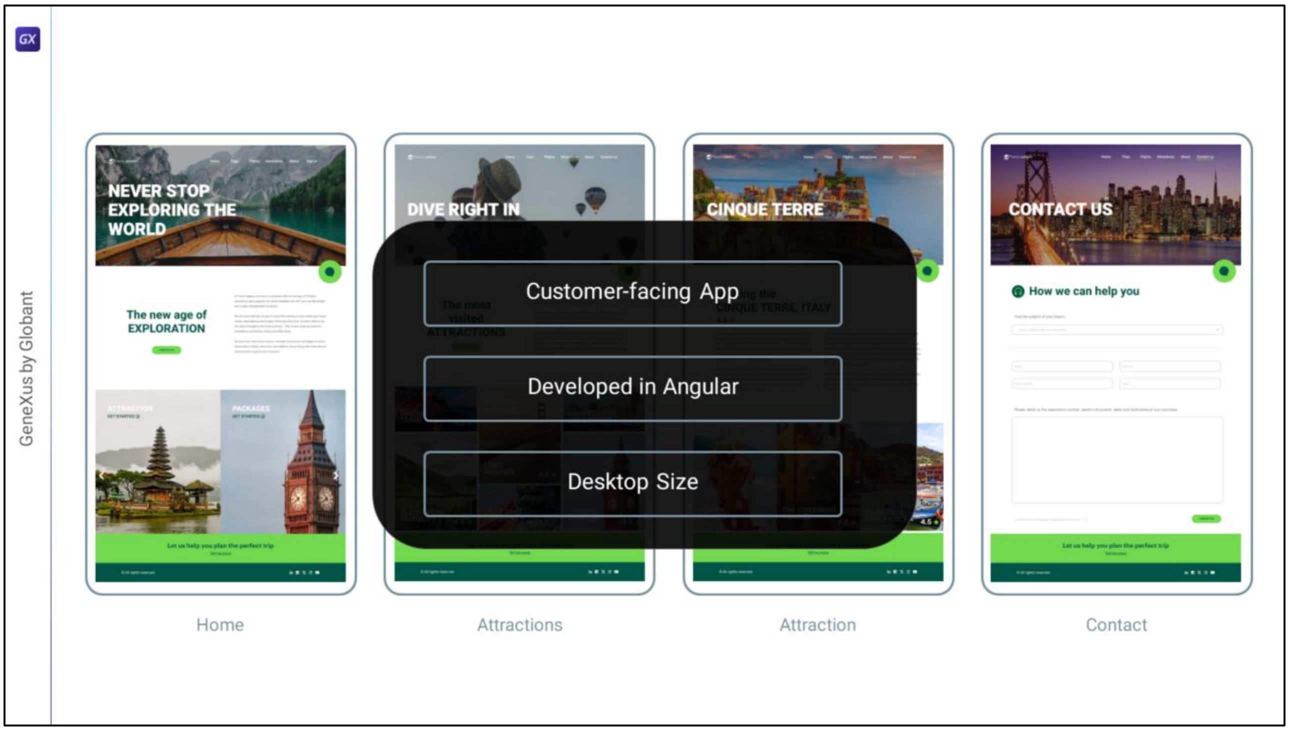
Layouts in Figma

GeneXus by Globant

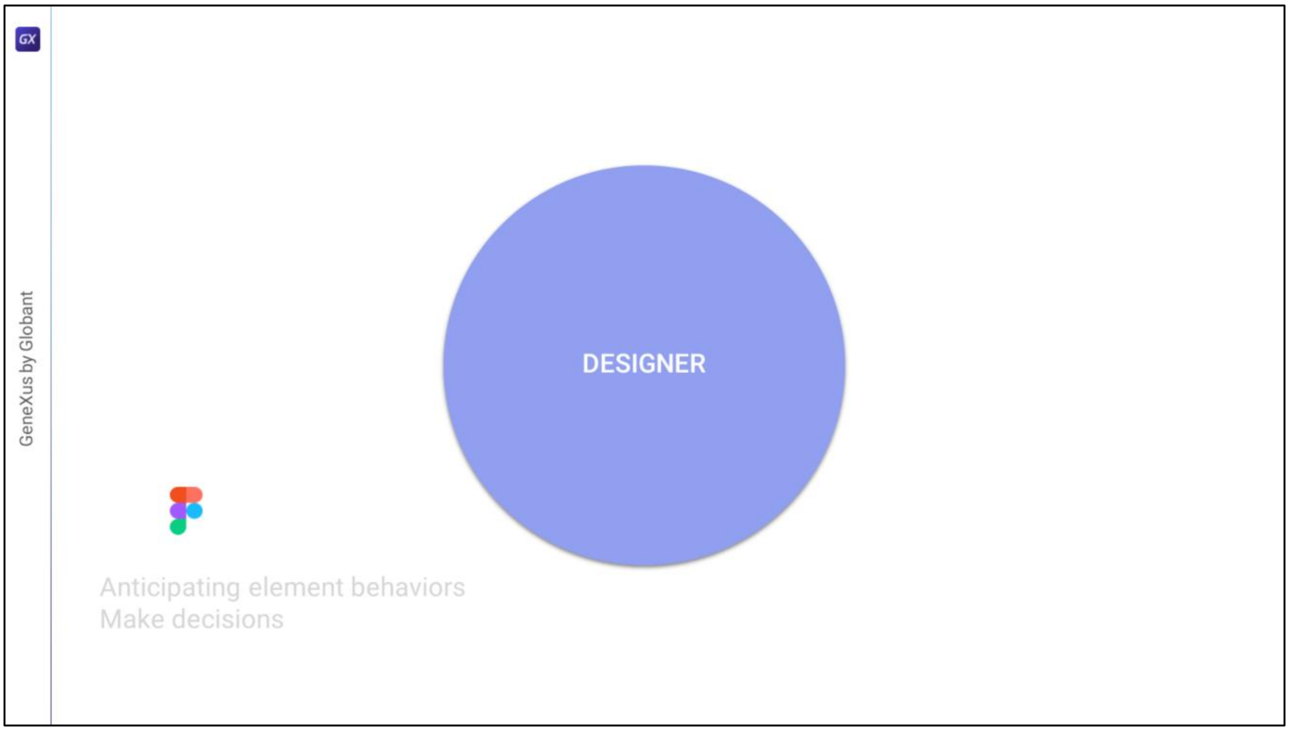
Cecilia Passalacqua

In this class, we are going to talk about Layouts in Figma and what happens to the design when we have small variations in the width of the screen.

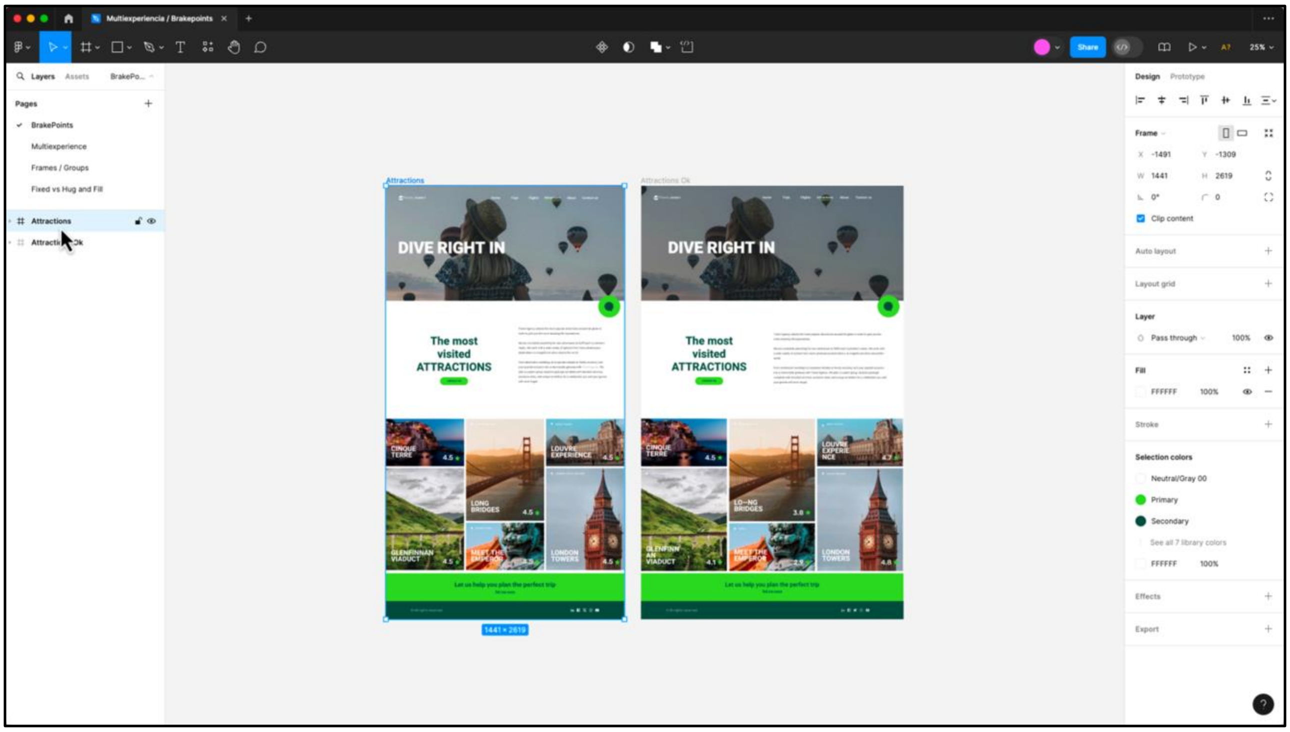


From the beginning of the course, we know that we are designing a customer-facing application that will be developed in Angular for Desktop size.

