenedor (aunque pueda resultar un poco contraintuitivo que diga estos ceros en lugar de un más esperable 100%)

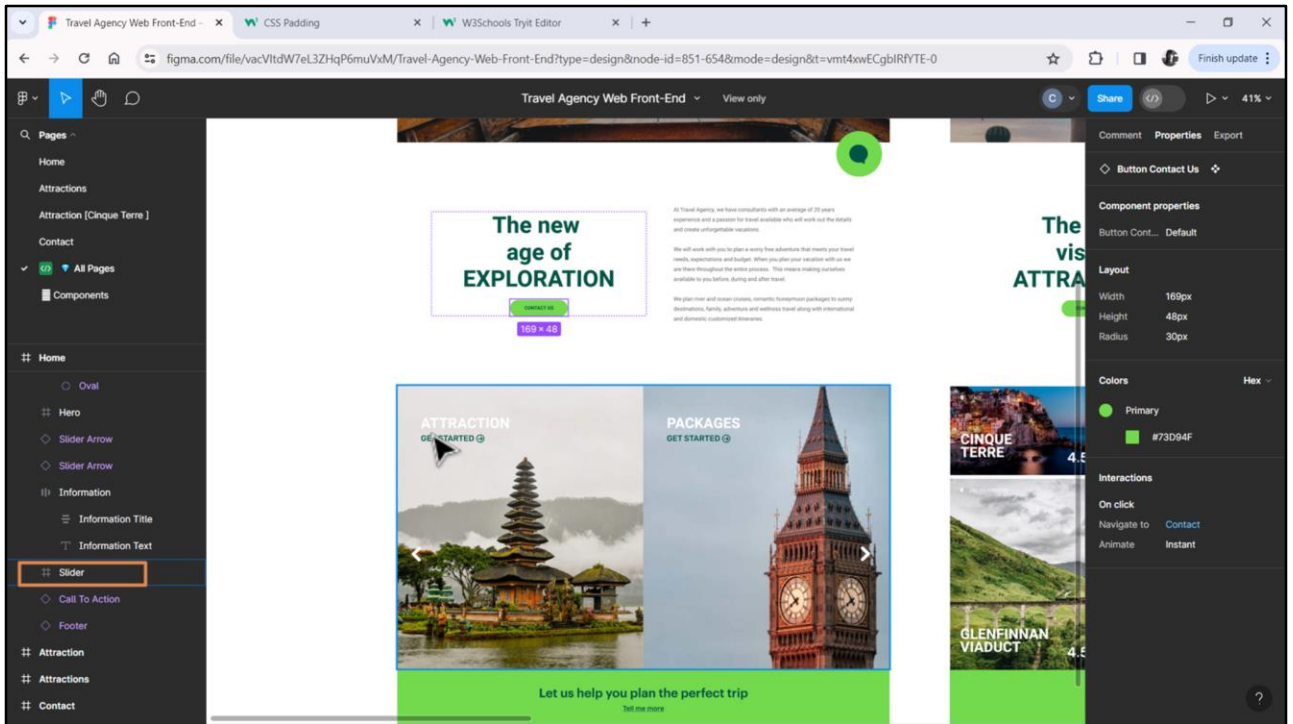
Para el alto, que ocupe el mismo espacio que el alto de esta celda está perfecto, porque indicamos que era de 48 dips justamente porque era el alto del botón. Pero para el ancho no es correcto.



Si vamos a ver a Figma, el ancho es menor, es de 169 píxeles.

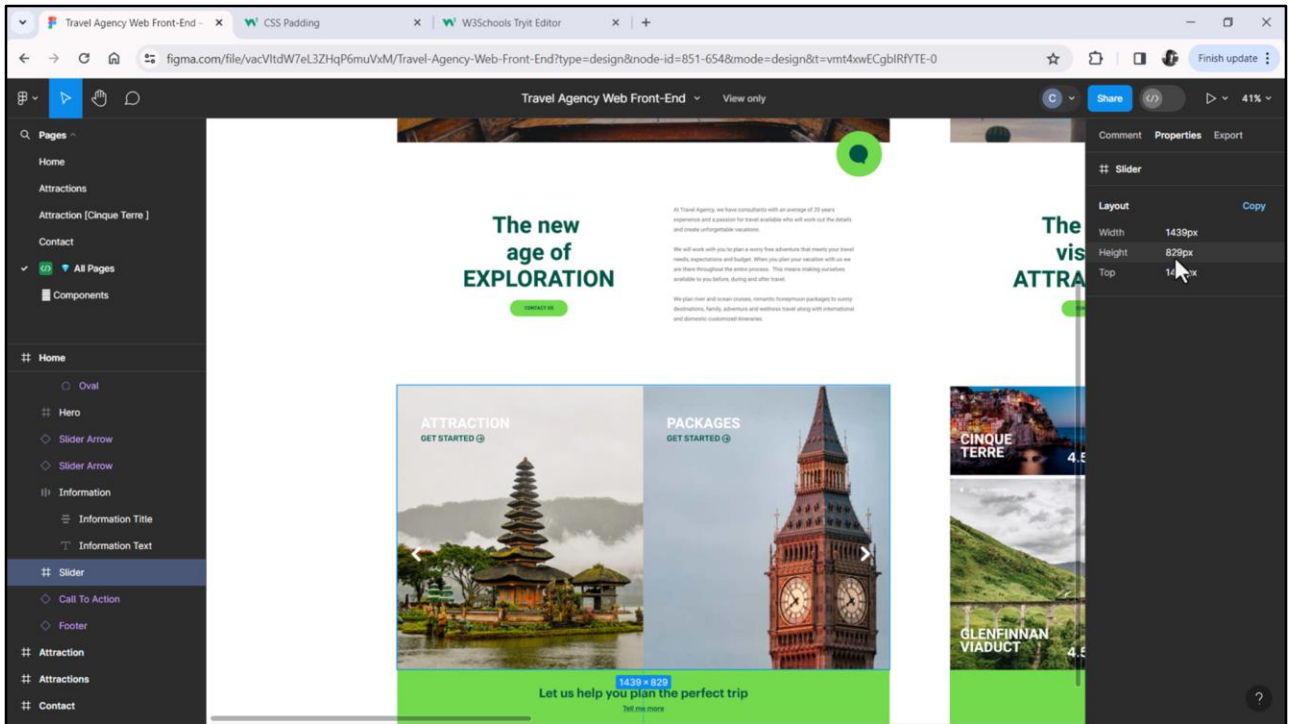


Luego veremos que ese sería el ancho mínimo, pero que si el "caption" del botón desbordara ese tamaño, el botón debería expandirse hacia los costados, sin superar el ancho del contenedor, claro. Pero eso lo veremos después, por ahora lo dejamos así.

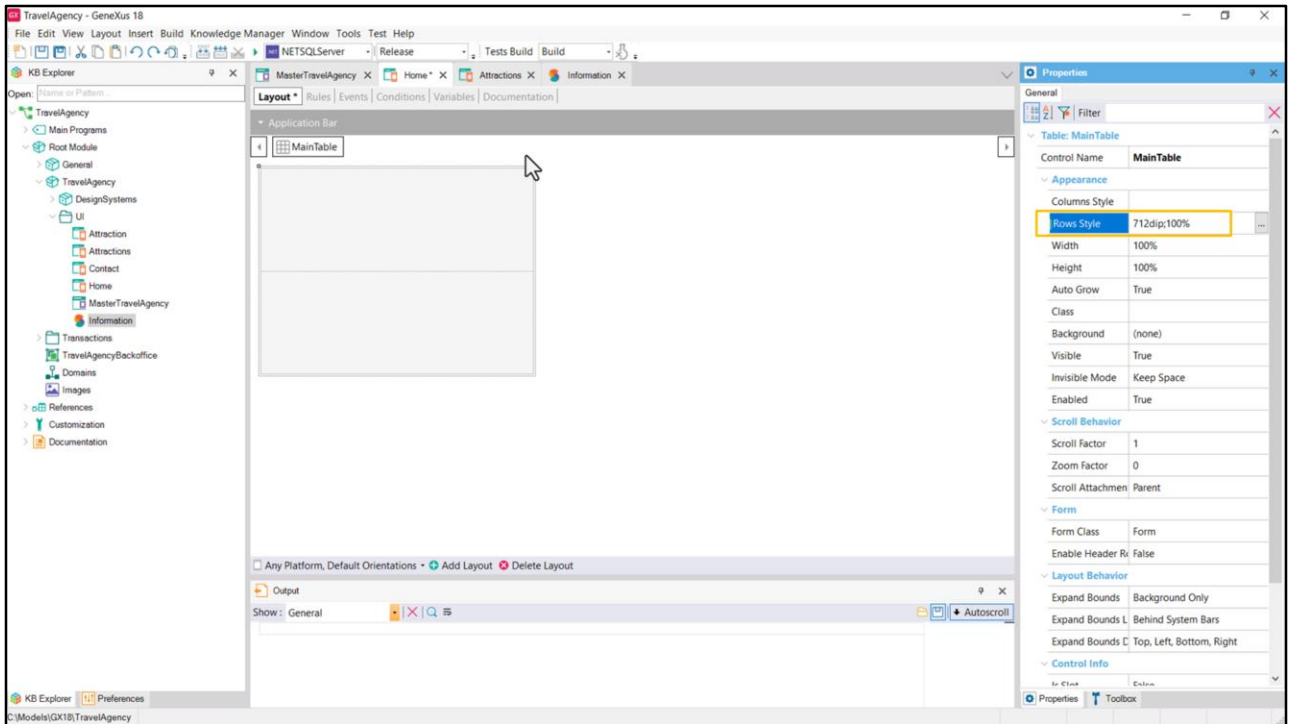


Antes de pasar a definir el estilo, quiero ver en ejecución cómo nos va quedando esto. Para ello tendré que instanciar el Stencil en el panel Home, primero, y luego en el de Attractions, que por ahora los tenemos vacíos.

