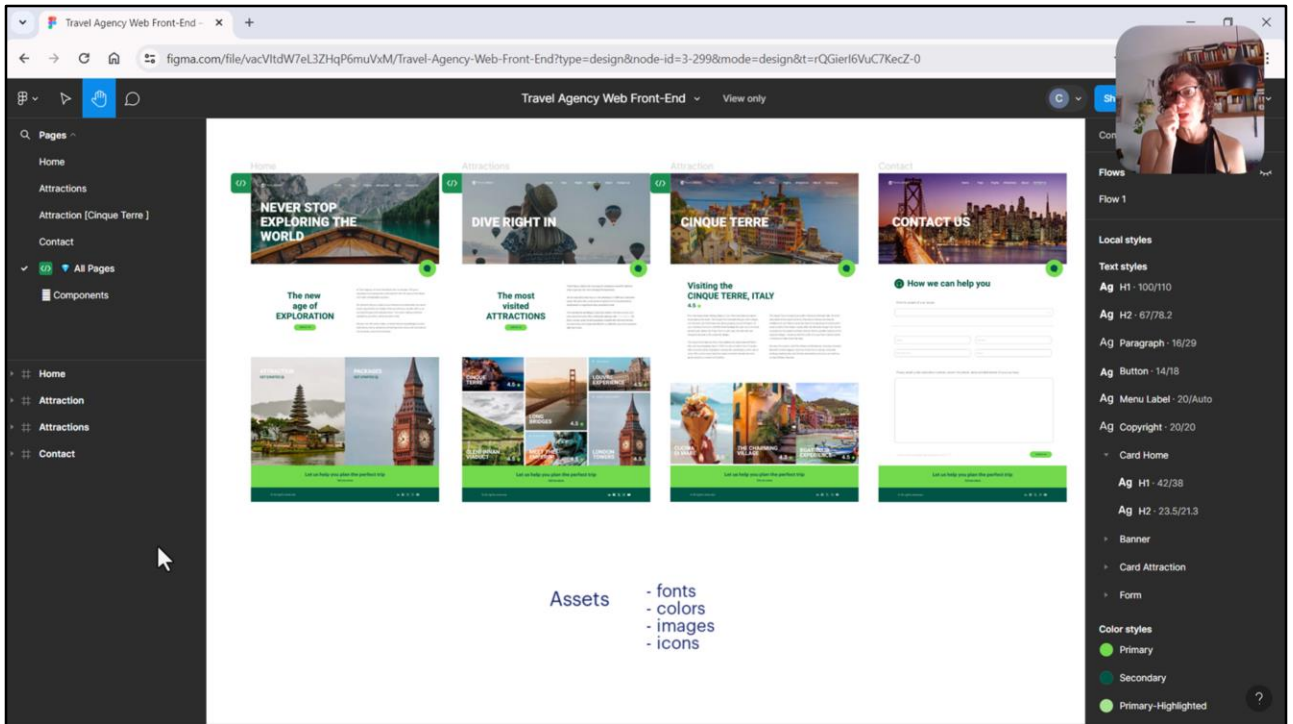


Assets

Fonts, Colors, Images



Cecilia Fernández



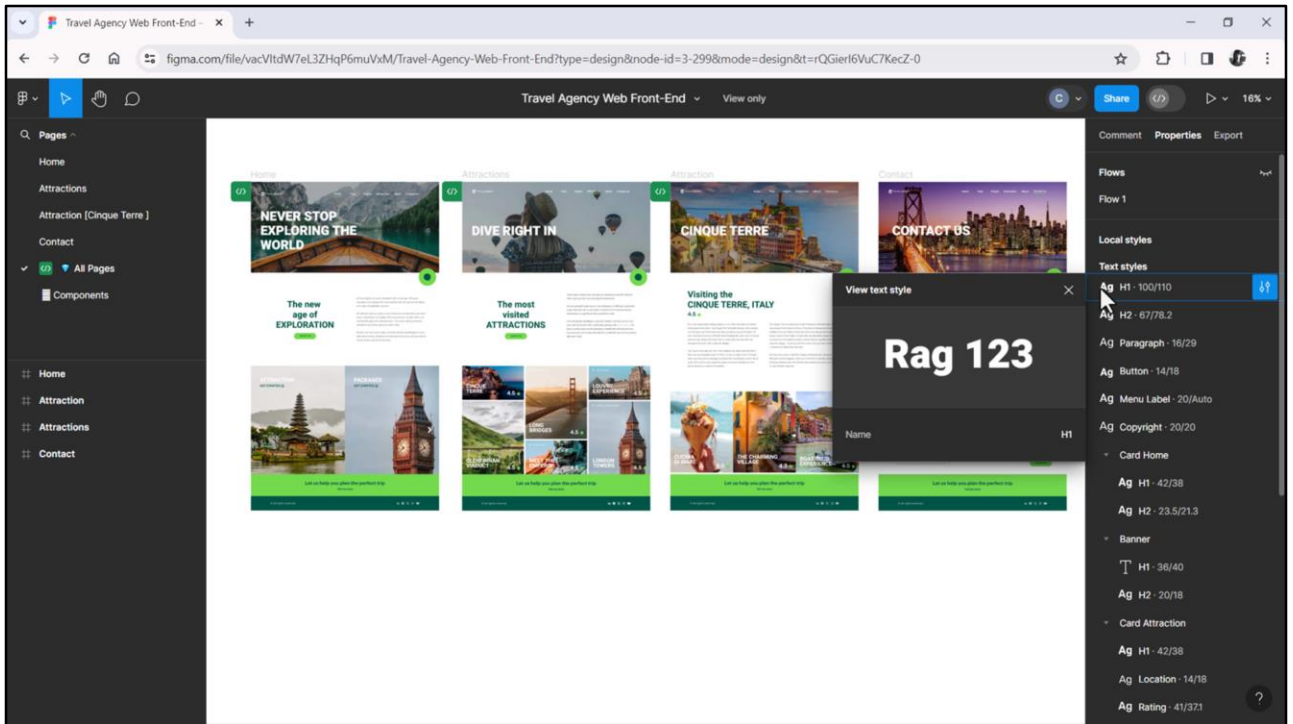
En el módulo anterior vimos cómo podíamos implementar estas dos porciones de layout, esta y esta, para que lucieran tal cual el diseño. Si usamos la metáfora del bosque, nos dedicamos a analizar dos hojas de dos árboles, pero casi no levantamos la mirada para ver el bosque como un todo.

Ahora que ya puedo dar como sabidos los **elementos básicos** que participan en el desarrollo de un Frontend, y por eso atendimos a estas dos hojas, justamente, podemos pensar cómo nos conviene comenzar las etapas iniciales del desarrollo de un proyecto de Frontend.

Para aceitar la etapa de desarrollo propiamente dicha, en general conviene integrar ya de entrada los assets, es decir, todos los recursos que deberemos utilizar en el proyecto, tales como fuentes, colores, imágenes, íconos, para no ir deteniendo cada vez que los necesitemos el desarrollo para irlos a buscar incorporarlos en la KB. Y además porque hacerlo todo de entrada, con la visión de conjunto, puede ayudar a ser más sistemático, es decir, más coherente.



Es así que si yo recién arrancara con el proyecto, lo primero que haría sería obtener las fuentes que voy a necesitar, e integrarlas a la KB.



Entonces acá, que estoy posicionada sobre la página que contiene todas las pantallas, puedo ver el listado de los estilos tipográficos utilizados en ellas... Pero en modo view (que es el gratuito) no conseguimos ver las fuentes que cada uno utiliza. Vemos apenas los nombres, y tamaño y alto de línea.

Si vamos a seguir en modo view, nos conviene pedirle, entonces, a nuestra diseñadora que nos comparta de alguna forma las fuentes que cada uno de esos estilos va a utilizar. Eso para no ponernos a inspeccionar nosotros los elementos de los frames buscando dónde se utiliza cada estilo tipográfico.

	A	B	C	D	E	F	G	H
1	Name	Font	Style	Size	Link			
2	H1	Heebo	Black	100	https://fonts.google.com/specimen/Heebo			
3	H2	Heebo	Bold	67				
4	Paragraph	Heebo	Regular	16				
5	Button	Heebo	ExtraBold	14				
6	Menu Label	Heebo	Medium	20				
7	Copyright	Rubik	Regular	20	https://fonts.google.com/specimen/Rubik			
8	Card Home/ H1	Heebo	ExtraBold	42				
9	Card Home/ H2	Rubik	Medium	23.5				
10	Banner / H1	Graphik	Semibold	36	https://commercialtype.com/catalog/graphik			
11	Banner / H2	Rubik	Medium	20				
12	Card Attraction / H1	Heebo	ExtraBold	42				
13	Card Attraction / Location	Rubik	Regular	14				
14	Card Attraction / Rating	Rubik	Medium	41				
15	Form / Regular Text	Graphik	Regular	20				
16	Form / Place Holder	Heebo	Regular	20				
17								

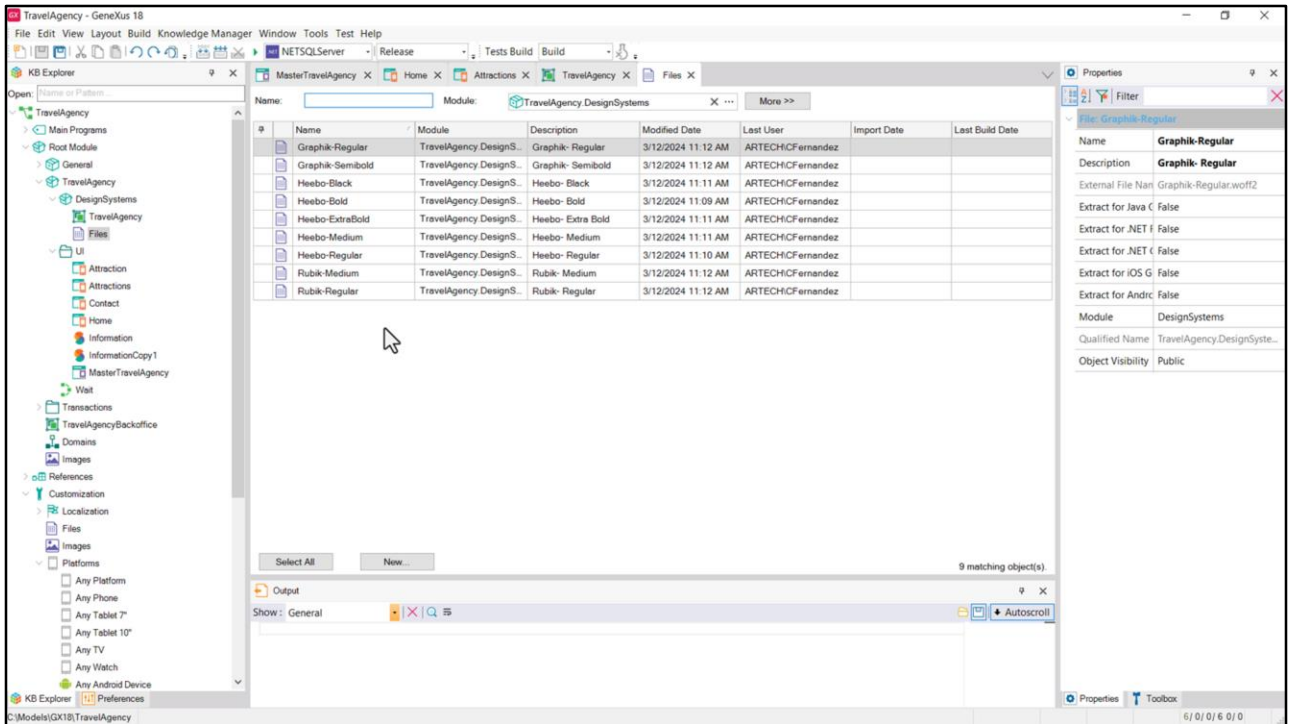
Entonces Chechu me envió esta planilla, donde vemos todos los estilos tipográficos y en esta columna la familia de fuentes. E incluso me compartió links para que pueda descargarme esas familias. La Graphik no es gratuita, pero yo ya la tenía. Si uds. no cuentan con esa familia, utilicen entonces la Rubik o la Heebo en estos dos casos.

The screenshot shows a Google Sheet with a table of font styles and a Windows File Explorer window open over it. The table lists various text elements and their corresponding fonts and styles. The File Explorer window shows a folder named 'Resources > Fonts' containing several font files in WOFF2 format. A file named '.ttf' is highlighted in a box.

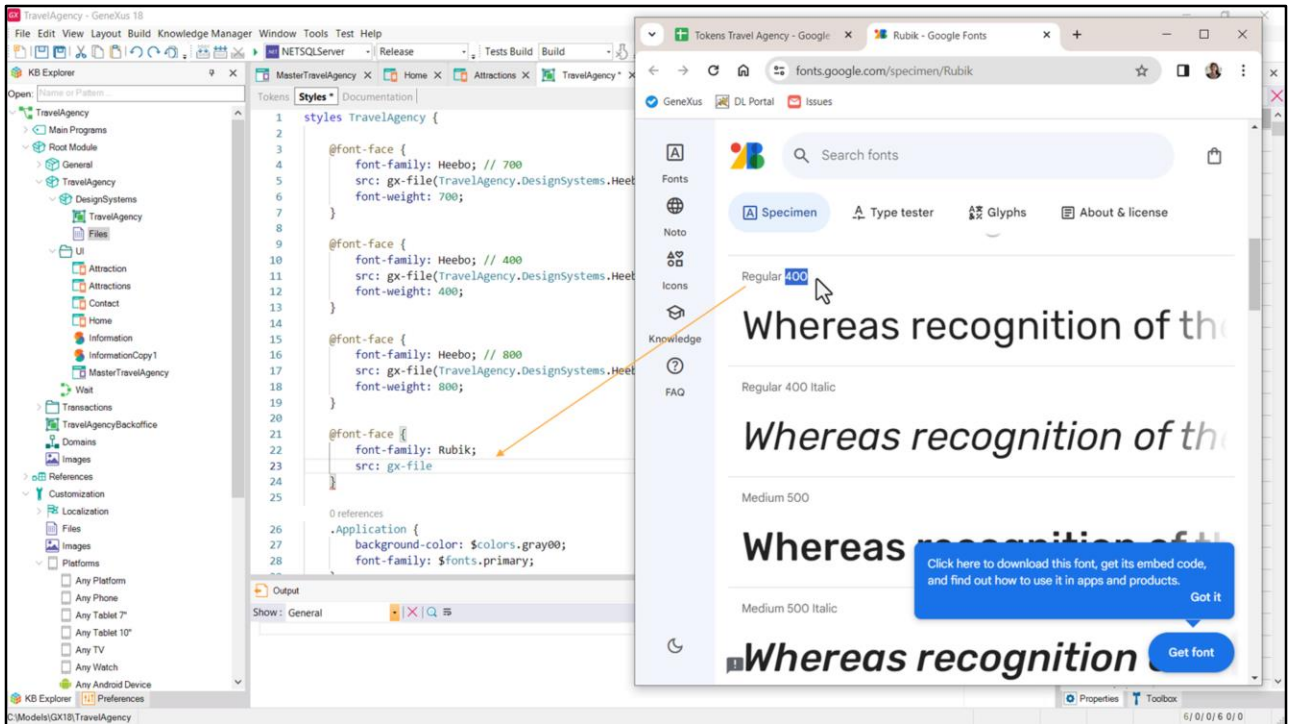
	A	B	C
1	Name	Font	Style
2	H1	Heebo	Black
3	H2	Heebo	Bold
4	Paragraph	Heebo	Regular
5	Button	Heebo	ExtraBold
6	Menu Label	Heebo	Medium
7	Copyright	Rubik	Regular
8	Card Home/ H1	Heebo	ExtraBold
9	Card Home/ H2	Rubik	Medium
10	Banner / H1	Graphik	Semibold
11	Banner / H2	Rubik	Medium
12	Card Attraction / H1	Heebo	ExtraBold
13	Card Attraction / Location	Rubik	Regular
14	Card Attraction / Rating	Rubik	Medium
15	Form / Regular Text	Graphik	Regular
16	Form / Place Holder	Heebo	Regular
17			

Aquí yo ya me había descargado antes los archivos para las fuentes, ¿se acuerdan hace muchos videos atrás, que les conté que me los había descargado en formato woff2 porque es muy liviano y nos sirve para Angular? La desventaja es que no se puede utilizar para aplicaciones nativas. En cambio podríamos haber descargado las fuentes en formato TrueType (ttf) y allí sí servirían exactamente las mismas para la aplicación nativa.

Hay que sopesar pros y contras para decidir. Yo por ahora voy a dejar las que son mejores para Angular, y cuando pasemos a pensar la aplicación nativa, si es que llegamos, entonces allí veremos cómo haríamos: si vamos a integrar dos versiones de la misma fuente (una woff2 y una ttf), o vamos a modificar la aplicación Angular para que pase a usar la misma versión que la de la nativa, es decir, ttf.

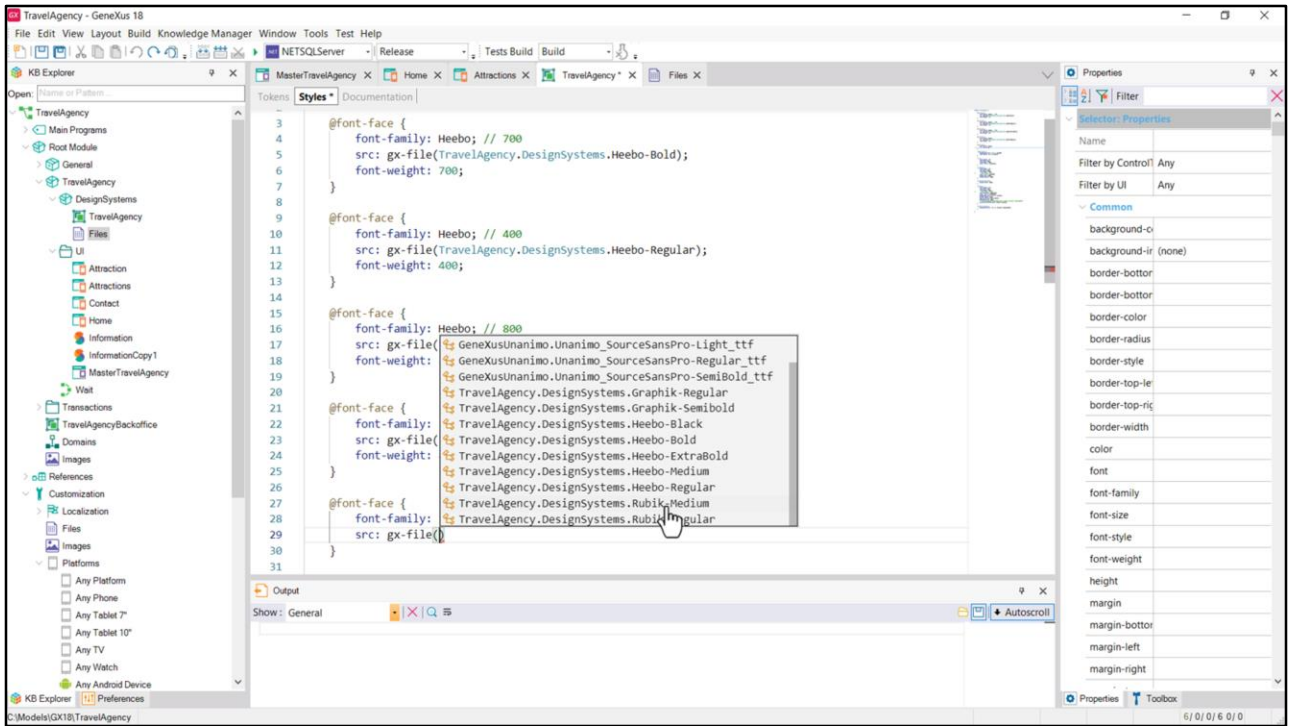


Paso siguiente será importar los archivos de fuentes dentro de GeneXus. Ya lo tengo hecho (lo habíamos hecho en uno de los primeros videos, ¿se acuerdan?).

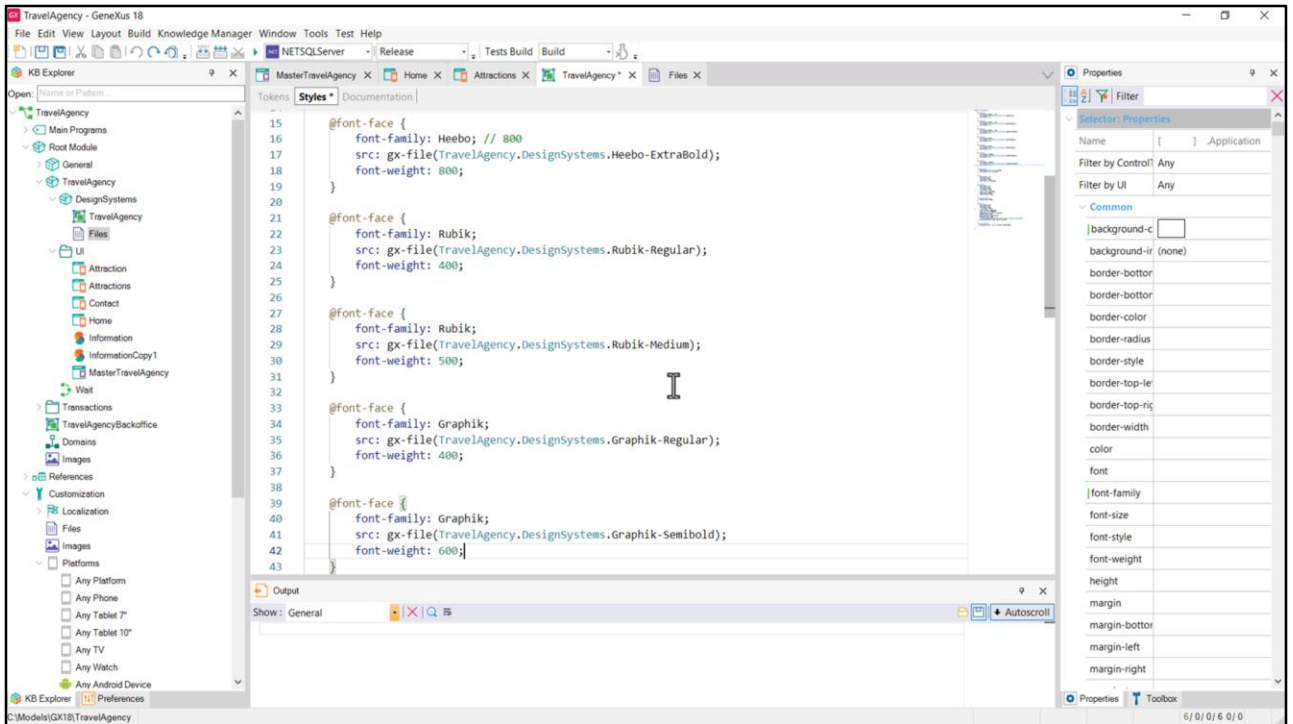


Lo que seguiría, entonces, sería declarar reglas font-face para incorporar todas esas fuentes en nuestro DSO. Aquí teníamos estas 3 declaraciones, completemos con todo el resto.