What we don't know (and we can't know) is the exact size of each Desktop screen that will display my application in the future, as in this case, where we see 3 screens that possibly have small variations in size between one and another.

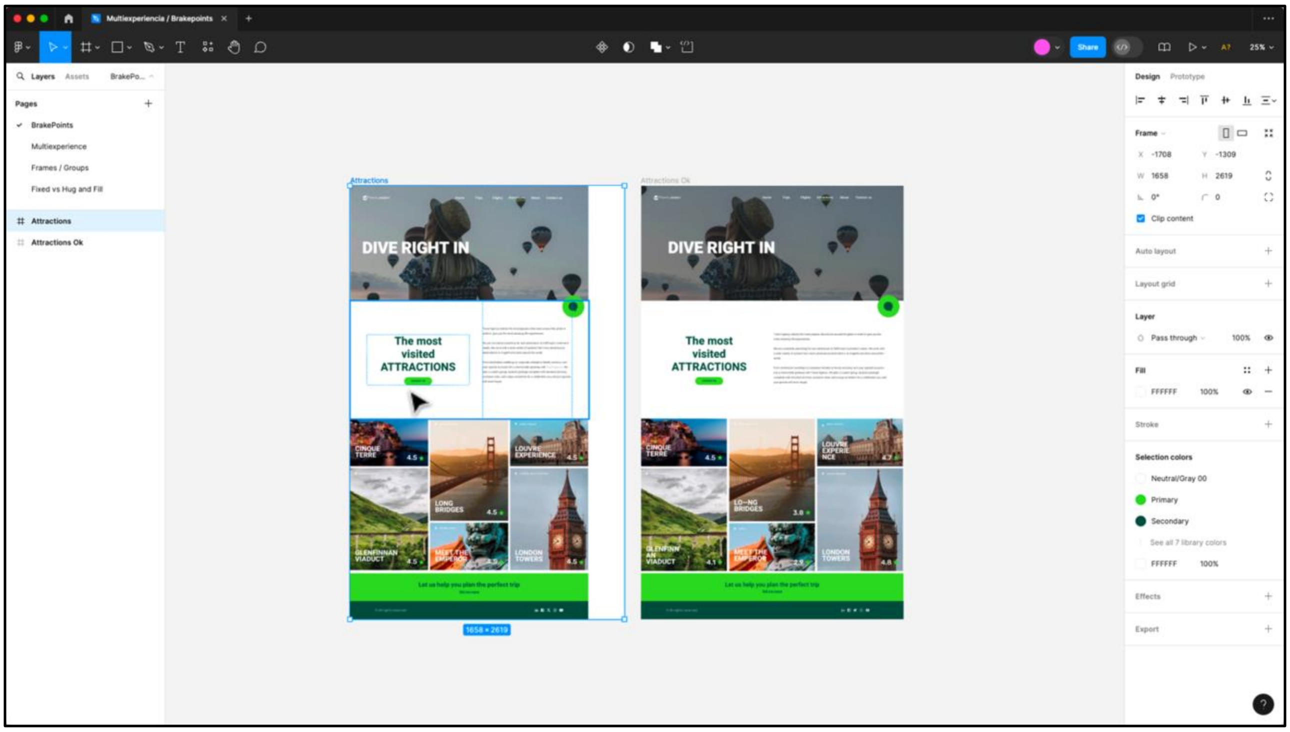


To solve this, as a designer I must anticipate the behavior of each of the elements on the screen and decide what I want to happen with each one in each case.

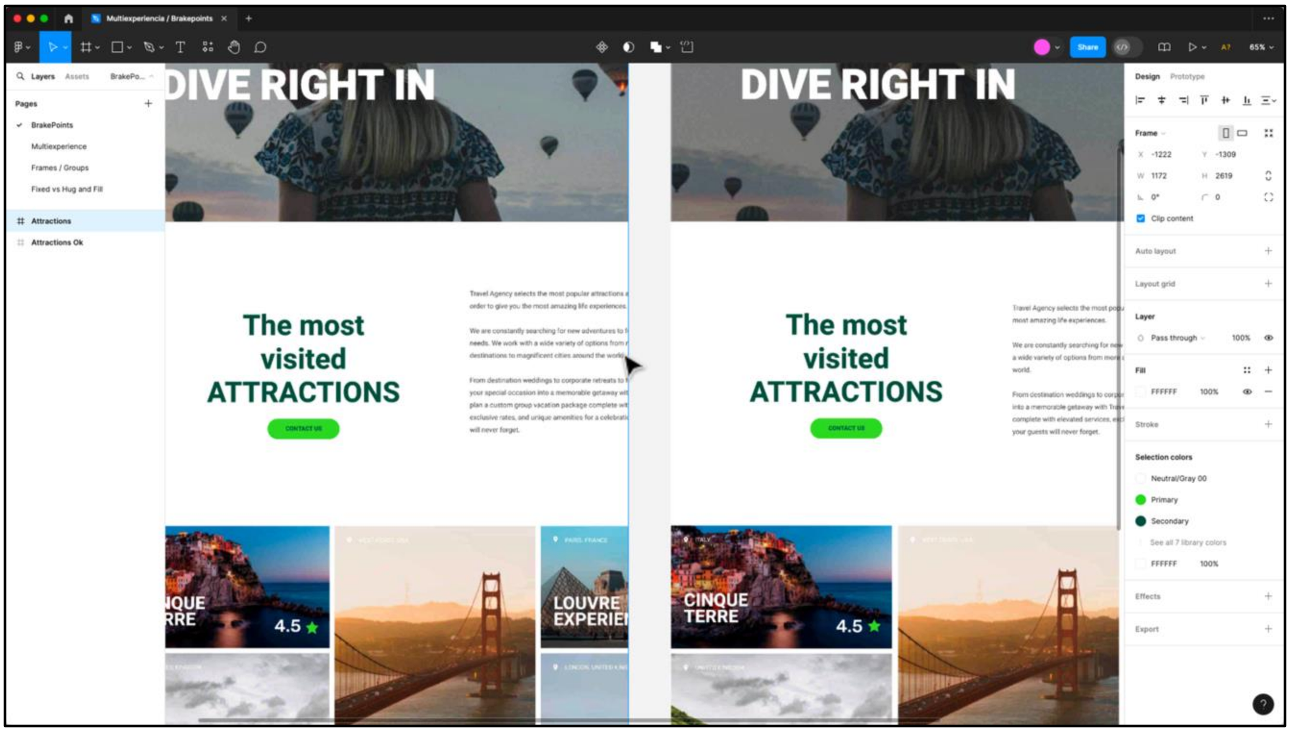


Frame
X -1491 Y -1309
W 1441 H 2619
0° 0
Clip content
Layer: Pass through 100%
Fill: FFFFFF 100%

Let's see it in Figma: here we see the Attractions screen of my application; one of them is in default state and the other has the settings that I mentioned.

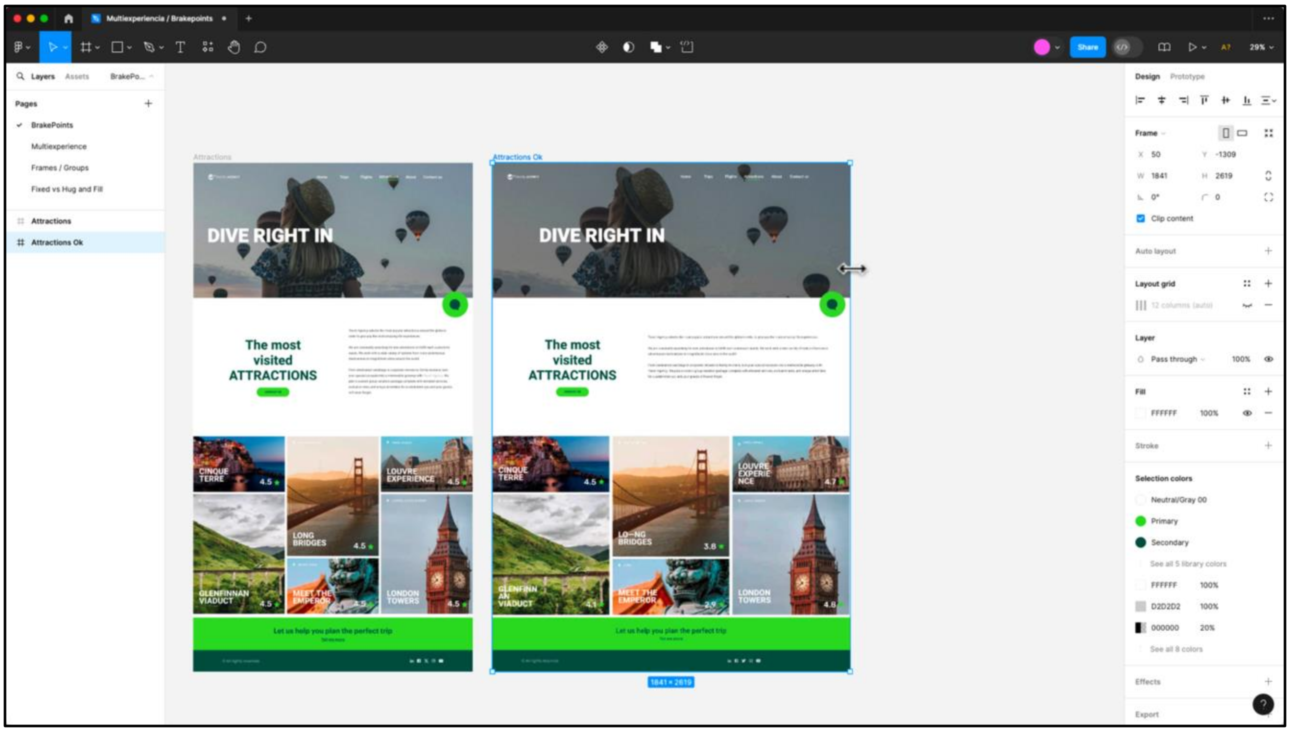


If we increase the width of the screen on the left we will notice that the elements do not change; in fact, they all move to the left. This suggests that these elements are somehow "pinned" to one of the sides.



