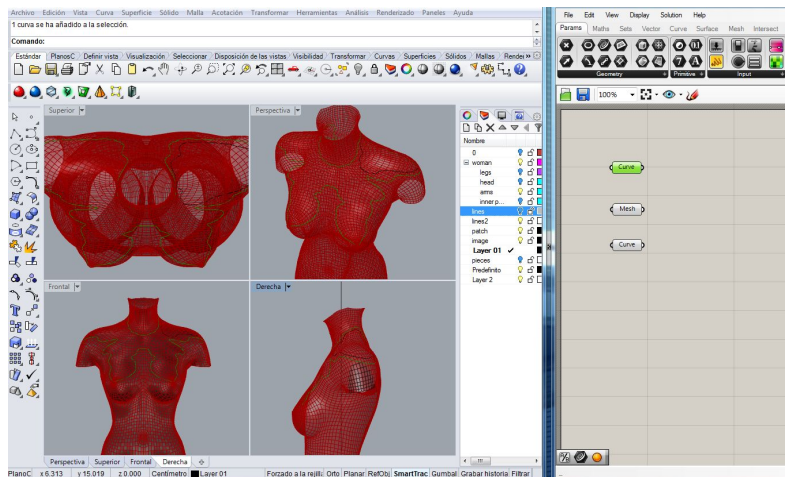
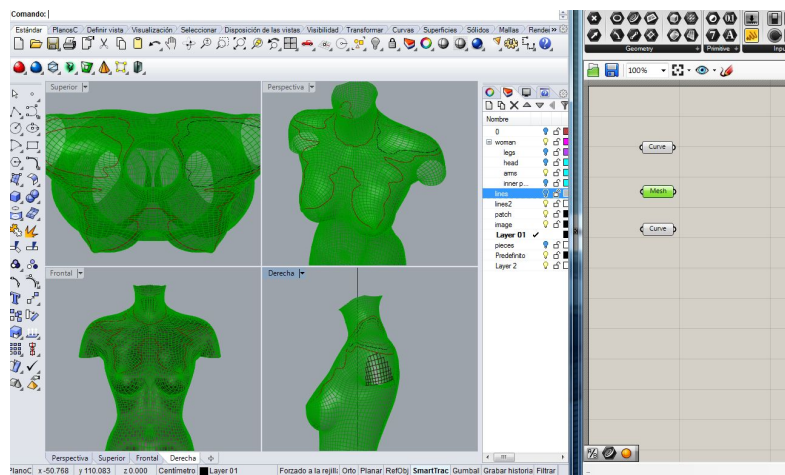


Once we have the body's stl with some curves on it, here we have 3 main components:

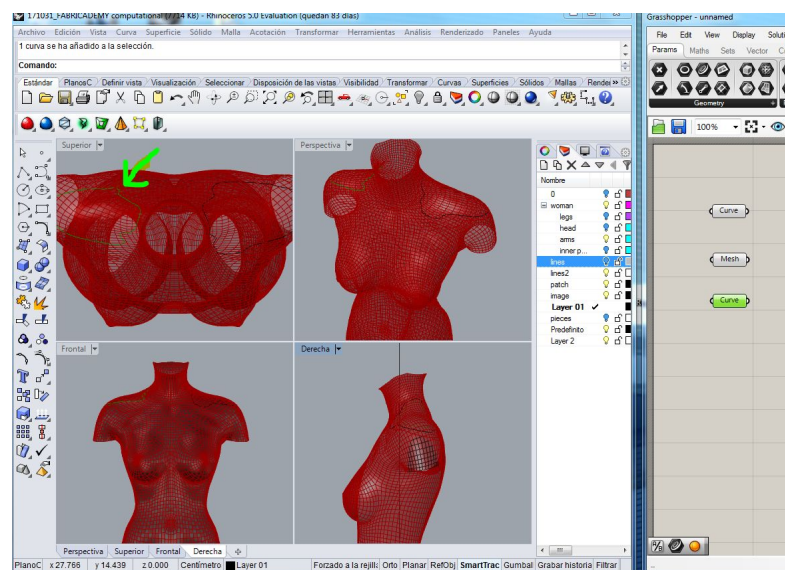
1. curve: Is a curve on the body: Marked in green