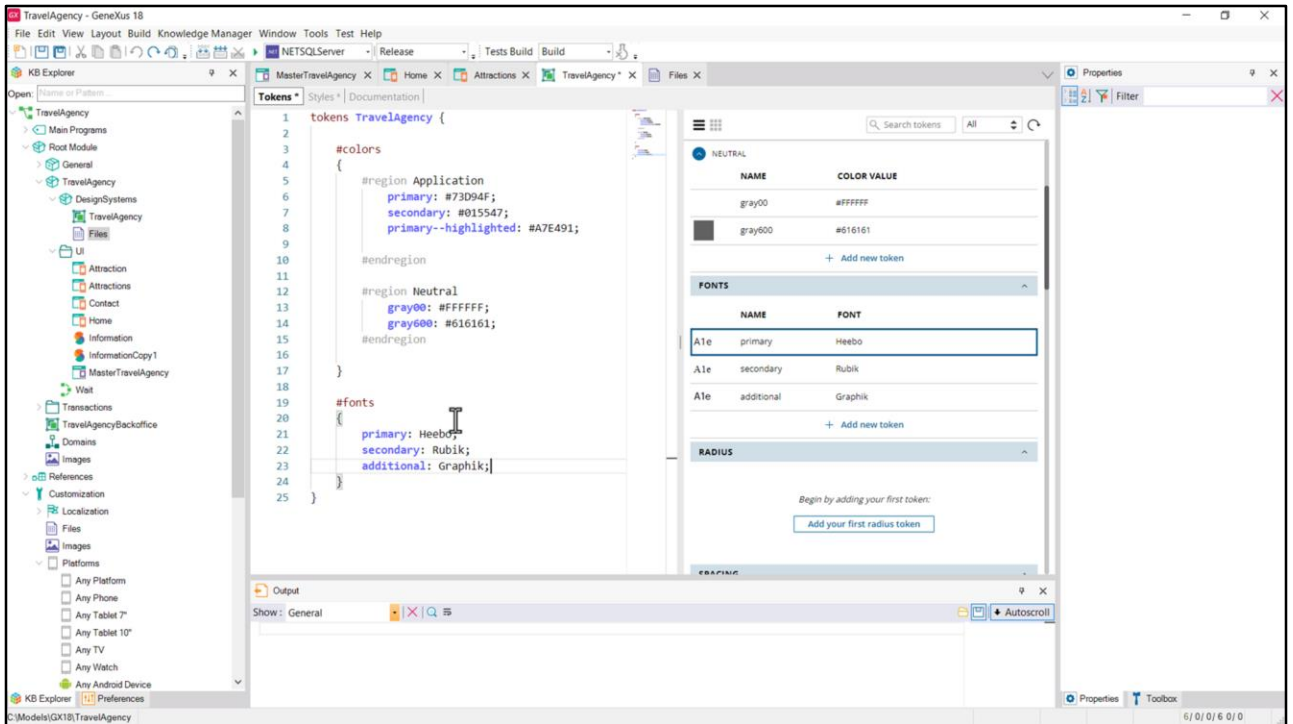
La Rubik-Regular, que es de peso 400, y el archivo es este de acá.



La Rubik... Medium, que es de peso 500.

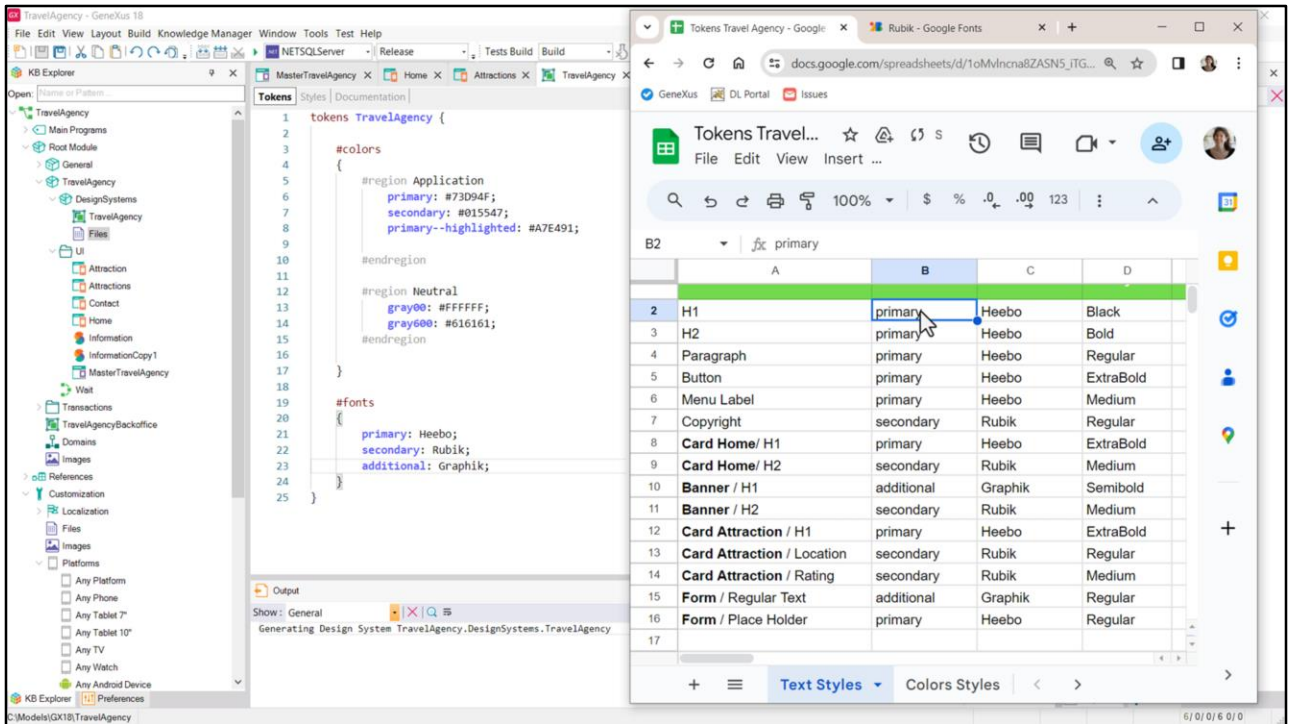


La Graphik... vemos que tenemos dos, la semi-bold y la regular. La regular es de 400, la semi-bold de 600.



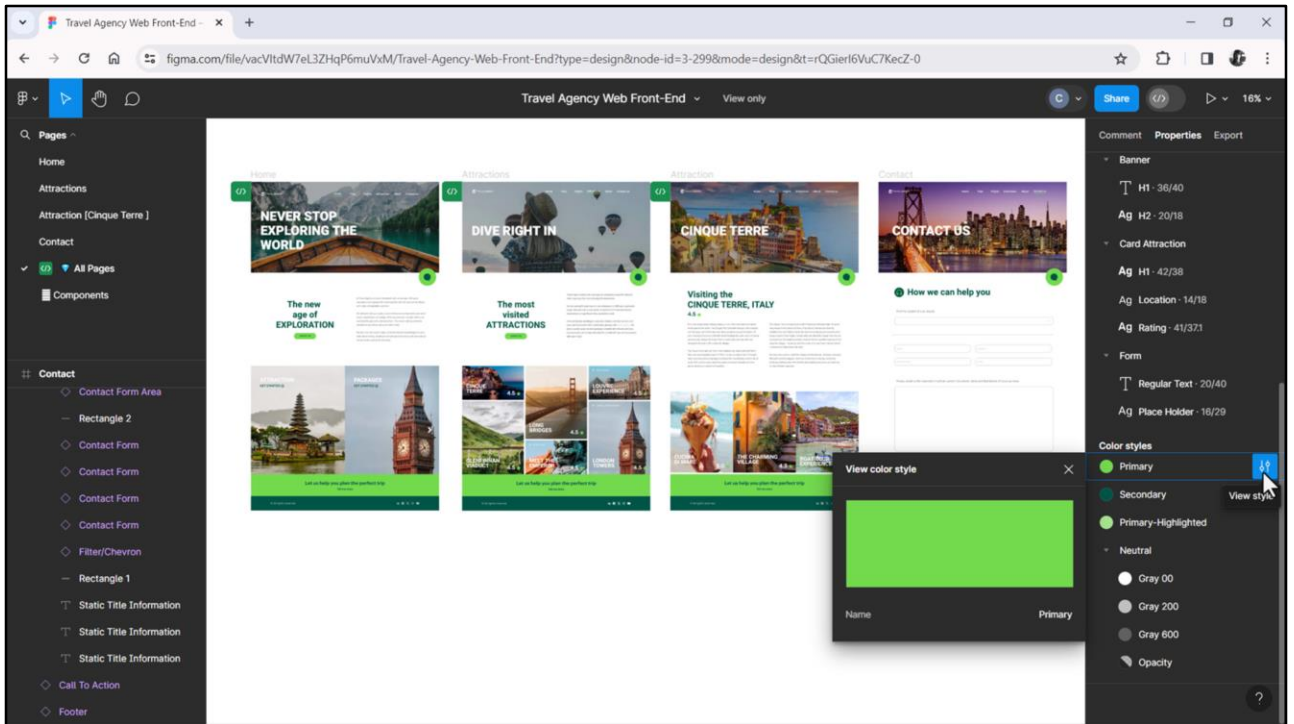
Tenemos 3 familias de fuentes, la Heebo es claramente la principal. Esta otra, la Rubik, se usa en varios casos, así que podríamos considerarla como una fuente secundaria, y esta, la Graphik, solo en dos casos.

Ya habíamos creado el token primary para la fuente primaria, y ahora voy a agregar dos tokens más para las otras dos.



Podríamos venir a la planilla de Chechu y en lugar de la fuente concreta, colocar el token, para cuando creamos las clases. Y con esto ya tenemos entonces todo listo, inicializado, para que nos sea menos engorroso incorporar las clases luego.

Colors



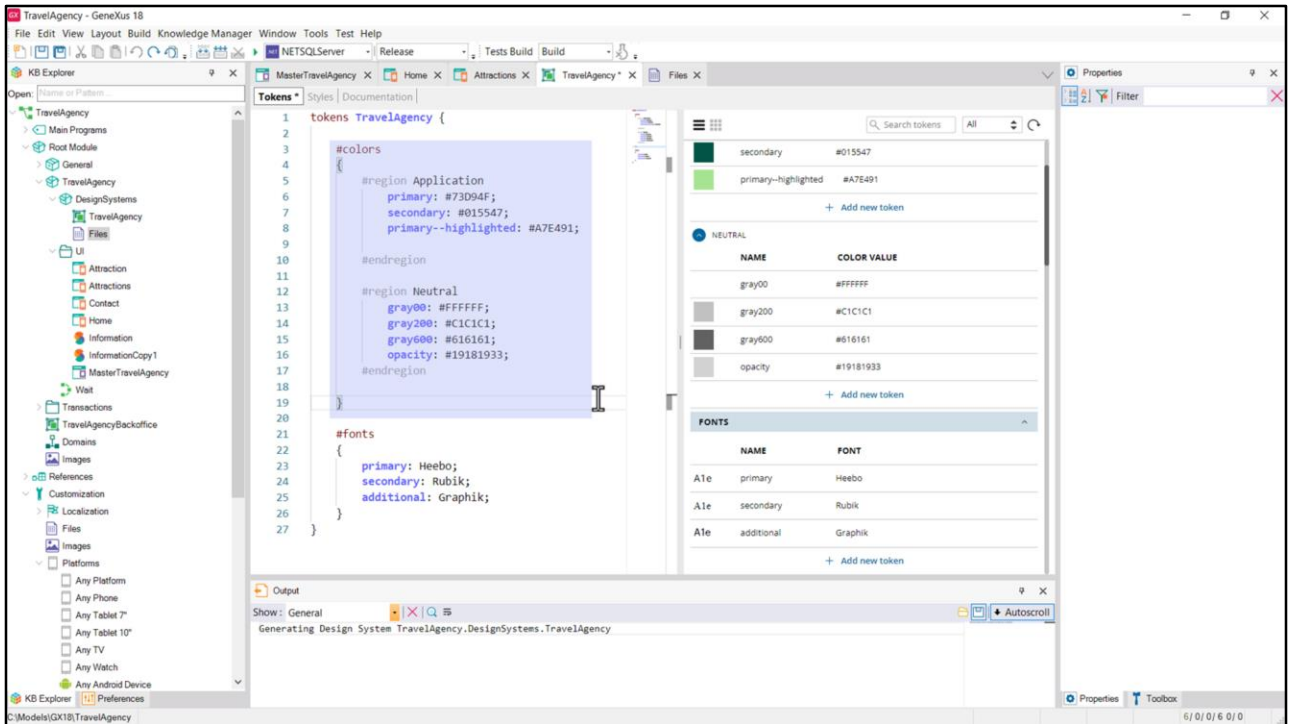
Luego vendríamos a Figma a ubicar los estilos de color, para poder definirlos como tokens en nuestro DSO.

Otra vez, aquí solo vemos los nombres pero no el valor.

The image shows a split-screen view of a design workflow. On the left, a Figma design tool displays a web page layout for a travel agency. The layout includes a header, a main hero section with two large images, and a grid of smaller images below. On the right, a Google Sheet is open, showing a color palette with columns for Name, Value (hex code), and Opacity. The sheet lists various colors used in the design, such as Primary, Secondary, and Neutral colors, along with their corresponding hex codes and opacity values.

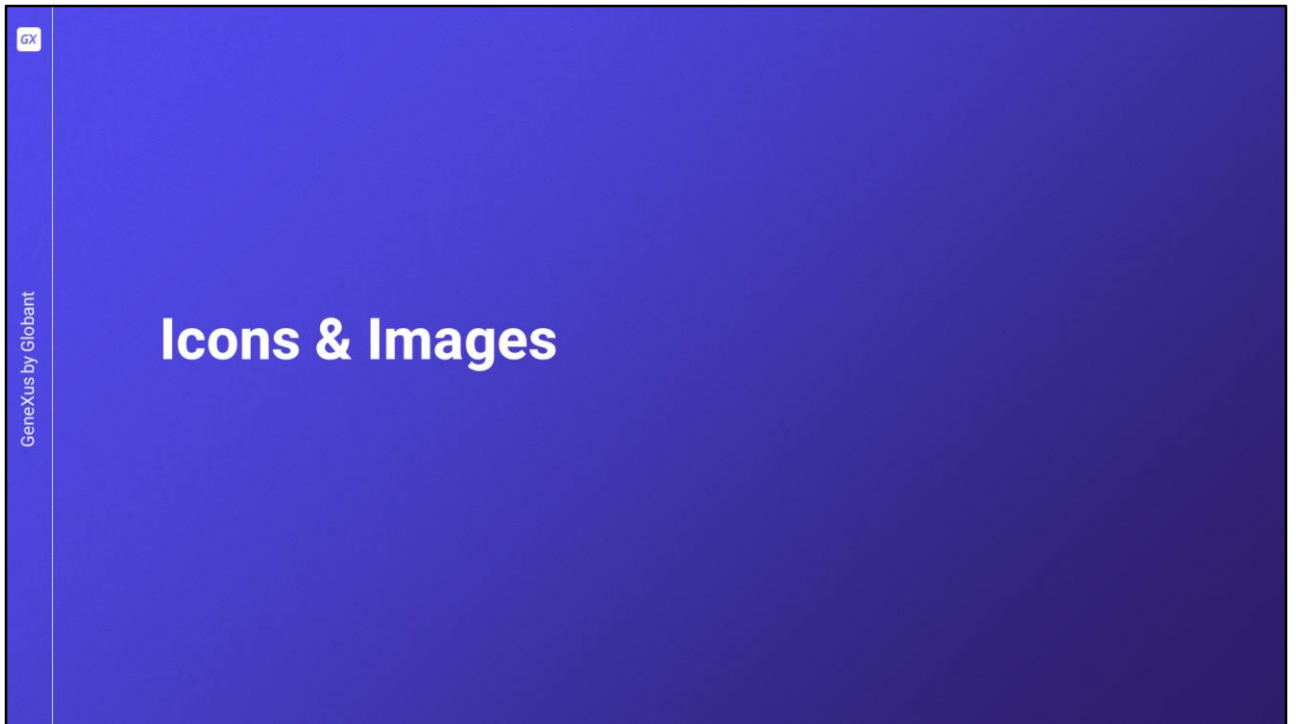
	A	B	C	D
1	Name	Value	Opacity	
2	Primary	#73D94F	100%	
3	Secondary	#015547	100%	
4	Primary-Highlighted	#A7E491	100%	
5	Neutral /Gray 00	#FFFFFF	100%	
6	Neutral /Gray 200	#C1C1C1	100%	
7	Neutral /Gray 600	#616161	100%	
8	Opacity	#191819	33%	
9				
10				
11				
12				
13				
14				
15				
16				
17				

Por lo cuál Chechu me los listó también en la planilla, en esta otra pestaña, para que yo no tenga que salir a buscar, inspeccionando cada elemento de las páginas hasta encontrarlos.

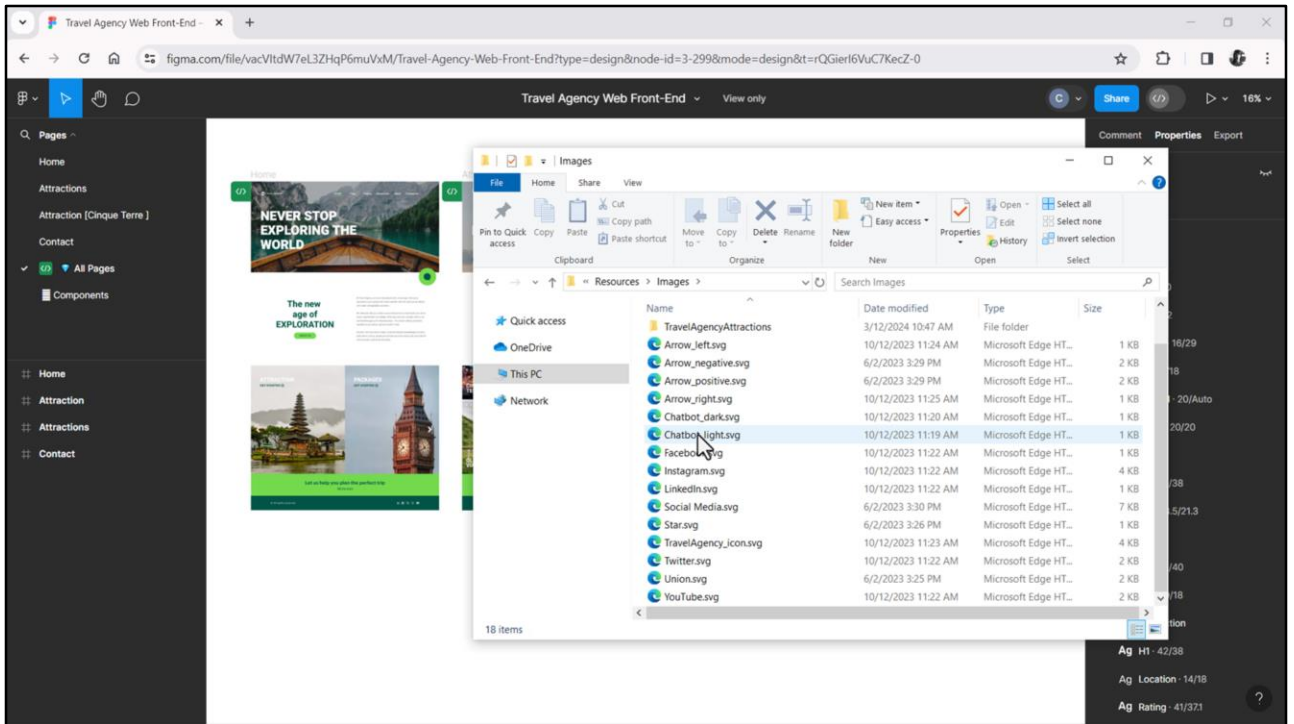


En los videos anteriores habíamos definido los que habíamos ido necesitando y los habíamos separado en dos regiones. Ahora definiremos los que nos faltan, que son dos: gray200... y opacity... que tiene 33 porciento de opacidad, así que agregamos el 33.

Bien, podemos preguntarnos ahora mismo si nos alcanza con este tan básico nivel de abstracción, el de estos tokens globales que lo único que hace es definir la paleta de colores que usará la aplicación, pero no muestra los colores organizados en relación a la función que cumplirán en el sistema. En el video siguiente abordaremos ese tema, particularmente, que se vincula más con la idea de **sistema** de diseño. Por ahora introducimos la paleta de colores.



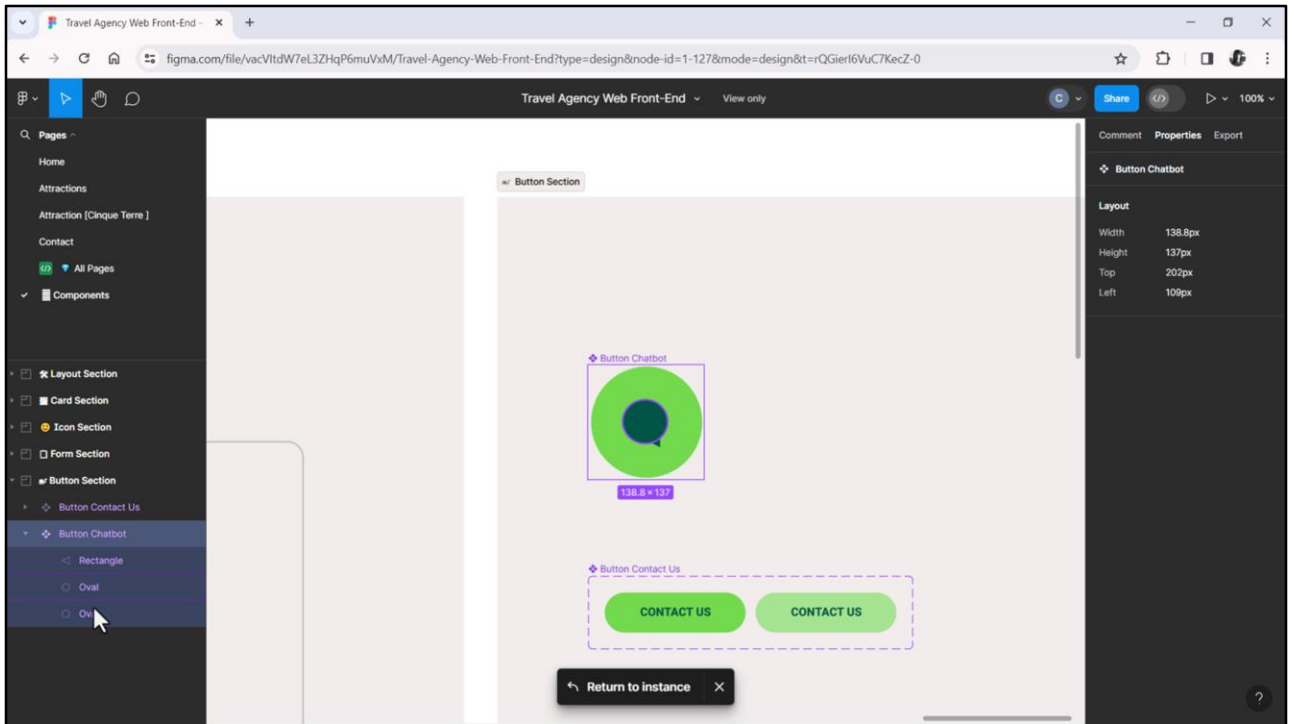
También sería bueno ya introducir en la KB los íconos e imágenes que se necesitaremos.