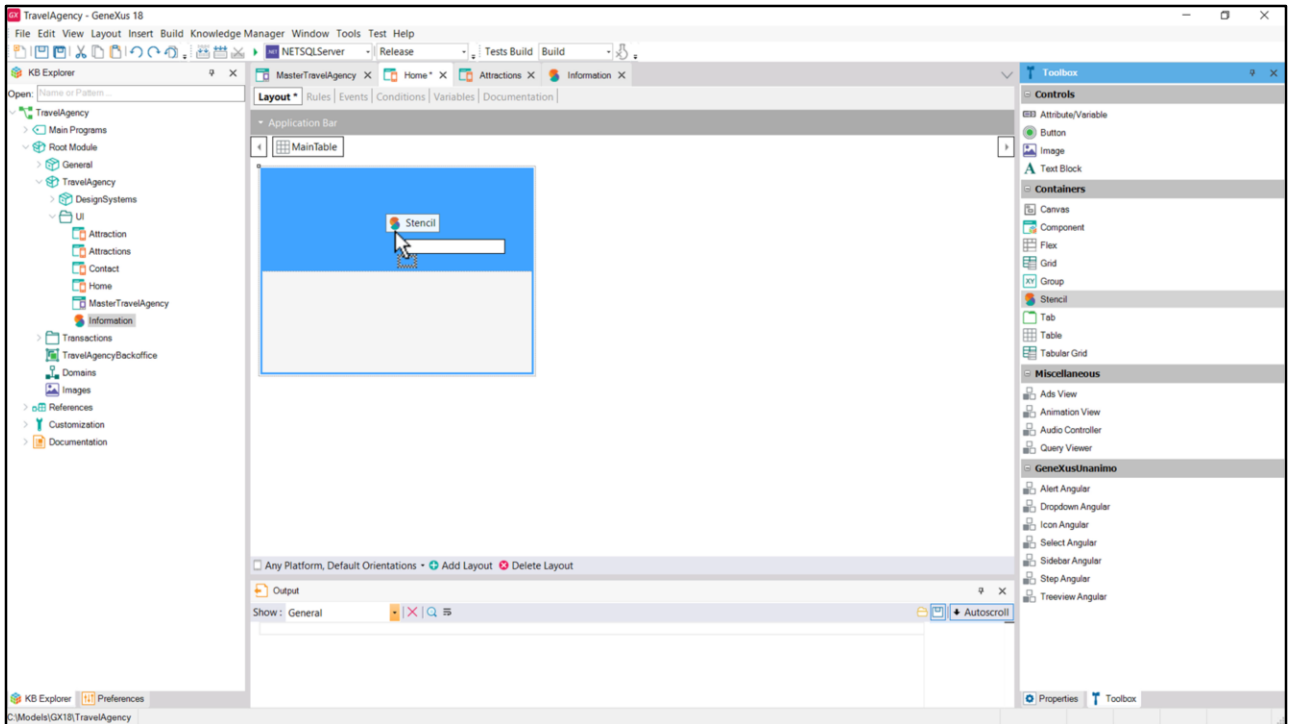
El Home tendrá además de este elemento, este otro, que aquí Chechu llamó Slider, y es un frame con subelementos.



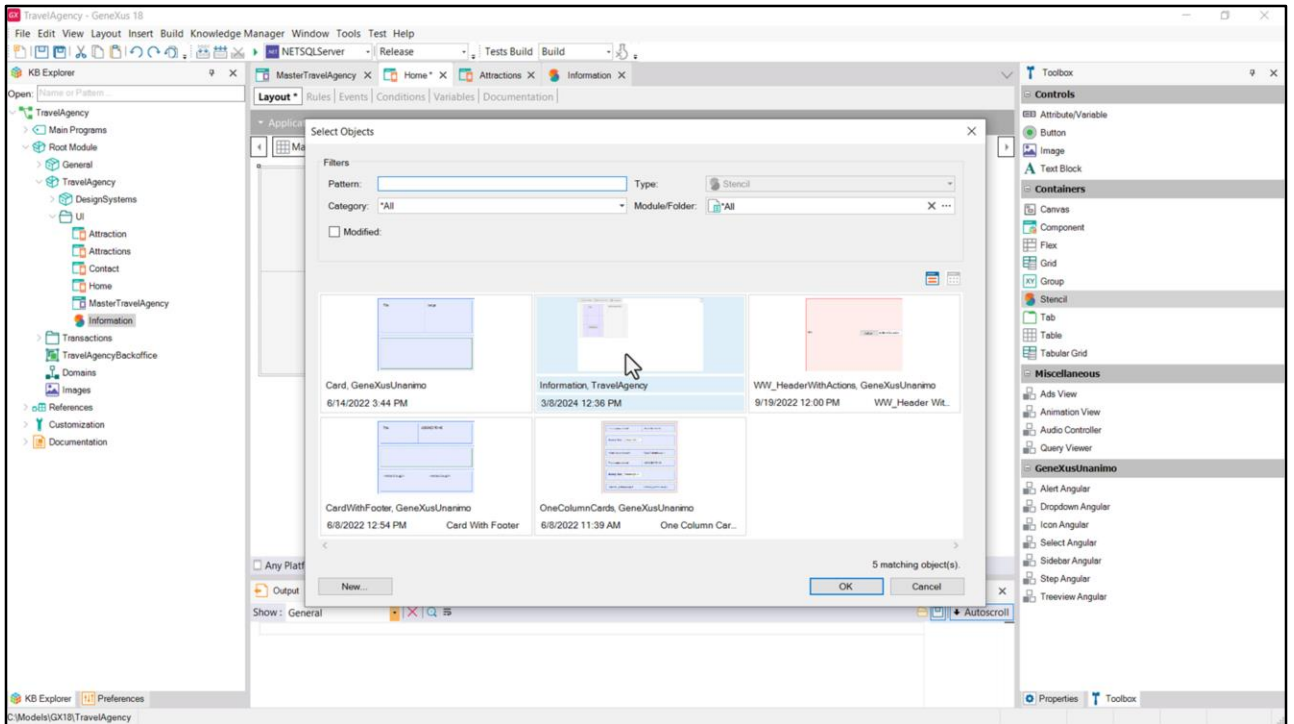
Su alto será de 829 píxeles. Y el de Information era de 712.



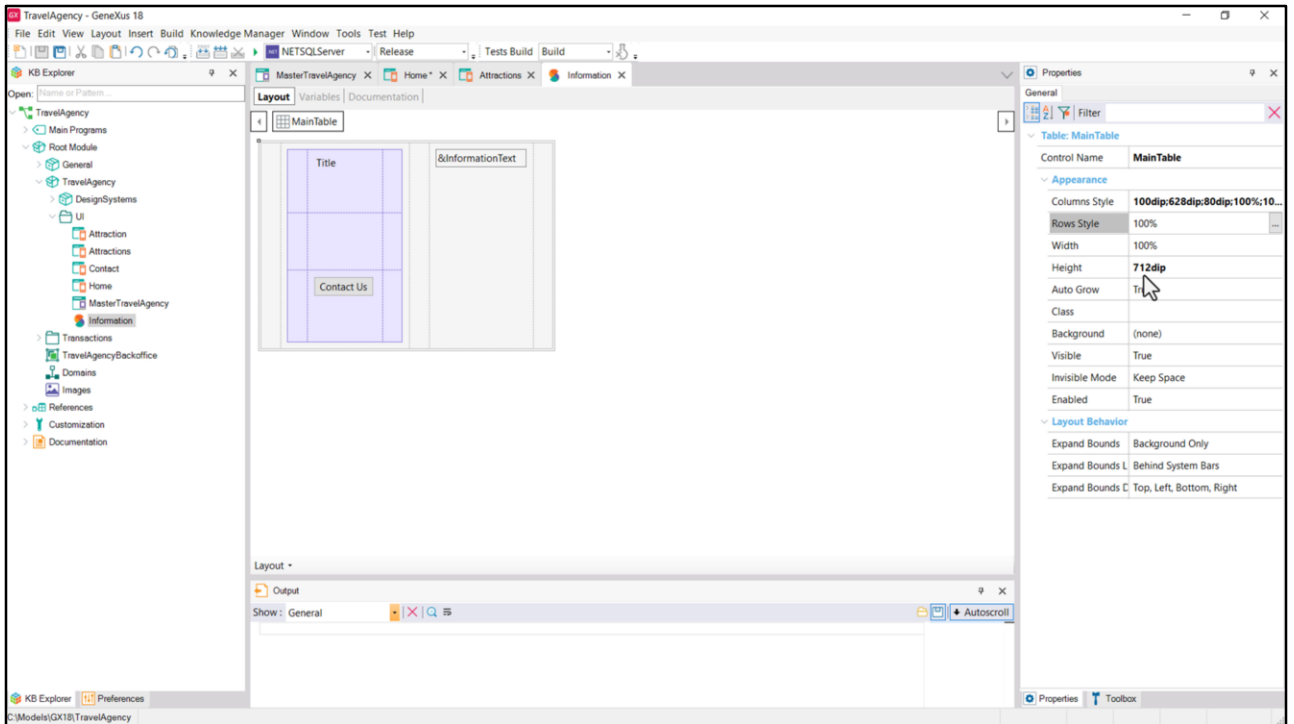
Entonces a la tabla main le agrego una fila... la primera de las dos filas que por ahora tiene va a ser de 712 dips y la segunda que podría ser de 829, la voy a dejar de 100%. Después analizaremos esto. El 100% restante de, evidentemente, el alto de la tabla Main.



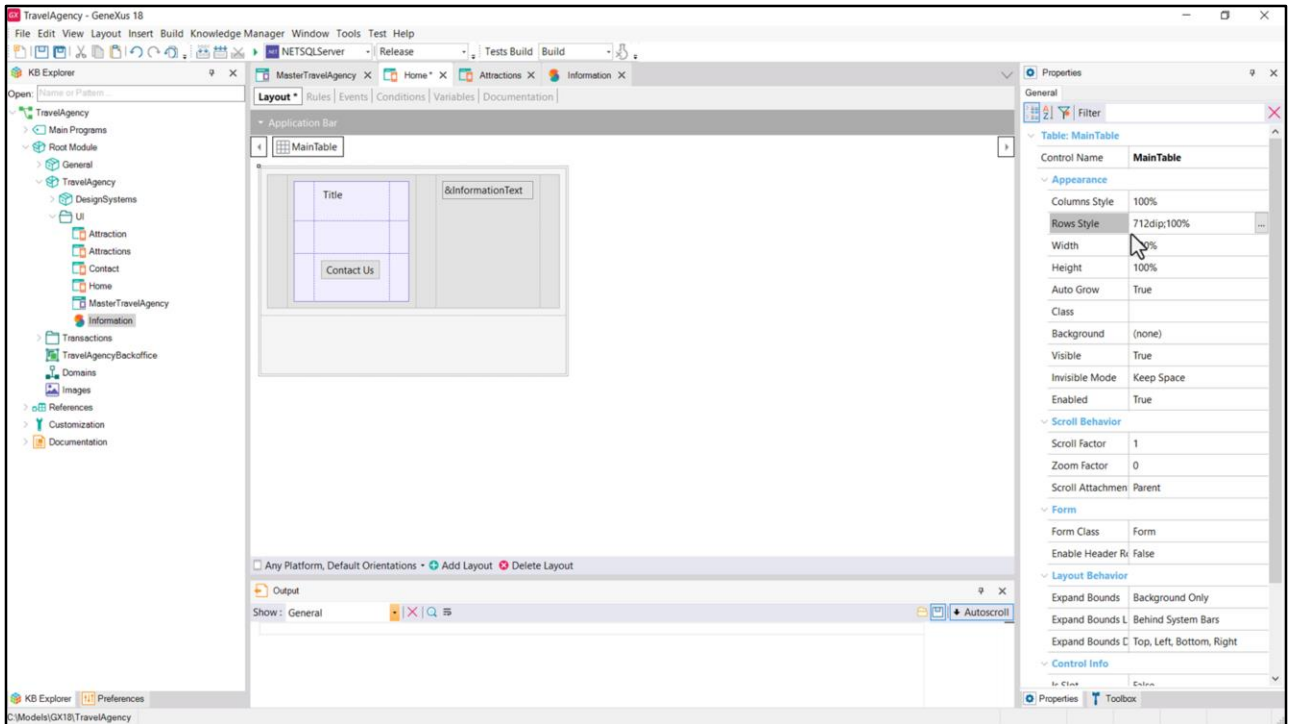
Y a continuación lo que voy a hacer es arrastrar un control de tipo Stencil dentro de la primera de esas filas...