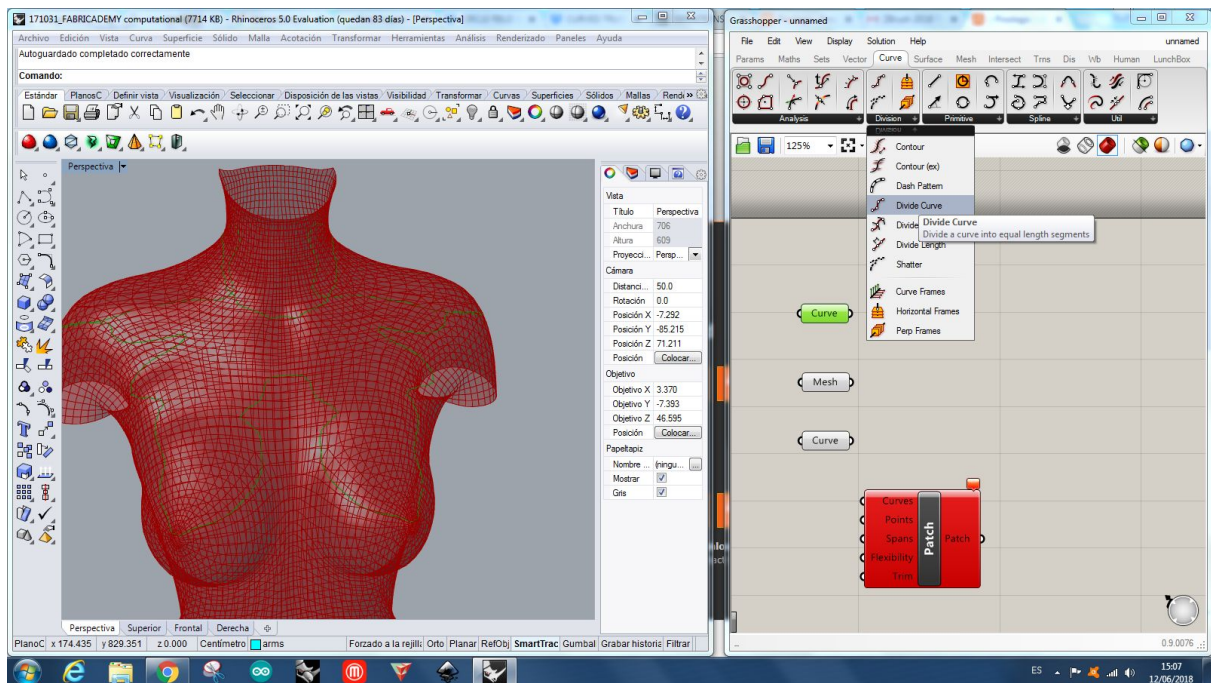
2. Mesh: Is the body



3. curve for patch

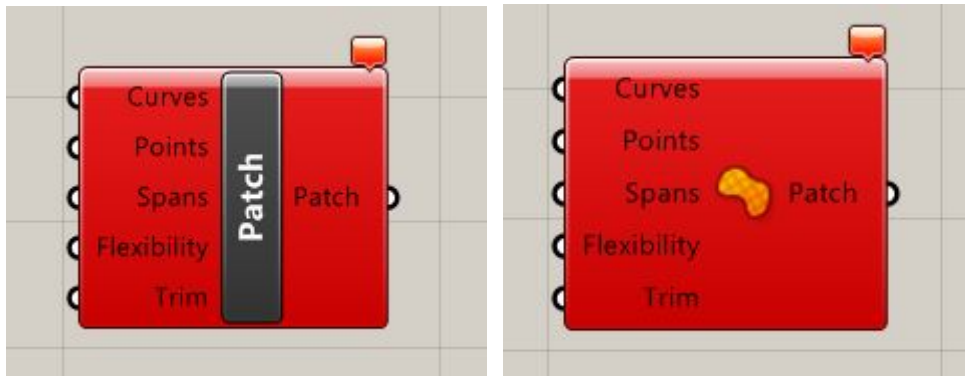


The main purpose is to cut or build a new surface and remesh the result (z brush). We will use a path component



We need a *patch component*. The difference between the boundary surface and the patch is:

- Boundary surface is planar over a planar curve
- Patch is no planar. created over a 3d dimensional curve