Yo ya tengo descargado todo eso. Se los voy a mostrar primero en el folder, y luego les mostraré cómo fue que obtuve cada cosa de Figma.

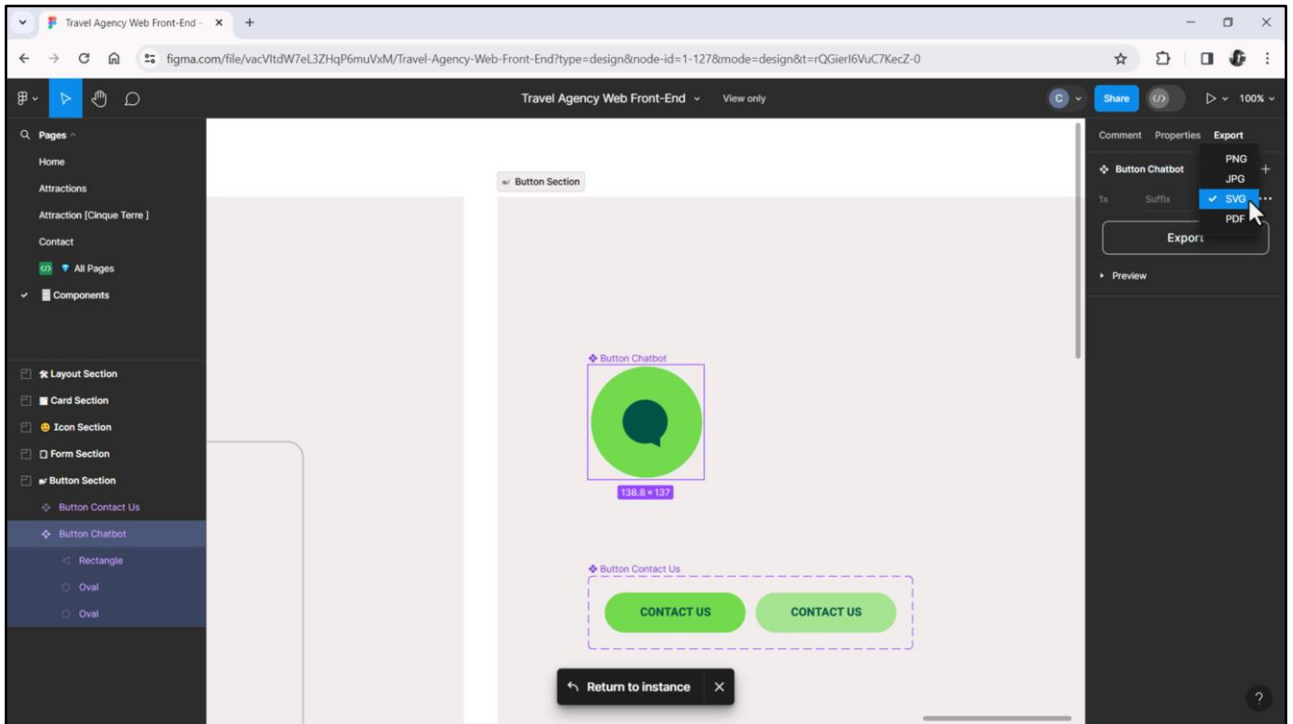
Aquí vemos los íconos... Por ejemplo, el del chatbot, en modalidad light, porque luego, más adelante veremos la dark, o por ejemplo, la estrella... Vemos que se trata de archivos SVG.

Una posibilidad es pedirle a nuestra diseñadora que nos pase todos estos recursos para no tener que hacer nada de esto que vamos a hacer ahora. Pero la otra opción es que lo descarguemos nosotros mismos desde el archivo en Figma.



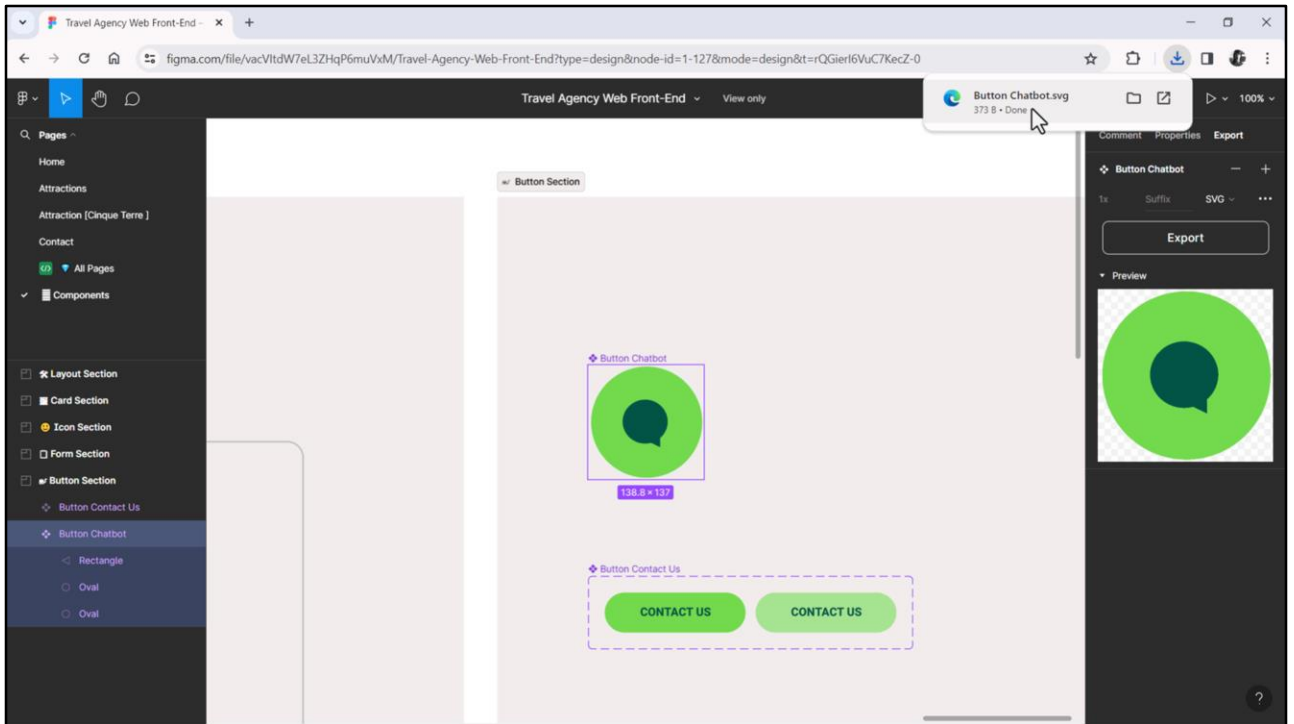
Por ejemplo, veámoslo con la imagen del chabot. Vemos que el elemento está modelado como un componente, y cuando lo vamos a inspeccionar, vemos que no es una imagen, sino que Chechu lo armó con estos elementos simples.

Nosotros lo necesitaremos como imagen, porque lo implementaremos en GeneXus como un botón con esa imagen justamente.

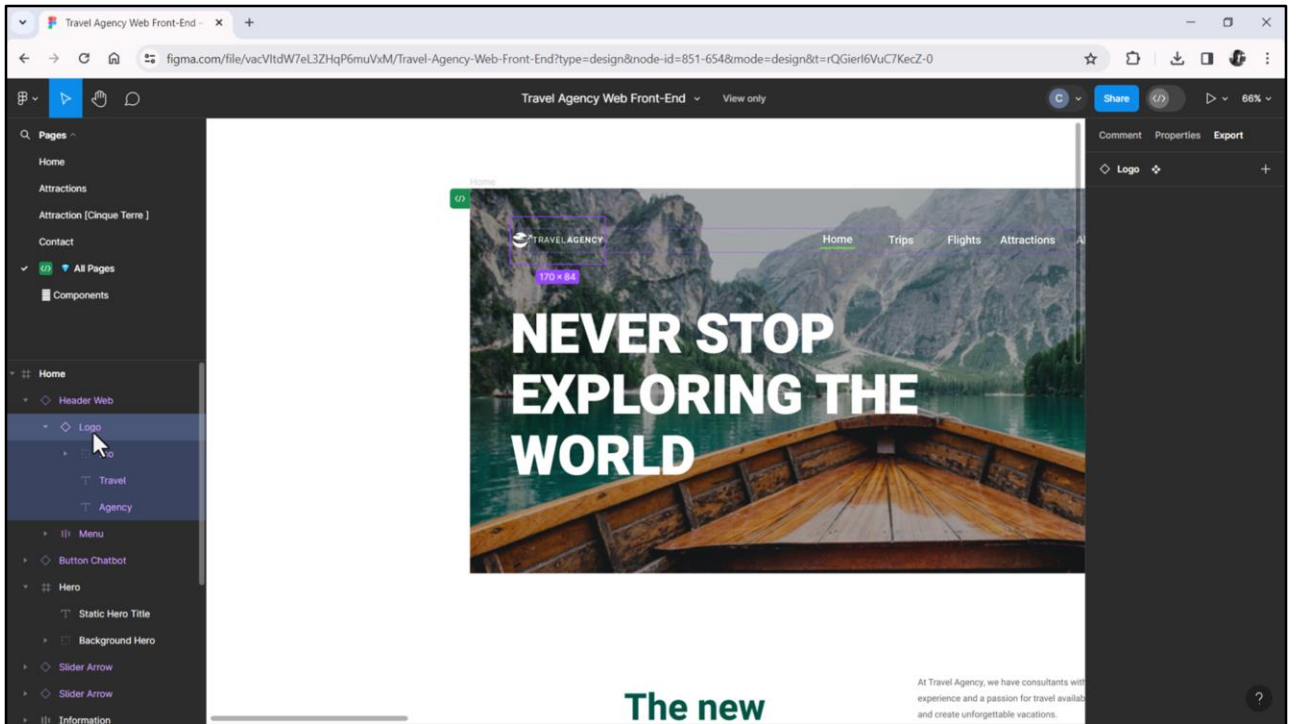


Dado que se trata de una imagen simple, no compleja, es que optamos por el formato SVG, que tiene varias ventajas frente a formatos como el png o el jpeg, entre ellas: que los archivos SVG escalan sin perder calidad, dado que son gráficos vectoriales, y que tienen menor tamaño (lo que ayuda a reducir el tiempo de carga de la página). Esas son dos ventajas muy importantes.

Entonces, los íconos, que son imágenes simples, y todas las imágenes simples que encontremos en el diseño deben ser todos archivos SVG.

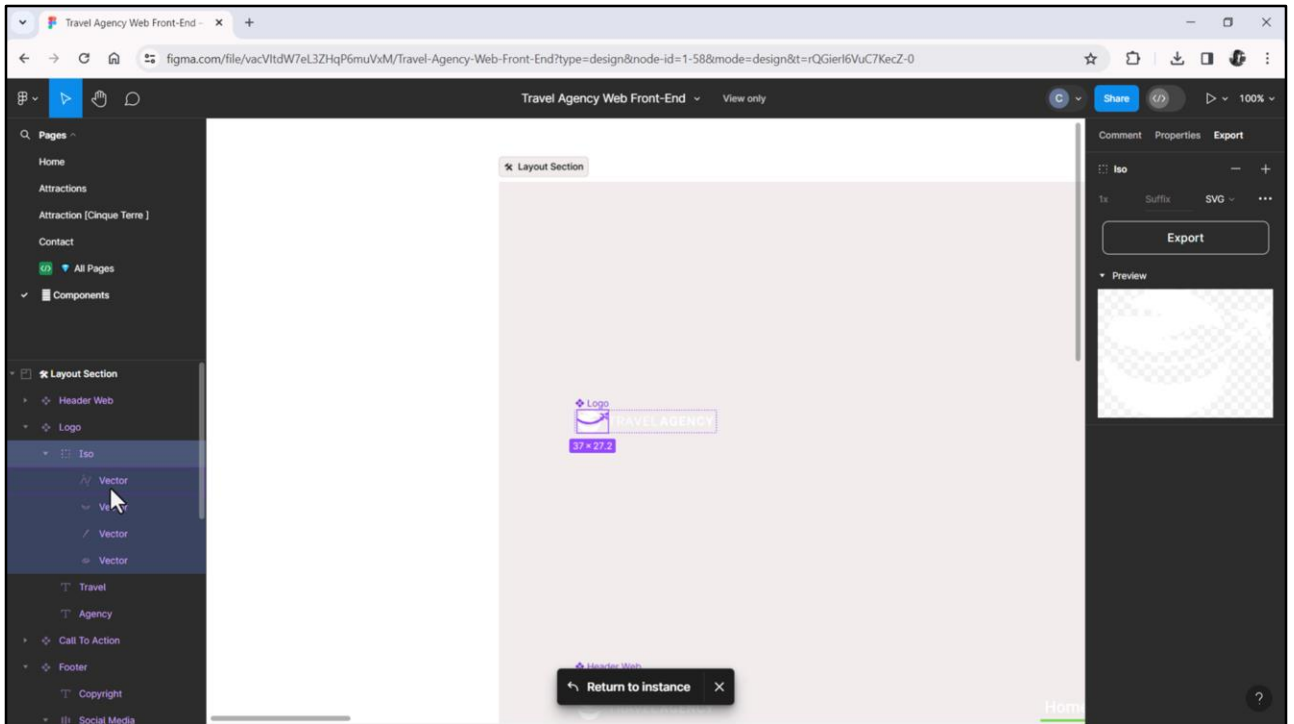


Lo que hacemos entonces es presionar aquí para descargarlo. Y allí lo vemos.

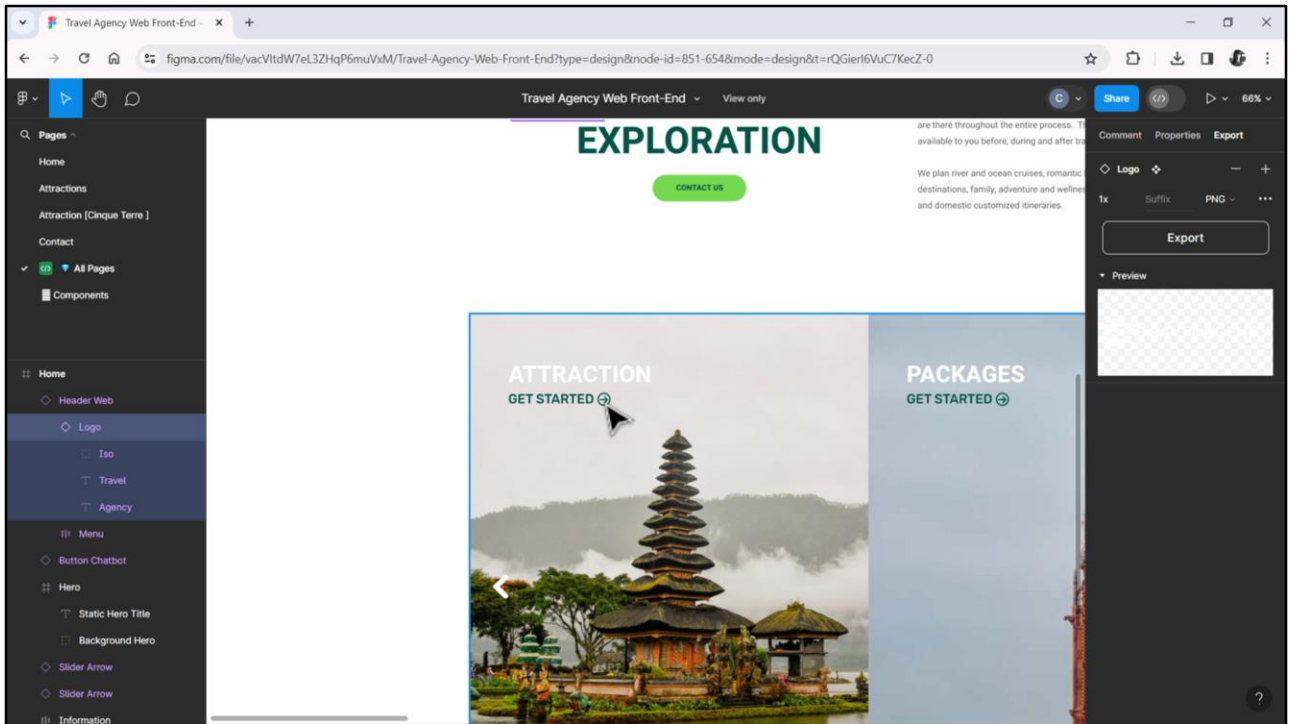


Bueno, entonces tendríamos que recorrer todas las pantallas buscando los íconos para descargarlos, así como hicimos con el del chatbot. Por ejemplo, tenemos acá, en el Header, el logo. Está compuesto el logo por tres elementos: uno es el ícono del logo, el texto “Travel” y el texto “Agency” (lo voy a agrandar para que se vea mejor). ¿Ven?

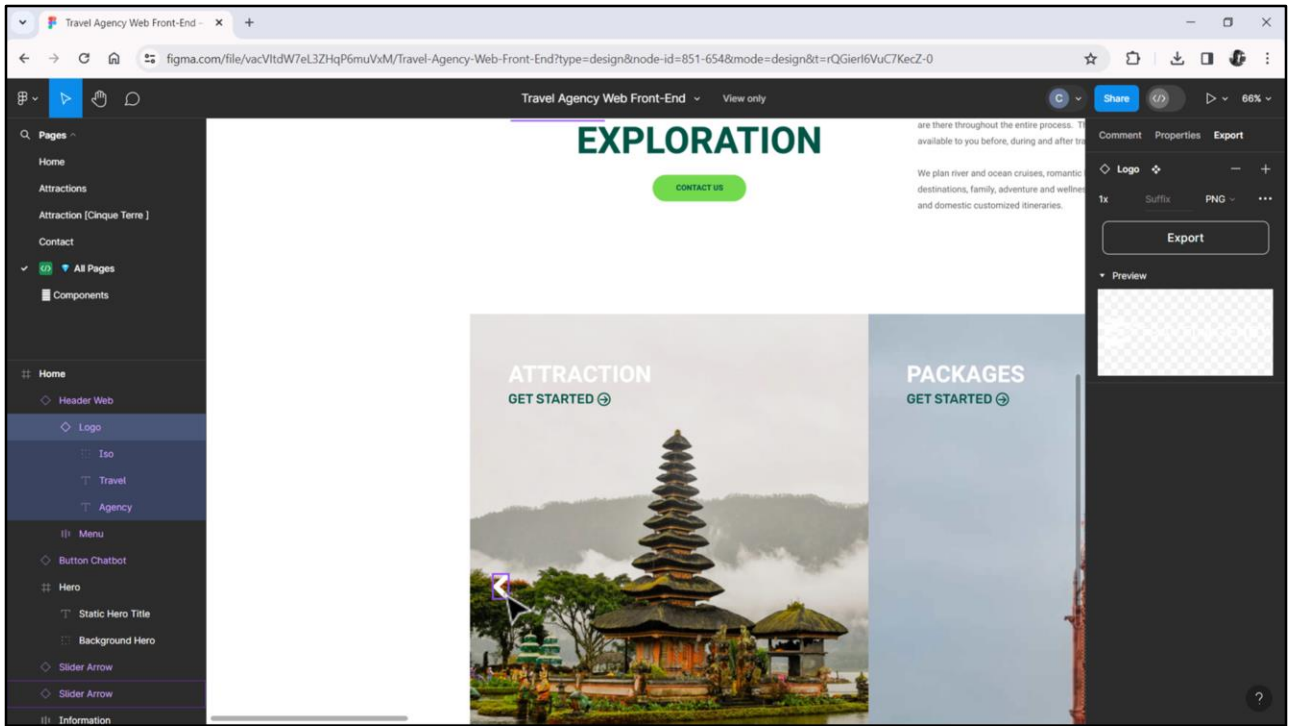
Bueno, podría descargarlo todo entero: el iconito con las dos palabras, exportando desde acá (bueno, vemos que es un componente).



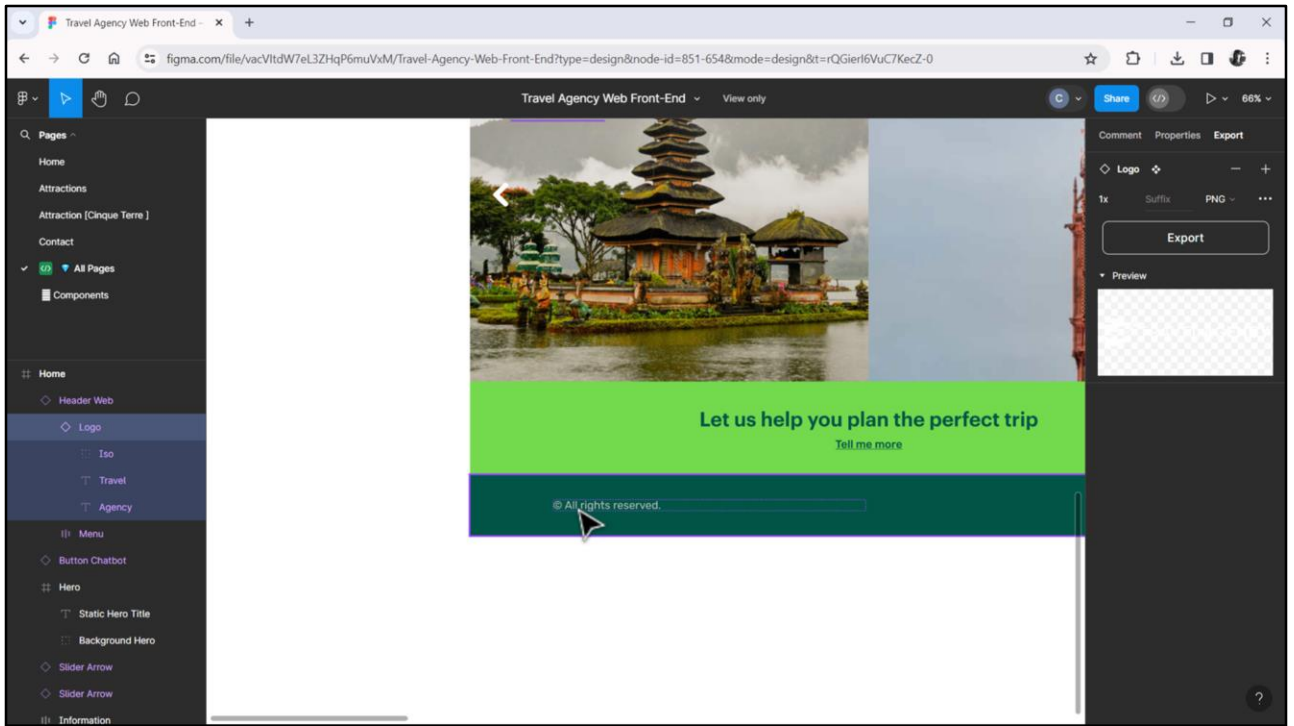
O podría solamente exportar el iconito propiamente dicho, que ven que también Chechu lo armó con estos elementos de acá. Es un poco la misma idea que la del chatbot, y entonces puedo descargarme solamente eso, que es lo que hice, e igual que en el caso del chatbot, presionar acá y entonces me va a descargar ese ícono como SVG.



