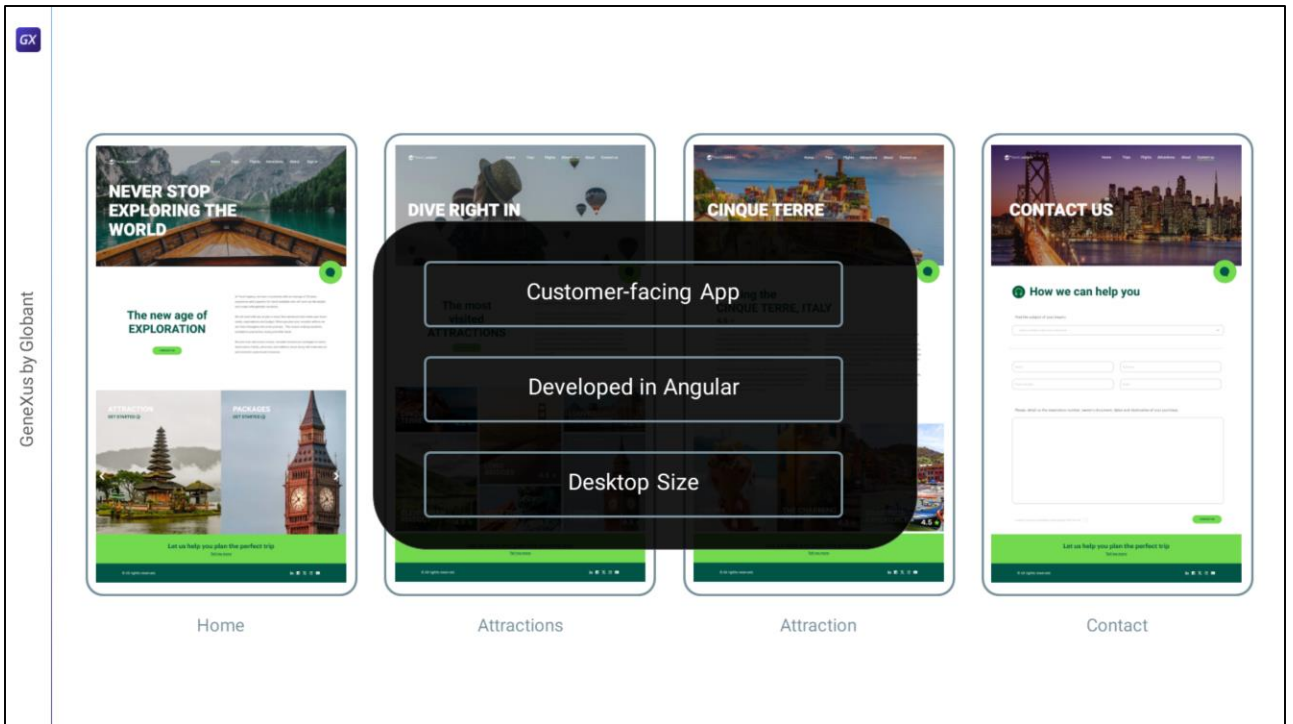


Layouts in Figma



Cecilia Passalacqua

En esta clase vamos a estar hablando sobre Layouts en Figma y qué pasa con el diseño cuando tenemos pequeñas variaciones en el ancho de la pantalla.



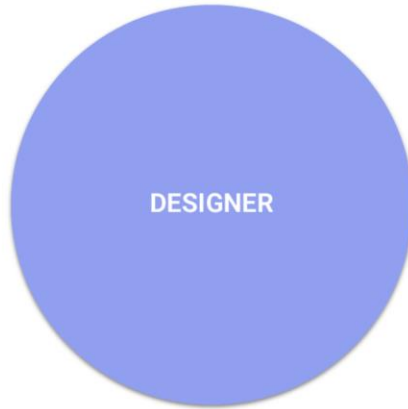
Desde el principio del curso sabemos que estamos diseñando una aplicación customer-facing que será desarrollada en Angular para tamaño Desktop.



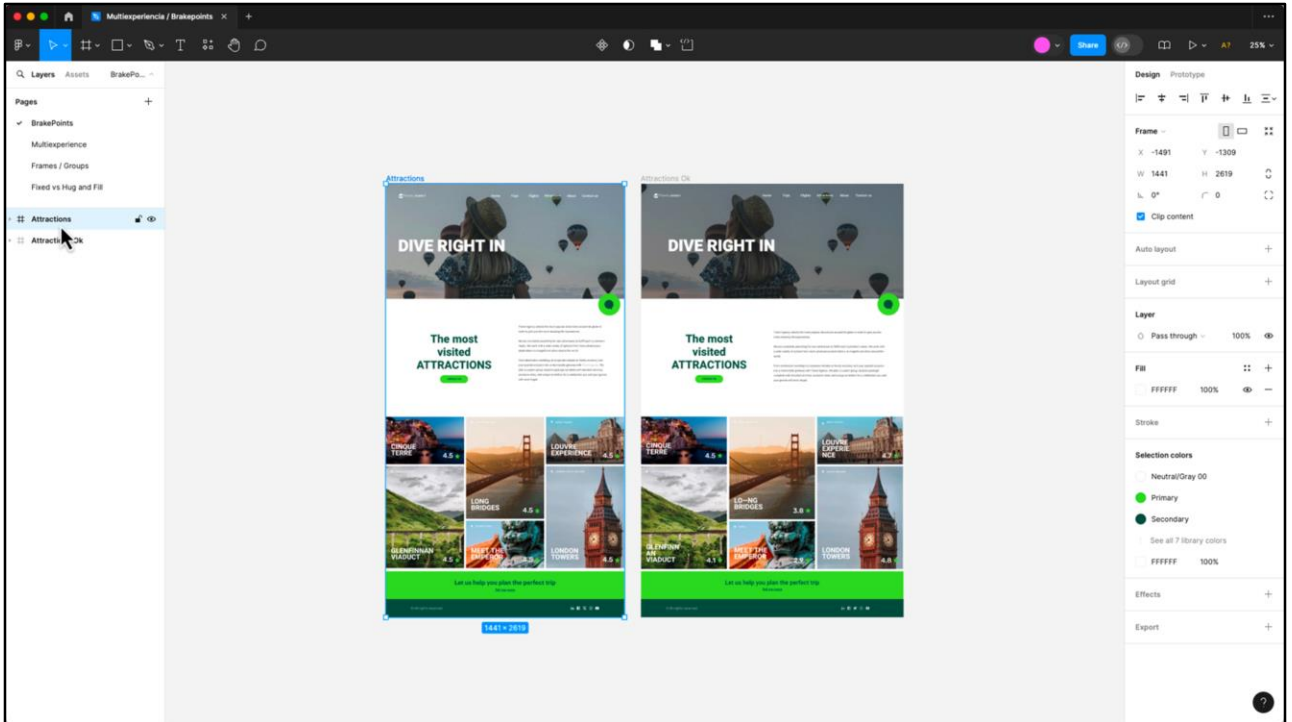
Lo que no sabemos (y no podríamos saber) es qué medida exacta van a tener cada una de las pantallas Desktop que visualicen mi aplicación en el futuro como en este caso, donde vemos 3 pantallas que posiblemente tengan pequeñas variaciones de tamaño entre una y otra.



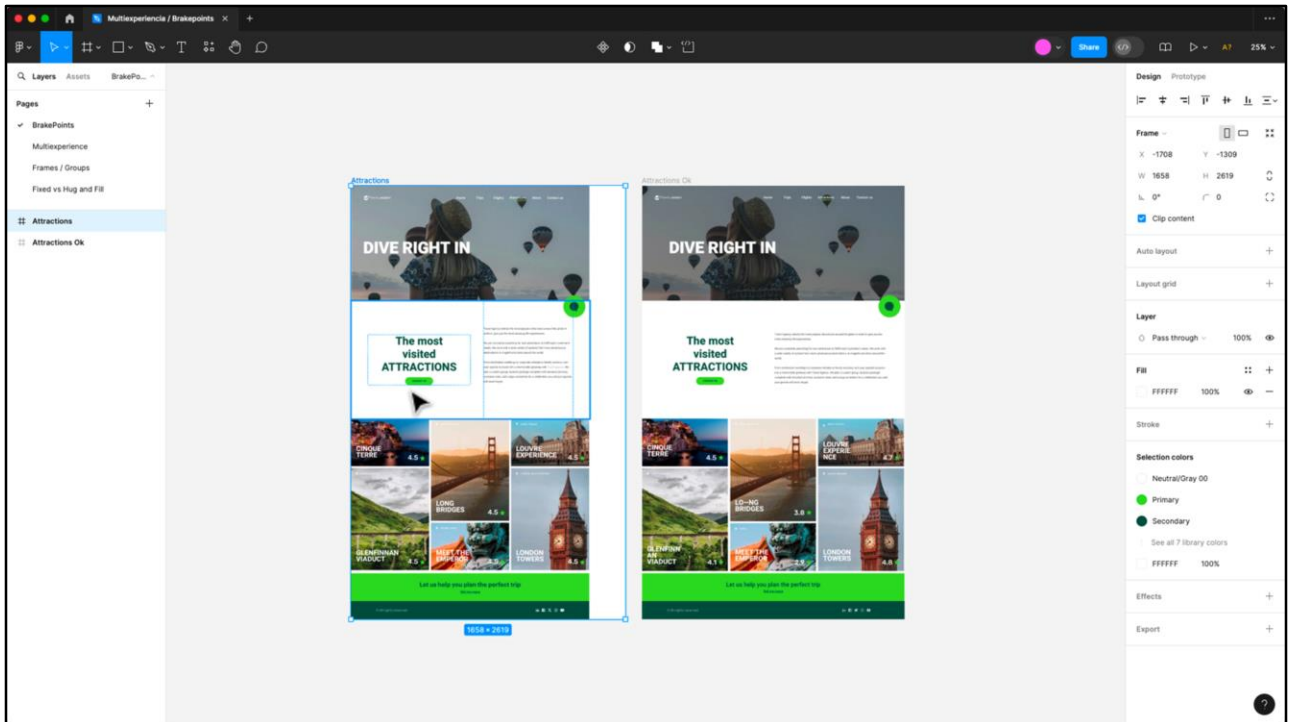
Anticipating element behaviors
Make decisions