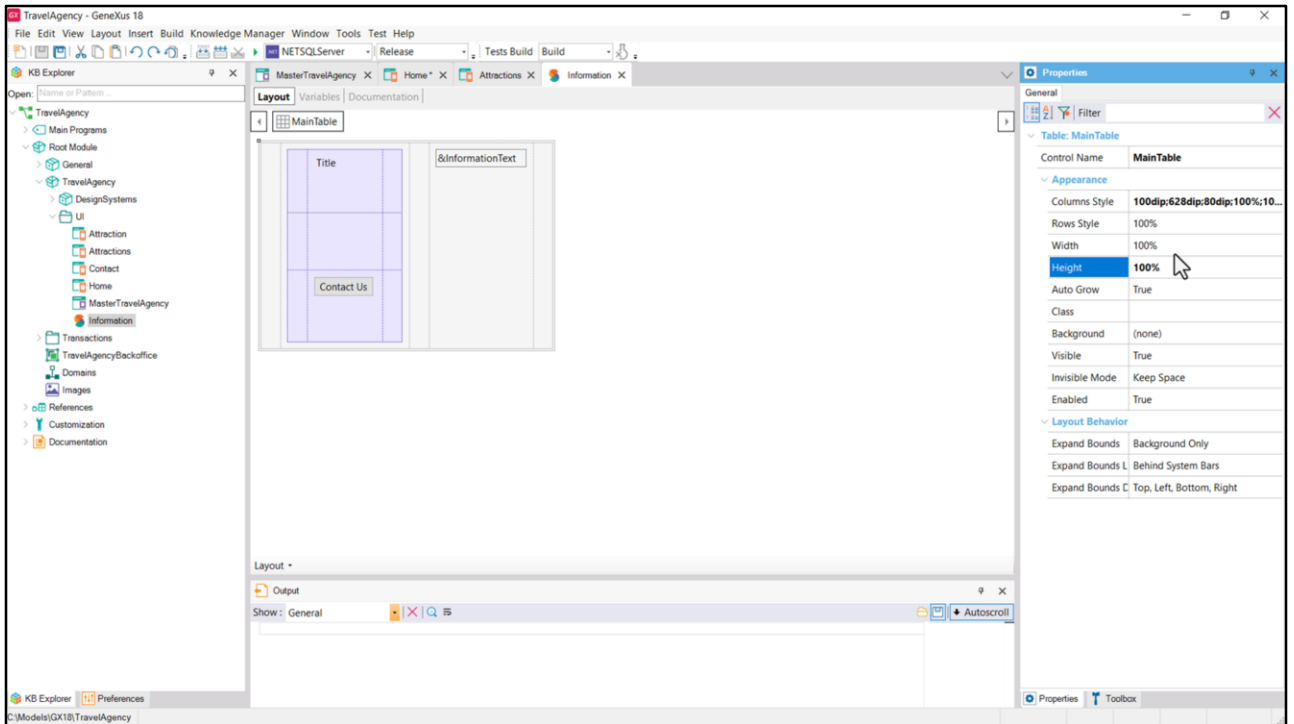
... y elegir el nuestro, Information.



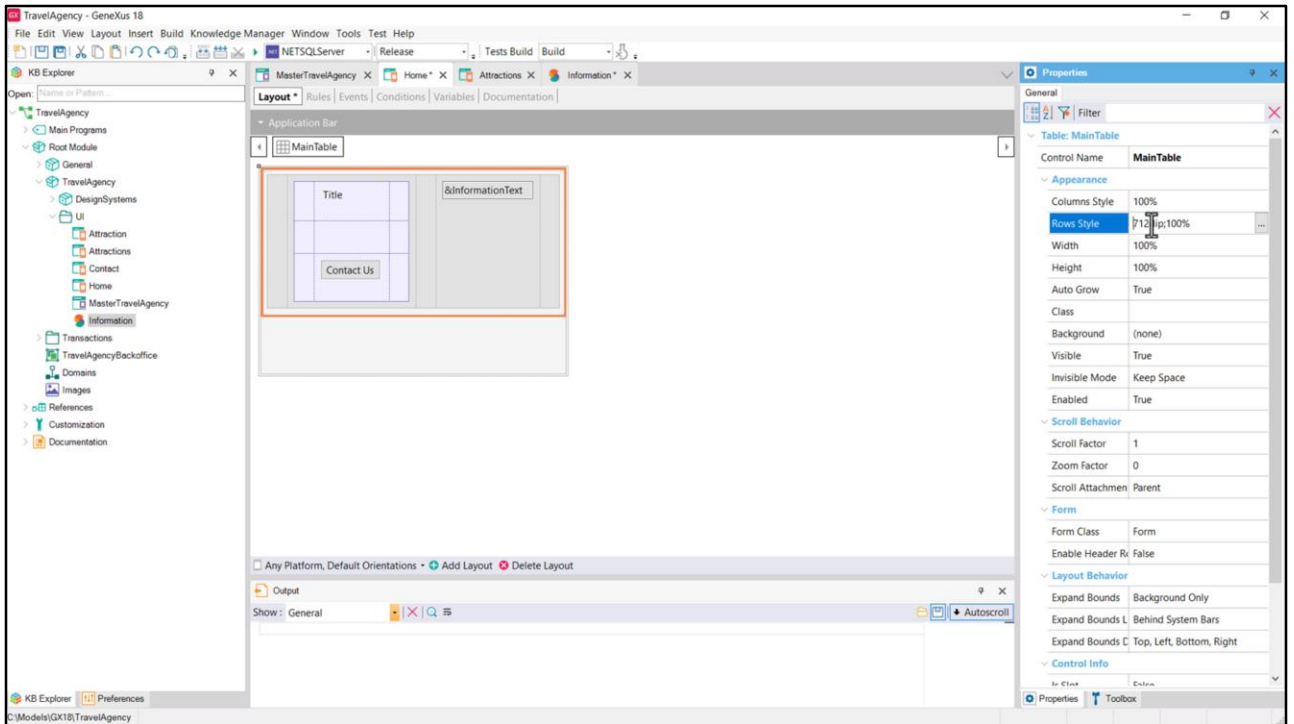
Si ahora Chechu nos advirtiera que cambió el alto de 712 píxeles o dips por 800, por ejemplo, ¿qué tendríamos que hacer? Tendríamos que venir a cambiarlo aquí...



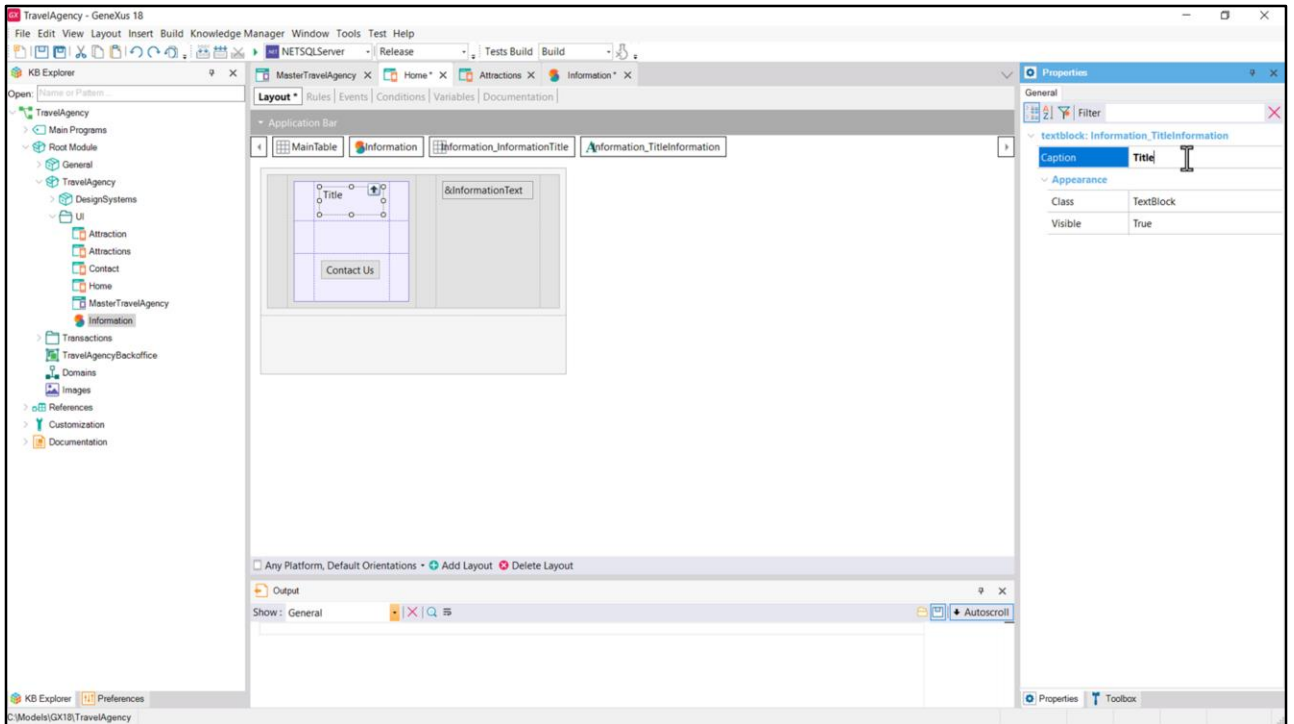
Y luego venir también aquí, a cambiarlo a nivel del alto de la fila. Y si después colocamos ese mismo Stencil en el panel de atracciones lo mismo.