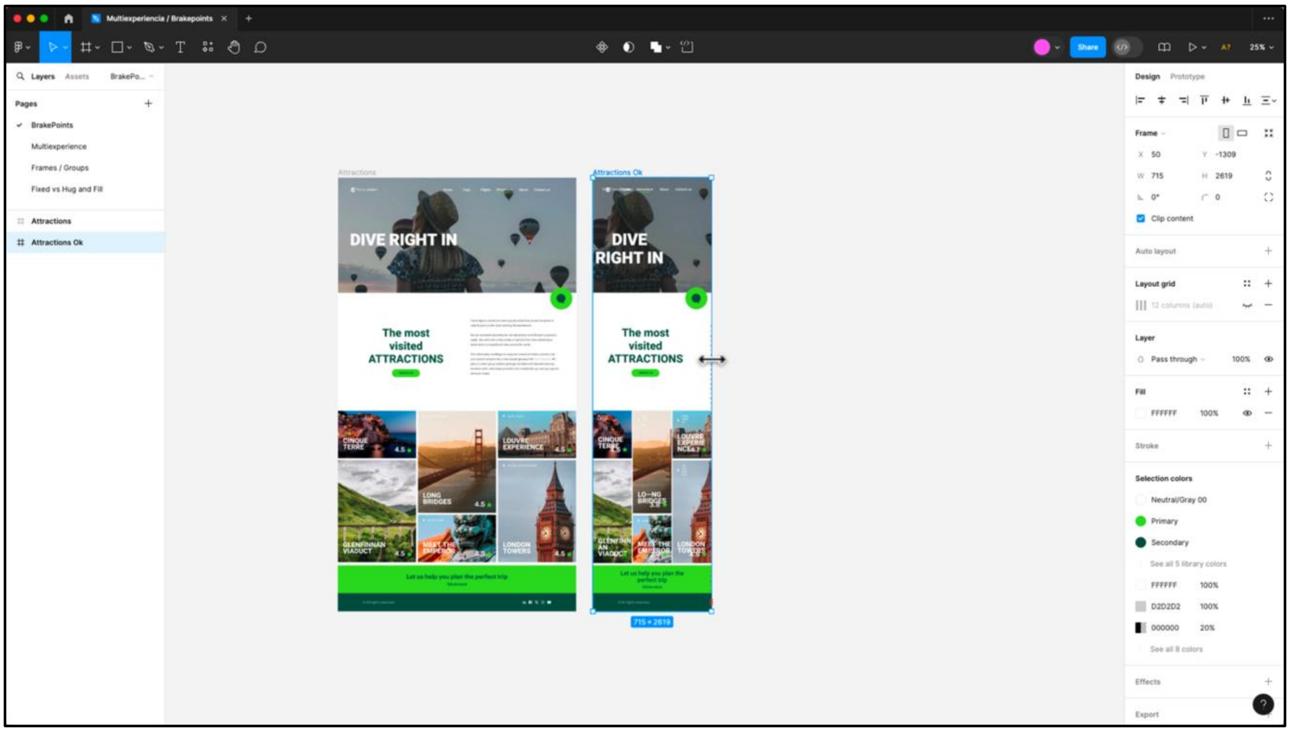
X -1222, Y -1309, W 1172, H 2619

On the other hand, if we decrease the size of the frame, we will note that the screen is cut off.



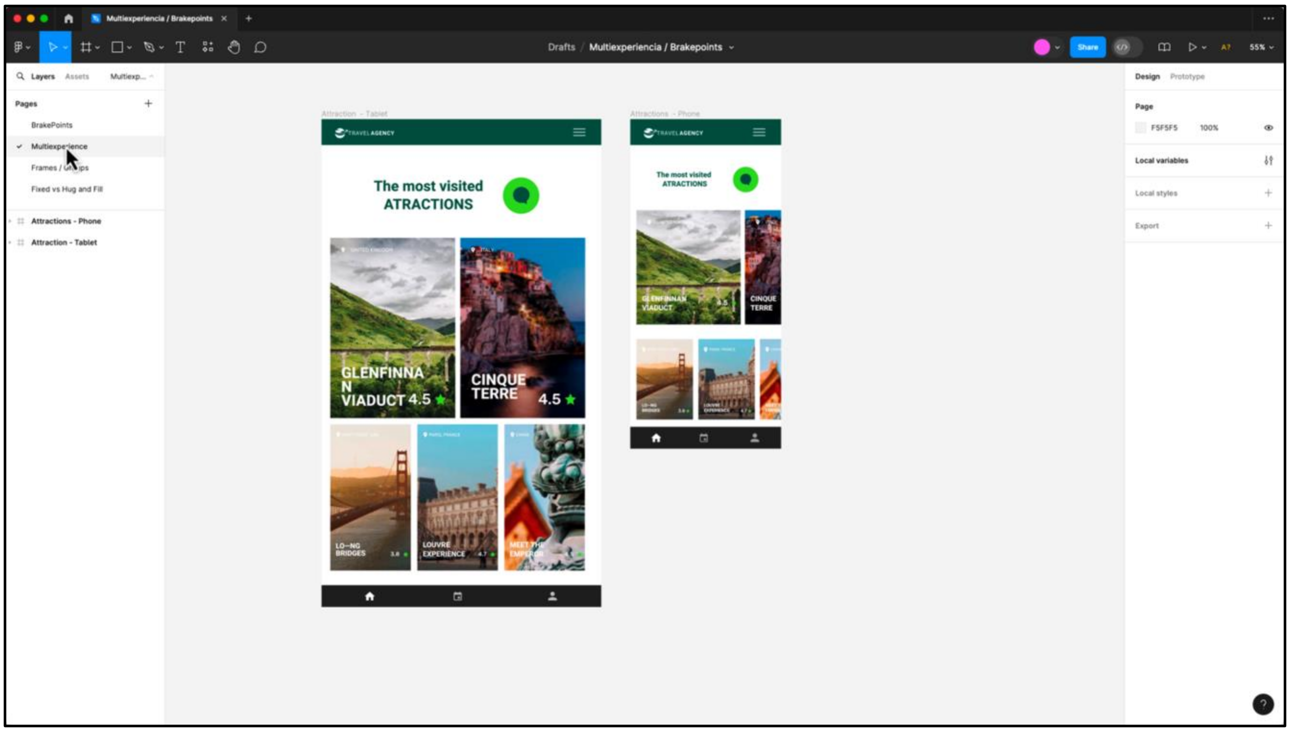
Let's now analyze what happens with the screen on the right. For this case, we make the necessary adjustments so that, if we enlarge or shrink the width of the frame, the elements are scaled to fill the new space.

Of course, these adjustments will be functional if the size variations are small; we cannot expect the same design to readapt from desktop to mobile size.

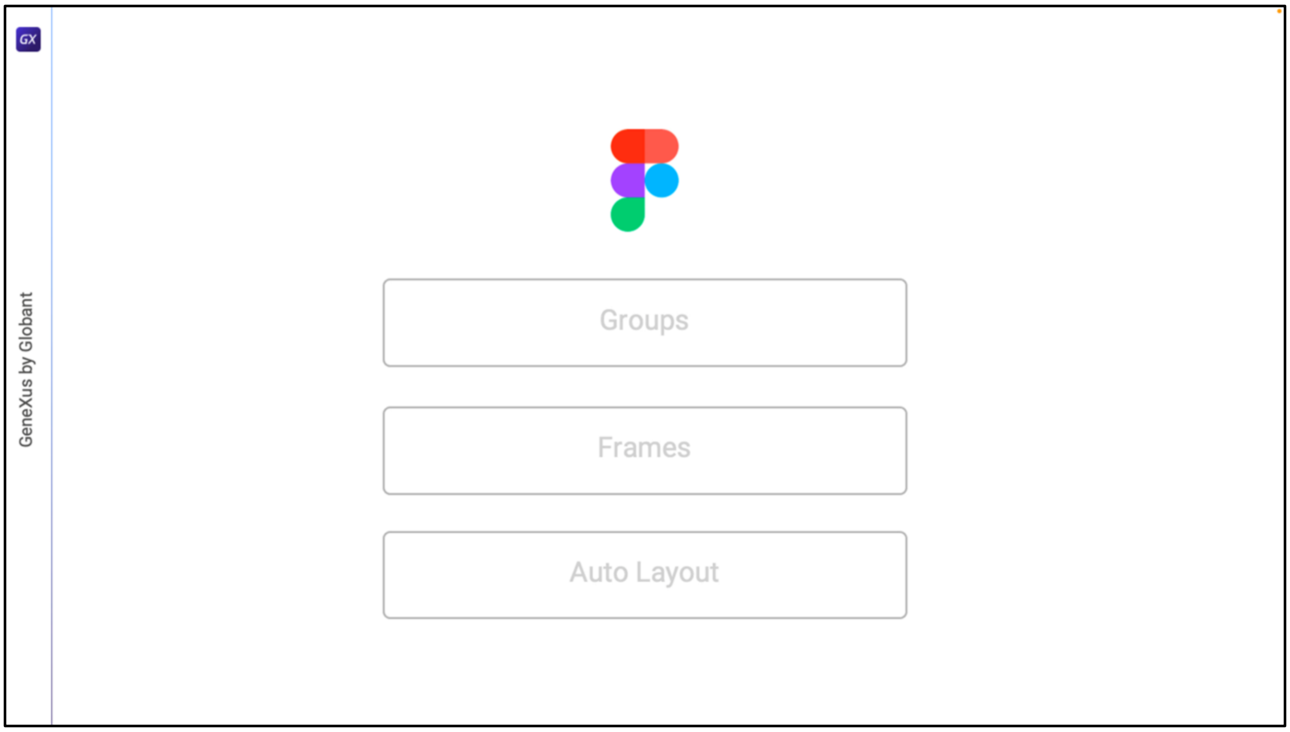


Logically, if the change is abrupt, the design will break, as we see here.