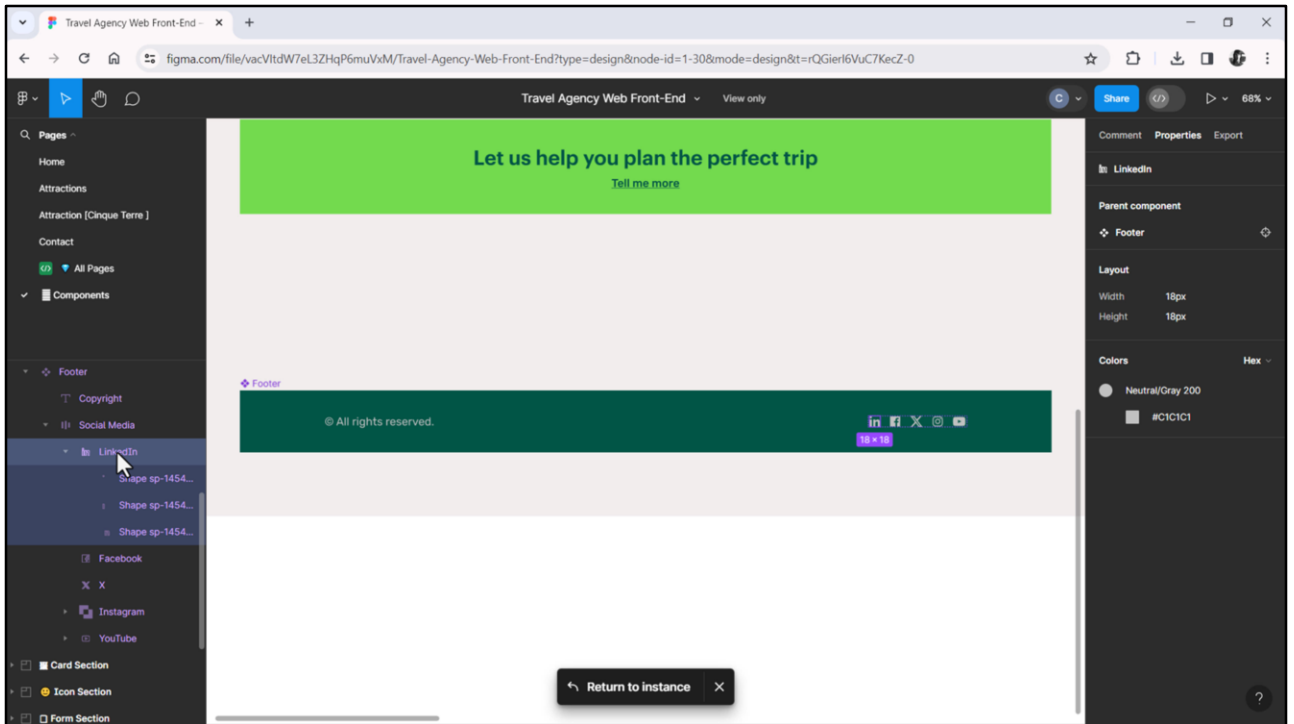
Y bueno, y así sucesivamente. Les voy a mostrar algún otro. Algunos que sea un poquito diferente. Vean que tenemos esta flechita de acá, por ejemplo...



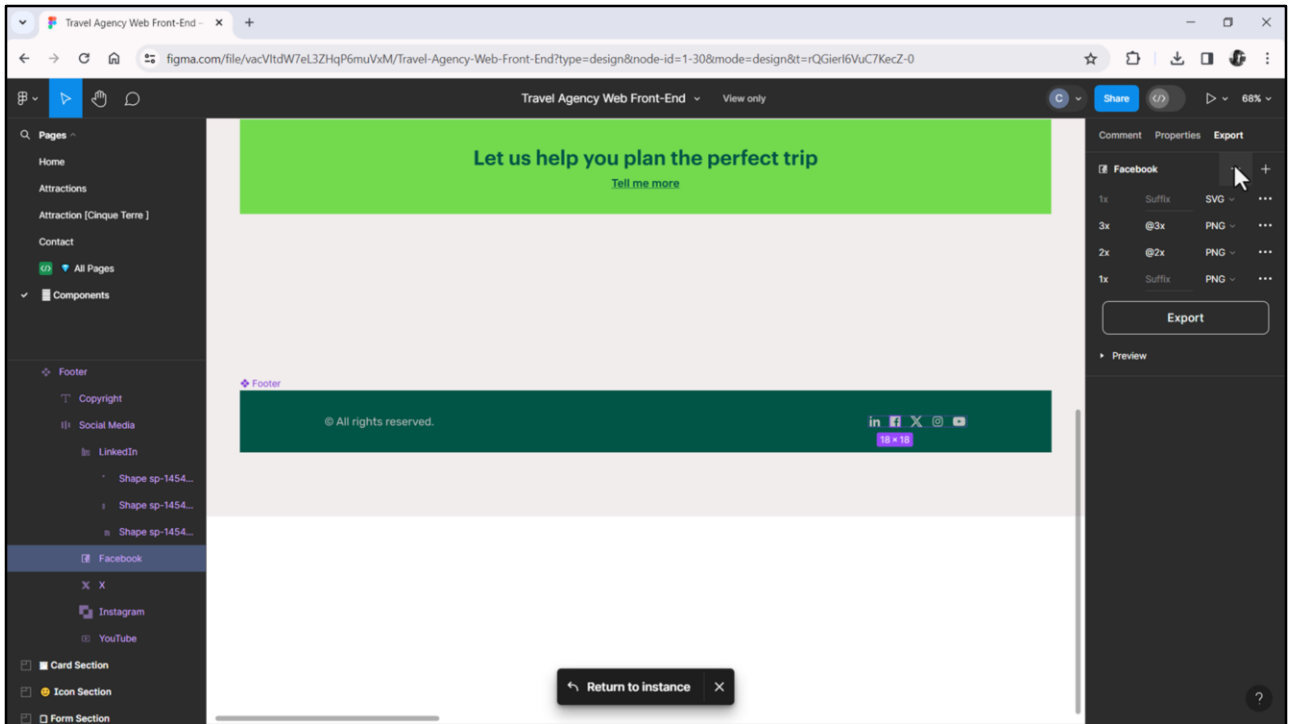
...tenemos estas flechas.



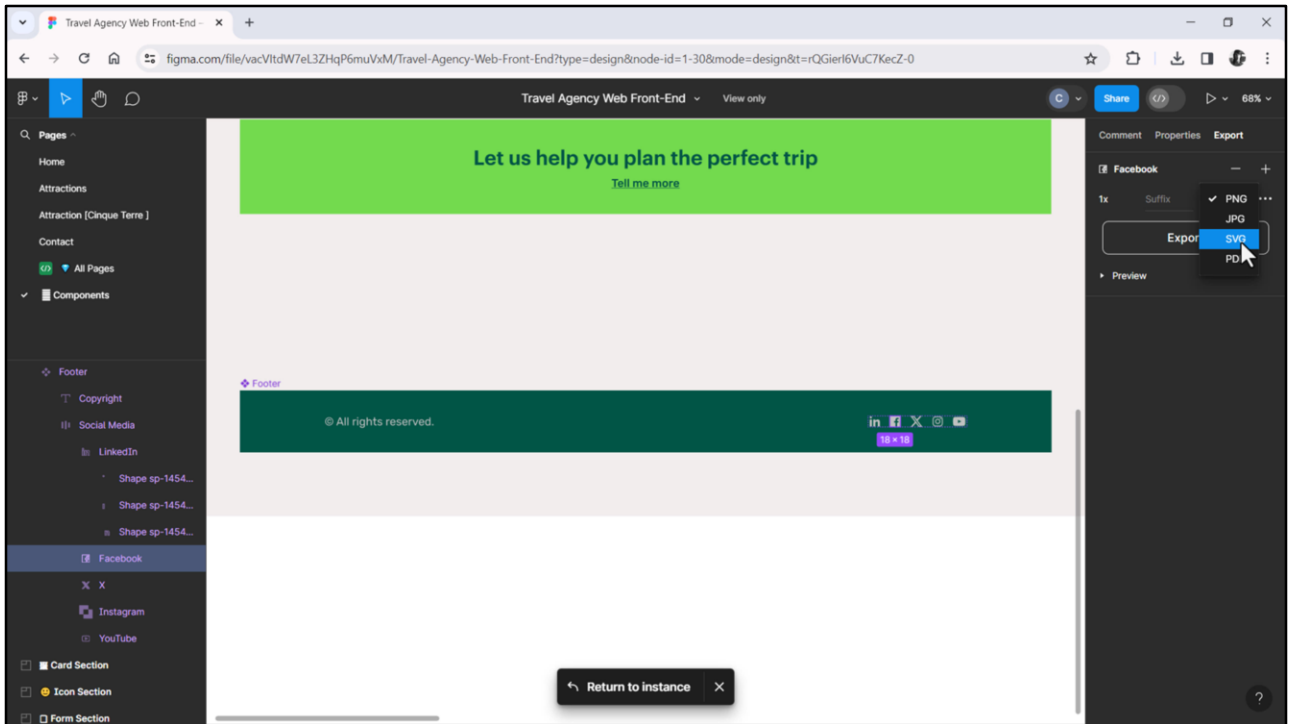
Esto es texto...



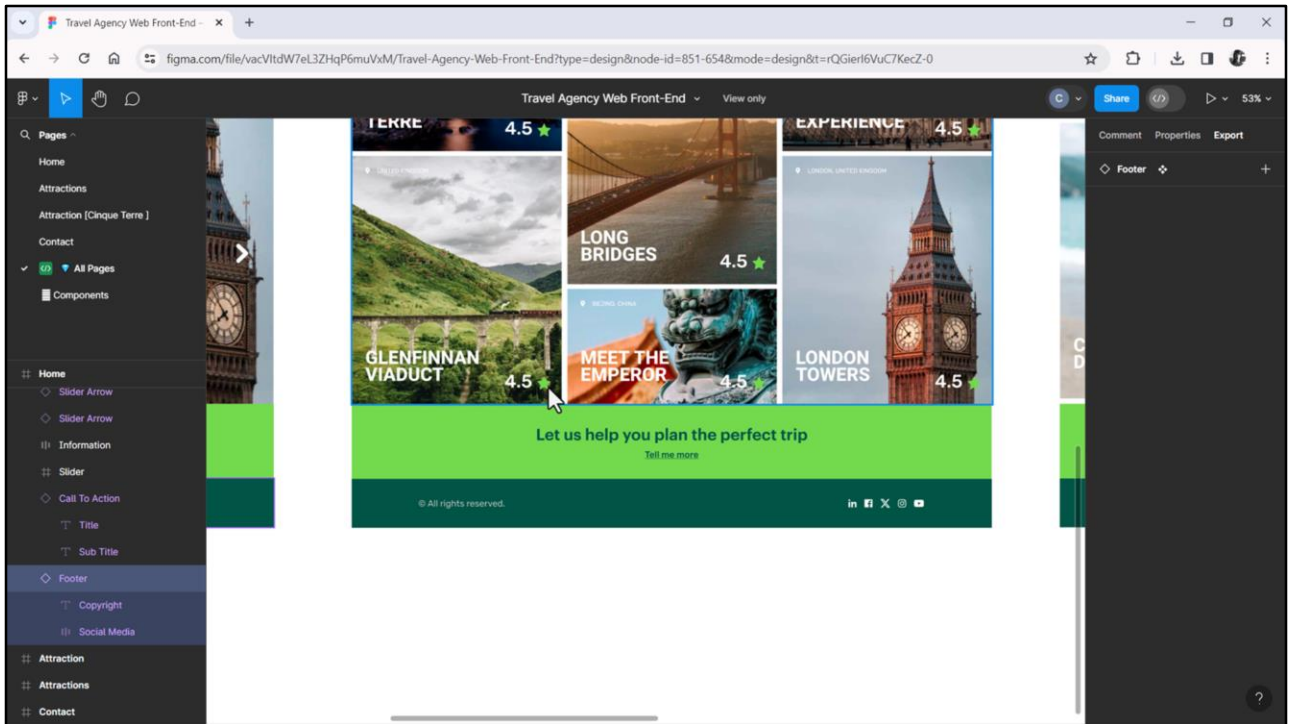
...pero acá tenemos por ejemplo los íconos de las redes sociales, que es un componente, está en el footer. Vamos hacia acá, y lo vemos acá, ¿no? Tenemos entonces... este es el de LinkedIn...



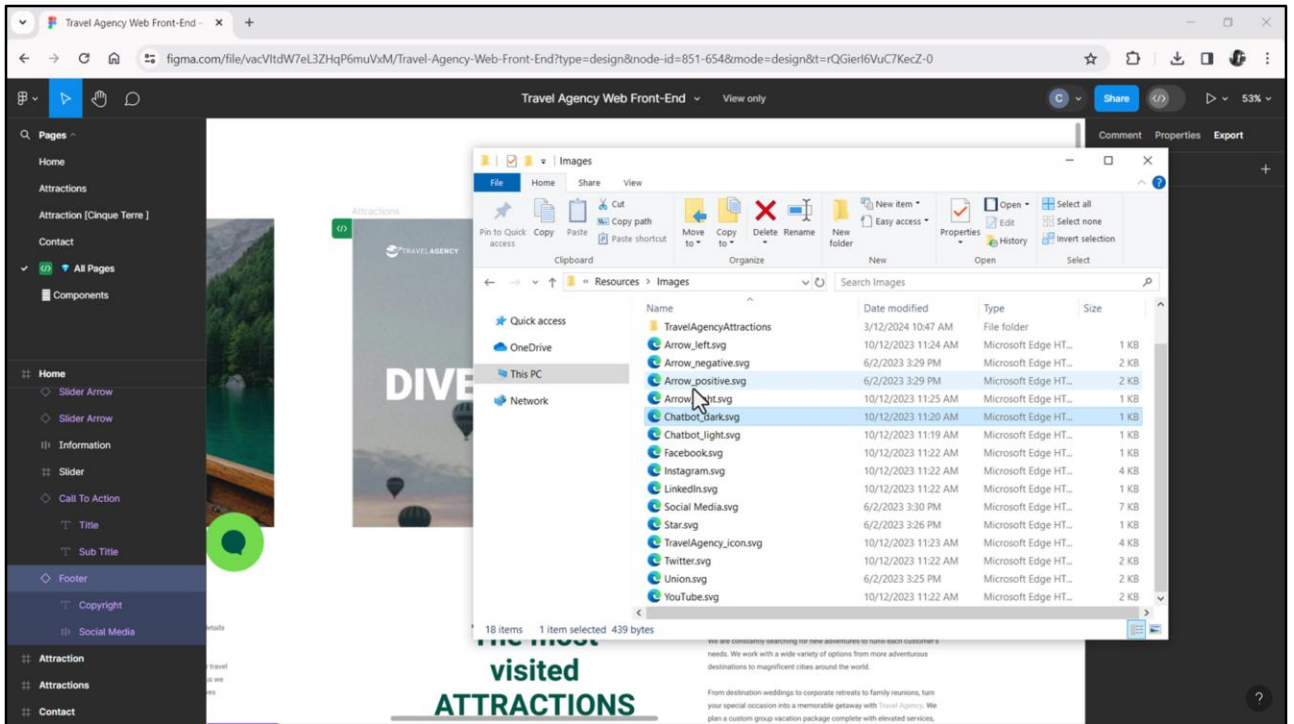
...este es el de Facebook... y bueno, misma cosa. Bueno, acá ven que tiene entradas para las distintas densidades.



Las voy a quitar, para que me quede solamente una, que va a ser el formato SVG y exporto entonces el ícono en ese formato.

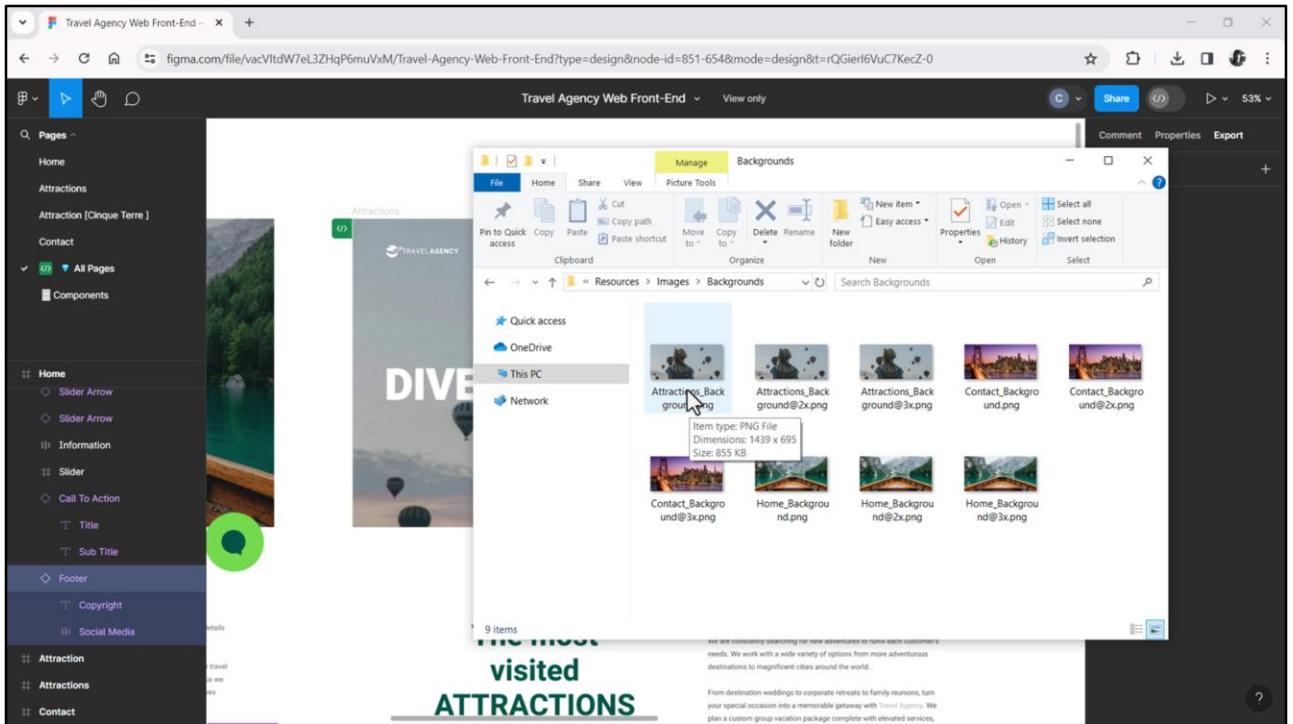


Y bueno, así seguiríamos. A ver si hay algo más que valga la pena mencionar... la estrellita ya la habíamos visto... bueno, pero así sería, ¿no?, tendríamos que investigar bien las páginas para descargar todos esos íconos para ya tenerlos ahí disponibles.



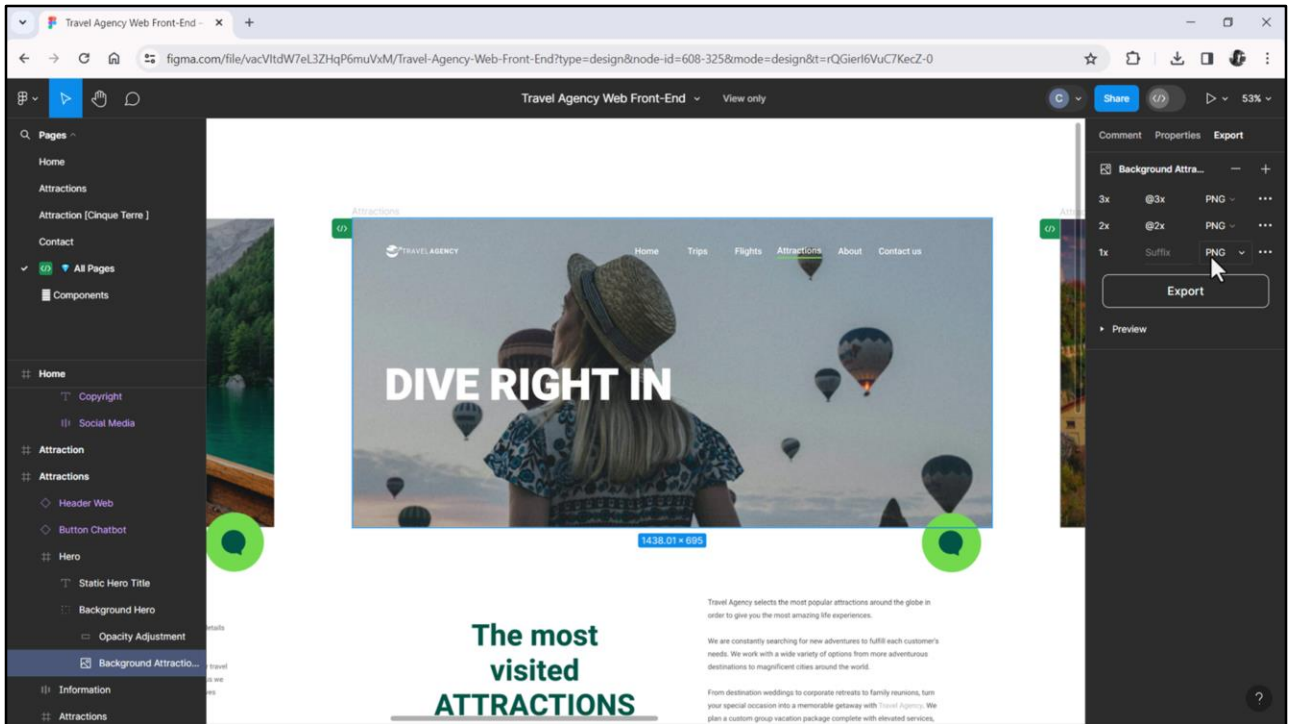
Y eso es lo que tengo, entonces, acá.

Bien, pero por otro lado, decía, hay imágenes que no van a ser SVG porque no son imágenes simples sino que son imágenes complejas.