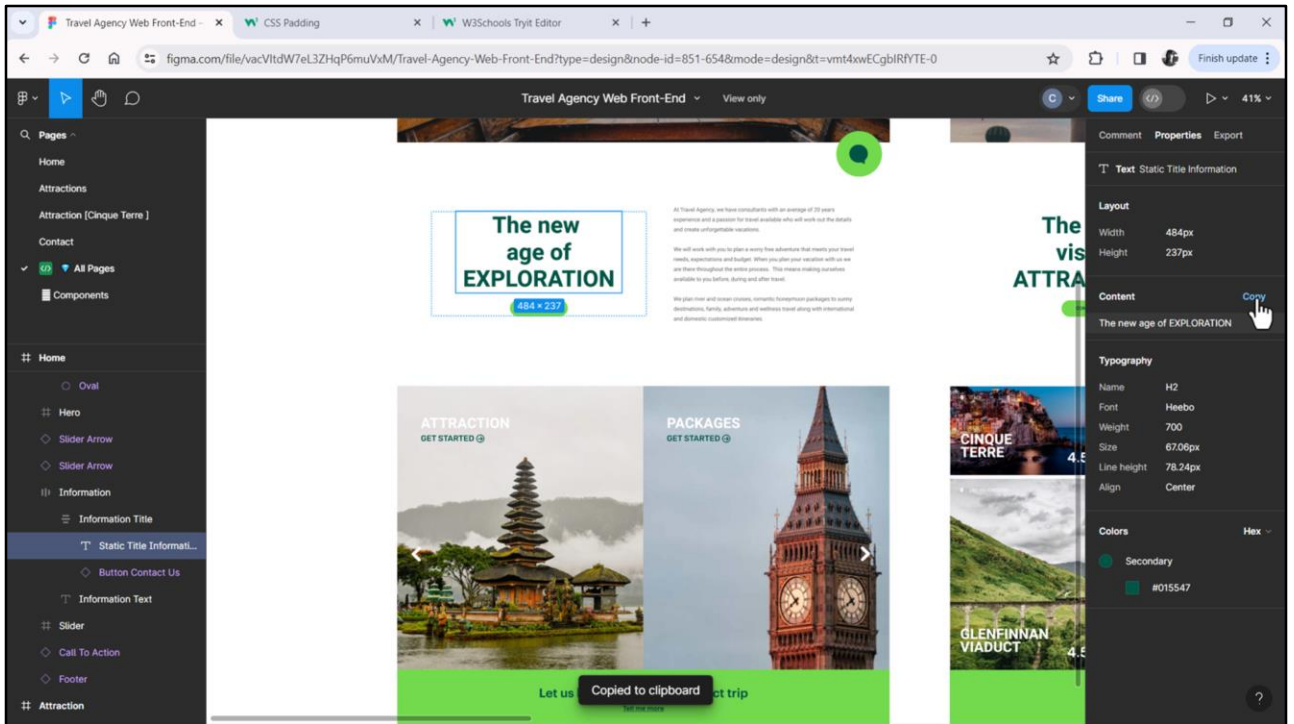
Entonces para evitar esas redundancias vamos a elegir dejar esos 712 dips en un lado solo, y va a ser a nivel de la fila del contenedor de ese Stencil. Entonces, aquí vamos a cambiar esto por 100%. De modo tal de que ahora, si tengo que cambiar a 800, esto va a estar bien, estamos diciendo que el alto de la tabla va a ser del 100% del contenedor...



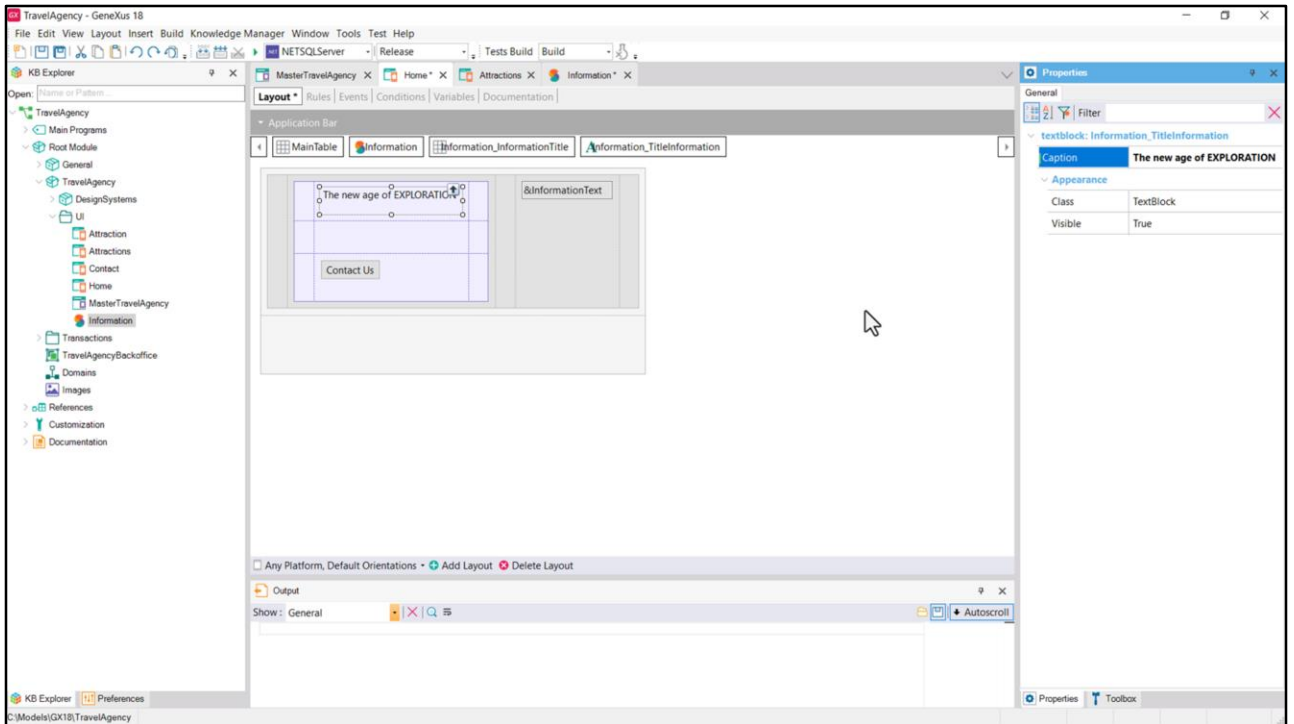
...y el contenedor en este caso es esta celda que corresponde a la primera fila de la tabla. O sea que acá cambiaría por 800 dips. Por supuesto lo mismo tendría que hacer en el panel de atracciones.



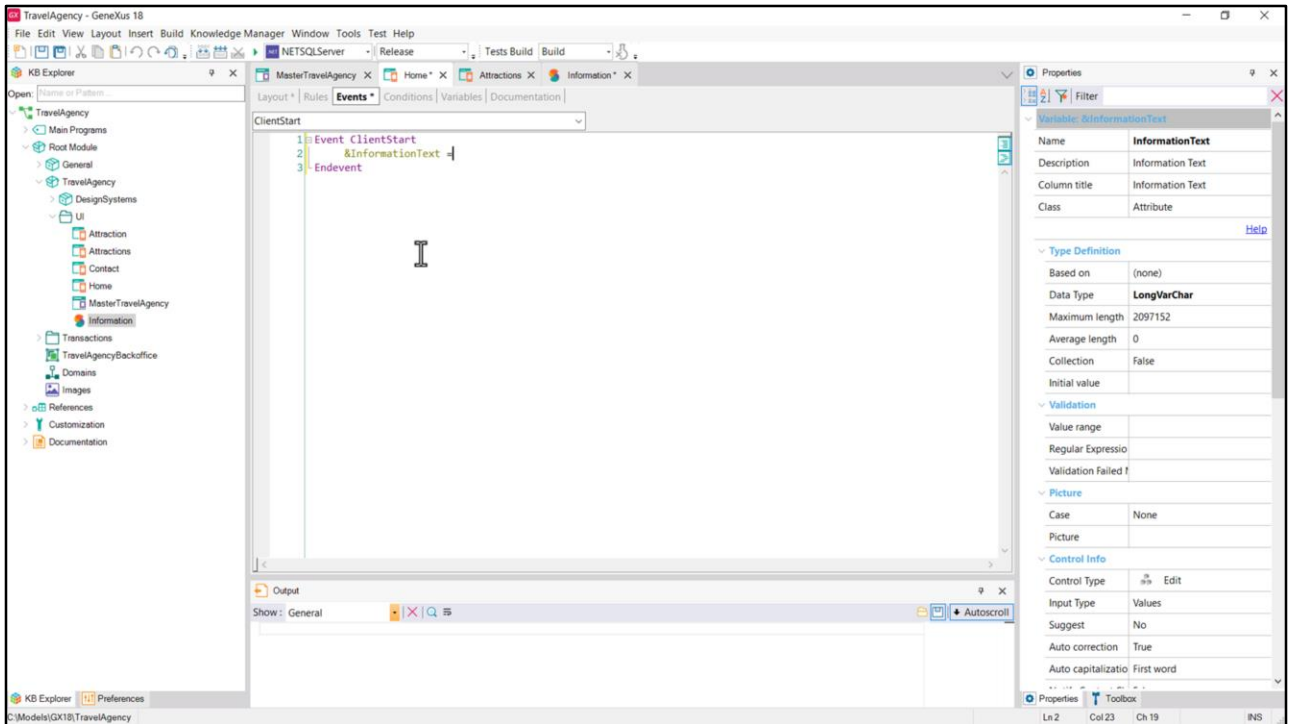
¿Qué nos queda para poder ejecutar? Asignarle valor al TextBlock y a la variable. Podría asignar ambos contenidos en el evento Start del cliente, o, por ejemplo, al TextBlock podría directamente asignárselo aquí, cambiando este Title...