For these cases, we must define breakpoints where the design of the screen is completely different.



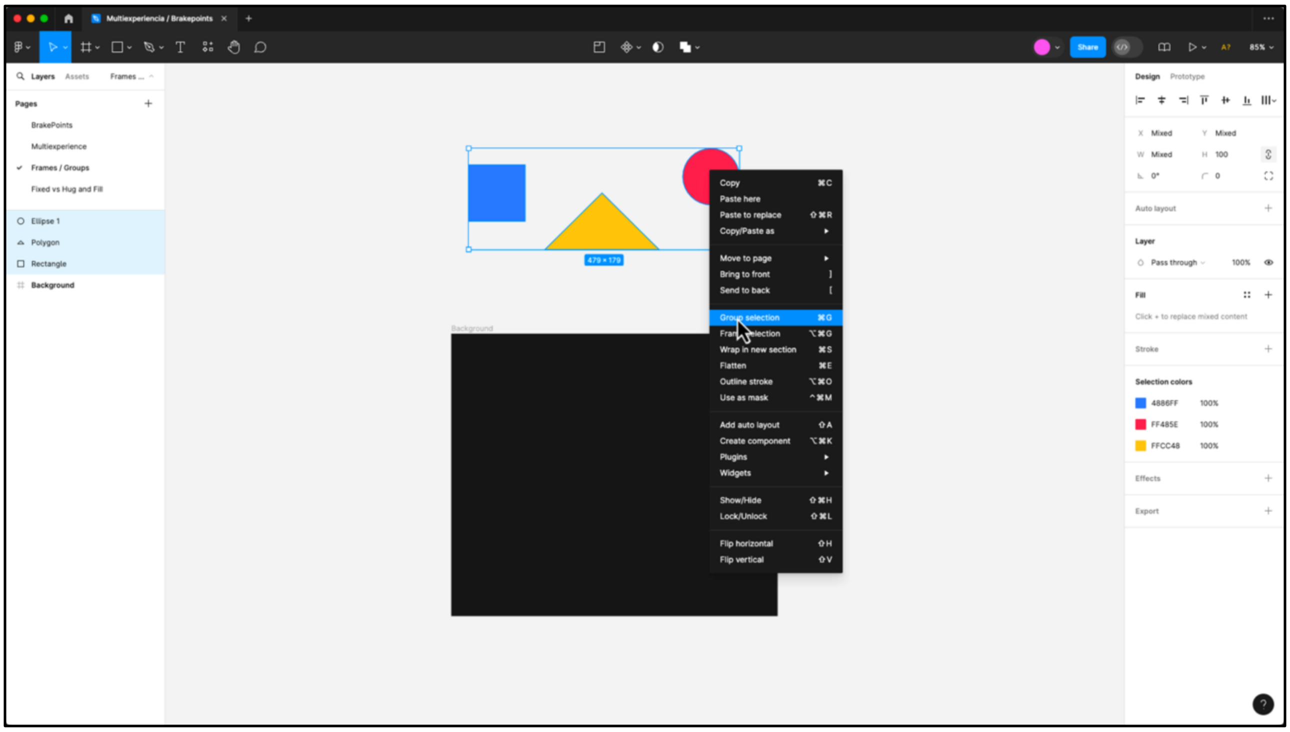
Here, for example, are the design variations of the Attractions page for Tablet and Phone.


GeneXus by Globant
Groups
Frames
Auto Layout

These are all design decisions and I have to make them known to the frontender.

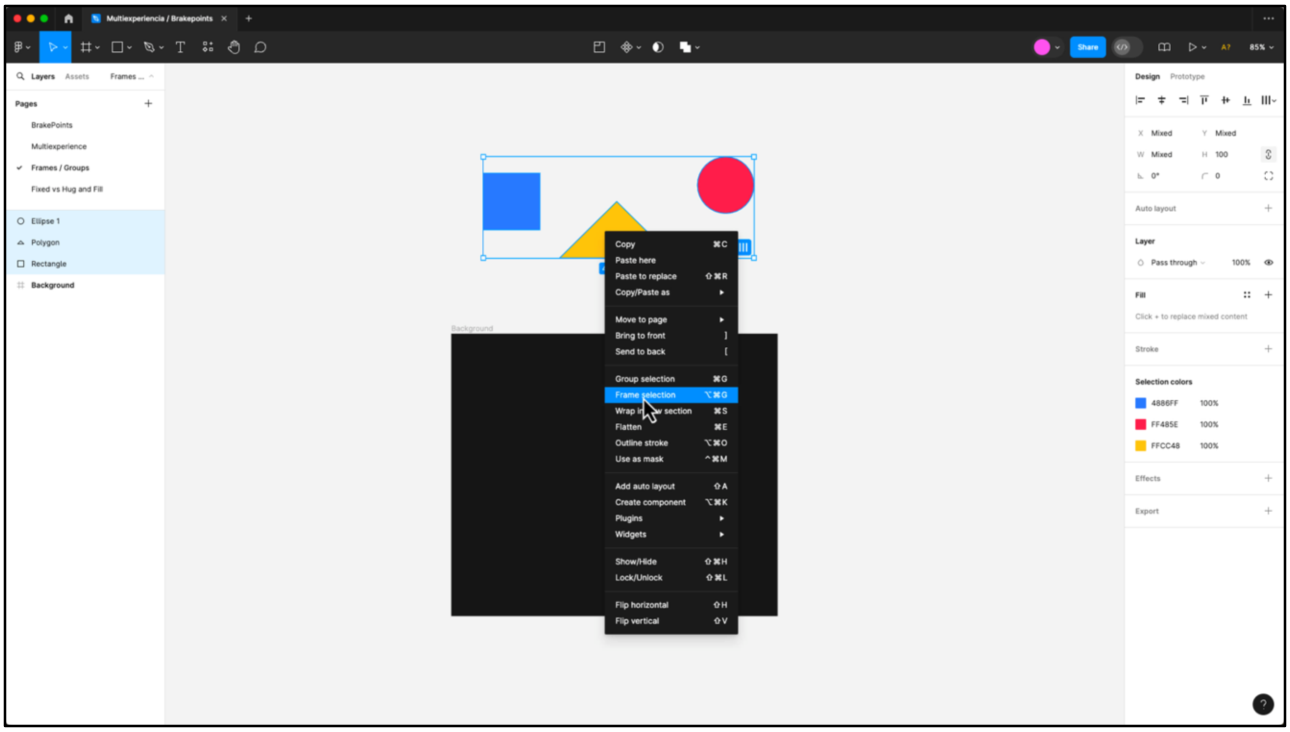
Before we start looking at the properties that allow us to define these elements' behaviors, we have to see the ways I have in Figma to group these elements and the properties of each one.

We are going to talk about Groups, Frames and Autolayout.



In this case, I have 3 independent ways. If I want to group them I can do it using groups or Frames. Let's see their differences.

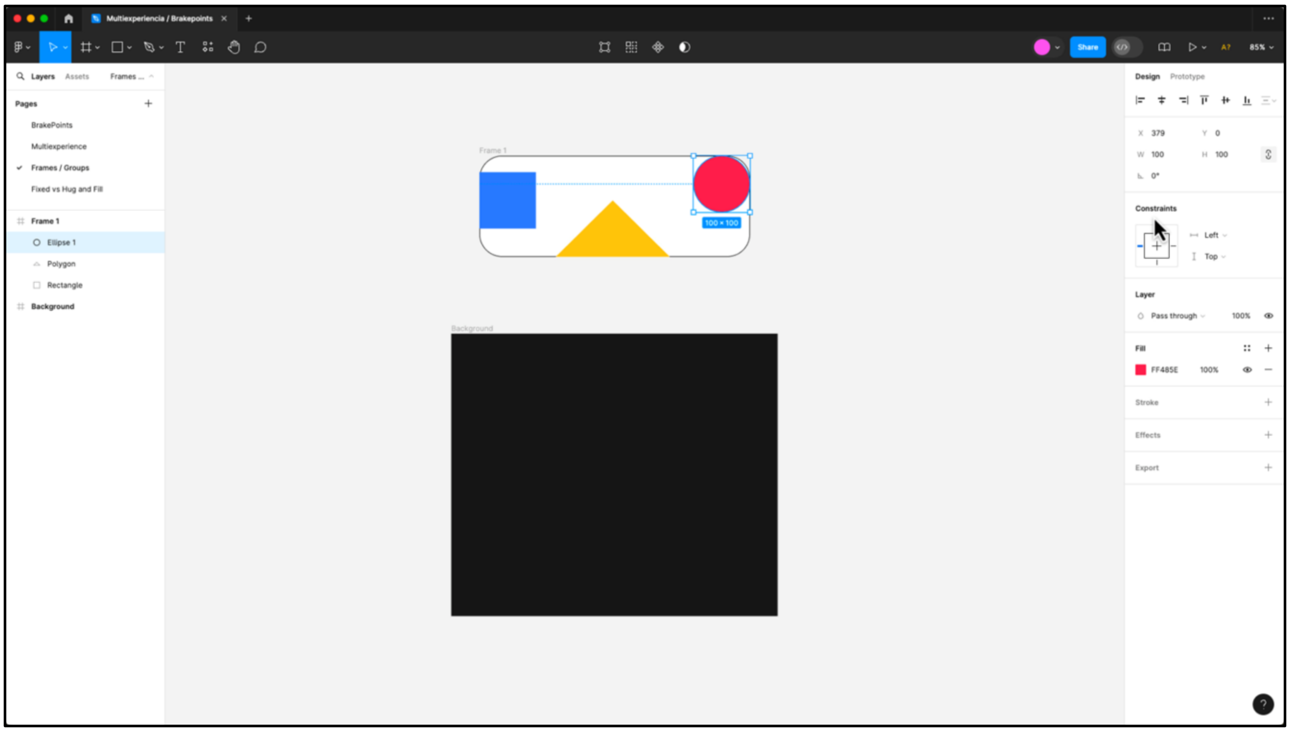
Groups are generally used to move a set of elements. They are displayed in the layers panel with a square of dotted lines. They have no background and if I select the elements inside the group (by double-clicking), I can only see their width, height and position properties.