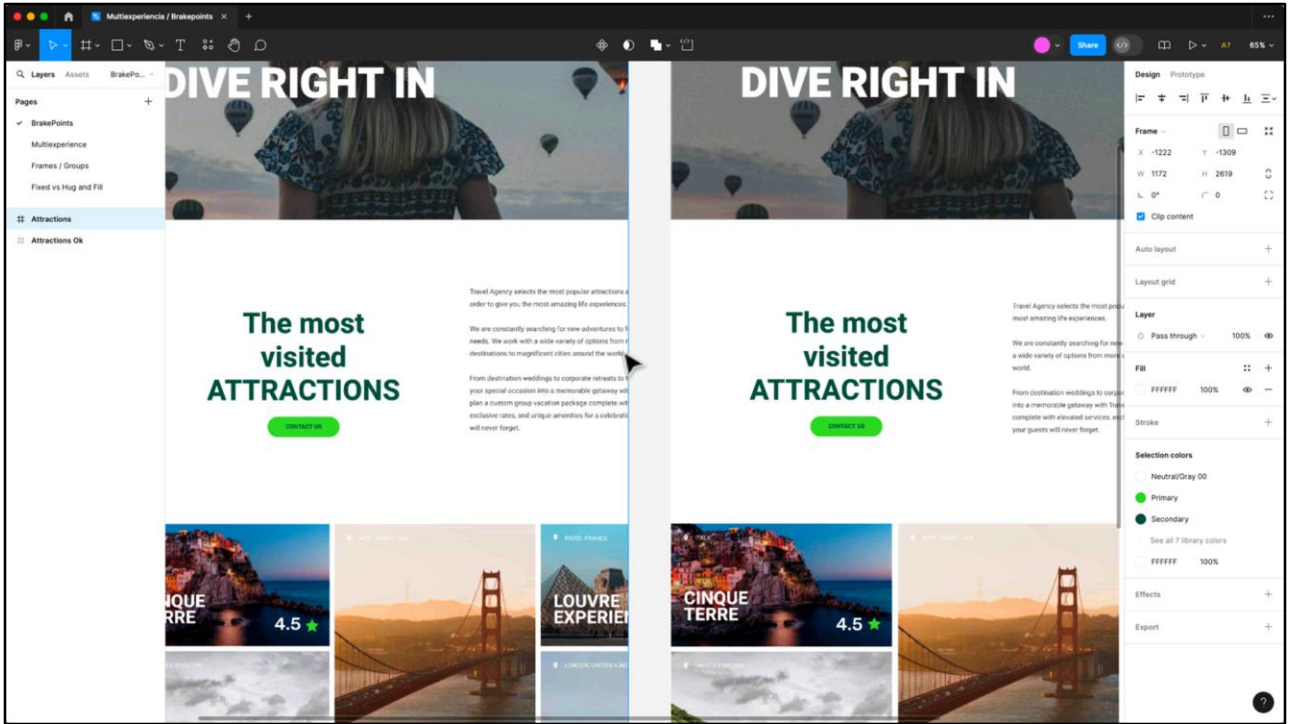
Para resolver esto, yo como diseñadora debo prever el comportamiento de cada uno de los elementos de la pantalla y decidir qué quiero que suceda con cada uno en cada caso.



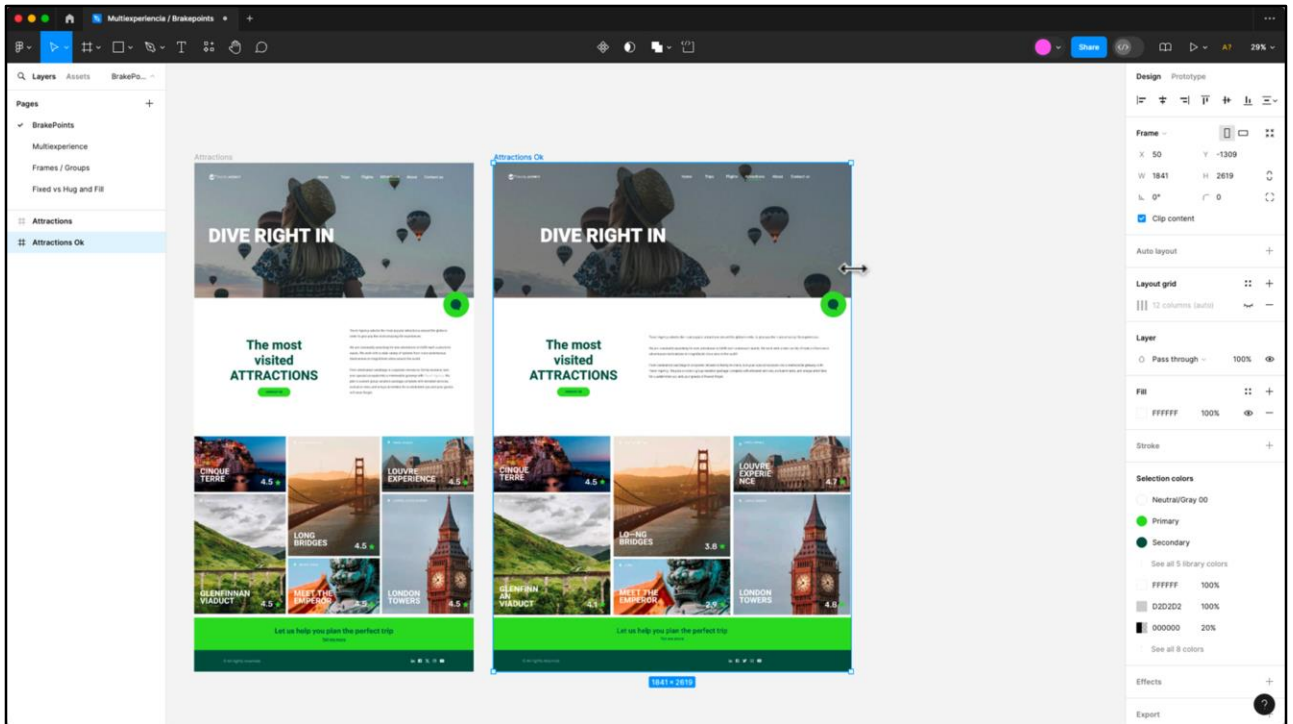
Vamos a verlo de forma aislada directamente en Figma: acá vemos la pantalla de Attractions de mi aplicación, una de estas está en estado default y la otra cuenta con los ajustes que les mencionaba.



Si incrementamos el ancho de la pantalla de la izquierda vamos a observar que los elementos no se modifican, de hecho se mueven todos hacia la izquierda. Esto me indica a mí que esos elementos de alguna forma están “pineados” para uno de los lados.