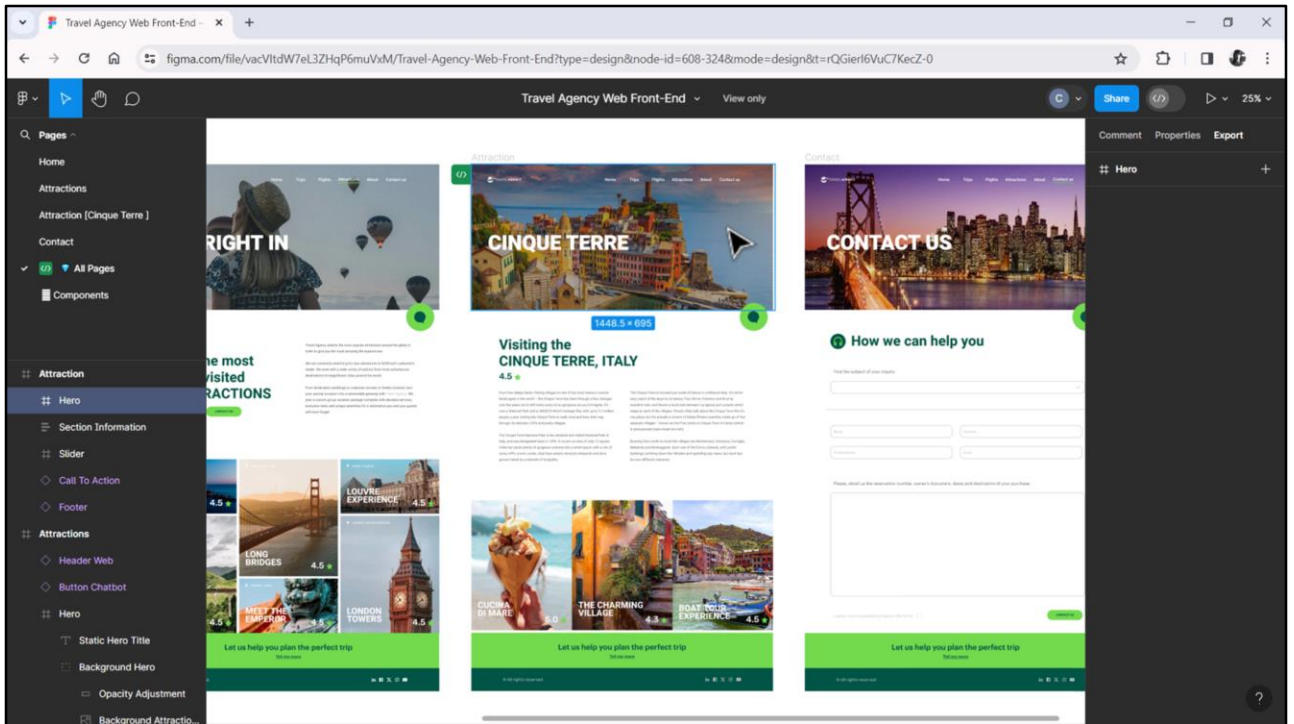
Por ejemplo, vemos aquí estas imágenes, que son imágenes Hero, es decir, las que van a ser parte del Header de cada página.

Como ven, me descargué varias versiones de cada imagen. ¿Por qué? Porque conviene tener versiones por densidad para que, si el dispositivo es de baja resolución, no pierda tiempo bajando imágenes más pesadas. Y si es de alta resolución, no se vea pixelado. Entonces dependiendo de la resolución del dispositivo pueda elegir cuál de las imágenes según su densidad utilizar. Entonces en este caso no vamos a descargar las imágenes en formato svg, sino en algún otro formato.

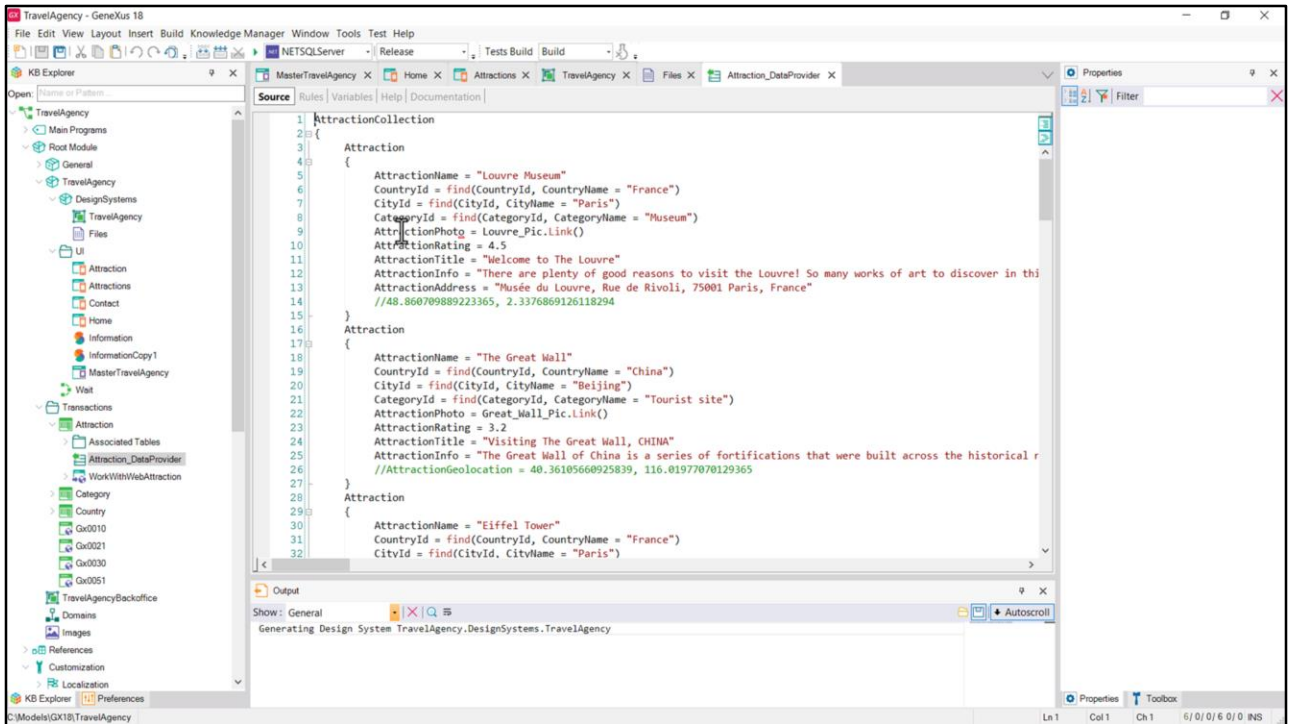


Por ejemplo, veamos acá que tenemos el elemento Hero, del cual queremos solamente la imagen. Esta de aquí. Y acá vemos que ya tenemos la posibilidad de descargarla con densidad 1x, 2x, 3x, con formato png. Entonces, si hacemos Preview nos aseguramos de que es la imagen que queremos, justamente, descargar. Podemos quitar y agregar opciones a la imagen. Por ejemplo acá tenemos otra más. La voy a quitar porque no la necesitamos. Bien, tenemos entonces las tres posibilidades y lo que hacemos es exportar, y vean que nos exporta las 3 imágenes dentro de un zip, con las 3 densidades.

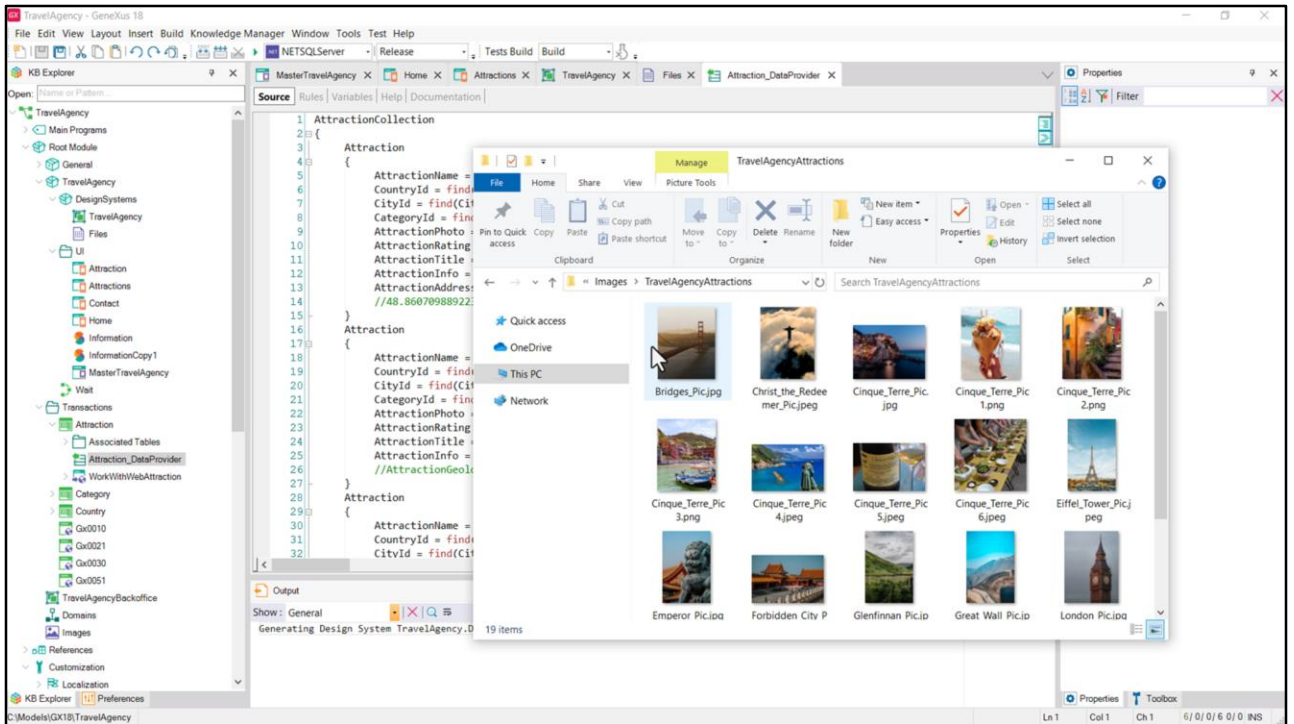
Bueno, y haríamos lo mismo para descargar esta imagen. Y también esta otra imagen.



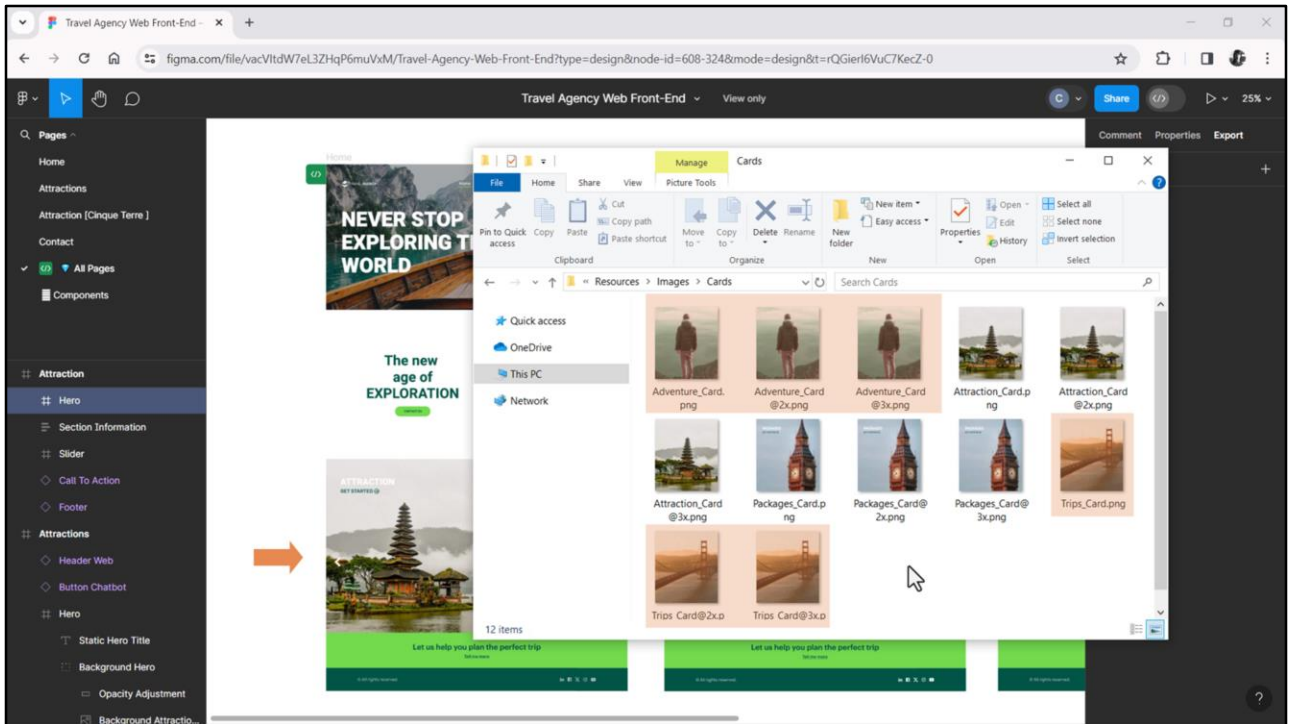
¿Por qué esta, la de Attraction, no? Porque esta imagen de aquí, Hero, va a ser tomada de la base de datos. Va a ser la imagen, la foto, de la atracción turística que se ha seleccionado desde este otro panel.



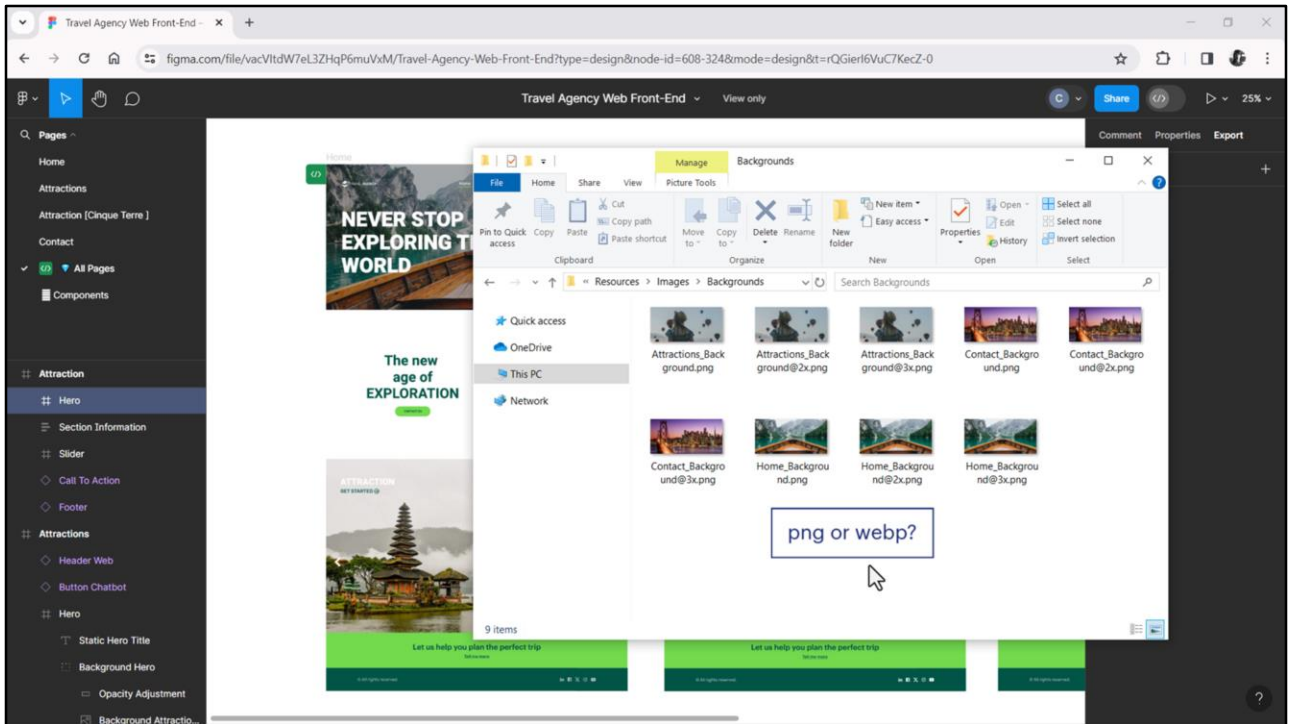
Recuerden que estas imágenes son cargadas en la base de datos a partir del Data Provider que tenemos en el Backoffice, justamente para cargar los datos de la transacción Attraction. Ven que está la foto entre los datos.



De todos modos yo tengo esas imágenes también aquí, las que cargó este Data Provider.



Y por otro lado también tengo las imágenes estas, Cards, que ¿a qué imágenes corresponden? A las que no son tomadas de la base de datos. Estas son de la base de datos, estas son de la base de datos... son estas de acá, que son 4, porque tenemos estas dos, y observen que esto es un carrusel, hay 2 más. Bueno, pero las descargué todas siguiendo el mismo criterio, ¿verdad? Tengo 3 versiones por cada una, para cada una de las resoluciones. Bueno, claro, acá tendríamos que hablar con Chechu porque yo me descargué las imágenes que ahora no se están viendo de un archivo anterior que me había pasado Chechu en el que se veían esas otras 2 imágenes, que acá no las estamos viendo. Entonces, bueno, o le preguntamos a la diseñadora qué pasa con esas imágenes para que nos las envíe si es que no las tenemos modeladas en el archivo o le pedimos que nos deje visibles estas imágenes.



Bien, entonces ya tengo en el folder Images aquí imágenes, aquí imágenes y aquí los íconos de la aplicación, es decir, ya tengo acá todos los assets que voy a necesitar dentro de la KB. Los íconos como imágenes svg y las imágenes más complejas como pngs.

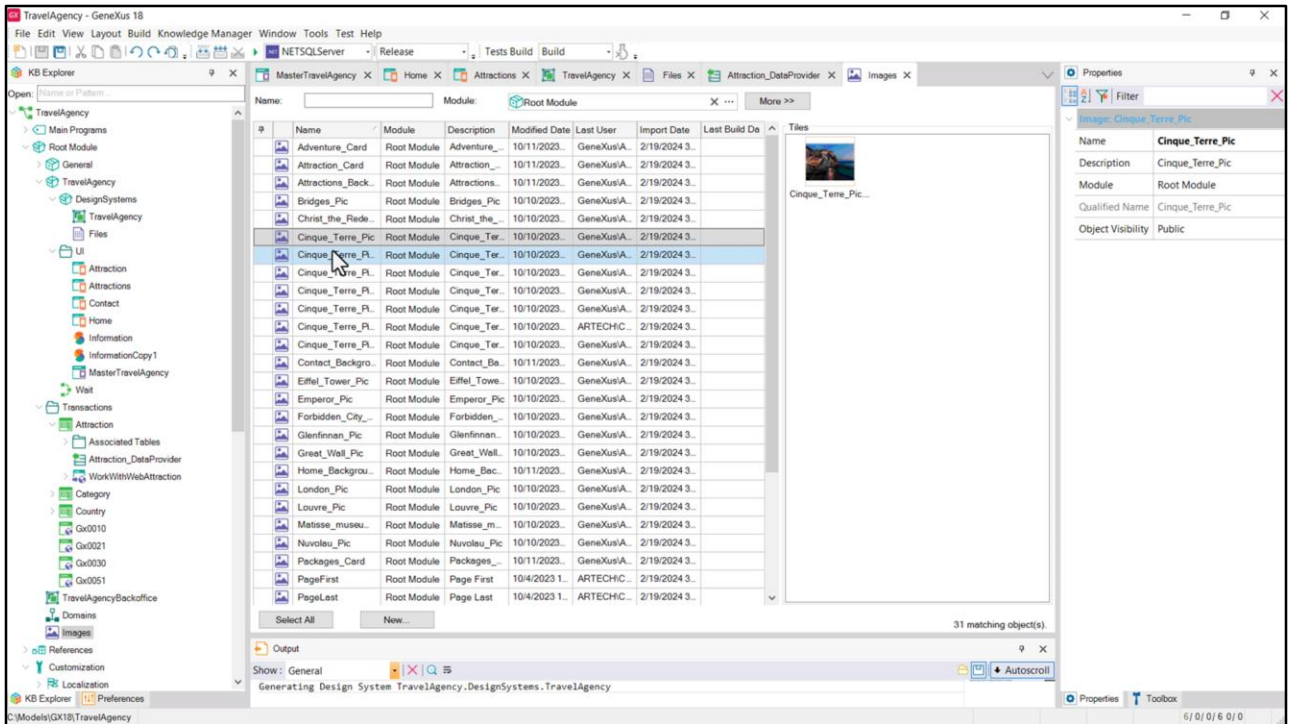
Por ahora como pngs. ¿Por qué?

En realidad resulta mucho más conveniente que el formato png el webp, dado que se logra la misma calidad que con el png pero es un formato más comprimido, por tanto pura ventaja podría uno pensar. Si no fuera que GeneXus no admite subir por el momento imágenes a la KB con este formato. La única forma de utilizarlas es si están en la base de datos o se consumen de servicios.

En cambio la versión Web de GeneXus sí soportará webp, pero la versión Win, que es en la que estamos no. Si estuviéramos utilizando la versión Web de GeneXus, lo que haríamos sería utilizar algún conversor online de formato png a webp, porque vieron que Figma no me ofrecía ese formato para la descarga. Y algo que no dije y es importante: es soportado por las aplicaciones nativas.

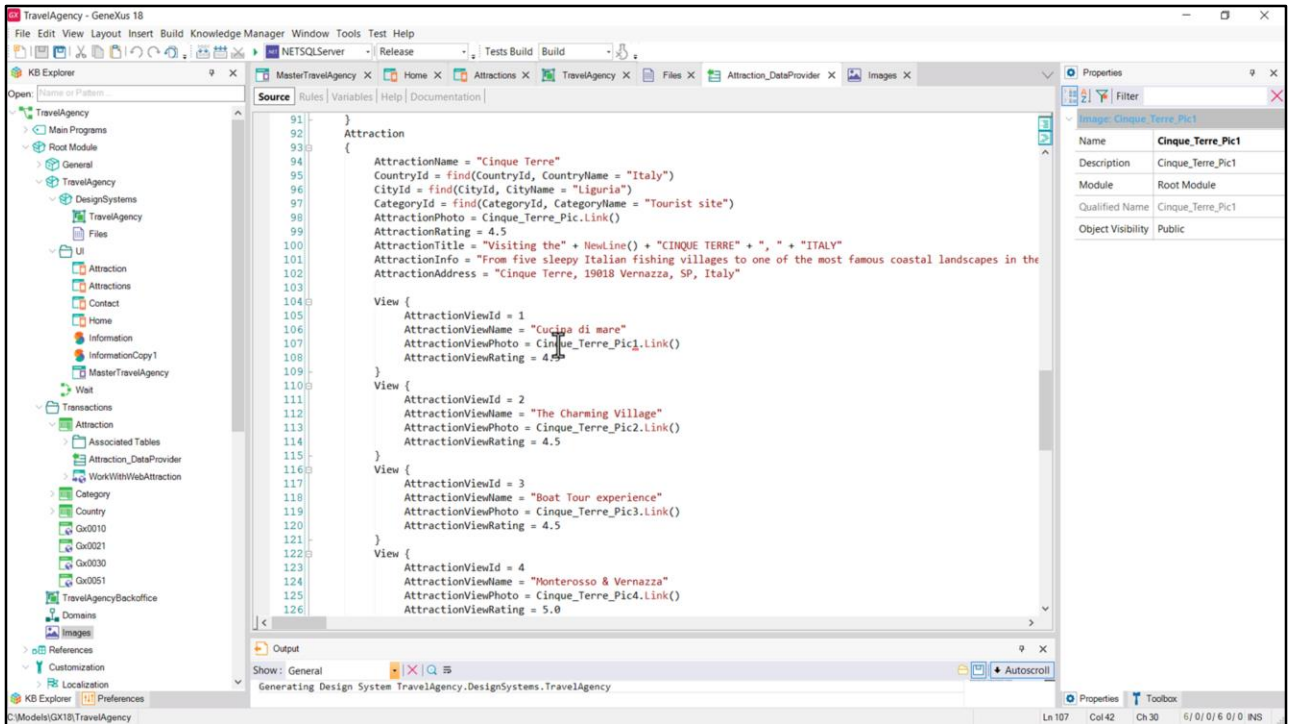
Pero bueno, nos quedamos por ahora en el escenario en el que estamos. No tenemos otra que utilizar el formato png.

Bien, entonces, paso siguiente, ¿qué nos queda? Insertar todas estas imágenes e íconos que recabamos del diseño como assets en nuestra KB.