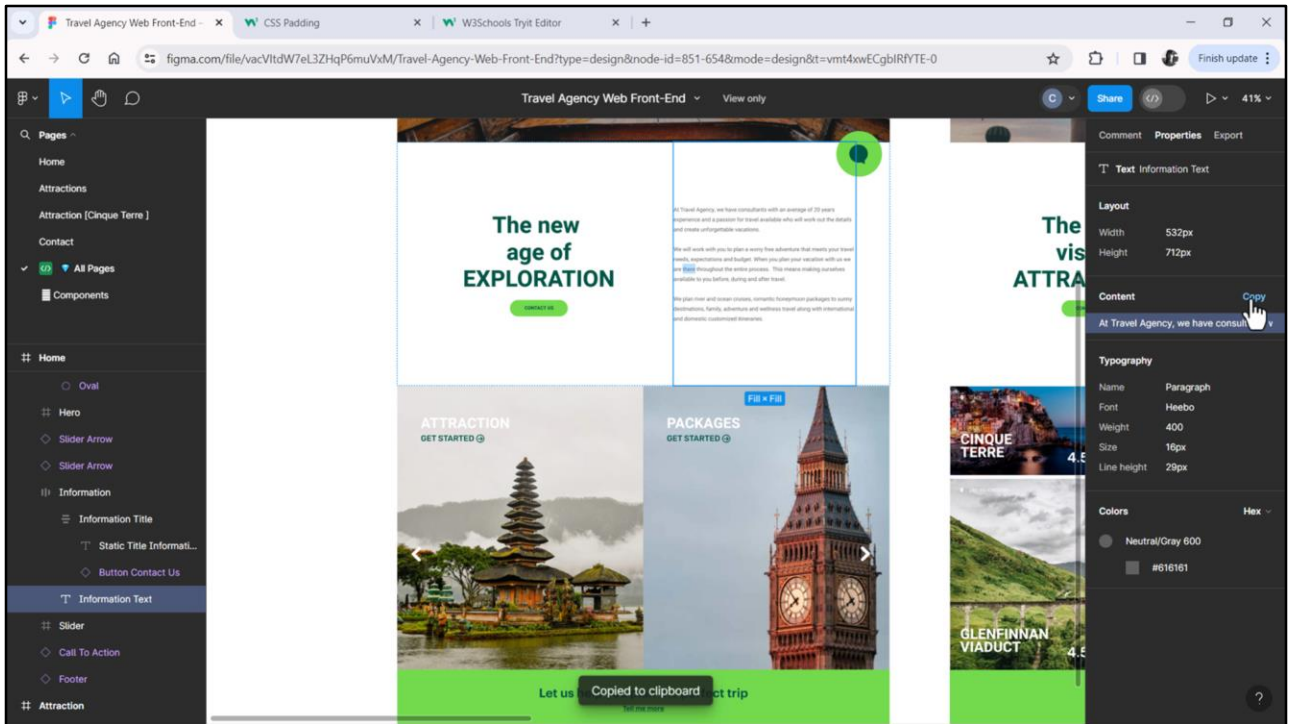
...por este contenido que puedo copiarlo directamente del archivo en Figma... ven que queda copiado al clipboard...



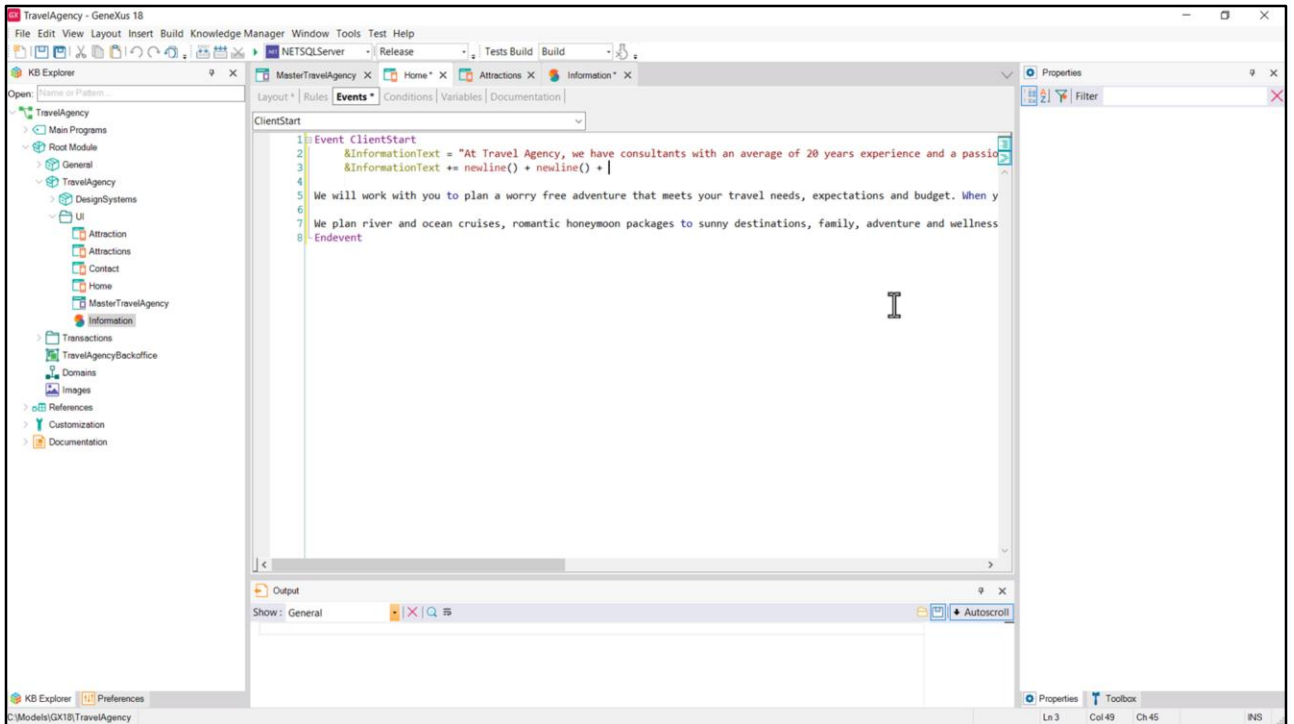
...y entonces vengo aquí y directamente lo pego.



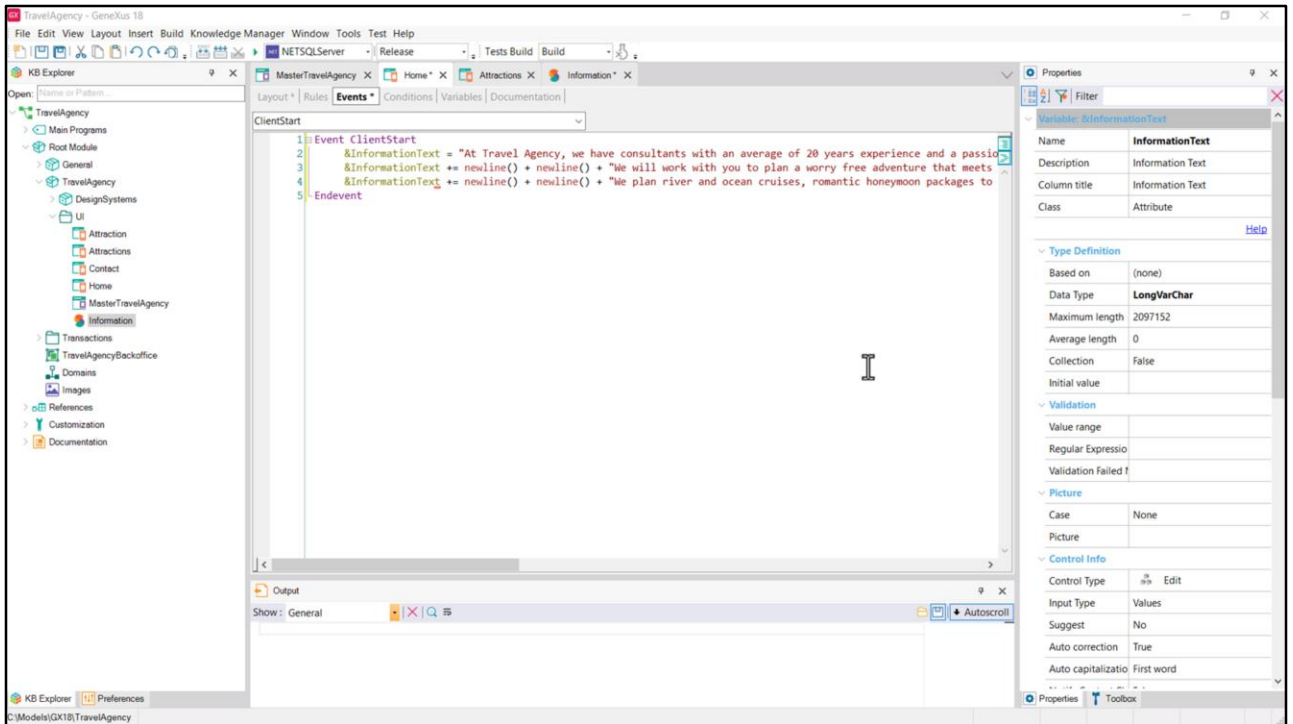
Y el de la variable sí no tengo más remedio que hacerlo en el evento ClientStart... entonces a la variable InformationText le asigno...



... el contenido que copio de aquí...

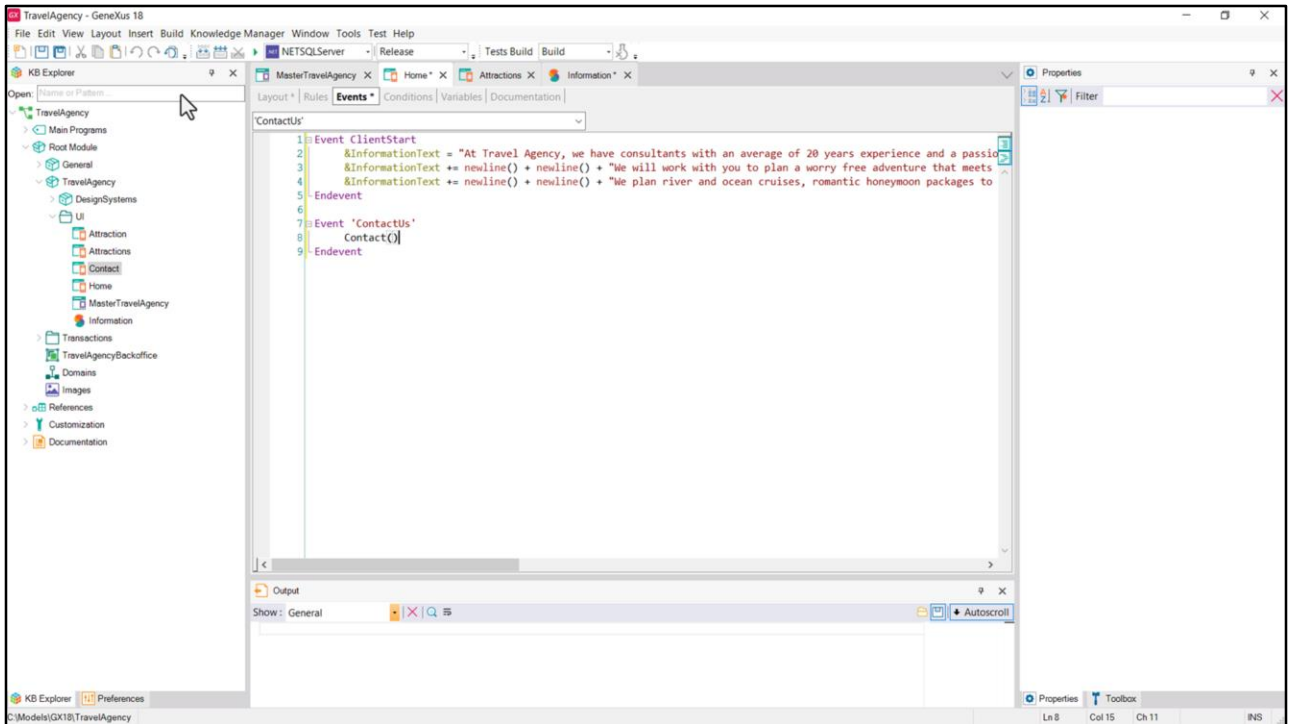


Y tengo que separarlo en las frases... Luego... a su contenido, a lo que tenía, le agrego un salto de línea, otro salto de línea...