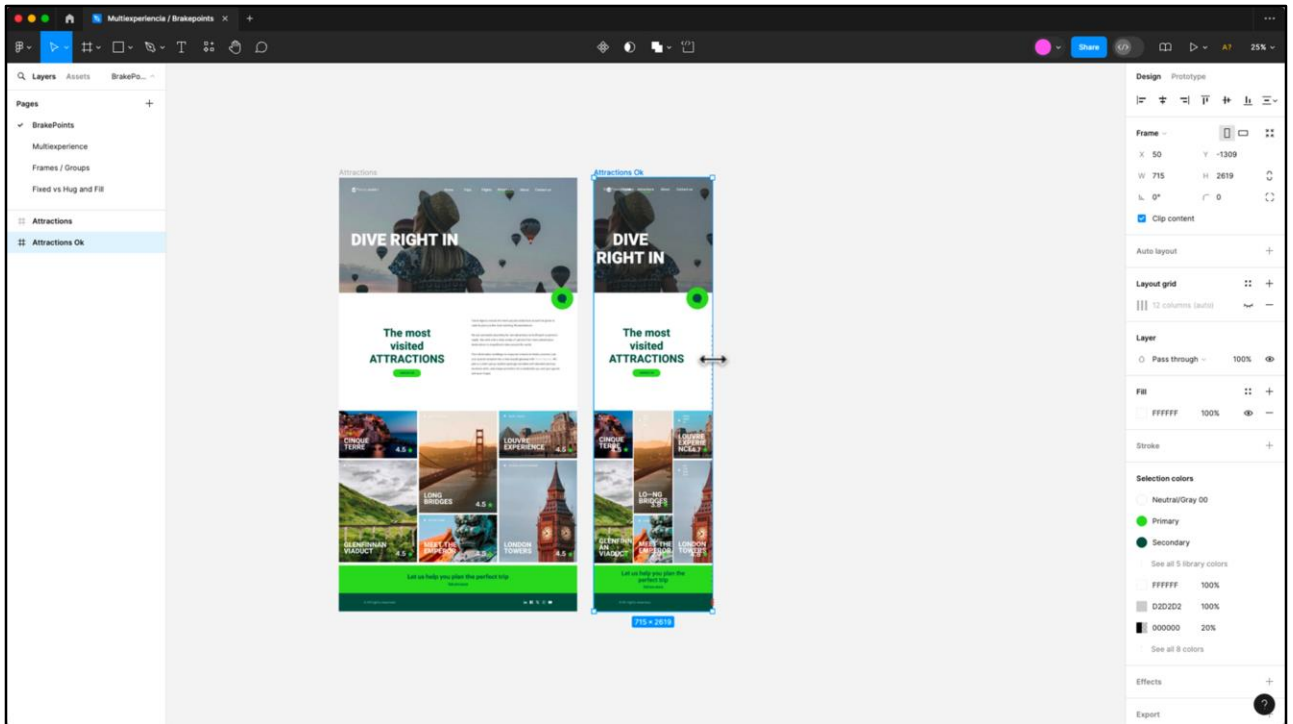


Por el contrario si disminuimos el tamaño del frame, vamos a observar que la pantalla directamente se corta.



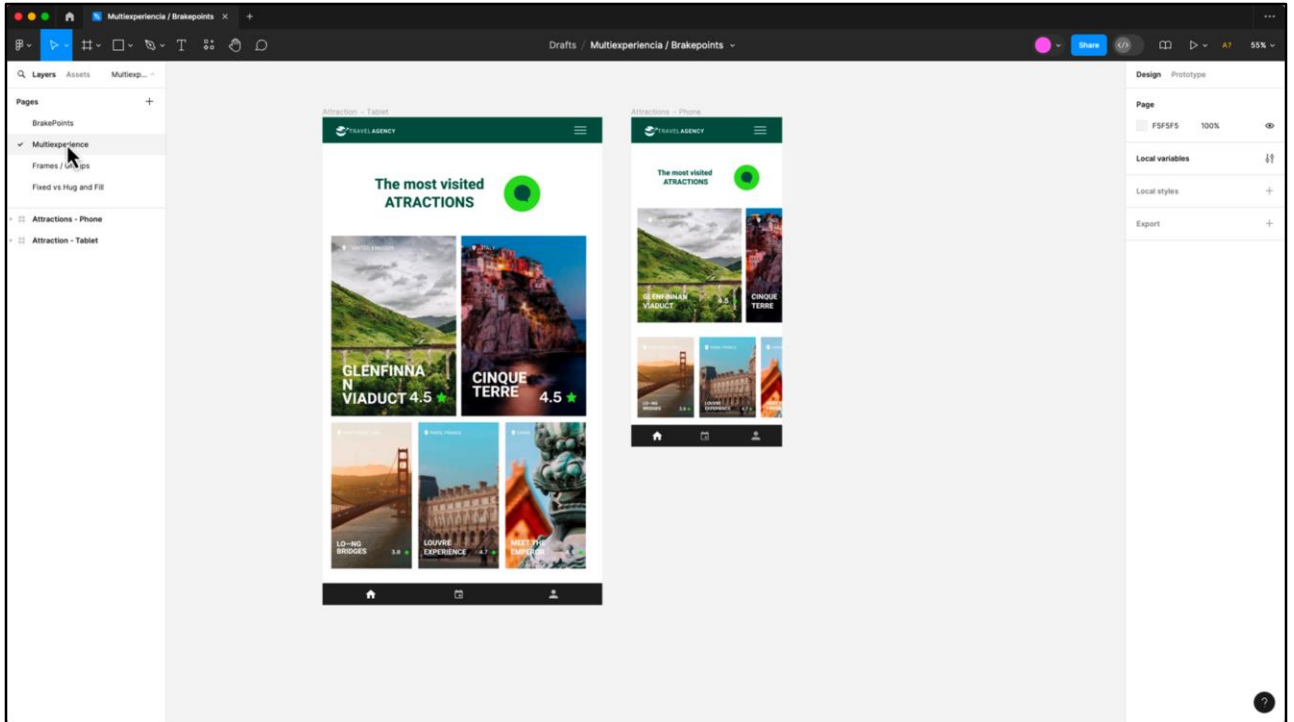
Analicemos ahora qué sucede con la pantalla de la derecha. Para este caso, realizamos los ajustes necesarios para que, si agrandamos o achicamos el ancho del frame, los elementos se escalen completando el nuevo espacio.

Por supuesto que estos ajustes van a ser funcionales si las variaciones de tamaño son finitas, no podemos pretender que un mismo diseño se readapte de un tamaño desktop a mobile.

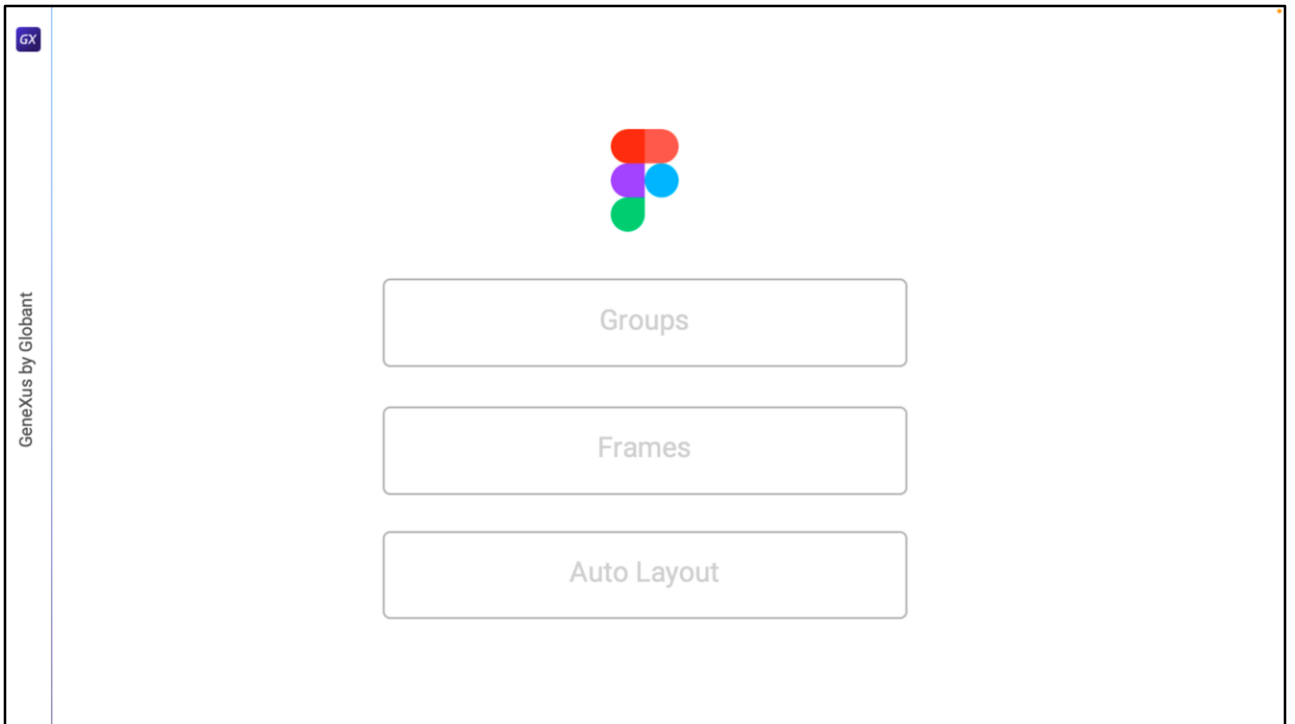


Lógicamente si el cambio es abrupto, el diseño se va a romper, como vemos aquí.

Para estos casos debemos definir breakpoints, puntos de quiebre donde directamente el diseño de la pantalla es otro.