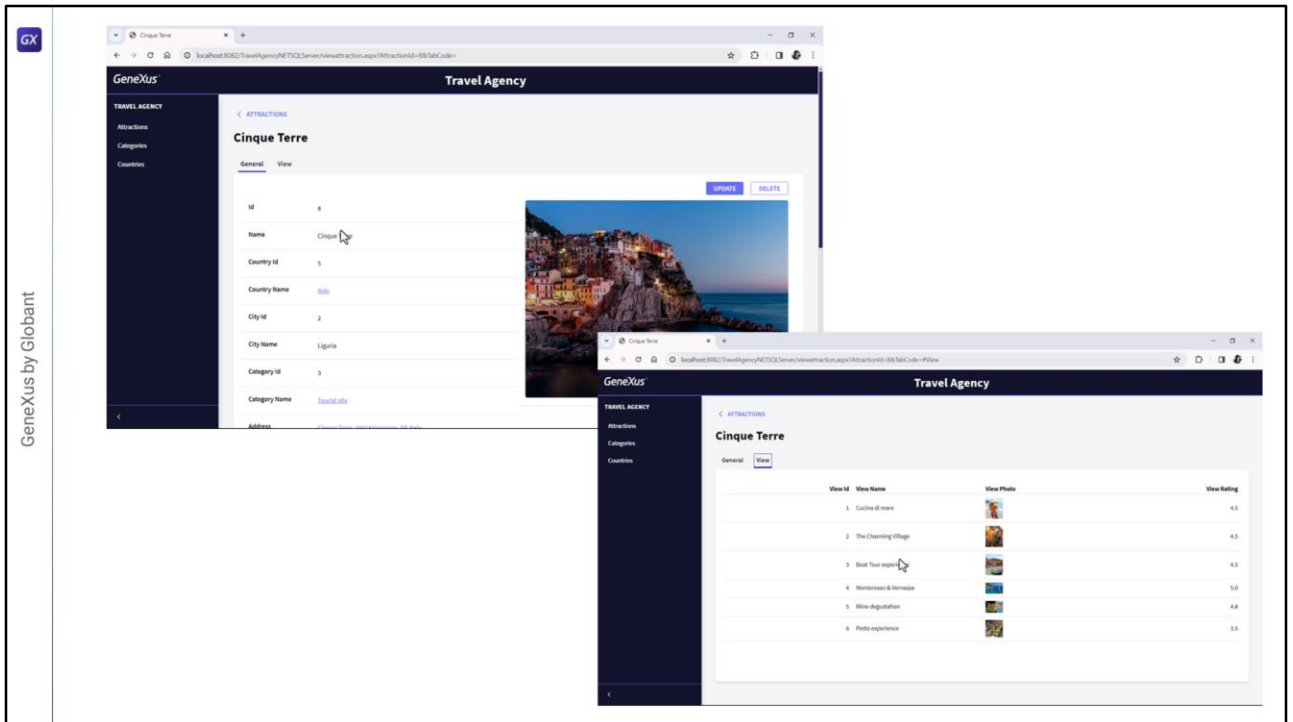


Yo las tengo insertadas desde el principio mismo, incluso desde antes de haber creado el módulo TravelAgency, por lo que están asociadas al Root Module.

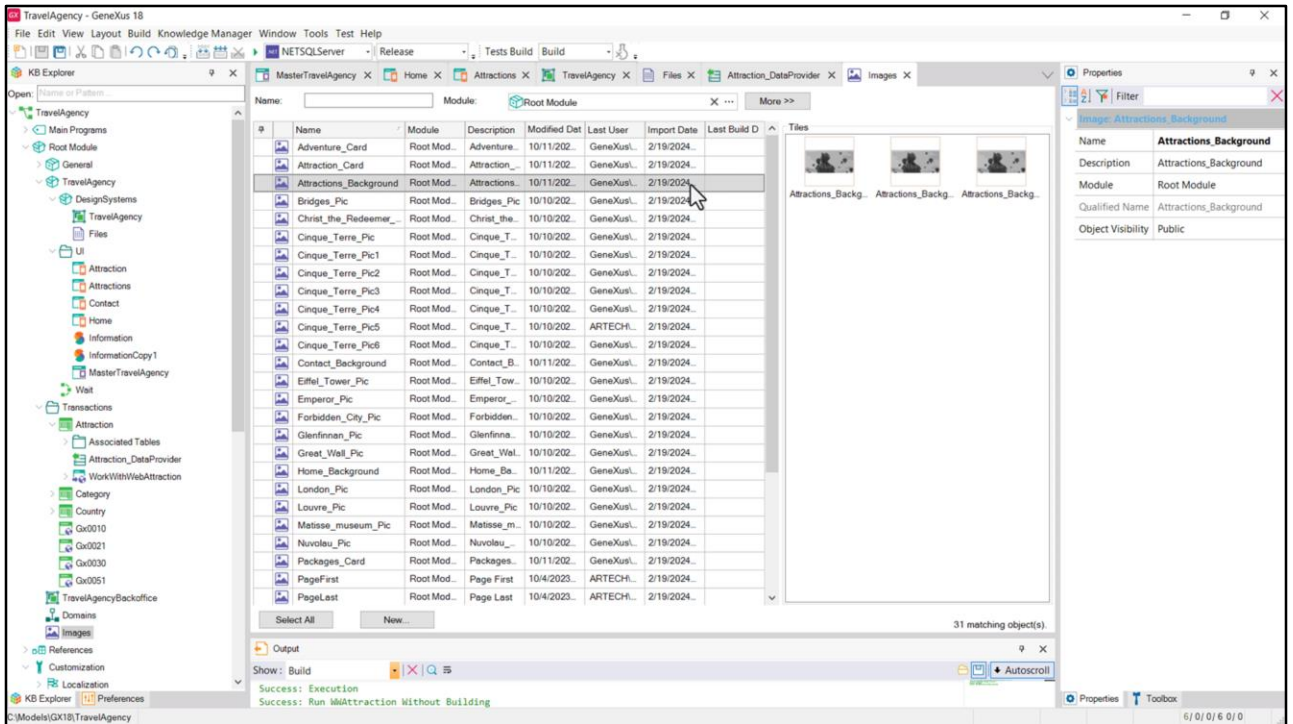
Todas corresponderán a objetos GeneXus de tipo Image. Por eso si me posiciono aquí, miren todo lo que aparece. De hecho estas son las imágenes que inserté en la KB cuando implementé el Backoffice... que uds. no lo vieron porque yo ya lo mostré hecho. Veamos esta, por ejemplo... y estas otras, que son las de las opciones de esa atracción turística...



Las utilicé en el Data Provider para poblar con datos la transacción Attraction inicialmente.

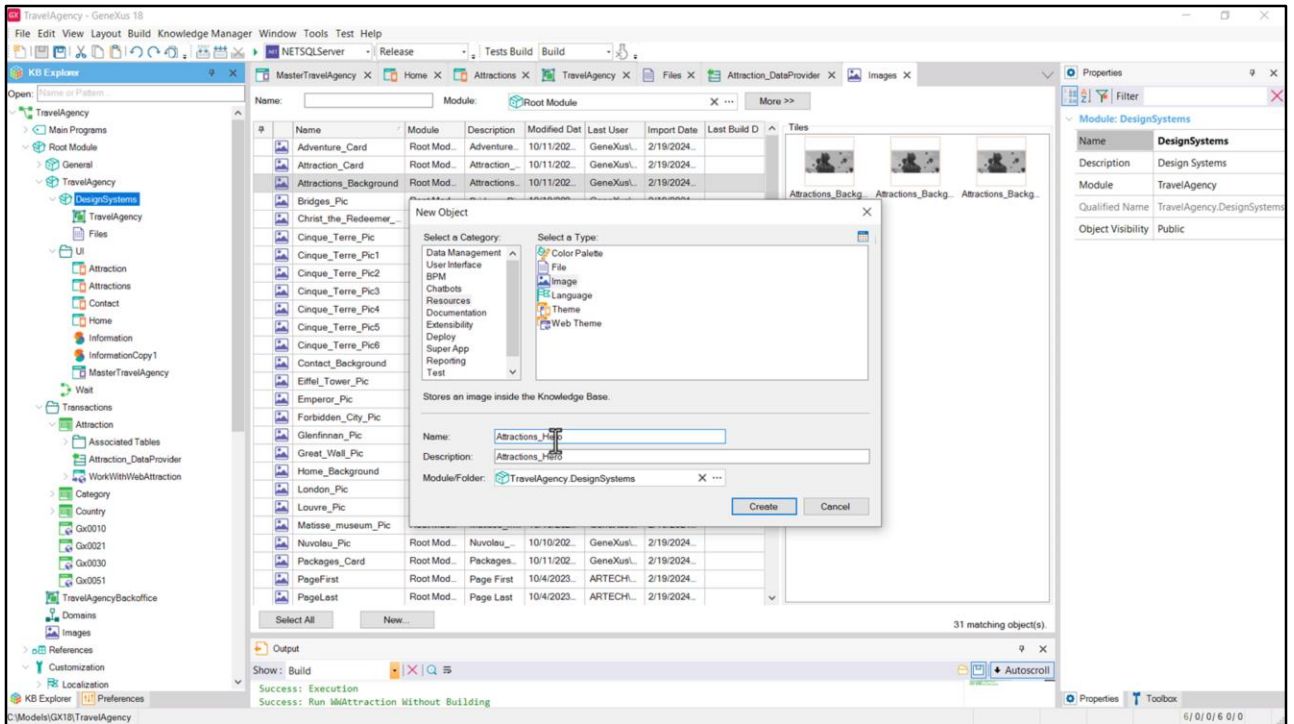


Acá las vemos en el Backoffice ya cargadas.

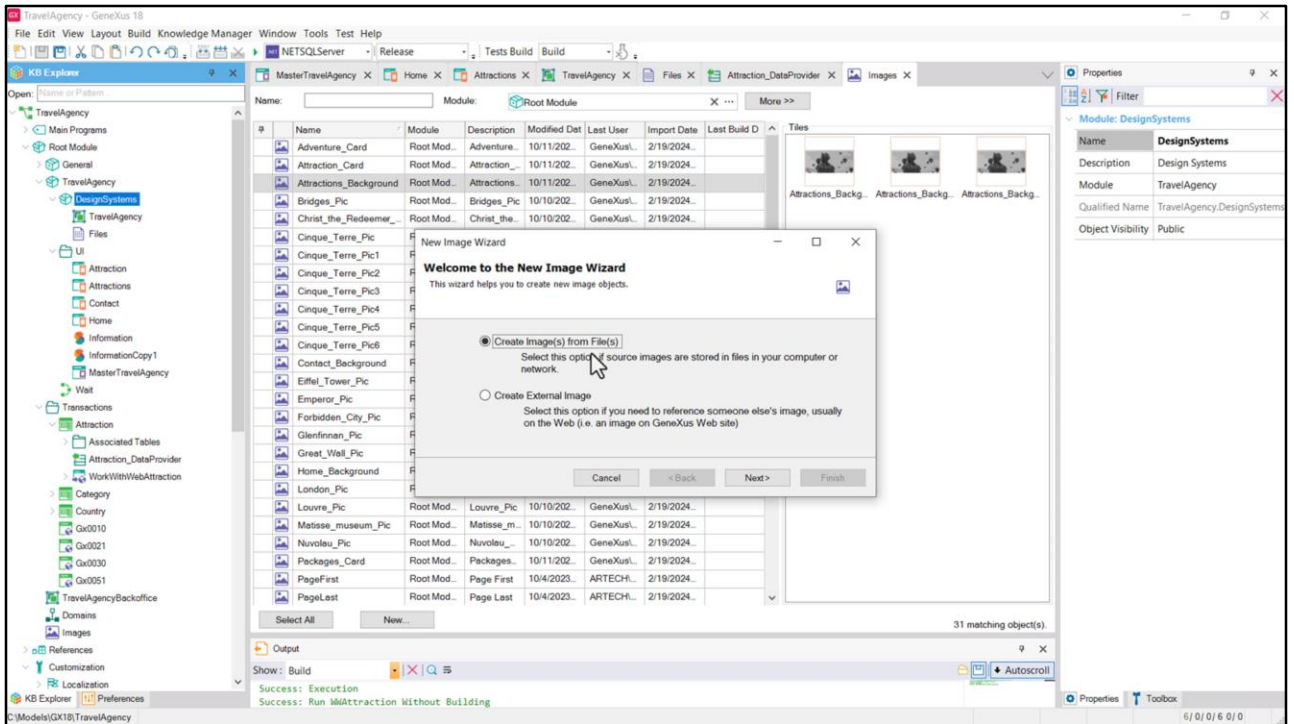


Pero en aquella etapa no solamente cargué las que utilizaría en la base de datos, como estas que terminan en Pic, sino que también cargué varias de las que veníamos viendo de nuestro diseño.

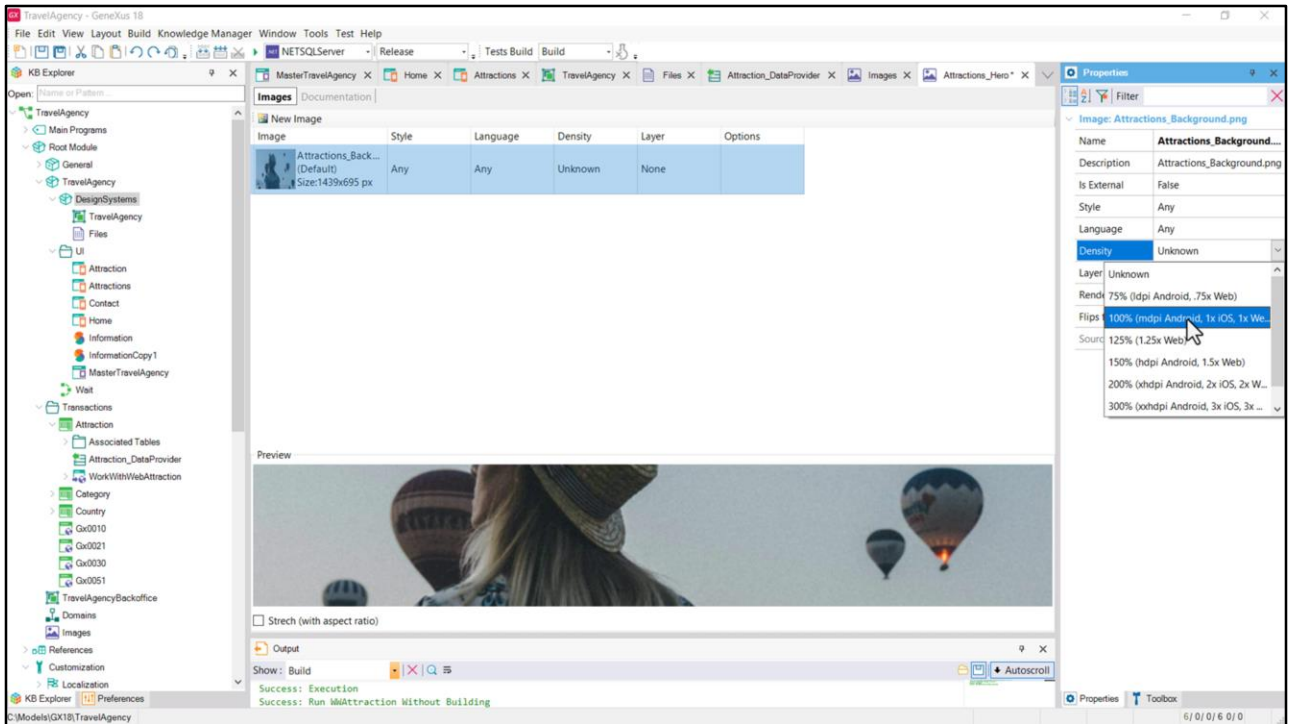
Bueno, pero para mostrarles cómo lo hice voy a elegir una, por ejemplo la del background de Attractions, esta, para mostrarles cómo se cargarían esos tres archivos dentro del mismo objeto de tipo Image. La mayoría de uds. esto ya lo sabe pero para los que no, vamos a contarlo.



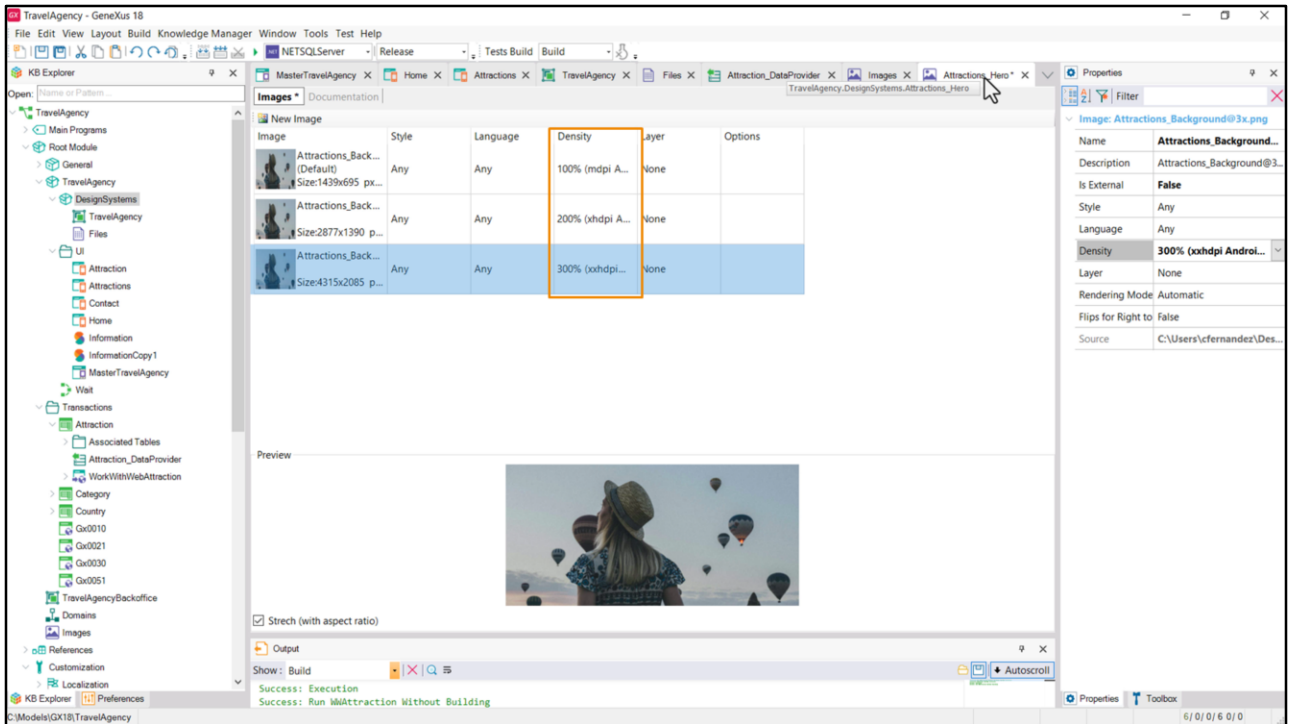
Como ya estoy a nivel del módulo voy a insertar la imagen Attractions Background, o Hero, o como sea que le quiera llamar, creando en el módulo DesignSystems un objeto GeneXus de tipo Image. Y le voy a dar por ejemplo, Attractions_Hero de nombre, en vez de background.



Y aquí nos pregunta si queremos crear la imagen a partir de archivos o de una fuente externa como un sitio web.



Y aquí vengo e ingreso por ejemplo el archivo que corresponde a la densidad 1x. Entonces lo que hago a partir de ahí...vemos que aparecen estas propiedades, y le coloco la densidad 1x.

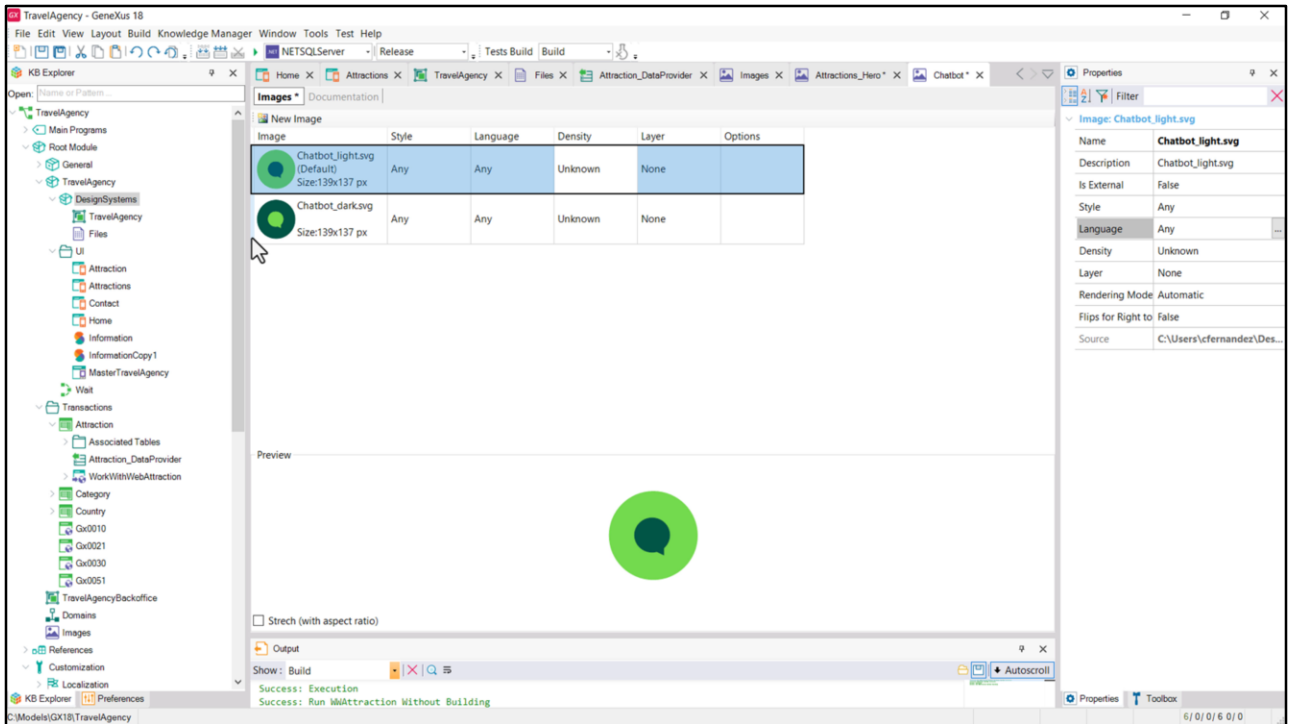


Y después lo que hago es agregar las otras 2 imágenes... la de 2x, le coloco la densidad 2x, y también la 3x, ¿sí?

Vemos que aparecen una serie de propiedades, entre las que está la Density, pero hay otras.

Hay que tener en cuenta que el objeto Imagen es uno solo, es este, de nombre Attractions_Hero, pero cuando se lo quiera utilizar en algún lugar de la KB, dependerá de estas dimensiones cuál de las imágenes concretas se termine utilizando. En este caso, por ejemplo, la única diferencia será en la densidad, porque las tres imágenes aplican para cualquier estilo, es decir para todo DSO de la KB, para todo lenguaje y sin especificación de layer. Es decir que se utilizará una, otra u otra de acuerdo a la resolución del dispositivo que esté ejecutando la aplicación.