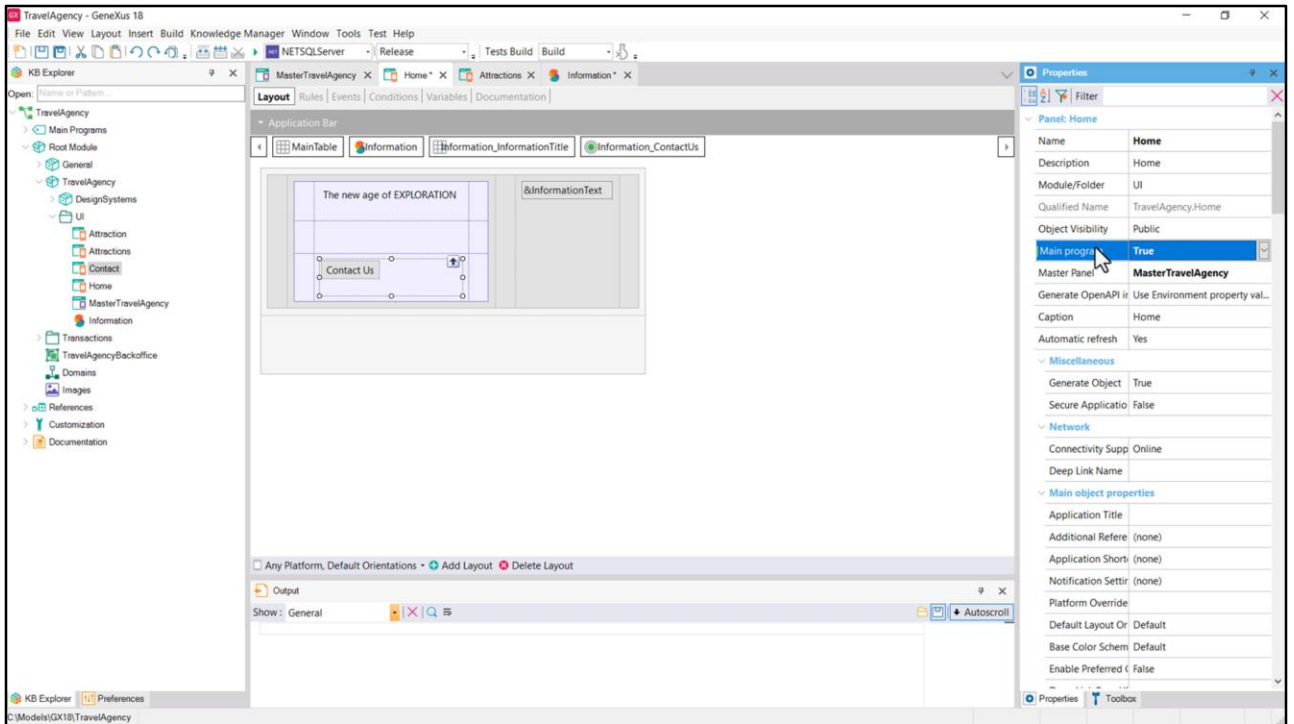


... y ahora sí, esto que seguía acá... y de la misma manera...

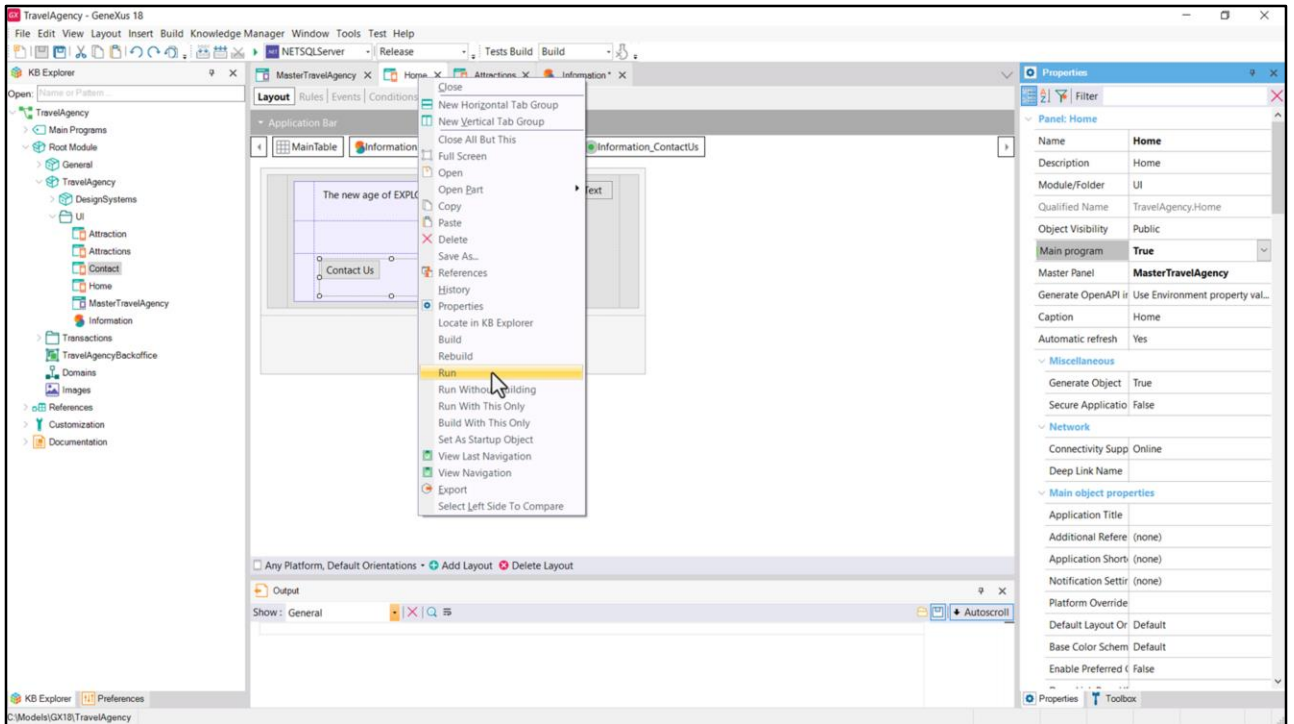
Una aclaración que no hice: el evento ClientStart se diferencia del evento Start en que el primero se ejecuta en el cliente, y el segundo en el servidor. Y se van a ejecutar en ese orden, primero el start del cliente, y después el del server. El contenido de la UI se puede actualizar en ambos, pero como en nuestro caso el contenido de la variable no depende de nada del server, y es un contenido estático, lo hacemos en el del cliente. Si estuviéramos implementando un Web panel en lugar de un Panel, no contaríamos con el ClientStart y ahí sí tendríamos que utilizar el Start.



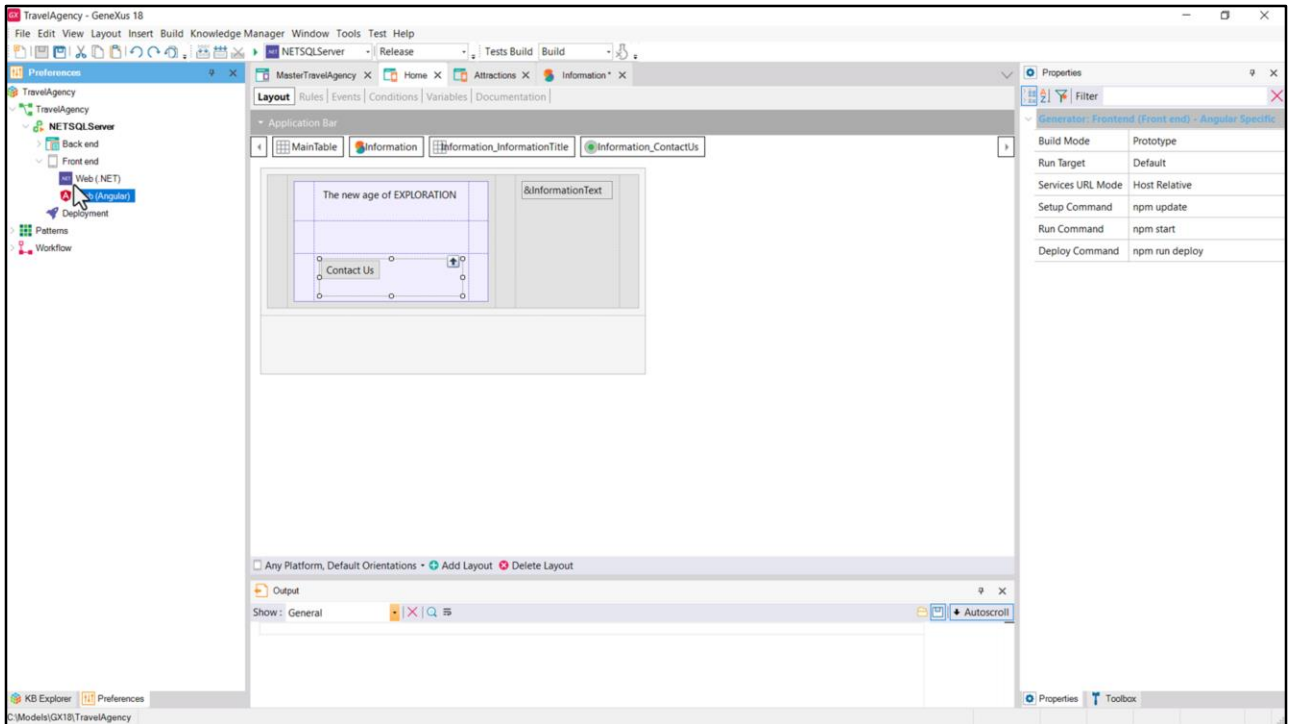
Bien, y por último, tenemos este evento ContactUs a nivel del botón y vamos a invocar a este panel... que por ahora está vacío.