Frames are containers. In addition to having the elements together, I have them in a space that contains them and that in turn has properties.

They are displayed in the layers panel with a frame icon.

To these frames I can give a background color and a border color, or even round their vertices.



X 379 Y 0
W 100 H 100
0°

Constraints
Left
Top

Layer
Pass through 100%

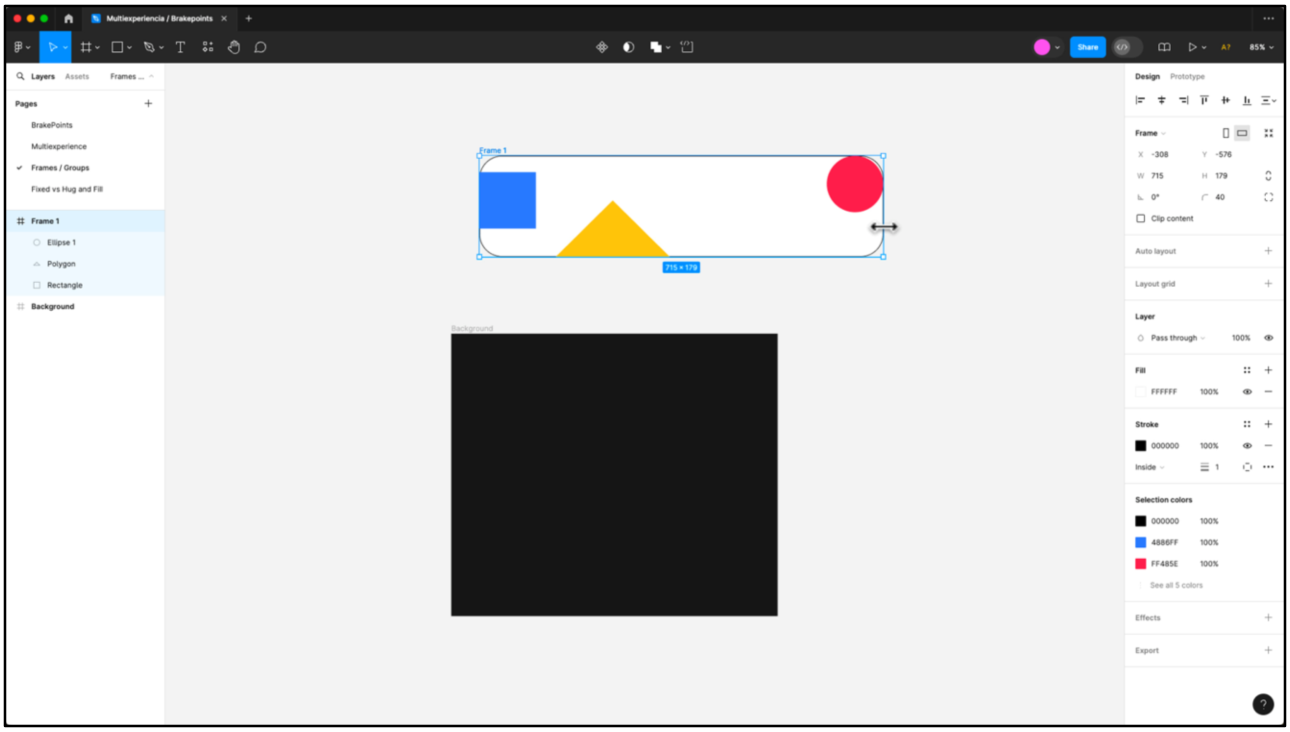
Fill
FF485E 100%

Stroke

Effects

Export

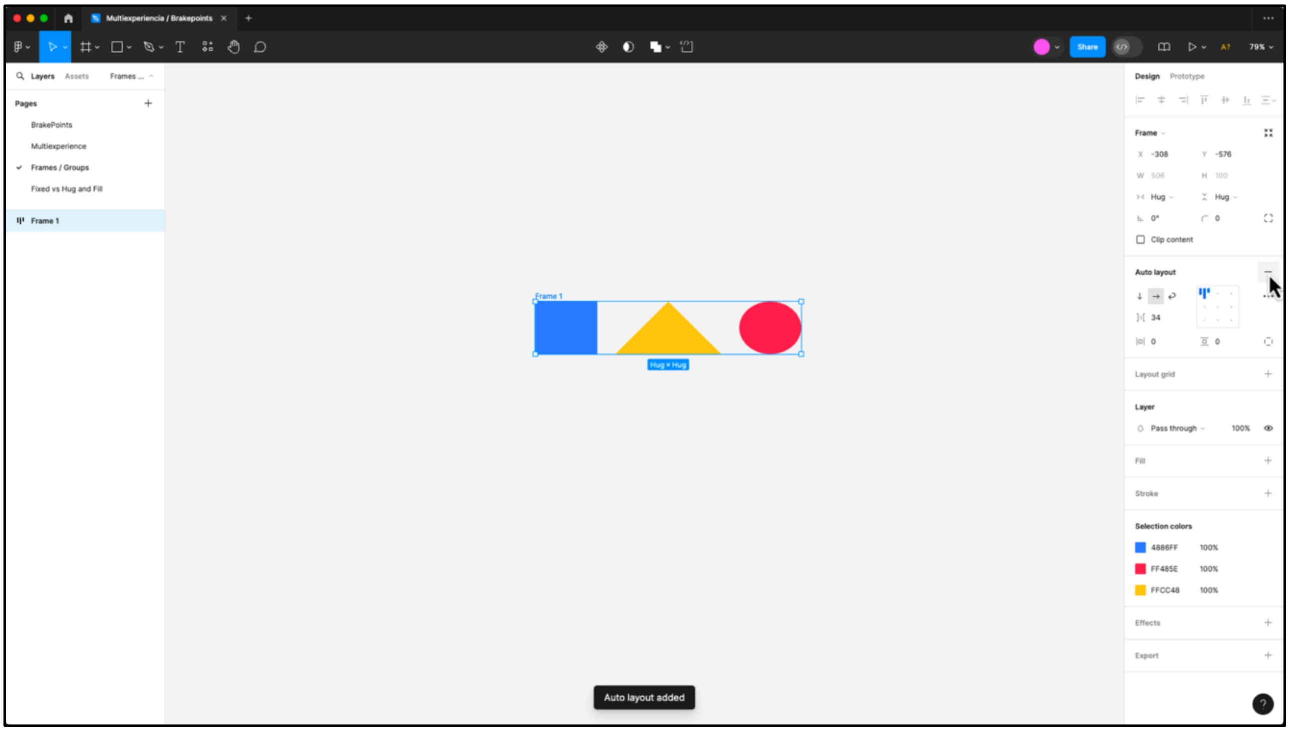
If I select the elements inside the frame, in addition to the properties I mentioned in the groups, a "constraints" section is added. By default all the elements are pinned to the left, just as they were in the Attractions screen without adjustments.



Frame
X -308 Y -576
W 715 H 179
0° 40
Clip content
Layer
Pass through 100%
Fill
FFFFFF 100%
Stroke
000000 100%
Inside 1
Selection colors
000000 100%
4886FF 100%
FF485E 100%
See all 5 colors

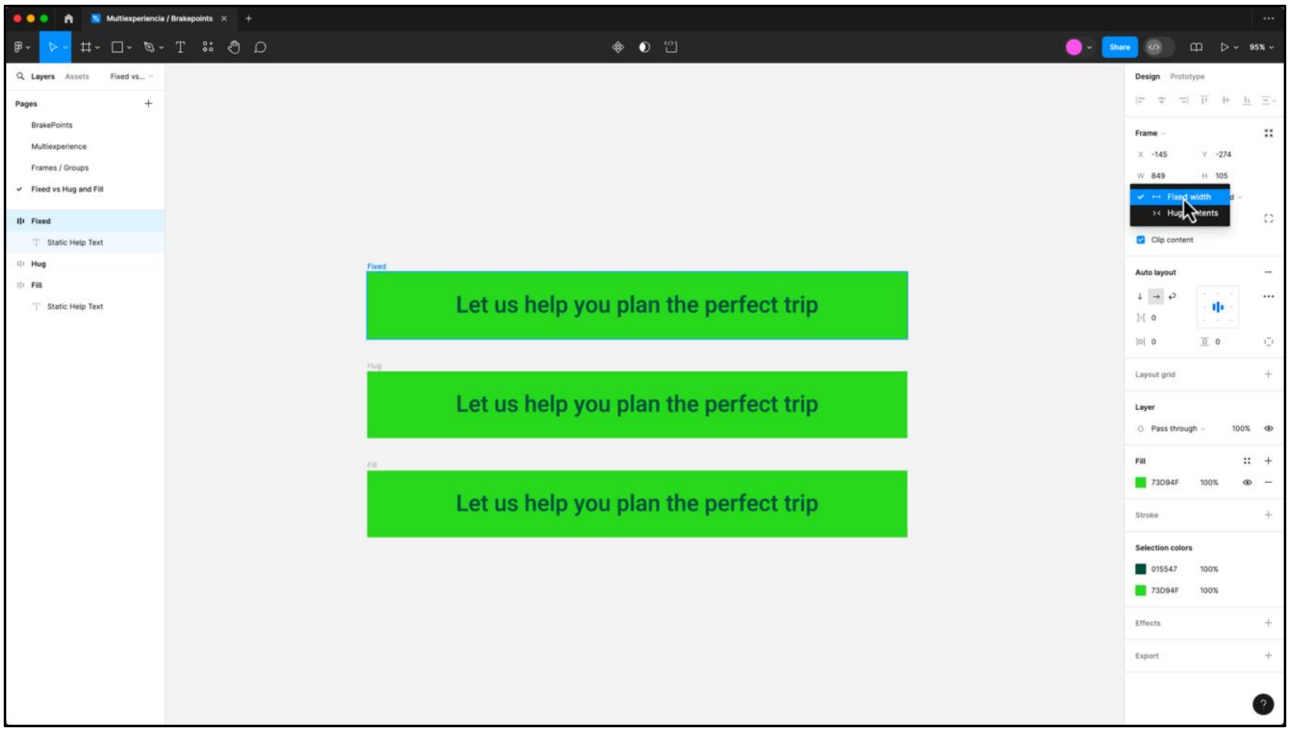
Constraints are some of the properties that will be used to define the behavior of the elements.