Ahora, yo podría hacer variar la imagen también por DSO, o por lenguaje si tengo textos a nivel de la imagen por ejemplo.

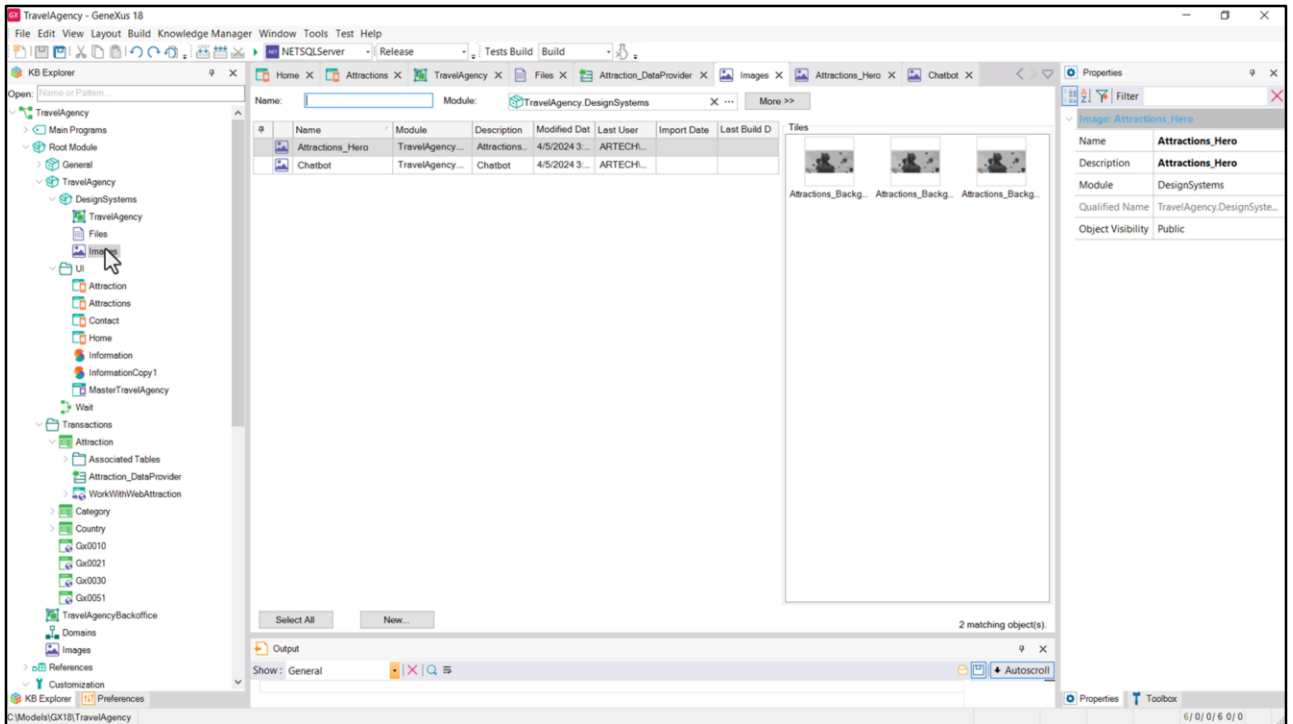


Voy a ingresar ahora la imagen del chatbot. Esta imagen, justamente por ser SVG, no requiere variación por densidad. Entonces nos quedará así, con un único archivo válido para cualquier circunstancia. Aunque, lo voy a dejar introducido, no va a ser fijo en realidad. Va a variar esta imagen del chatbot de acuerdo al modo: si es light o dark.

Nosotros todavía no hemos visualizado cómo debería quedar el diseño de la aplicación con el modo dark. Chechu ya lo hizo pero en otro archivo.

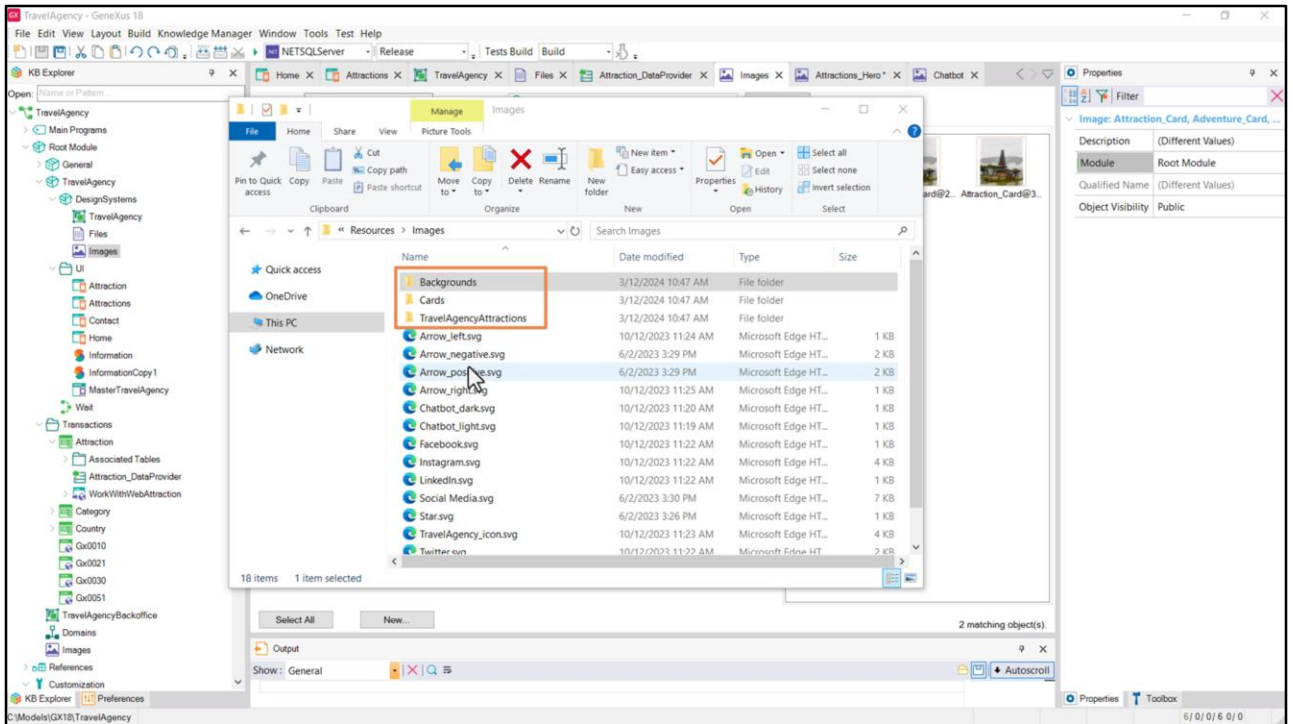
Entonces lo que voy a hacer es solamente para que ya quede expresado aquí, sin preocuparnos ahora, más adelante lo vamos a ver, es agregar otra versión de la imagen, que va a ser para el modo dark. ¿Cómo le voy a poder decir que una es para modo light y la otra para modo dark? Eso va a venir dado por las opciones, que no las tenemos habilitadas aún porque para que se nos habiliten vamos a tener que especificar que vamos a utilizar esos dos modos, ¿a nivel de quién?, del DSO.

Entonces ahora voy a dejar el archivo con estas dos opciones, voy a grabar. ¿Y cómo sabrá, entonces, cuando yo inserte esta imagen en alguno de nuestros paneles, cómo va a saber – bueno, de hecho va a ser en el Master Panel–, cómo va a saber cuál de estas dos utilizar? Bueno, como las dimensiones son las mismas va a utilizar la primera que encuentre. Bueno, voy a grabar esta otra.

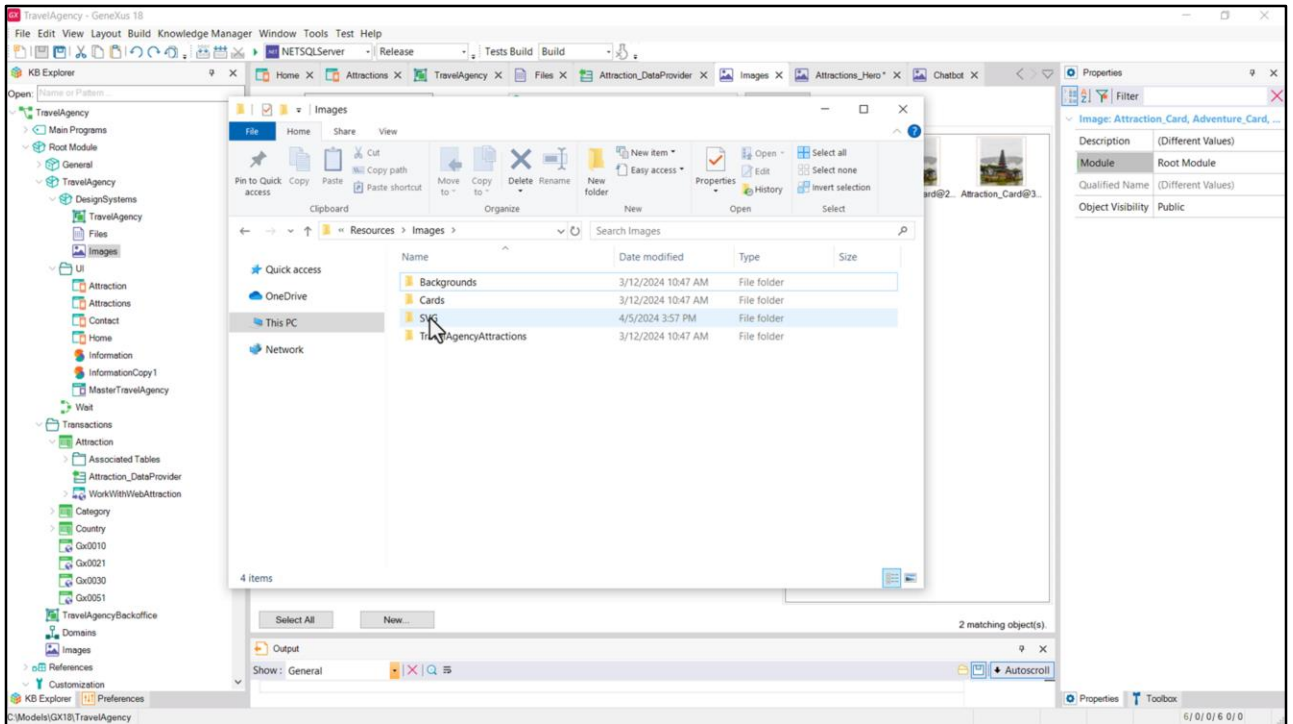


Ahí vemos que por colocarlos dentro del módulo Design Sytems nos aparece este nodo acá, separado de este de acá que era el del Root. Y las dos imágenes que creamos.

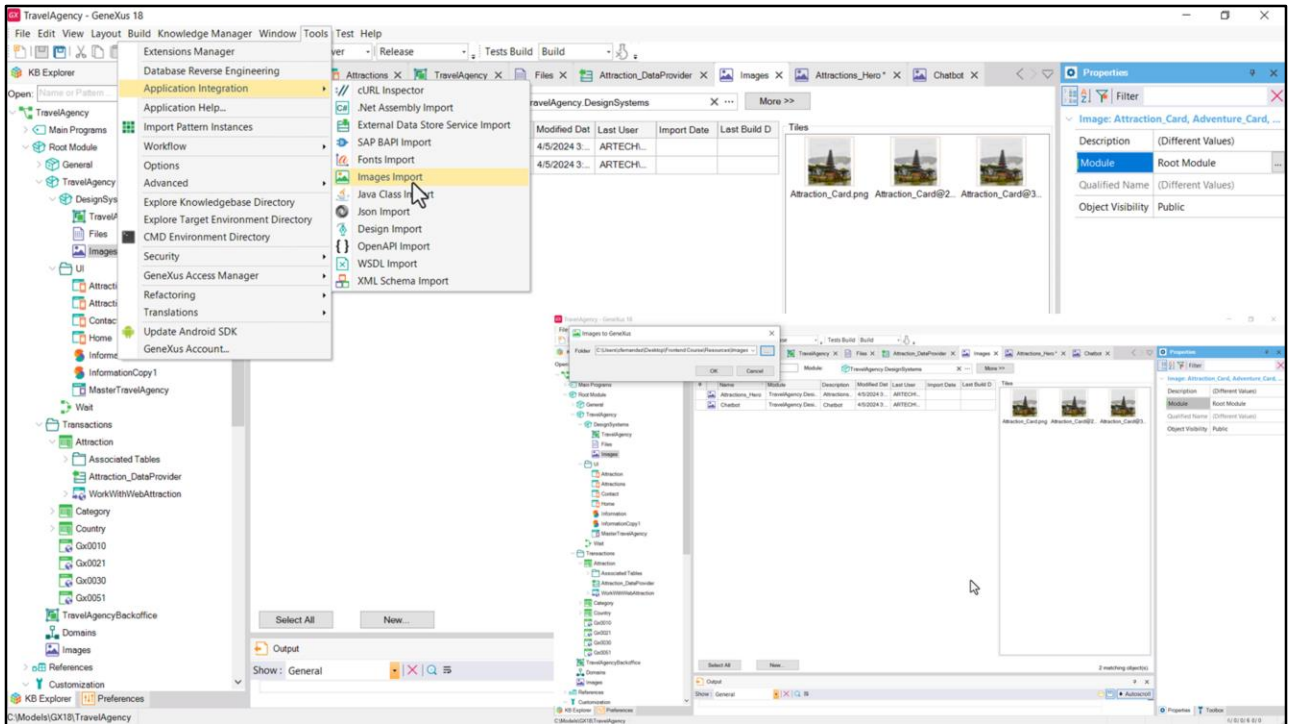
Bueno, trabajando así introduciríamos todas las imágenes. También podemos tener en cuenta que tenemos una herramienta para ello, que es esta de acá. Que nos permite importar las imágenes de un folder.



Así, por ejemplo, para insertar todos estos SVGs, teniendo en cuenta que no los tenía previamente insertados en la KB (tenía insertadas las imágenes, las más complejas, las pngs, pero no los SVGs).



Bueno, entonces, teniendo en cuenta esto, voy a crear un folder, para que me quede más ordenado, SVGs, y allí voy a mover todos estos elementos.



Y ahora voy a venir acá de vuelta, e importar todo lo de ese folder. Las importó, ¿pero las importó dónde?

The screenshot displays the GeneXus IDE interface for a project named 'TravelAgency'. The main window shows a table of imported images with the following columns: Name, Module, Description, Modified Date, Last User, Import Date, and Last Build Date. The 'Module' column for all entries is 'Root Mod.', which is incorrect as they should be imported into the 'TravelAgency' module.

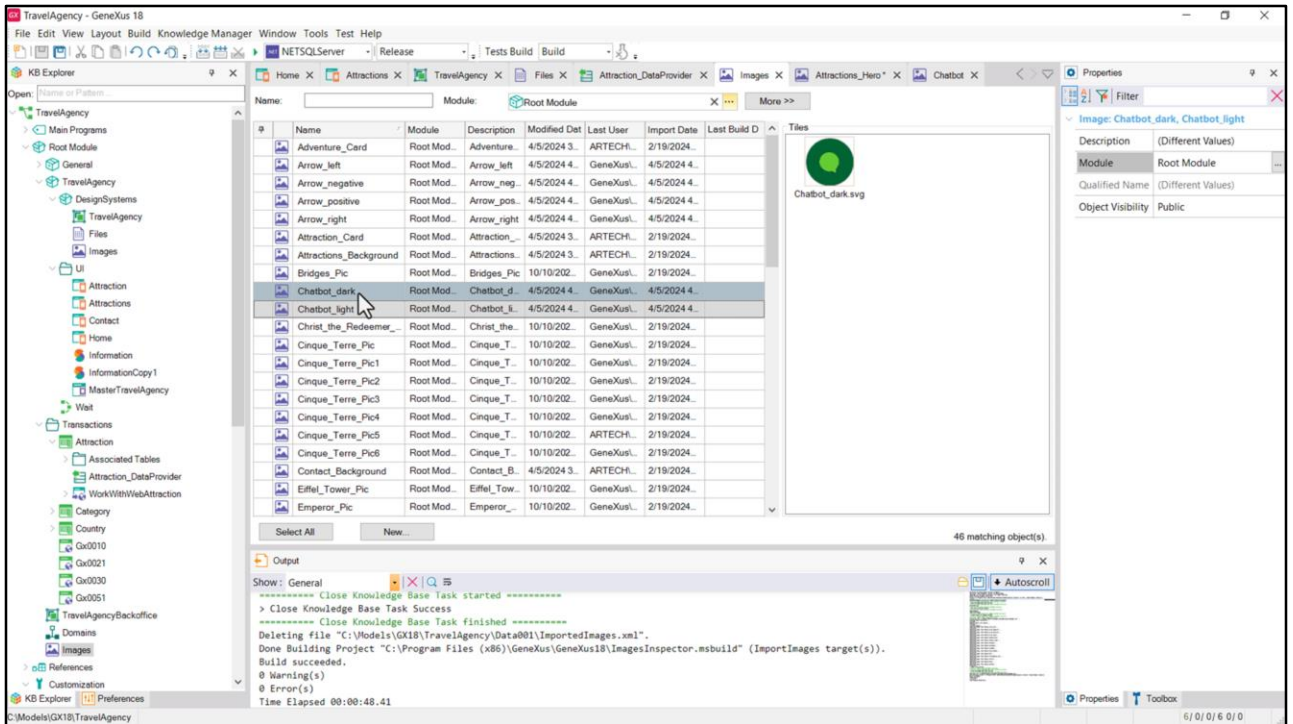
Name	Module	Description	Modified Date	Last User	Import Date	Last Build Date
Adventure_Card	Root Mod.	Adventure_	4/5/2024 3..	ARTECH...	2/19/2024..	
Arrow_left	Root Mod.	Arrow_left	4/5/2024 4..	GeneXusL..	4/5/2024 4..	
Arrow_negative	Root Mod.	Arrow_neg	4/5/2024 4..	GeneXusL..	4/5/2024 4..	
Arrow_positive	Root Mod.	Arrow_pos	4/5/2024 4..	GeneXusL..	4/5/2024 4..	
Arrow_right	Root Mod.	Arrow_right	4/5/2024 4..	GeneXusL..	4/5/2024 4..	
Attraction_Card	Root Mod.	Attraction_	4/5/2024 3..	ARTECH...	2/19/2024..	
Attractions_Background	Root Mod.	Attractions_	4/5/2024 3..	ARTECH...	2/19/2024..	
Bridges_Pic	Root Mod.	Bridges_Pic	10/10/202..	GeneXusL..	2/19/2024..	
Chatbot_dark	Root Mod.	Chatbot_d	4/5/2024 4..	GeneXusL..	4/5/2024 4..	
Chatbot_light	Root Mod.	Chatbot_l	4/5/2024 4..	GeneXusL..	4/5/2024 4..	
Christ_the_Redeemer_	Root Mod.	Christ_the_	10/10/202..	GeneXusL..	2/19/2024..	
Cinque_Terre_Pic	Root Mod.	Cinque_T_	10/10/202..	GeneXusL..	2/19/2024..	
Cinque_Terre_Pic1	Root Mod.	Cinque_T_	10/10/202..	GeneXusL..	2/19/2024..	
Cinque_Terre_Pic2	Root Mod.	Cinque_T_	10/10/202..	GeneXusL..	2/19/2024..	
Cinque_Terre_Pic3	Root Mod.	Cinque_T_	10/10/202..	GeneXusL..	2/19/2024..	
Cinque_Terre_Pic4	Root Mod.	Cinque_T_	10/10/202..	GeneXusL..	2/19/2024..	
Cinque_Terre_Pic5	Root Mod.	Cinque_T_	10/10/202..	ARTECH...	2/19/2024..	
Cinque_Terre_Pic6	Root Mod.	Cinque_T_	10/10/202..	GeneXusL..	2/19/2024..	
Contact_Background	Root Mod.	Contact_B	4/5/2024 3..	ARTECH...	2/19/2024..	
Eiffel_Tower_Pic	Root Mod.	Eiffel_Tow	10/10/202..	GeneXusL..	2/19/2024..	
Emperor_Pic	Root Mod.	Emperor_	10/10/202..	GeneXusL..	2/19/2024..	

The 'Output' window shows the following log:

```

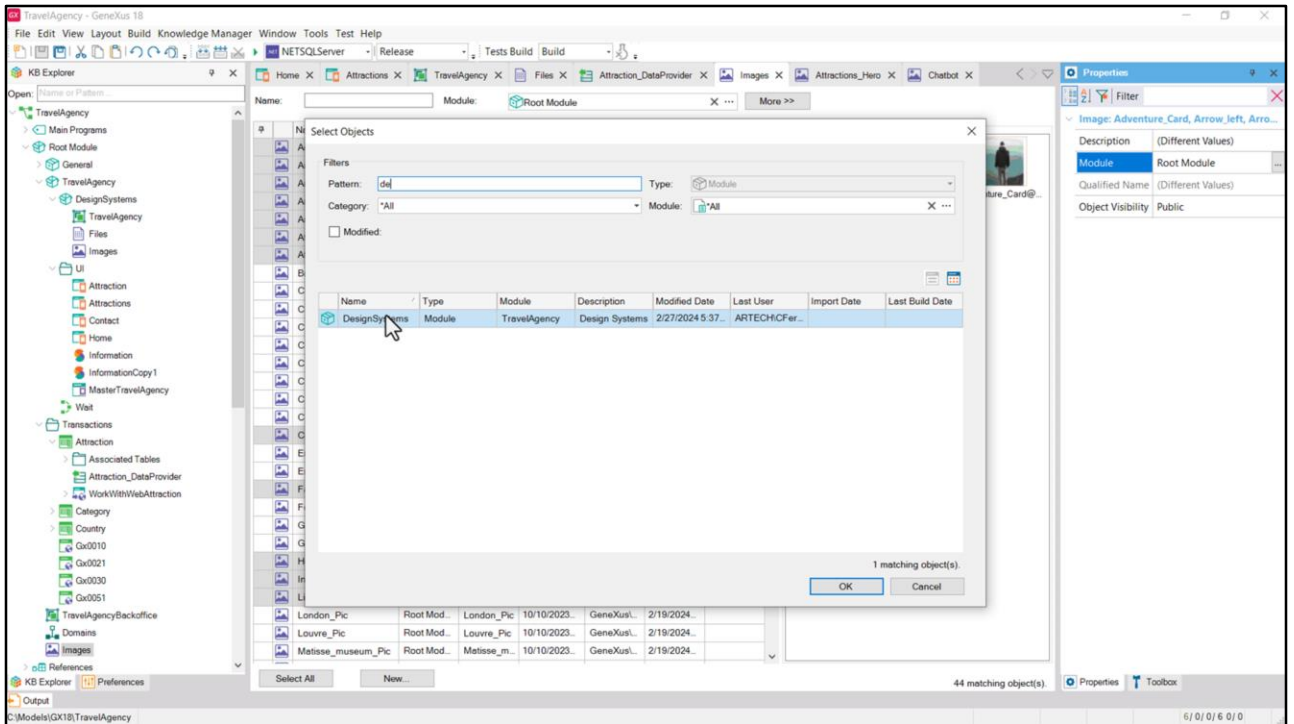
Show: General
----- Close Knowledge Base task started -----
> Close Knowledge Base Task Success
----- Close Knowledge Base Task finished -----
Deleting file "C:\Models\GX18\TravelAgency\Data001\ImportedImages.xml".
Done Building Project "C:\Program Files (x86)\GeneXus\GeneXus18\ImagesInspector.msbuild" (ImportImages target(s)).
Build succeeded.
0 Warning(s)
0 Error(s)
Time Elapsed 00:00:48.41
  
```

No las importó en nuestro módulo, claramente, sino que las importó en el módulo Root. Acá las vemos.