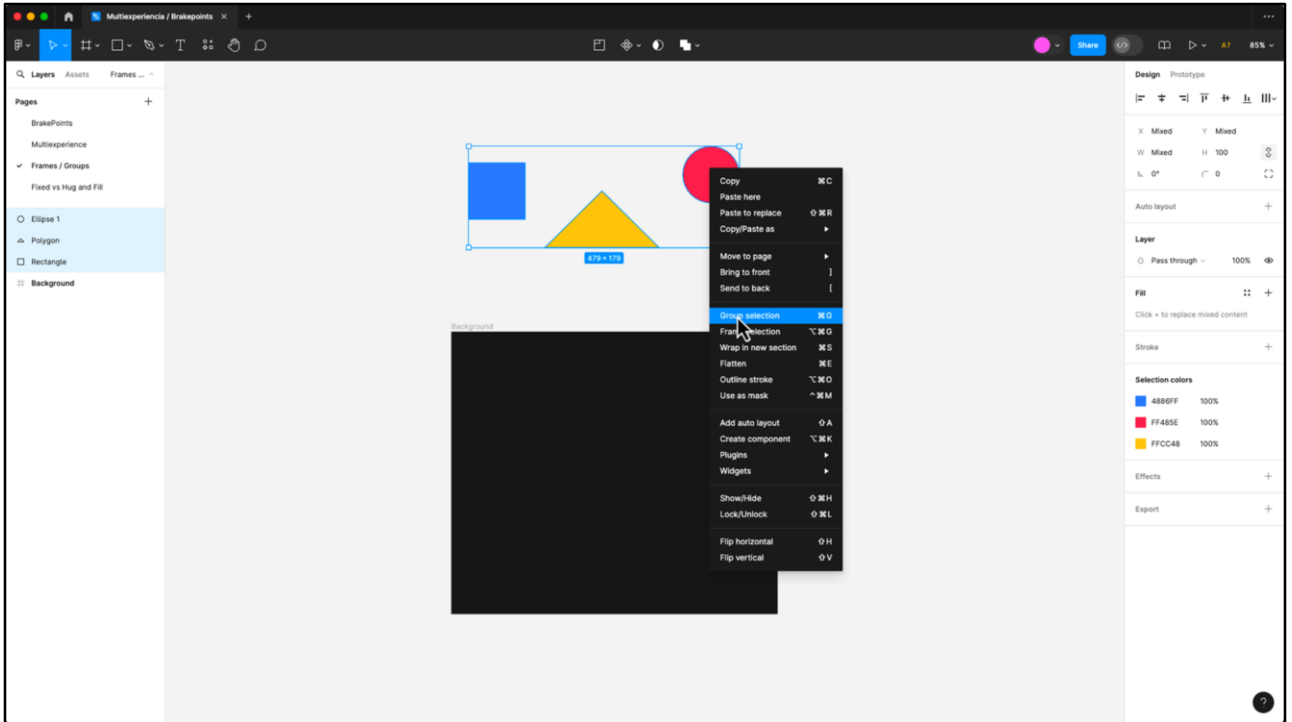
Aquí por ejemplo les muestro las variantes de diseño de la página Atractions para Tablet y Phone.



Todas estas son decisiones de diseño y tengo que dárselas a conocer al frontender.

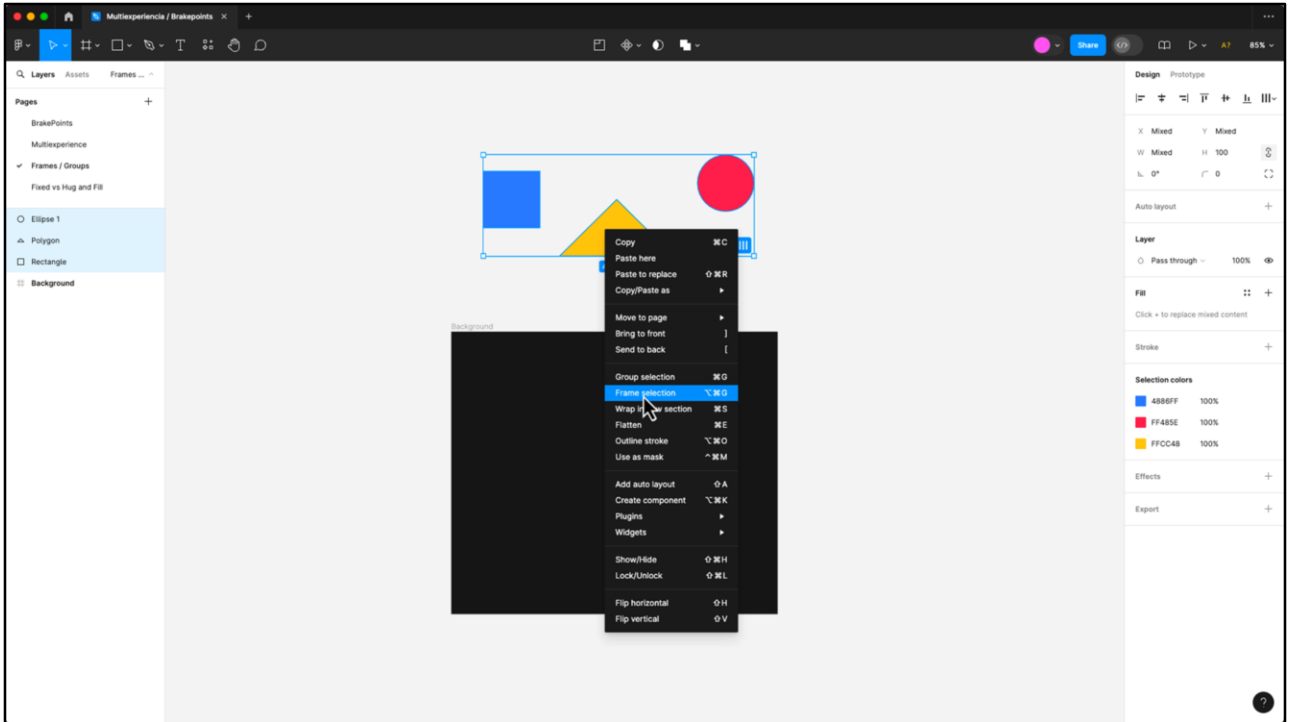
Antes de empezar a ver las propiedades que nos permiten definir estos comportamientos de los elementos tenemos que ver qué maneras tengo en Figma para agrupar dichos elementos y qué propiedades tiene cada una.

Vamos a hablar de Grupos, Frames y Autolayout.



En este caso tengo 3 formas independientes. Si quisiera agruparlas lo puedo hacer utilizando los grupos o los Frames. Veamos sus diferencias.

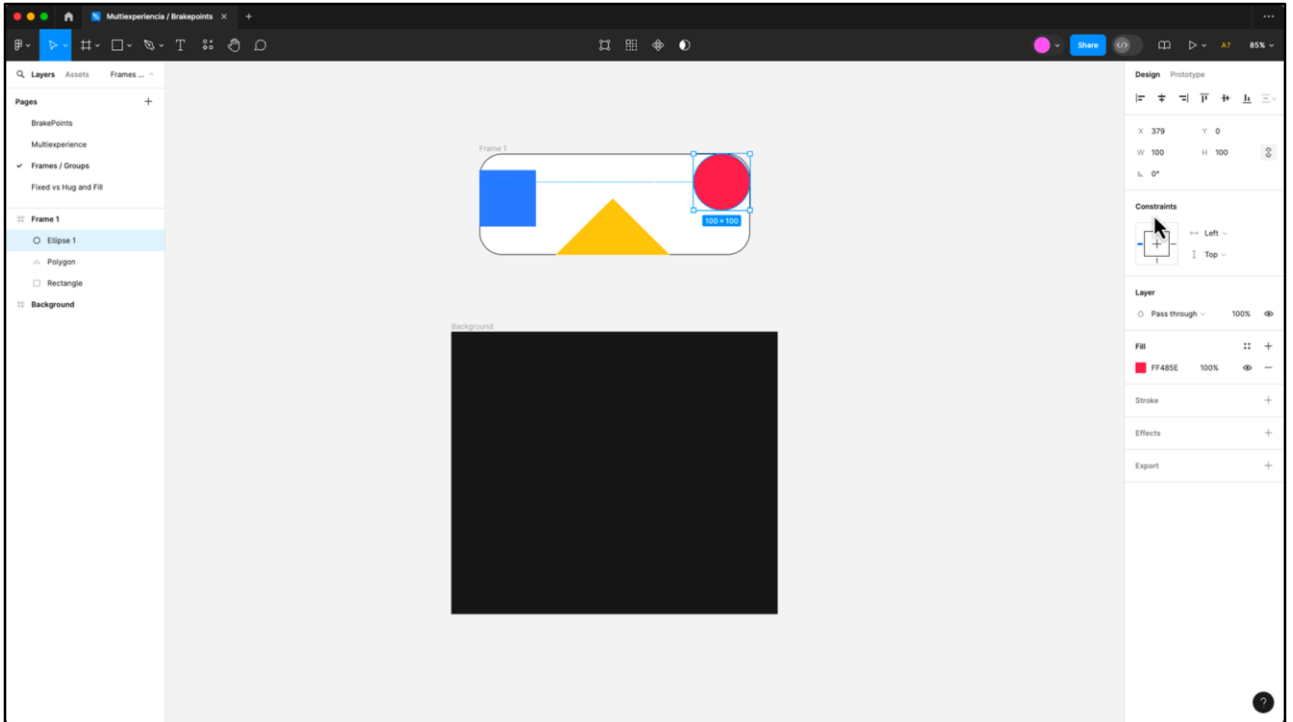
Los grupos se utilizan generalmente para mover un conjunto de elementos. Se visualizan en el panel de capas con un cuadrado de líneas punteadas. No tienen fondo y si selecciono los elementos dentro del grupo (haciendo doble clic) solo puedo ver sus propiedades de ancho, alto y posición.