For example, if we modify the width of this frame, all the elements remain on the left. However, if we pin the ellipse to the right, for example, we see that it stays on that side when we modify the size of the frame.



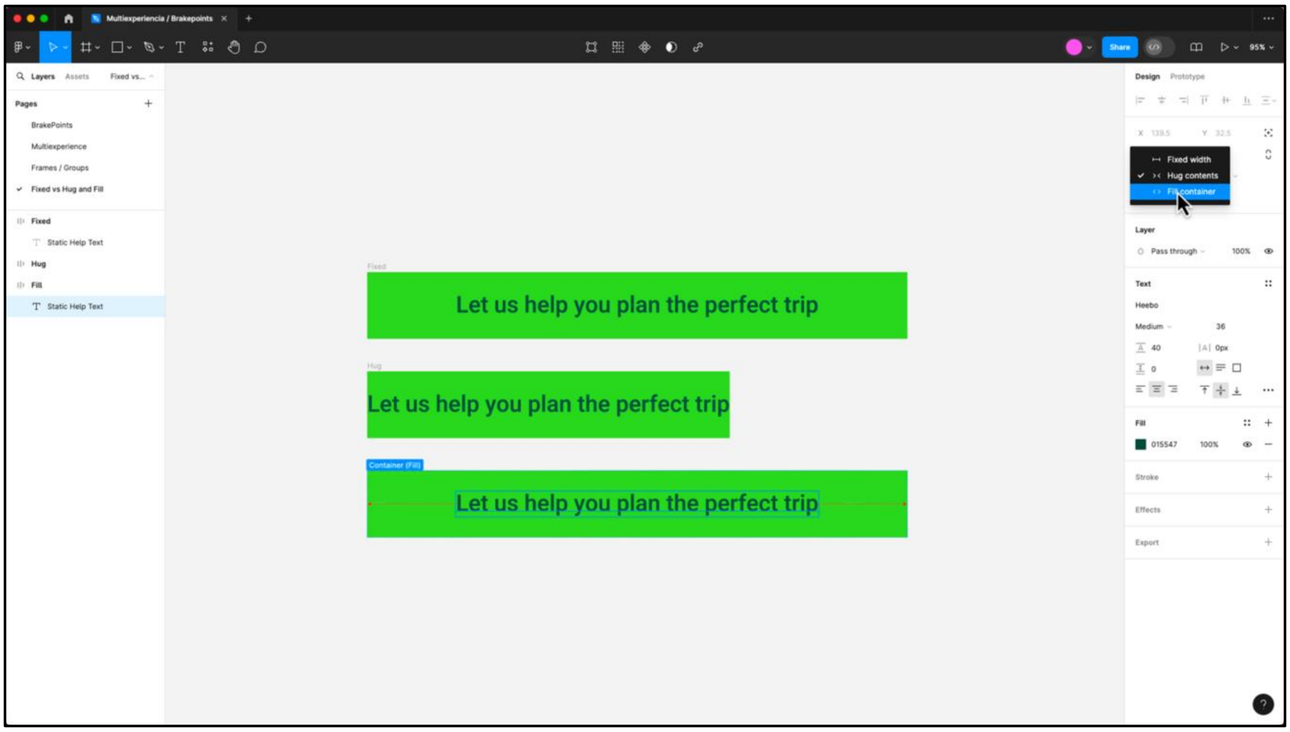
Another property that I can set to both Groups and Frames is AUTOLAYOUT.

This will allow us to organize this set of elements, defining in which orientation I want them to be directed, what is the distance between them, how I want them to be ordered among themselves and if they will have padding on their sides or top and bottom.

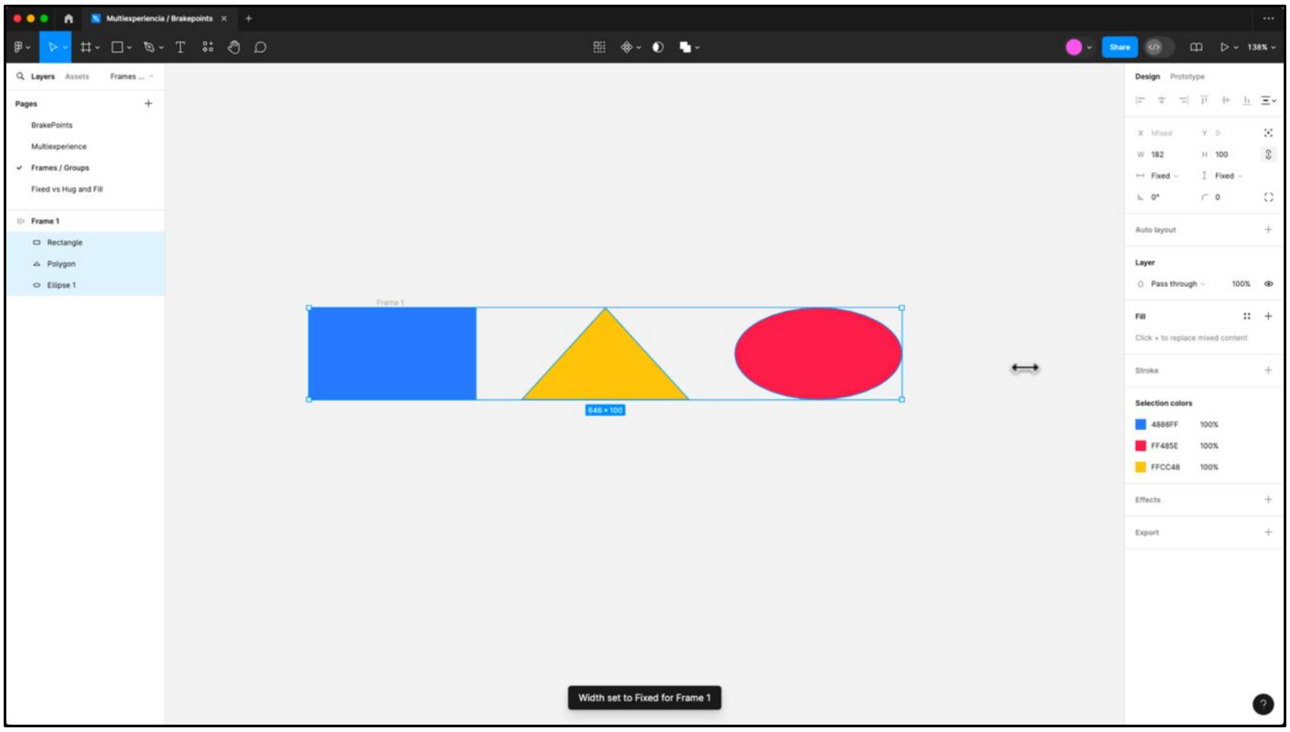

Fixed
Let us help you plan the perfect trip
Hug
Let us help you plan the perfect trip
Fill
Let us help you plan the perfect trip
Fixed width
Hug contents

In addition, when we use a container with autolayout and inside we have one or more elements, we can define how the container behaves in relation to its content and its content in relation to the container.

Within the container properties we have Fixed or Hug, that is to say, it adapts to its content.



For the case of the content, we find the same 2 options of fixed and hug plus the fill container option. In this example, we see that the text field is stretched to reach the edge of the container, without any major problems.