La desventaja de esta importación masiva es esto por ejemplo: que los dos archivos que en realidad deberían haberse importado como una única imagen, se importan como dos objetos imagen por separado.

Bueno, dado que esta ya la creamos nosotros en nuestro módulo Design Systems la podemos eliminar de aquí.



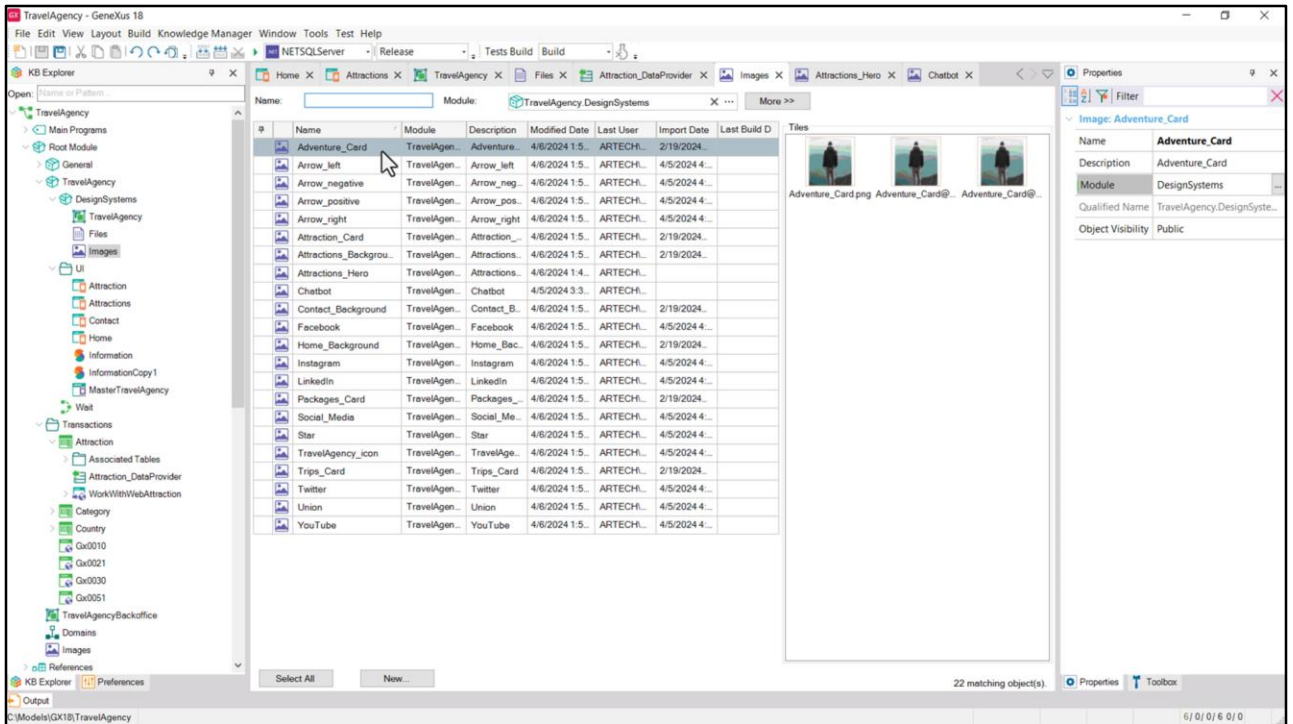
Y ahora lo que voy a hacer para terminar de dejar todos los assets ordenados es mover todas las imágenes que estarán ligadas a nuestro Design System al módulo DesingSystems.

The screenshot shows the GeneXus 18 IDE interface for a project named 'TravelAgency'. The main window displays a list of modules under the 'Root Module'.

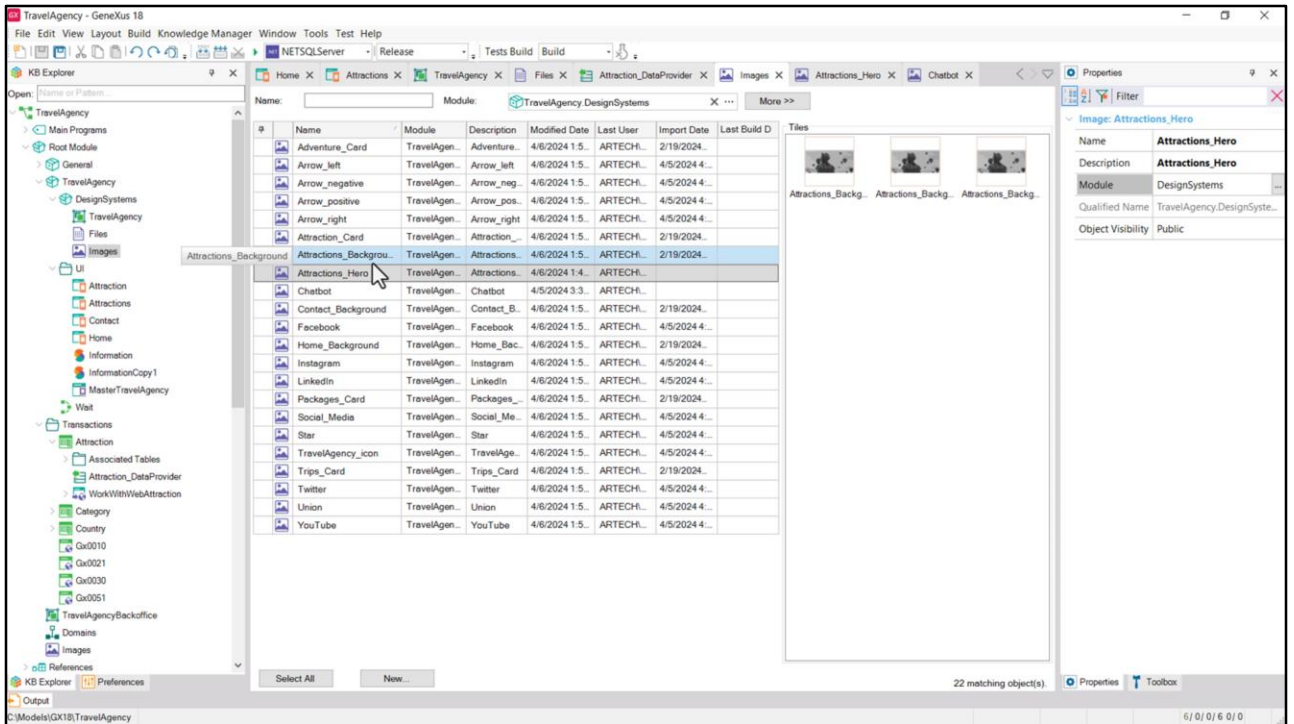
#	Name	Module	Description	Modified Date	Last User	Import Date	Last Build D
1	Bridges_Pic	Root Module	Bridges_Pic	10/10/2023...	GeneXus...	2/19/2024	
2	Christ_the_Redeemer...	Root Module	Christ_the...	10/10/2023...	GeneXus...	2/19/2024	
3	Cinque_Terre_Pic	Root Module	Cinque_Te...	10/10/2023...	GeneXus...	2/19/2024	
4	Cinque_Terre_Pic1	Root Module	Cinque_Te...	10/10/2023...	GeneXus...	2/19/2024	
5	Cinque_Terre_Pic2	Root Module	Cinque_Te...	10/10/2023...	GeneXus...	2/19/2024	
6	Cinque_Terre_Pic3	Root Module	Cinque_Te...	10/10/2023...	GeneXus...	2/19/2024	
7	Cinque_Terre_Pic4	Root Module	Cinque_Te...	10/10/2023...	GeneXus...	2/19/2024	
8	Cinque_Terre_Pic5	Root Module	Cinque_Te...	10/10/2023...	GeneXus...	2/19/2024	
9	Cinque_Terre_Pic6	Root Module	Cinque_Te...	10/10/2023...	GeneXus...	2/19/2024	
10	Eiffel_Tower_Pic	Root Module	Eiffel_Tow...	10/10/2023...	GeneXus...	2/19/2024	
11	Emperor_Pic	Root Module	Emperor_...	10/10/2023...	GeneXus...	2/19/2024	
12	Forbidden_City_Pic	Root Module	Forbidden...	10/10/2023...	GeneXus...	2/19/2024	
13	Glenfennan_Pic	Root Module	Glenfennan...	10/10/2023...	GeneXus...	2/19/2024	
14	Great_Wall_Pic	Root Module	Great_Wal...	10/10/2023...	GeneXus...	2/19/2024	
15	London_Pic	Root Module	London_Pic	10/10/2023...	GeneXus...	2/19/2024	
16	Louvre_Pic	Root Module	Louvre_Pic	10/10/2023...	GeneXus...	2/19/2024	
17	Matisse_museum_Pic	Root Module	Matisse_m...	10/10/2023...	GeneXus...	2/19/2024	
18	Nivola_Pic	Root Module	Nivola_P...	10/10/2023...	GeneXus...	2/19/2024	
19	PageFirst	Root Module	Page First	10/4/2023 1...	ARTECH...	2/19/2024	
20	PageLast	Root Module	Page Last	10/4/2023 1...	ARTECH...	2/19/2024	
21	PageNext	Root Module	Page Next	10/4/2023 1...	ARTECH...	2/19/2024	
22	PagePrevious	Root Module	Page Previ...	10/4/2023 1...	ARTECH...	2/19/2024	
23	selectRow	Root Module	select Row	8/15/2011 1...	GeneXus	2/19/2024	
24	Smithsonian_Pic	Root Module	Smithsonia...	10/10/2023...	ARTECH...	2/19/2024	

The right-hand side of the interface shows a 'Properties' panel for the selected 'Image: Adventure_Card, Arrow_Left, Arro...'. The 'Module' property is set to 'DesignSystems' and the 'Object Visibility' is 'Public'. Below the table, there are 'Select All' and 'New...' buttons, and a status bar indicating '24 matching object(s)'.

Entonces acá nos quedan solo las del Backoffice.

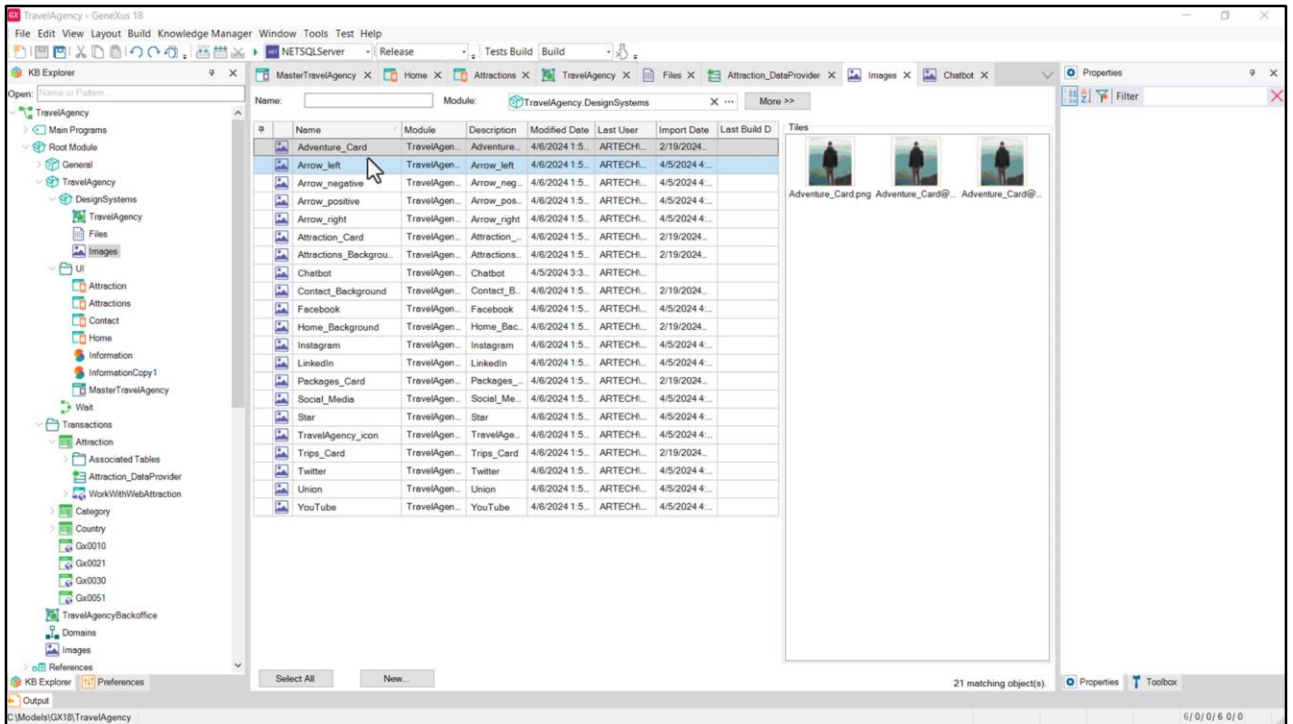


Y en el módulo Design Systems todas las otras. Por ejemplo, una, dos, tres, cuatro cards, que van a ser de la página Home.



Y acá, por ejemplo, tenemos la que creamos nosotros y esta que ya estaba, que es la misma, así que vamos a eliminar la que habíamos creado, que era para mostrarles nada más.

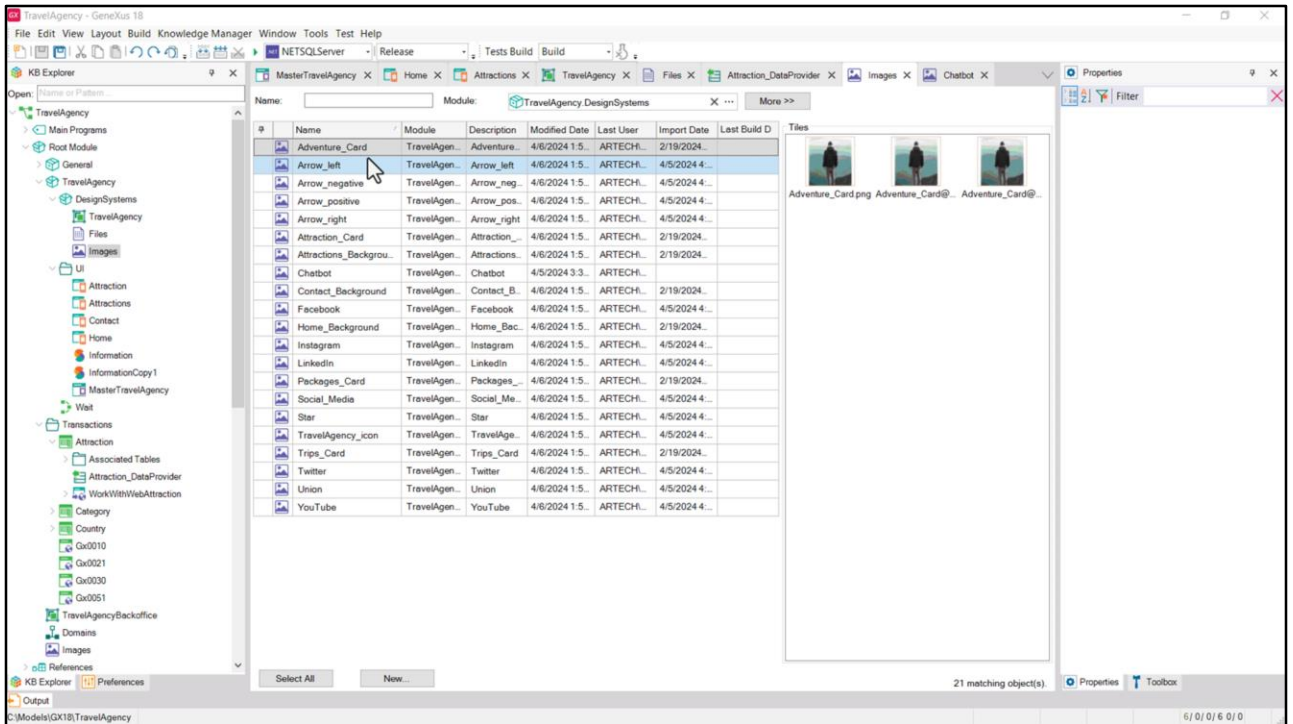
Bien, y ahora sí.



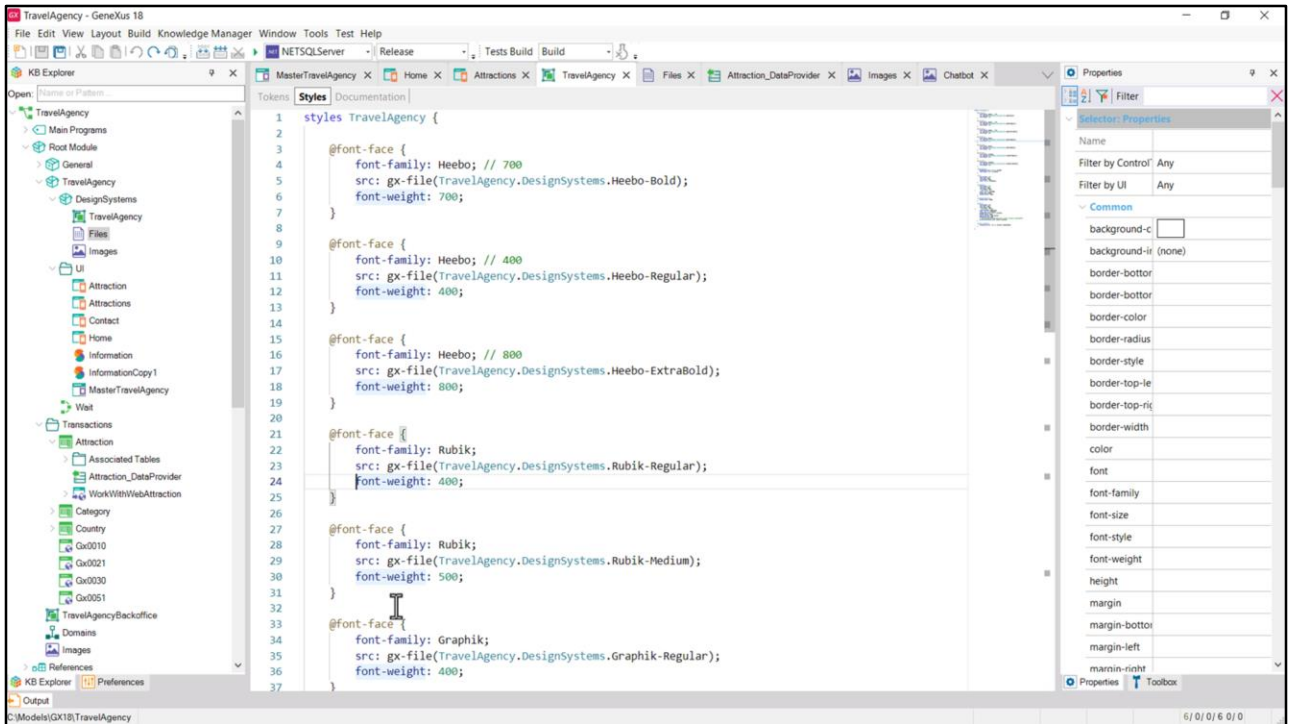
Para resumir:

En la etapa inicial del desarrollo nos conviene ya incorporar en GeneXus todos los assets, es decir, todos los recursos que tendremos que utilizar luego en nuestras pantallas y DSOs.

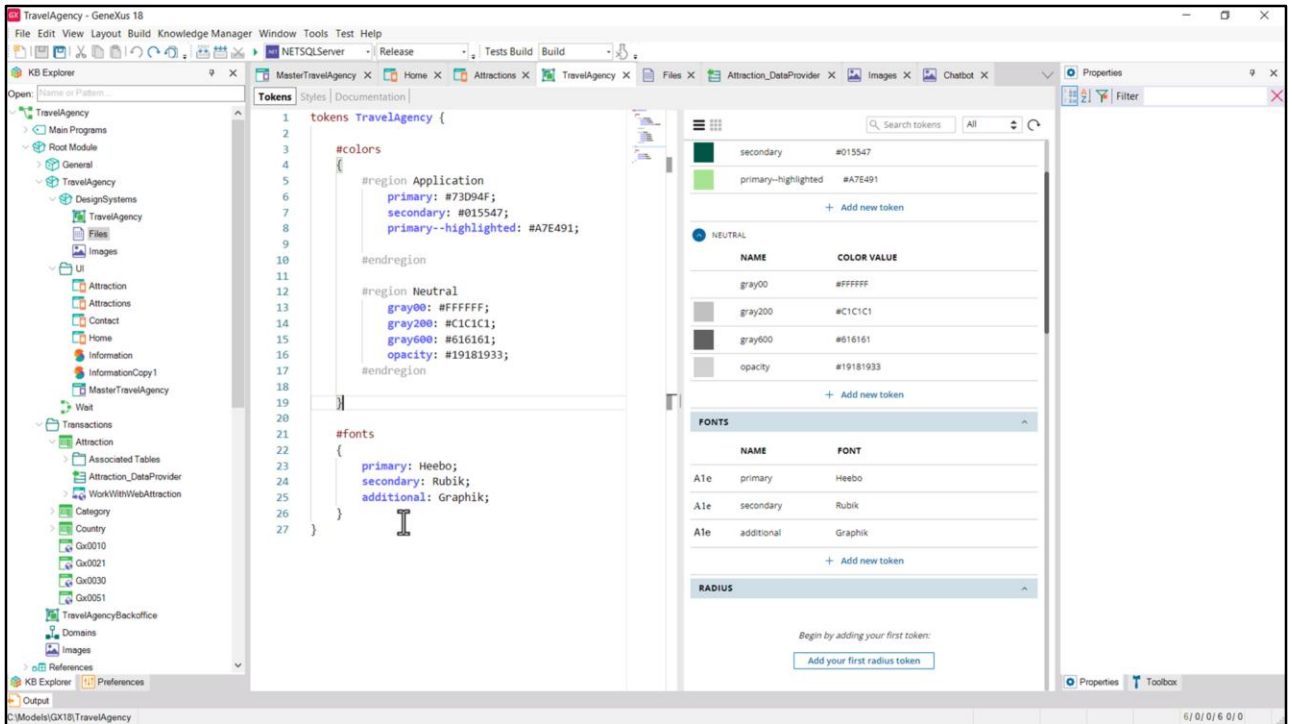
Entonces contar ya con las imágenes...



... con los archivos de fuentes no default...



...integradas al DSO a través de las reglas font-face... parece ser una buena idea.



Incluso también, si así lo decidimos, tenerlas tokenizadas a esas familias de fuentes...

Y por supuesto, incluir como tokens la paleta de colores de nuestra aplicación.

En el próximo video ampliaremos un poquito más esto, y pensaremos en una segunda etapa de tokenización de los colores que permitan modelar mejor el sistema, de un modo más semántico. Bueno, nos vemos ahí.

GX

GeneXus by Globant

GeneXus[™]
by **Globant**

training.genexus.com