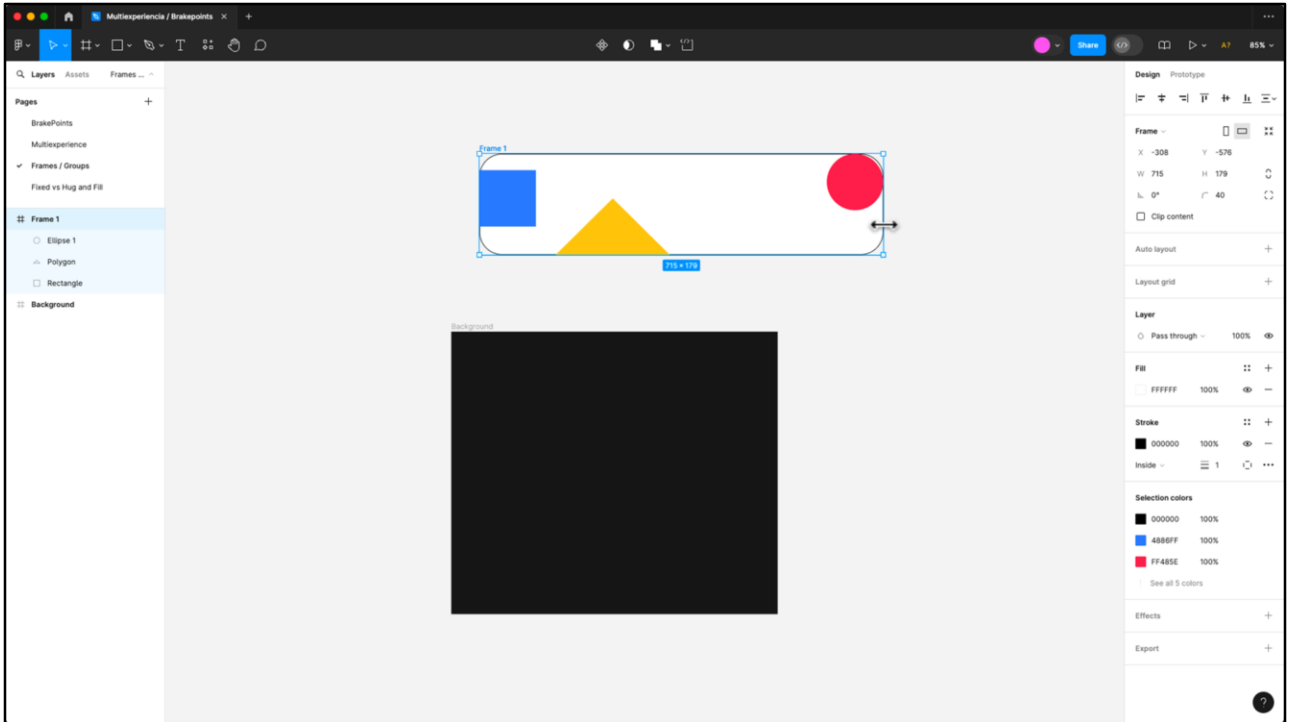
Los frames son contenedores. Además de tener los elementos juntos los tengo en un espacio que los contiene y que a su vez tiene propiedades.

Su visualización en el panel de capas es con un ícono de un marco.

A estos frames puedo ponerle un color de fondo, un color de borde o incluso redondear sus vértices.

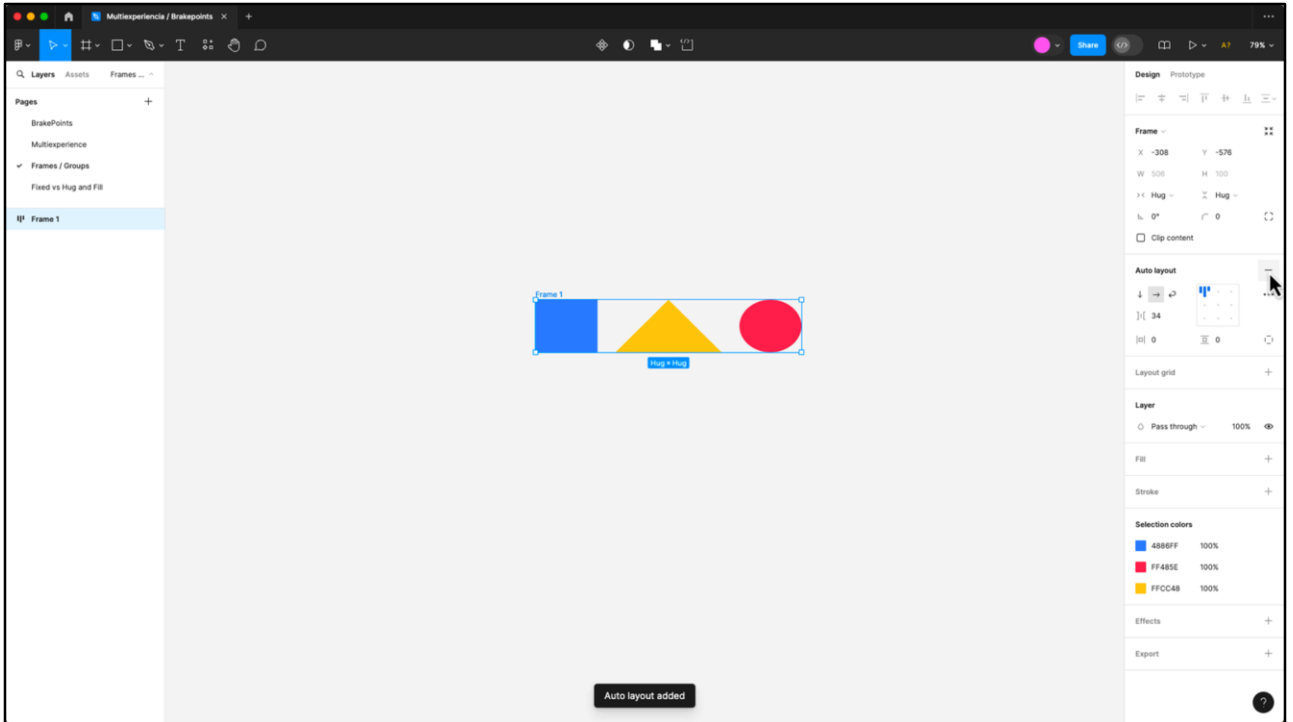


Si selecciono los elementos dentro del frame, además de las propiedades que mencionaba en los grupos se suma un apartado de “constraints”. Por defecto todos los elementos están pineados a la izquierda, igual que pasaba en la pantalla Atracciones sin ajustes.



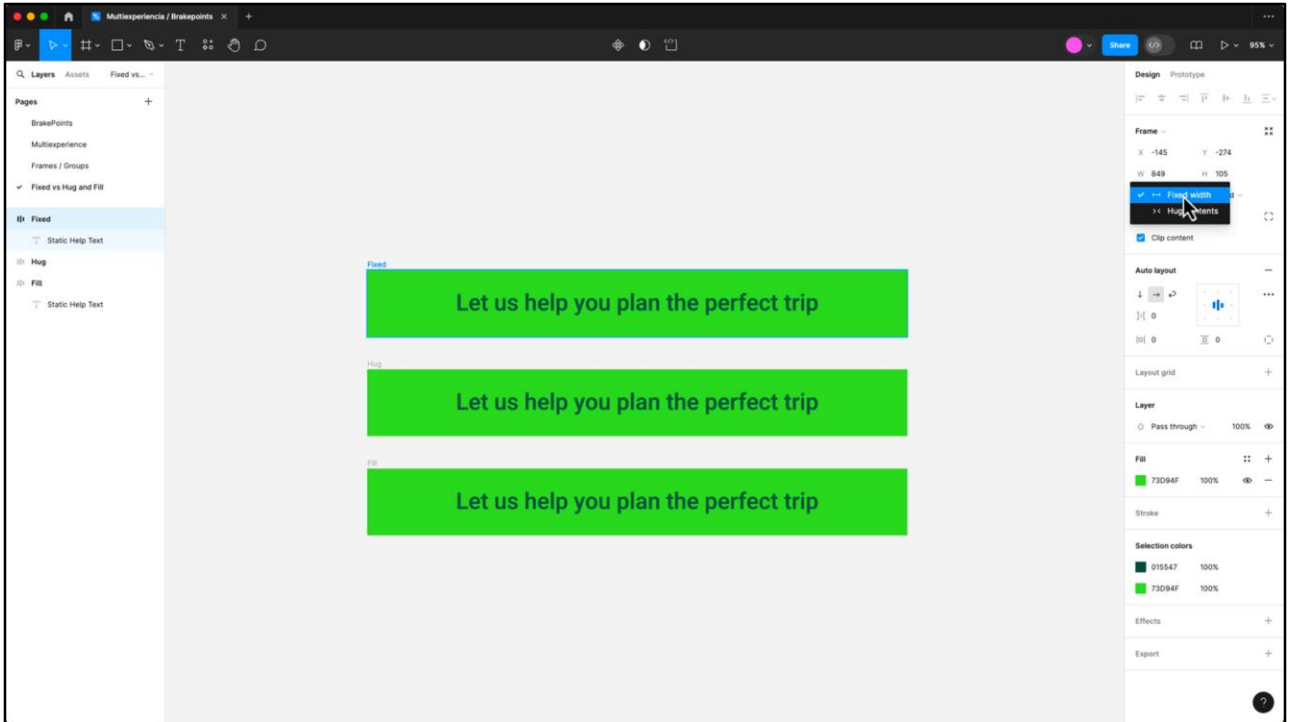
Las constraints son algunas de las propiedades que nos van a servir para definir el comportamiento de los elementos.