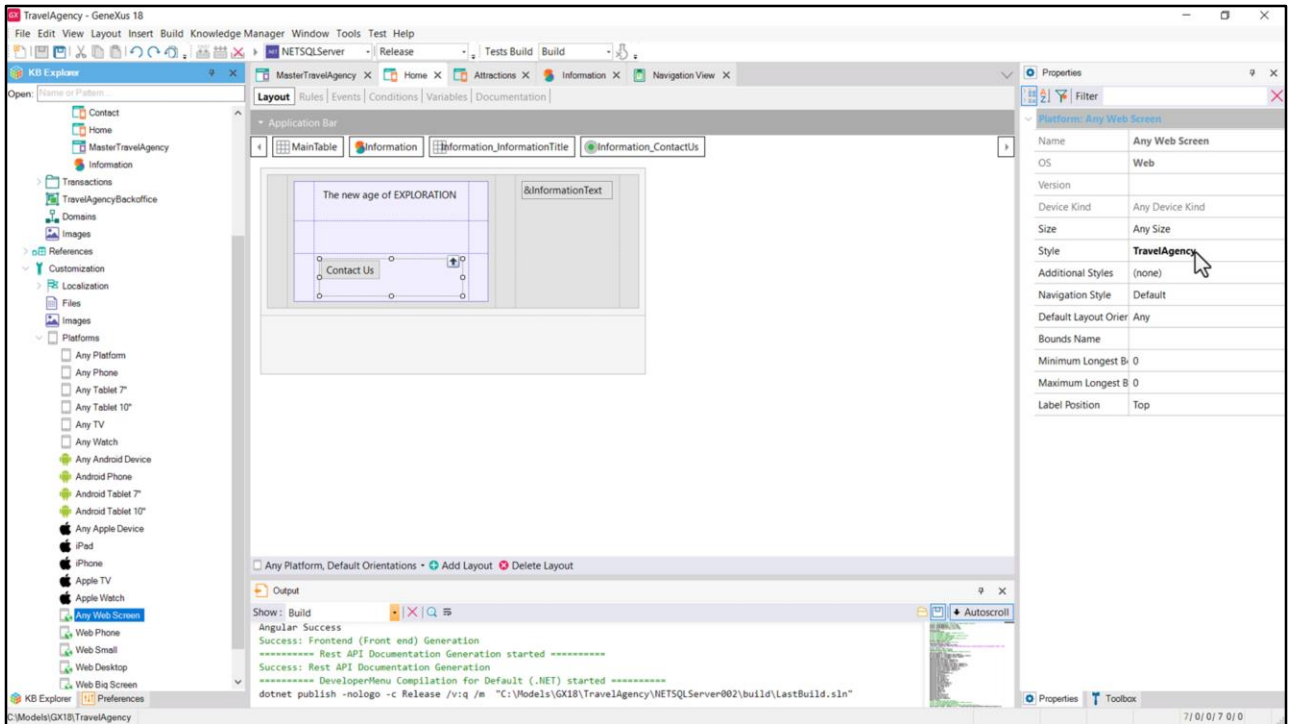
Bueno, con esto ya tenemos el objeto listo para ser ejecutado. ¿Qué nos hace falta para ejecutarlo en Angular? Por ejemplo, colocar este panel como programa Main.



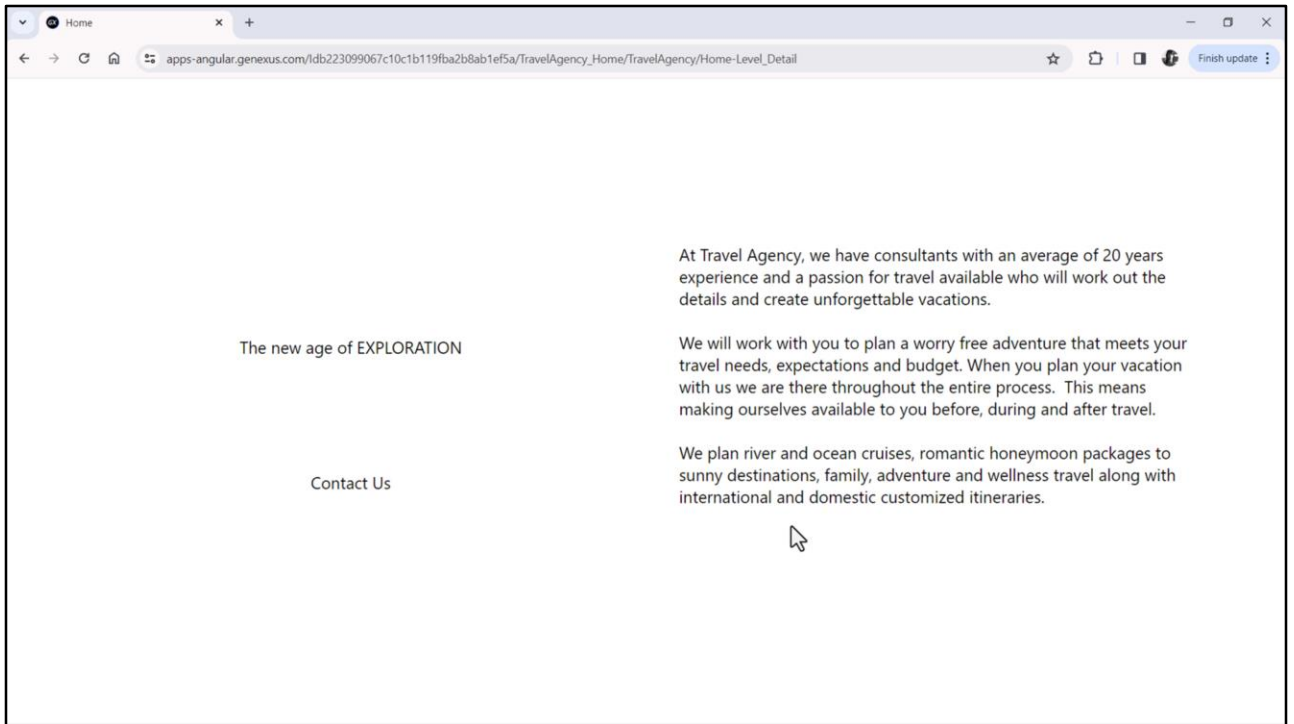
Y haciendo esto podemos hacer botón derecho sobre el panel / Run.



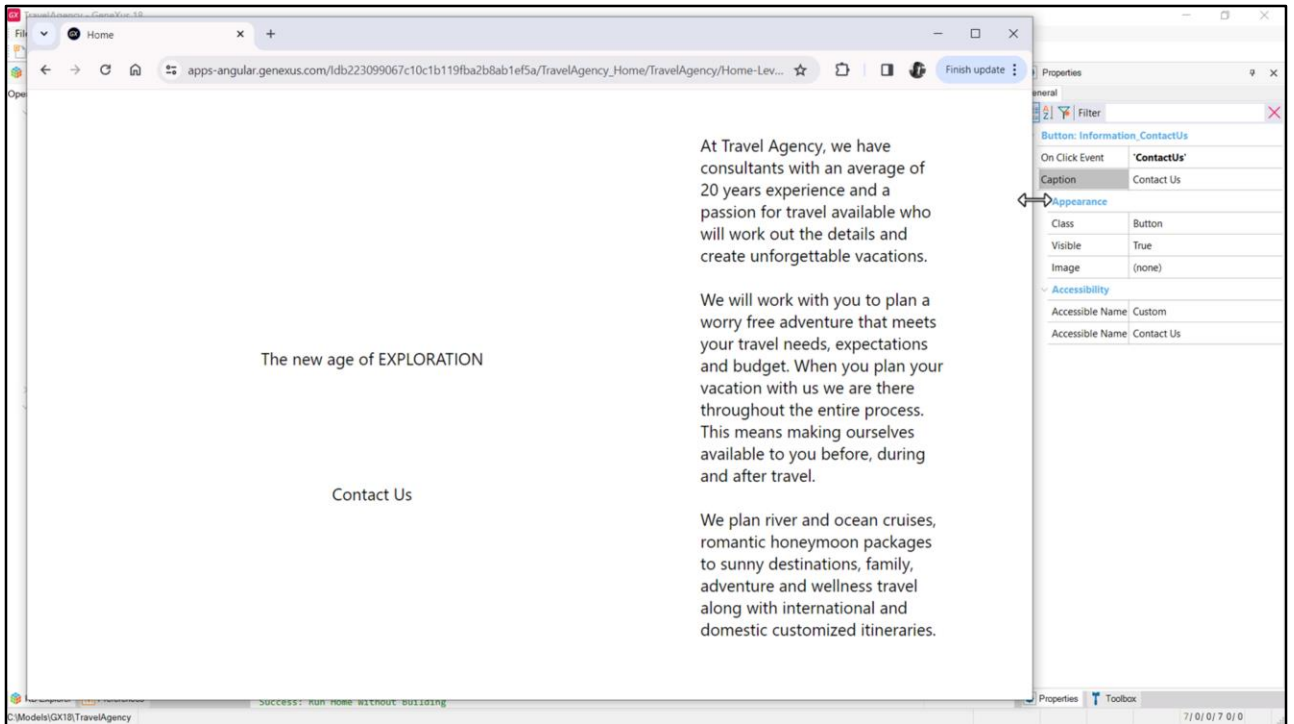
Sabiendo que tenemos el Front end Angular definido y que es preferible siempre para el desarrollo prototipar local y no en la nube, para ahorrarnos los tiempos de deploy cada vez que se actualice cualquier mínima cosita. Entonces...