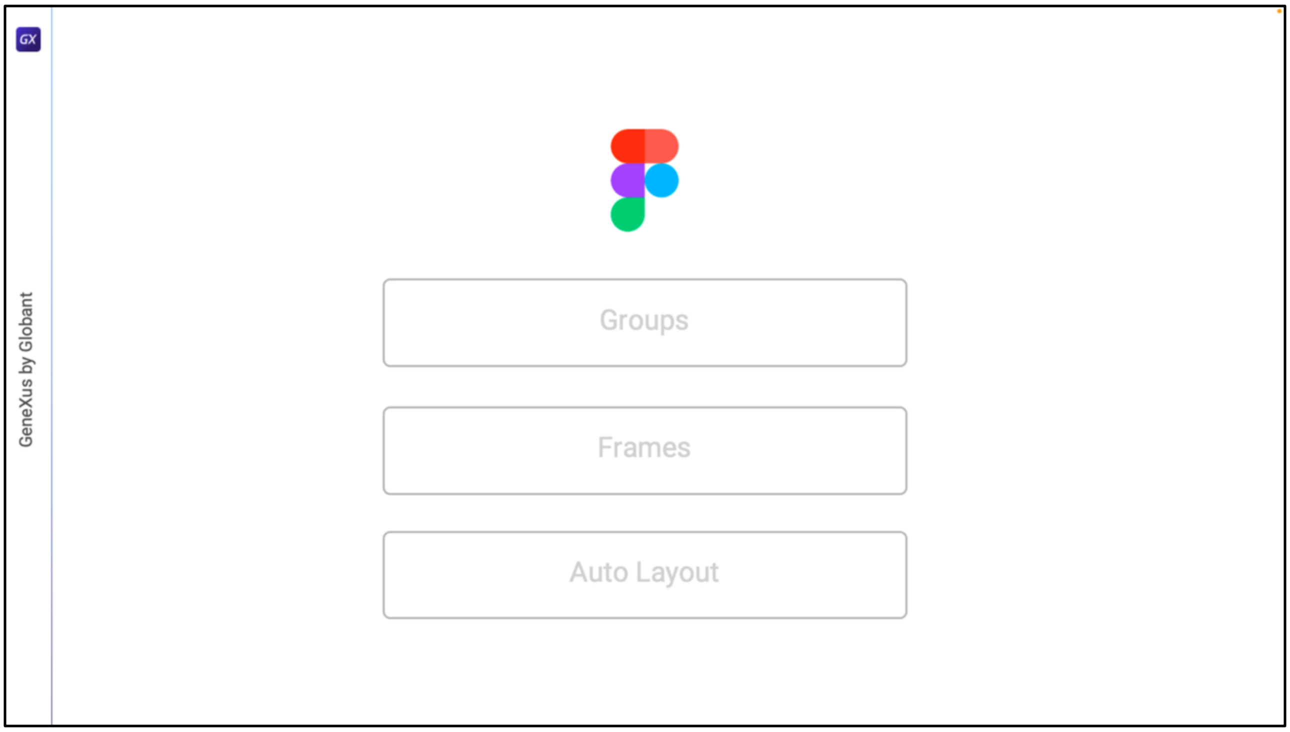


X Mixed Y 0
W 182 H 100
Fixed Fixed
0° 0
Pass through 100%
Click + to replace mixed content
4886FF 100%
FF485E 100%
FFCC48 100%

Width set to Fixed for Frame 1

But if we take the previous case as a reference, and each of the figures that are inside the Auto Layout were set to Fill container, when stretching that container they would be distorted completely.

That's why it is essential to analyze what happens in each particular case and remember that all the information we place in the design file will somehow be matched in GeneXus, as you will see later with Ceci.