Por ejemplo si modificamos el ancho de este frame, todos los elementos se mantienen a la izquierda. Sin embargo, si pineamos por ejemplo la elipse a la derecha vemos que se mantiene sobre ese lado al modificarse el tamaño del frame.



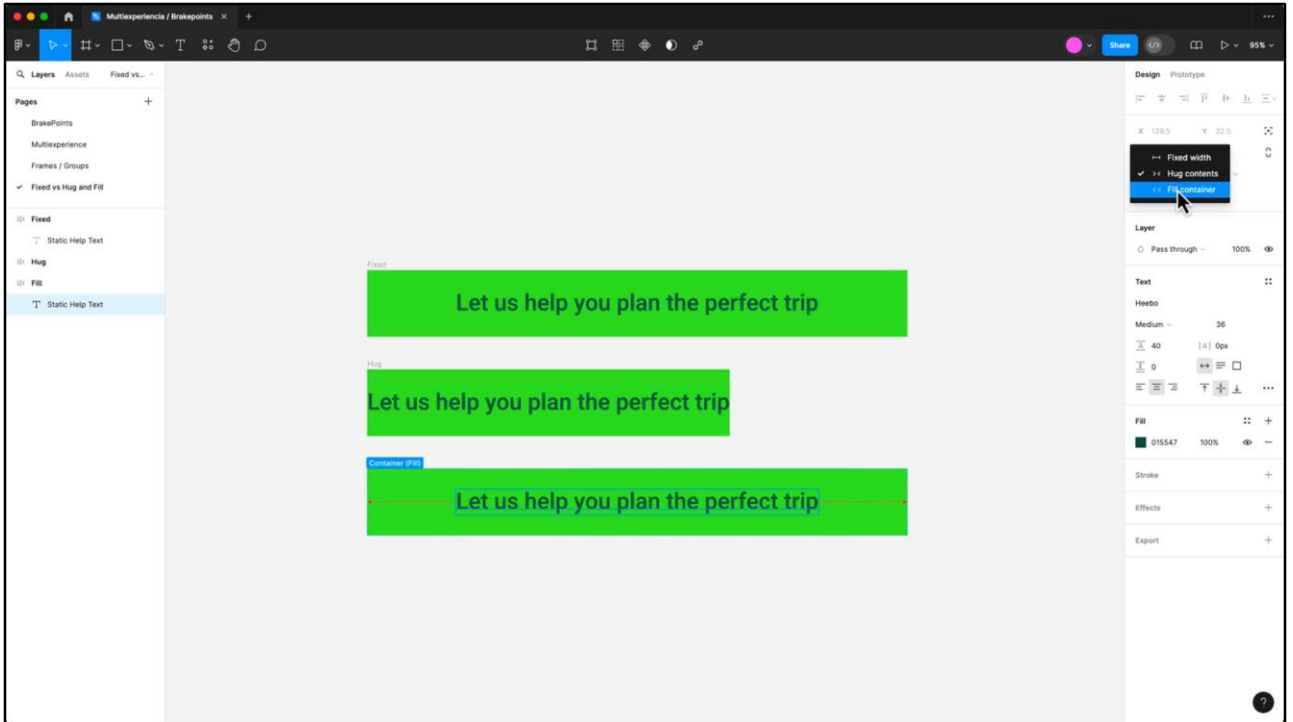
Otra propiedad que puedo colocarle tanto a los Grupos como a los Frames es el **AUTOLAYOUT**.

Este nos va a permitir organizar ese conjunto de elementos, definiendo en qué orientación quiero que se dirijan, cuál es la distancia entre ellos, cómo quiero que se ordenen entre sí y si van a tener padding a sus lados o superior e inferior

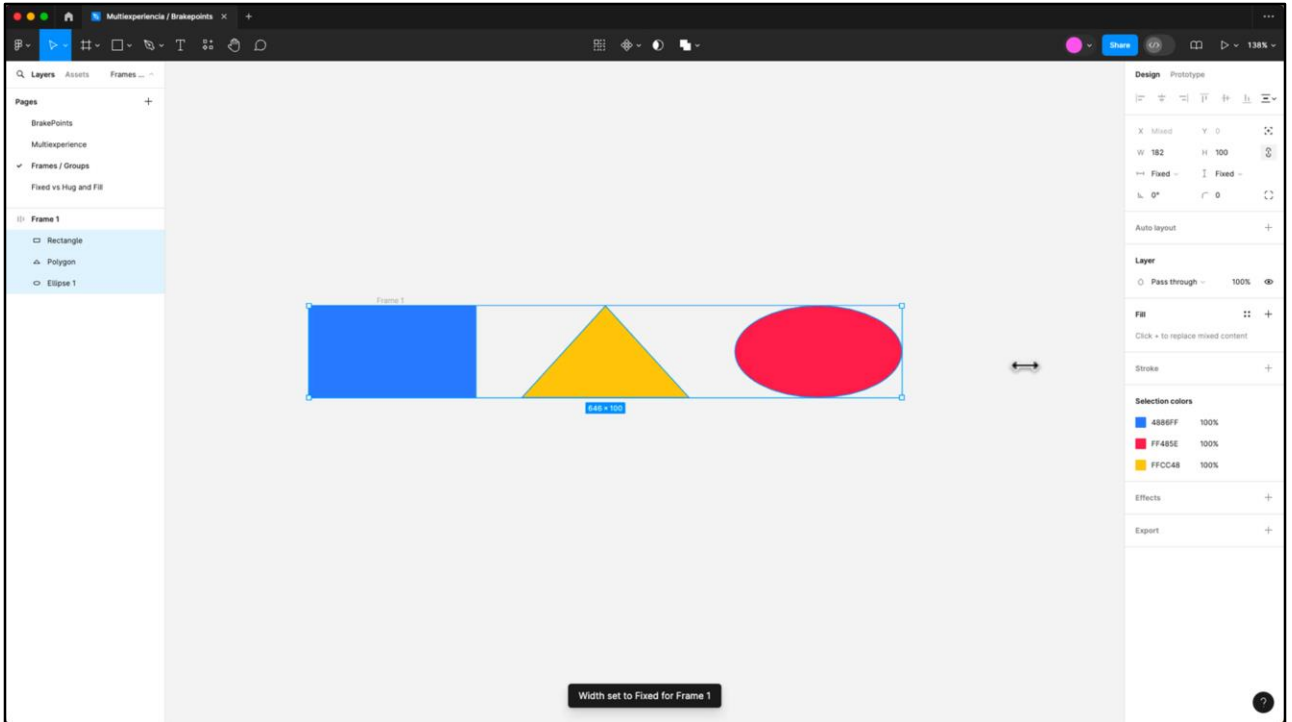


Además cuando utilizamos un contenedor con autolayout y dentro tengo uno o varios elementos, puedo definir cómo se comporta el contenedor en relación con su contenido y su contenido en relación al contenedor.

Dentro de las propiedades del contenedor tenemos las de Fixed, es decir, fijo o que sea hug, es decir que se adapte a su contenido.