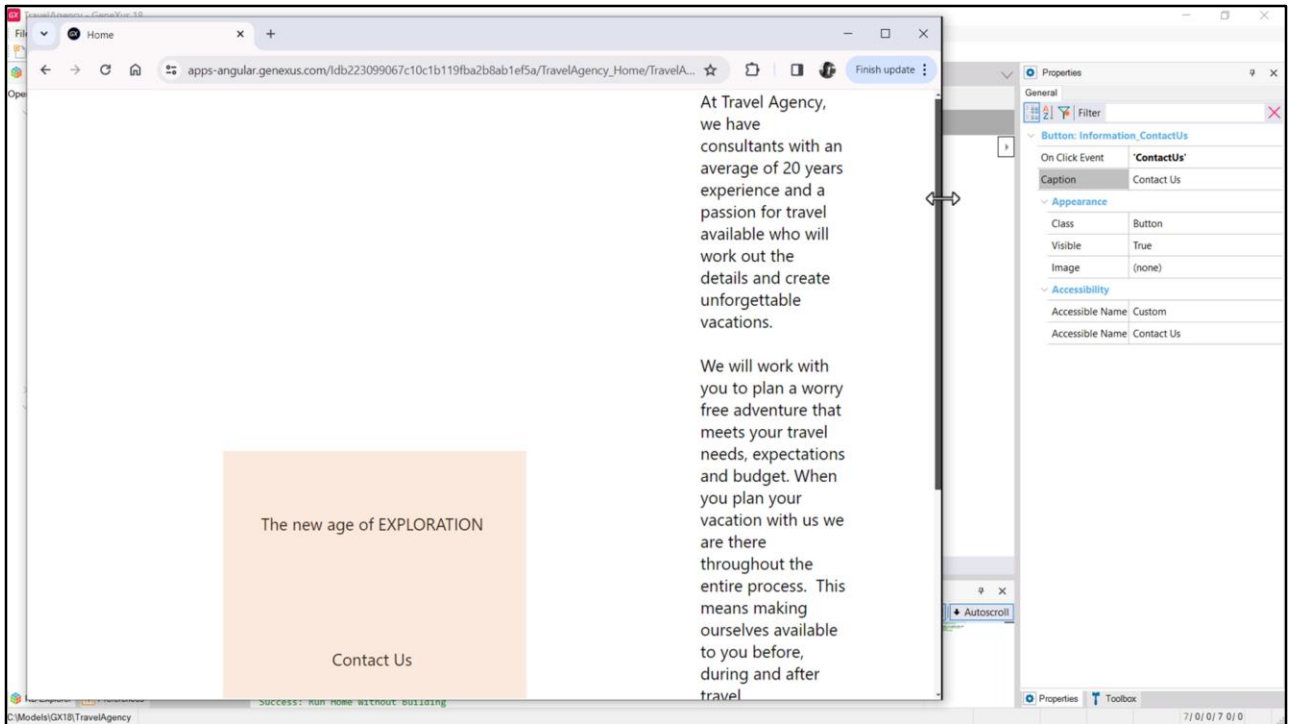
Antes de que termine recordemos que teníamos especificado que la plataforma Any Web iba a utilizar el DSO Travel Agency que por ahora está vacío.



Y así lo vemos en el navegador Chrome.

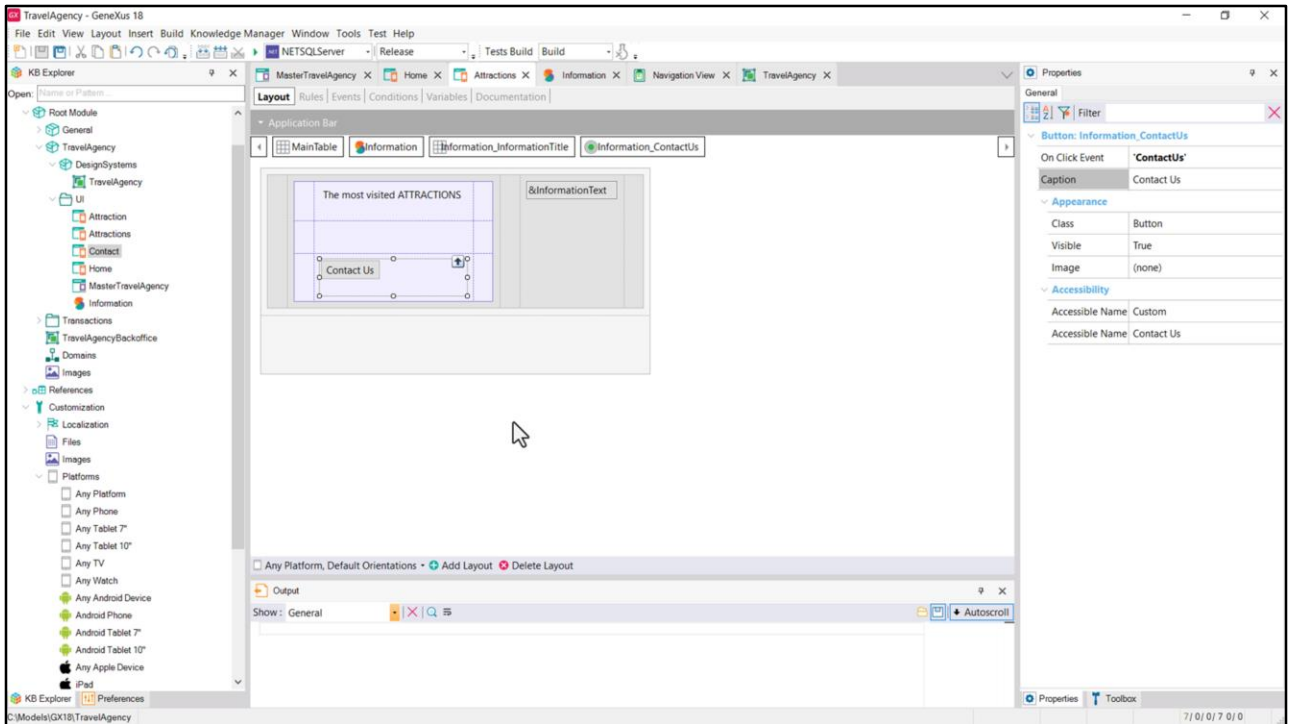


Veamos que si achicamos el ancho del navegador, tal como esperamos, crece el alto del texto de la derecha pero no se modifica nada de los demás.



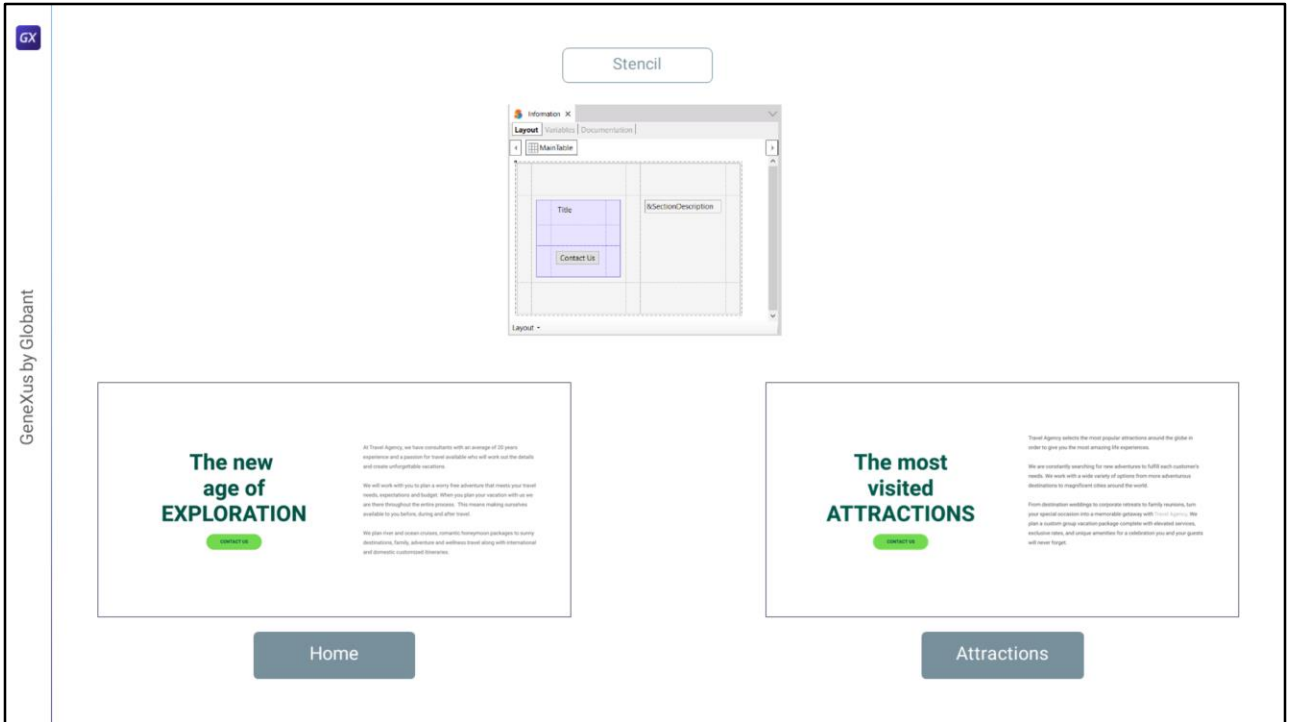
Nada a excepción del alto de la tabla, que es estirado para que el texto siga entrando y por eso vemos estos otros textos ir hacia abajo, dado que están verticalmente centrados.

Como nuestro DSO está vacío, claramente se le están aplicando a nuestros controles los defaults del navegador, tanto para el tipo de letra, su interlineado, el color, y todo lo demás. Sobre esto es que pasaremos a intervenir en los videos siguientes.



Por supuesto, haremos lo mismo que hicimos para el panel Home, ahora para el panel Attractions (lo reproduciré rápido para no aburrirlos y con eso damos por terminada esta clase).

Necesito aclarar algo antes de terminar: cuando se indica que un objeto es Main, esto se implementará como una aplicación completa que incluirá al objeto y a todos los involucrados a partir de él. Como aún no implementamos la parte del menú, desde el que quedarán todos nuestros objetos integrados es que hicimos esto de colocar el Home como main y por otro lado el de Attractions, para poder ejecutarlos ahora que todavía no los tenemos relacionados. Pero esto consume mucho más tiempo que si ambos objetos estuvieran integrados en la misma aplicación, porque así como están son literalmente dos aplicaciones independientes. Por ahora lo dejaré todo así porque nos vamos a concentrar en lo que sigue solo en el panel Home, pero luego haré que todos estén en el mismo programa, porque es super importante reducir los tiempos de desarrollo.



Bueno, y llegado a este punto ya estamos en condiciones de empezar a poner nuestras manos en la masa del Design System Object, para lograr que estas dos secciones de pantalla, tanto de Home como de Attractions, luzcan de esta manera.

GX

GeneXus by Globant

GeneXus[™]
by **Globant**

training.genexus.com