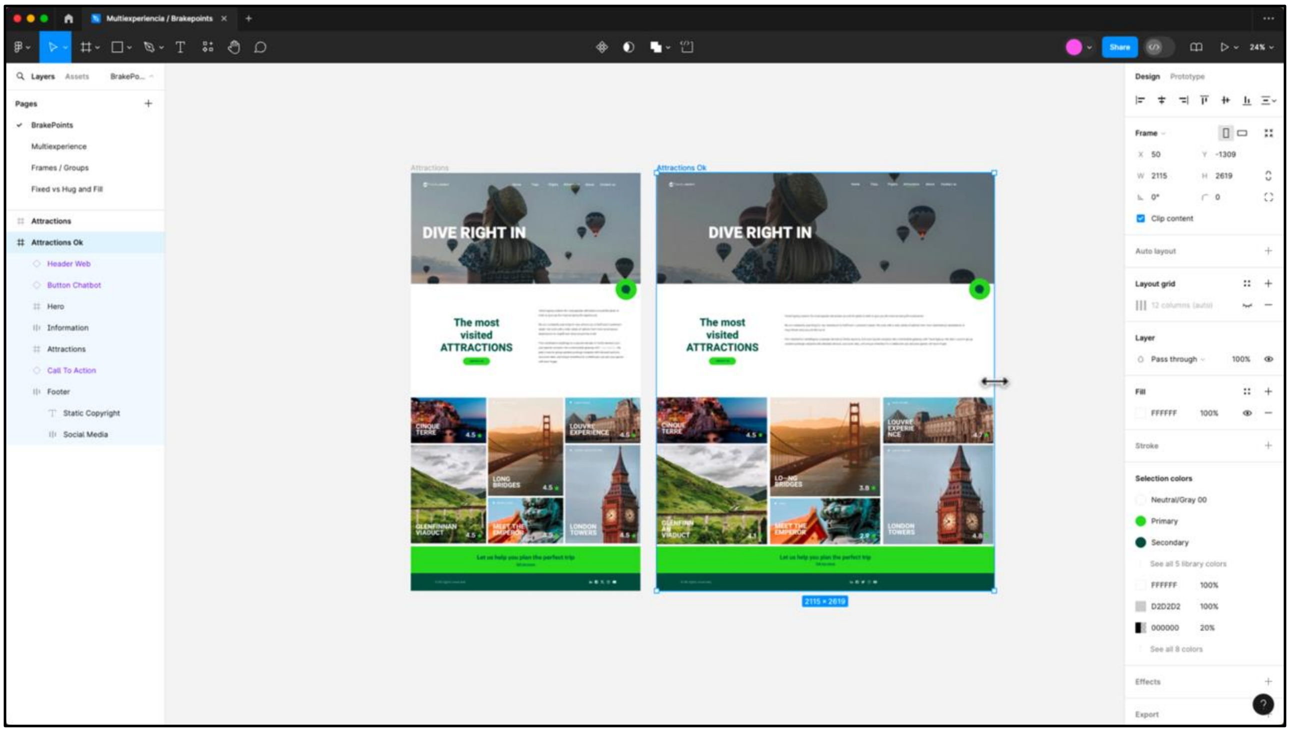

GeneXus by Globant

Groups

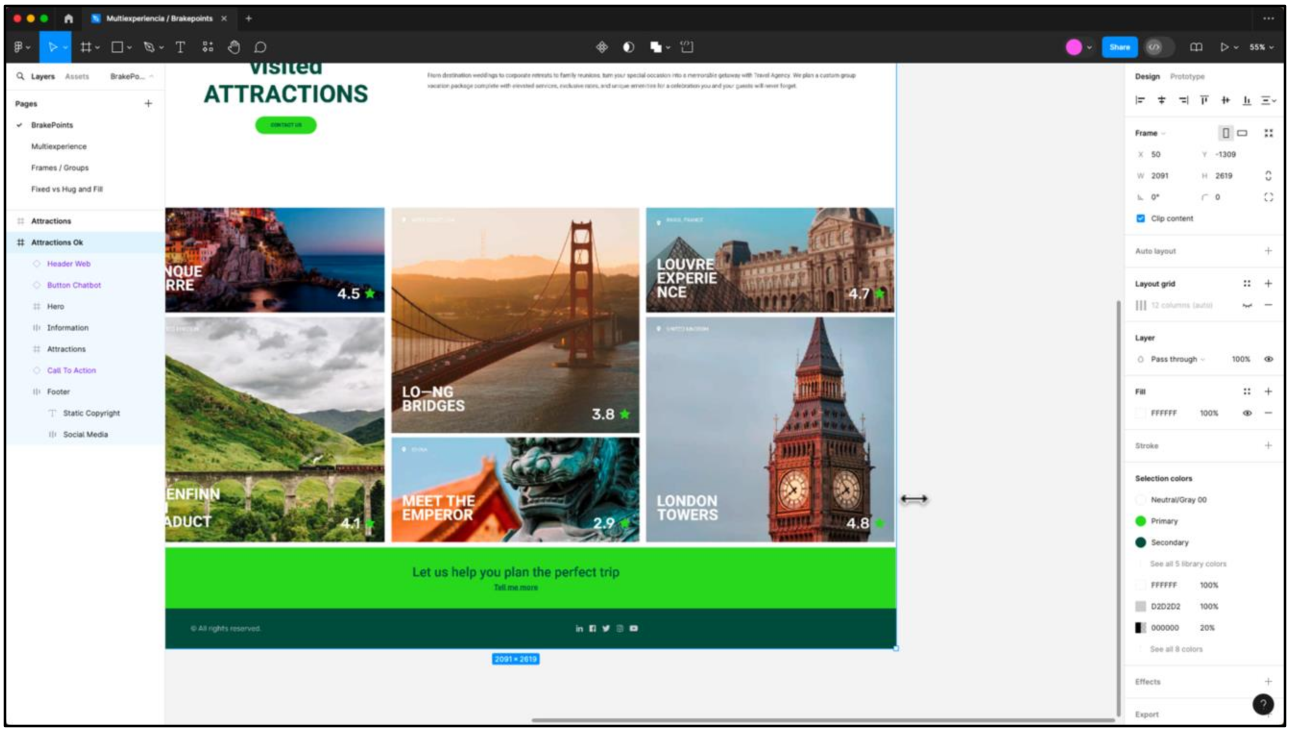
Frames

Auto Layout

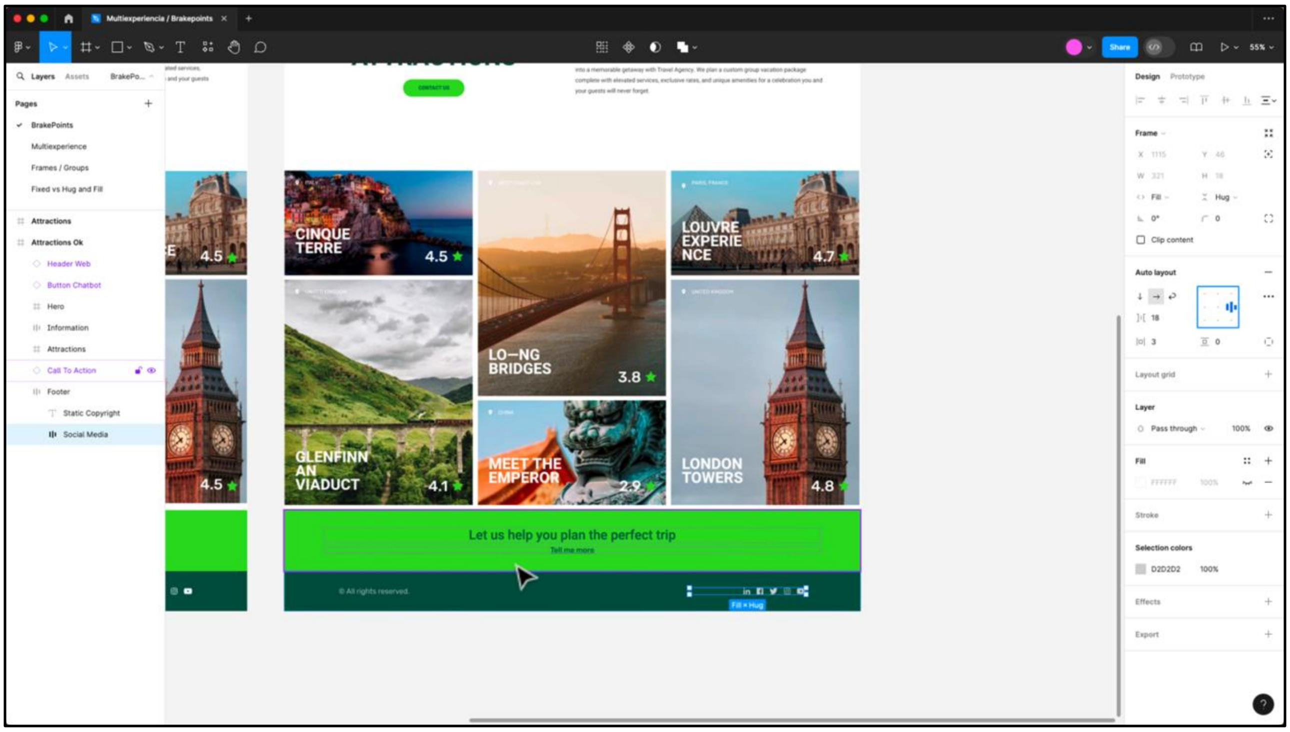
As a summary of this class, we could say that to define how each on-screen element will behave when there are small variations in their width, I'm going to work preferably with autolayout frames, constraints properties and the Fixed, Hug and Fill properties.



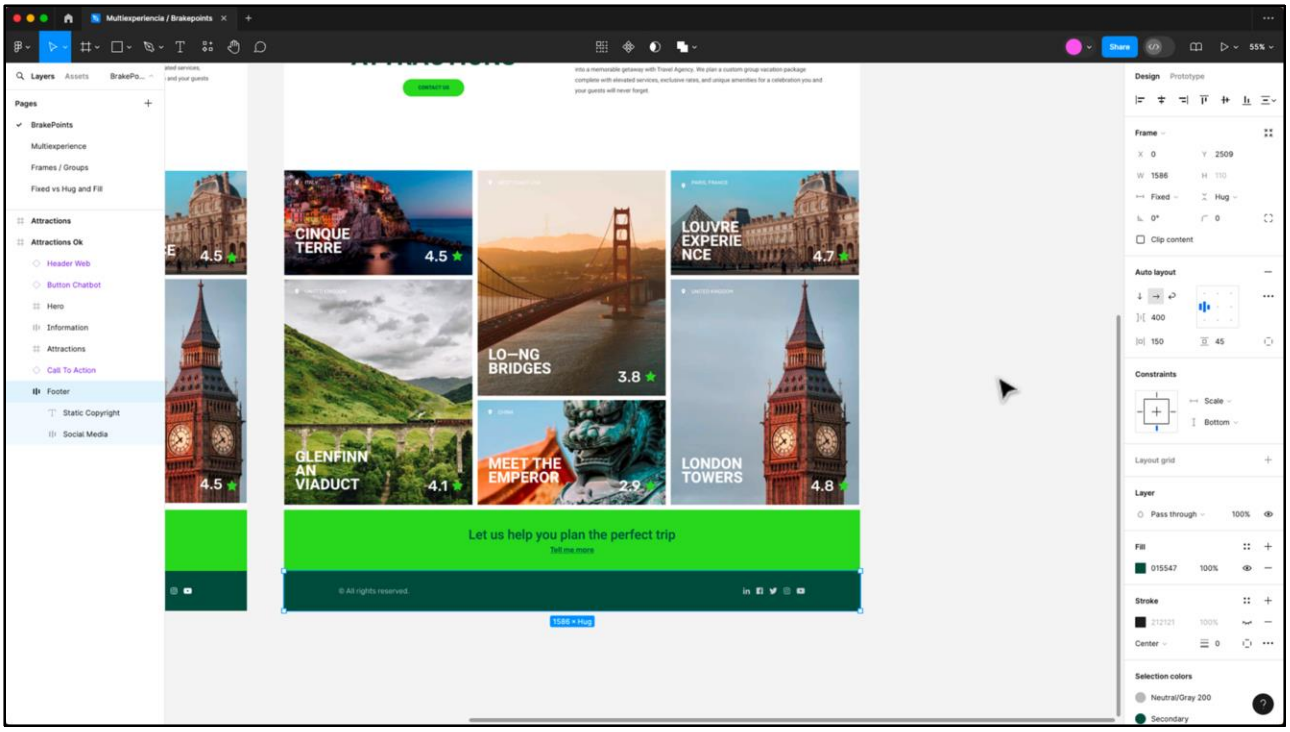
Now let's go back briefly to the initial screens. Did you notice at the beginning of the class that when I modified the screen settings something happened in the footer?



Let's take a closer look… can you see? Let's modify it.



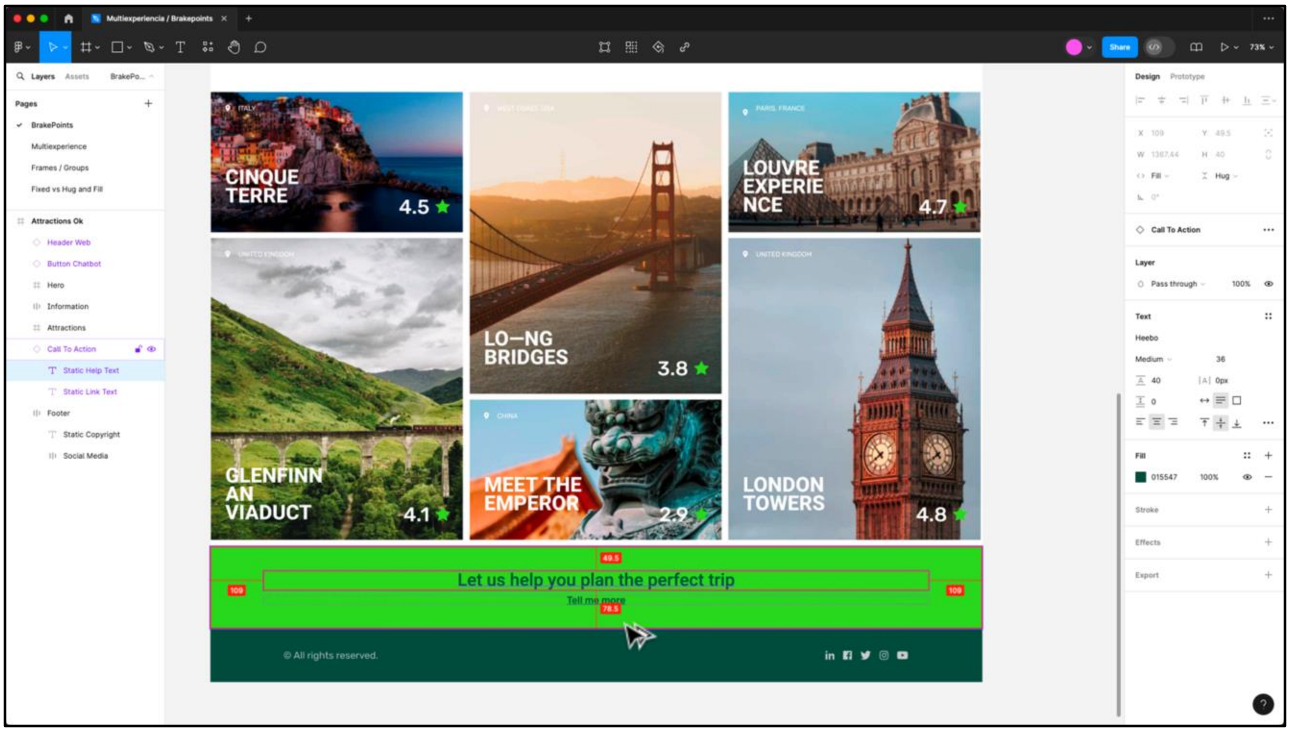
I think that if we set the set of social network icons with the Fill behavior, and inside the autolayout we pin it to the right...



X 0, Y 2509, W 1586, H 110, Fixed, Hug, 0°, 0

Fill 015547 100%

...and leave the Footer container fixed and pinned to the left, this behavior will be resolved.



Finally, I want to share with you a simple way to measure the distance between elements. If I select an element and press the "option" key in Mac or "alt" in Windows and hover the cursor over another element, it will show how far apart they are.

In the next video, you will work with Ceci on how to interpret this design file and implement it in GeneXus so that the application looks and behaves in the way we have just defined.

GeneXus by Globant

GeneXus™ by Globant

training.genexus.com