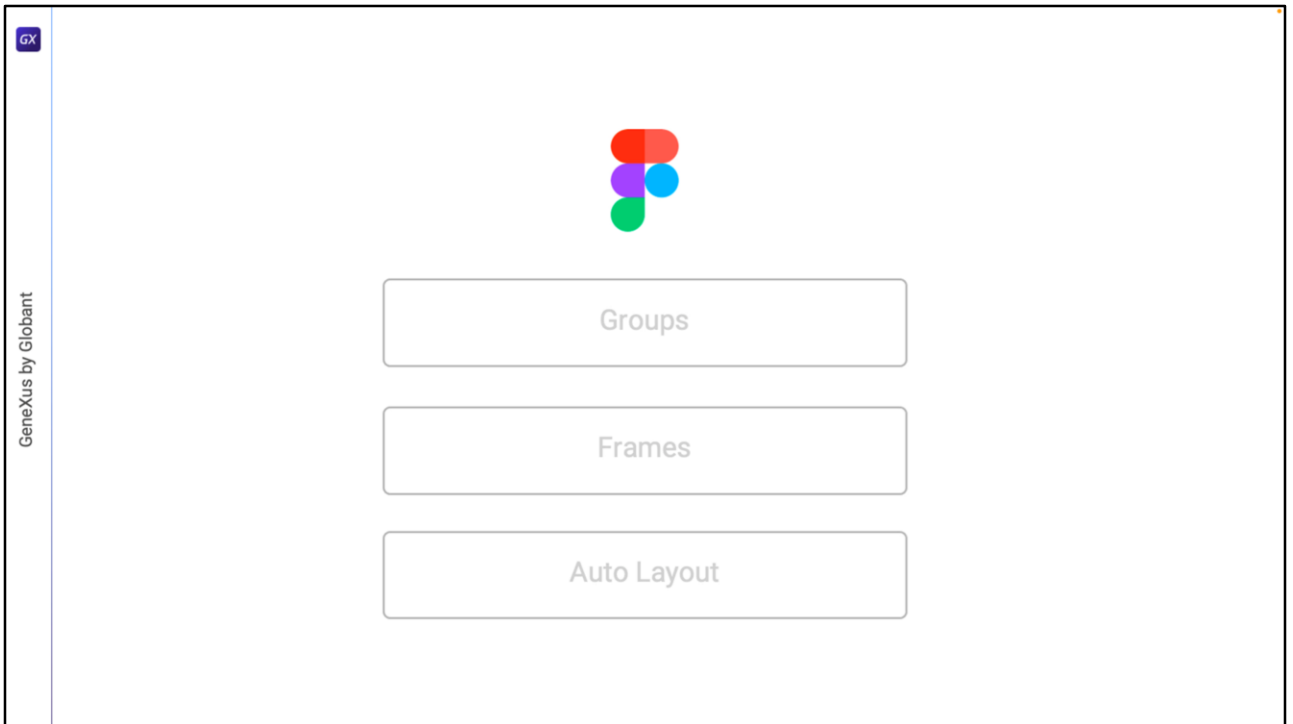


Para el caso del contenido, encontramos las mismas 2 opciones de fixed y hug más la opción de fill container, en este ejemplo que estamos viendo el campo de texto se estira para llegar al borde del contenedor, no presentando mayores problemas.

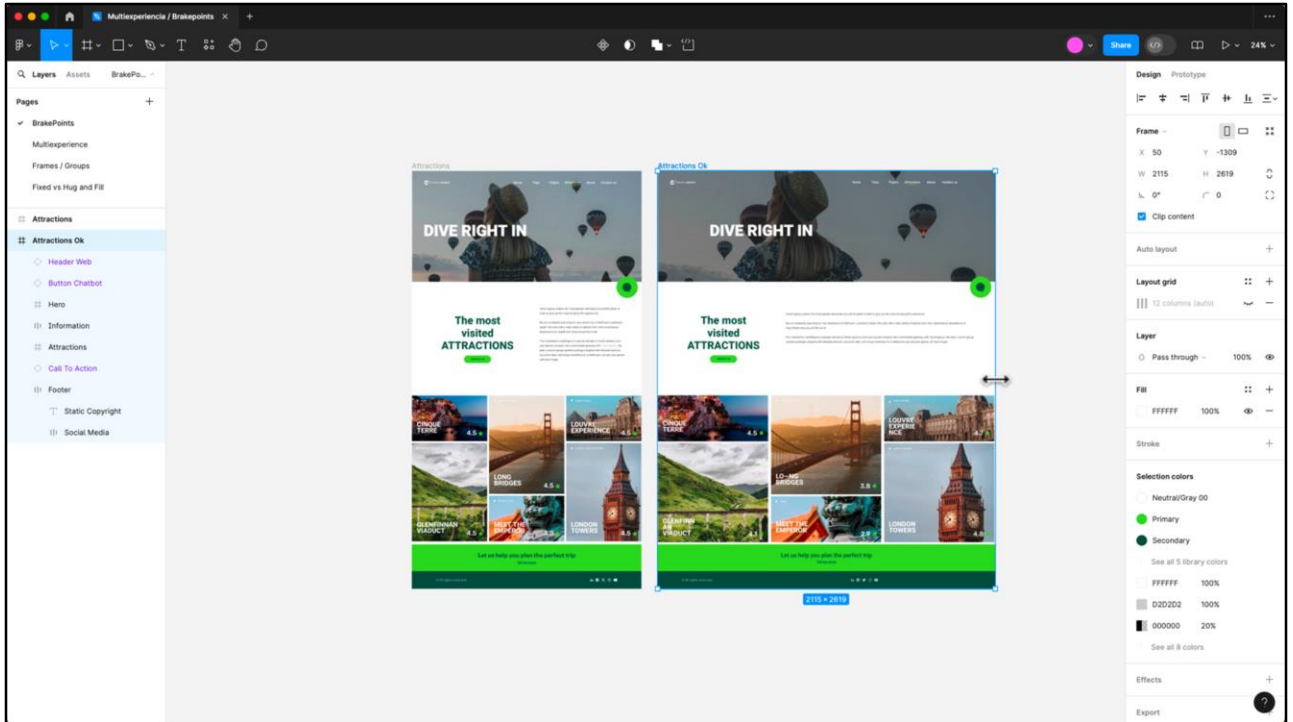


Pero si tomáramos como referencia el caso anterior, y a cada una de las figuras que están dentro del Auto Layout les colocáramos fill container, al estirar ese contenedor las deformaríamos por completo.

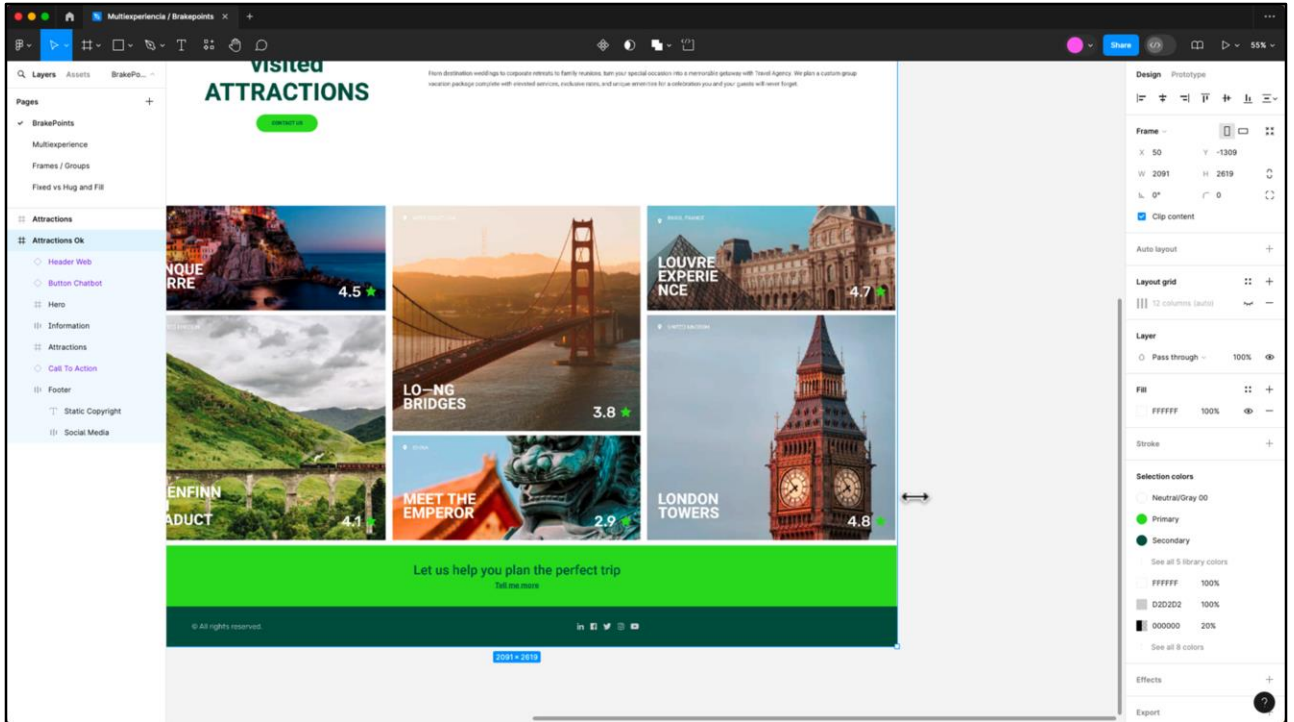
Por eso es fundamental analizar qué sucede en cada caso particular sin perder de vista que toda la información que coloquemos en el archivo de diseño de alguna forma se va a corresponder en GeneXus, como verán mas adelante junto a Ceci.



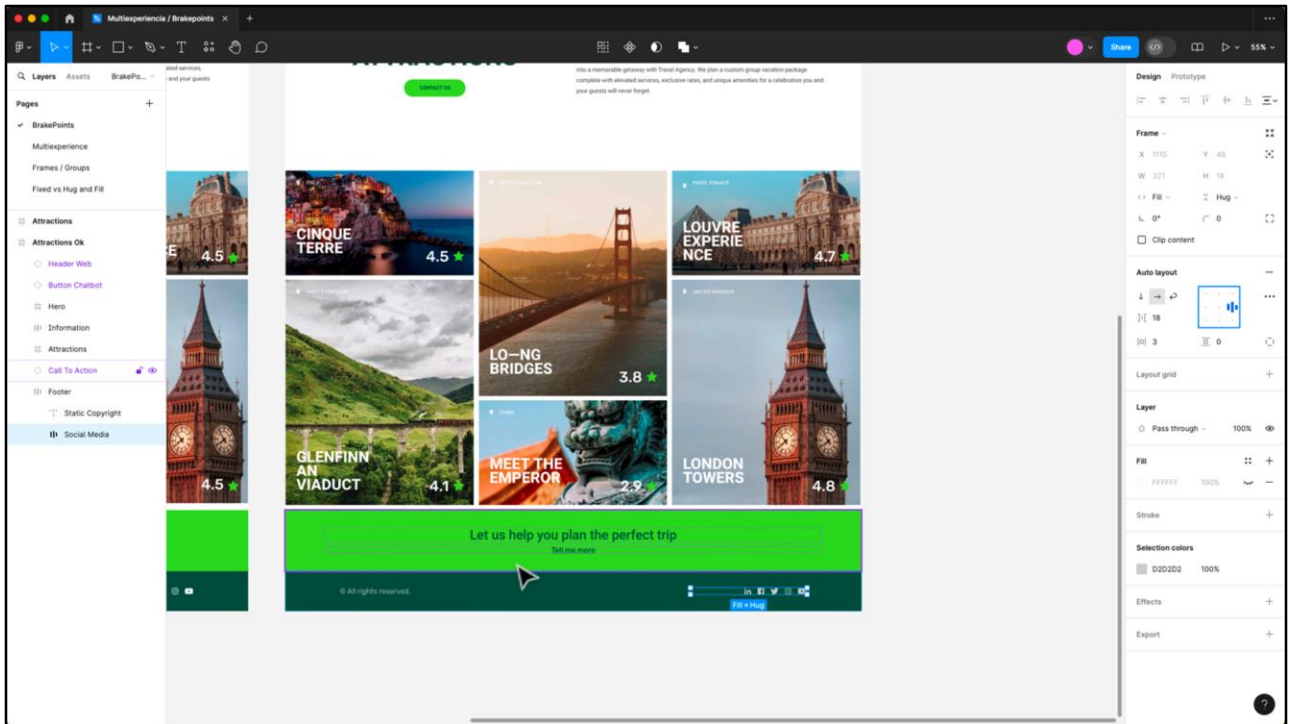
A modo de resumen de esta clase podríamos decir que para definir cómo se van a comportar cada uno de los elementos de las pantallas ante variaciones mínimas en su ancho voy a trabajar preferiblemente con frames con autolayout, las propiedades de constraints y las propiedades fixed, hug y fill.



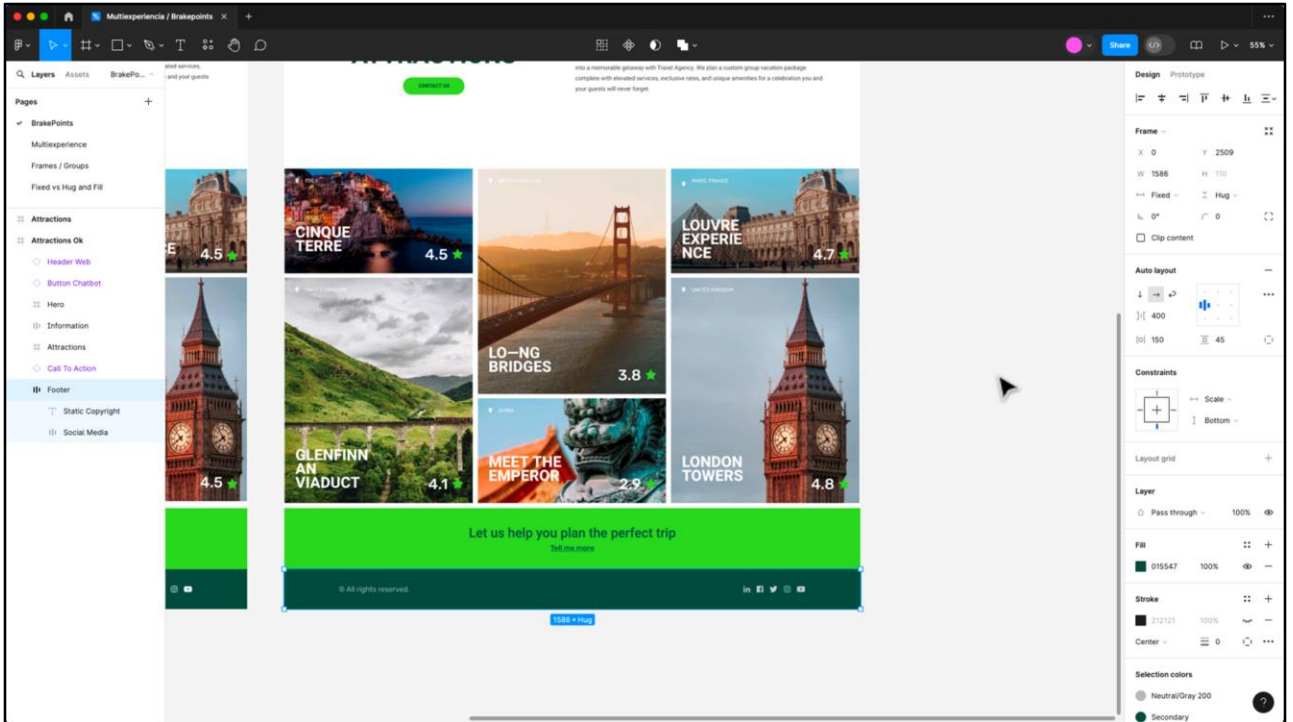
Ahora volvamos brevemente a las pantallas iniciales. ¿Notaron al principio de la clase que cuando modifiqué la pantalla con ajustes algo pasó en el footer?



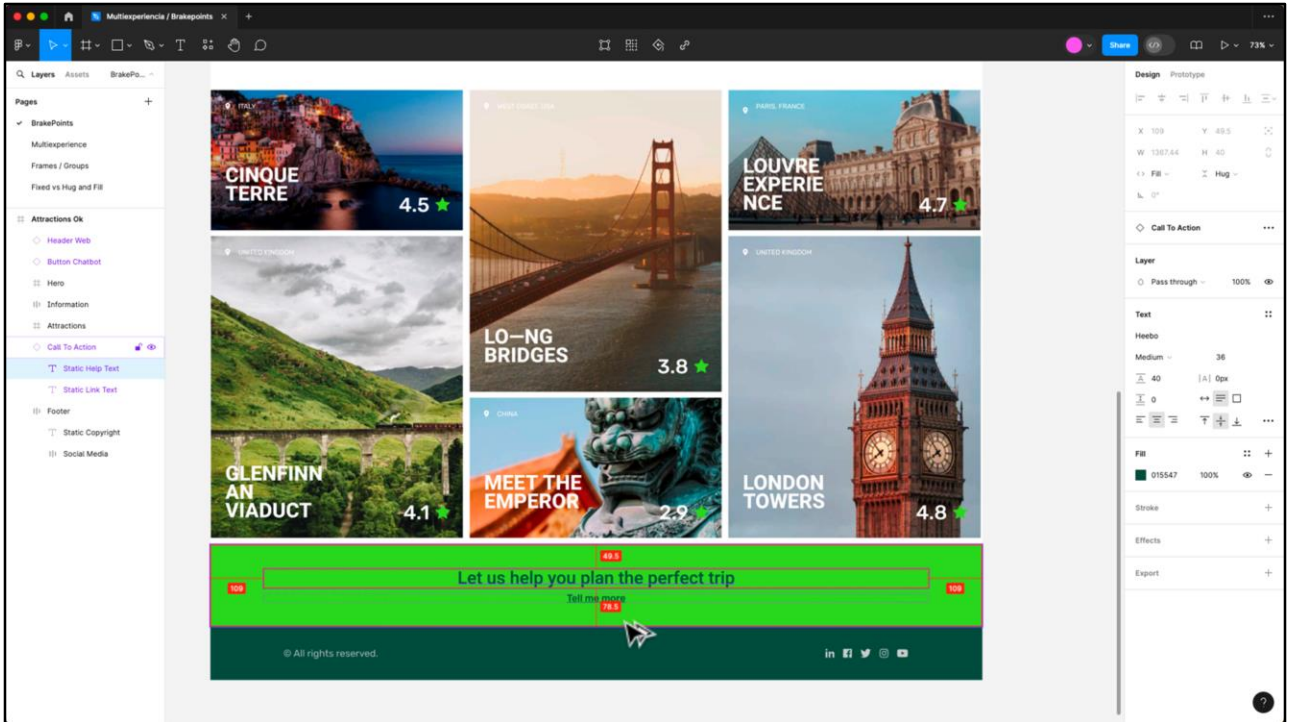
Mirémoslo más de cerca, ¿se dan cuenta? Vayamos a modificarlo.



Yo creo que si al conjunto de íconos de redes sociales lo ponemos con el comportamiento Fill, y dentro del autolayout lo pineamos a la derecha...



...y al contenedor Footer lo dejamos fixed y pineado a la izquierda, este comportamiento se va a resolver.



Por último les quiero compartir una forma simple de medir las distancias entre los elementos. Si selecciono un elemento y presiono la tecla "option" en mac o "alt" en windows y paso el cursor sobre otro elemento me va a decir a qué distancia se encuentran.

En el próximo video, trabajarán junto a Ceci en cómo interpretar este archivo de diseño e implementarlo en GeneXus para que la aplicación luzca y se comporte de la forma en que acabamos de definir.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com