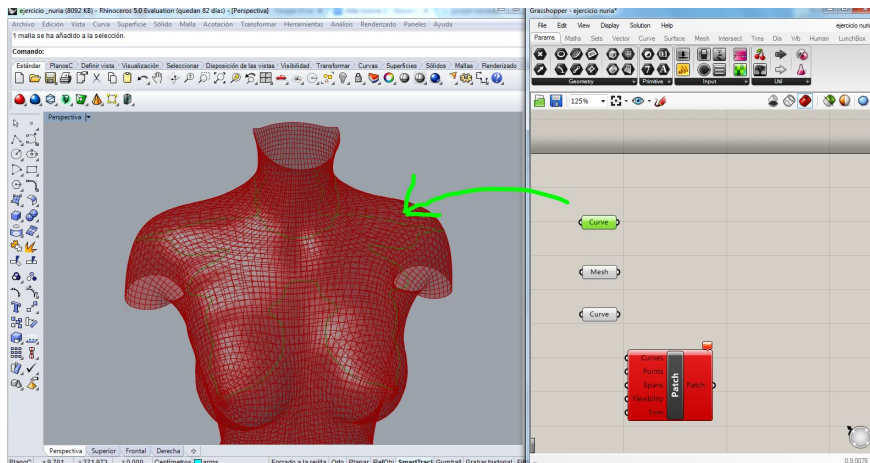


In the input we have:

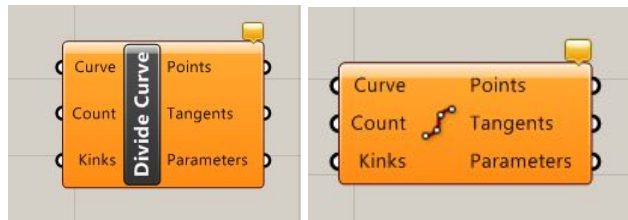
- *Curves*: We need to patch a curve
- *Points*: The points we want to patch (internally)
- *Sliders*: to increase the level of accuracy

## First step

We have the first curve and we want to subdivide this curve into a lot of curves



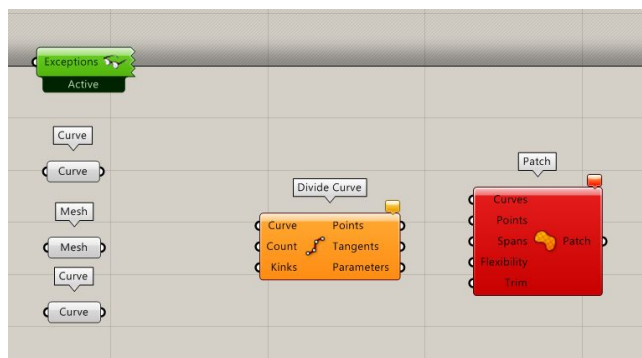
We will that with *Divide curve component*



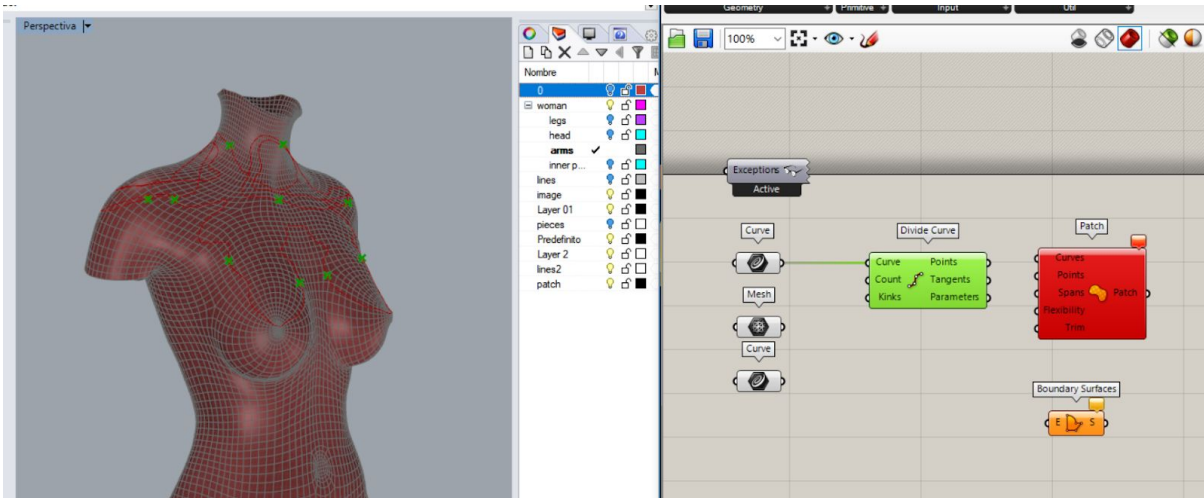
Here we place [bifocals](#) componen(to see icons and text)



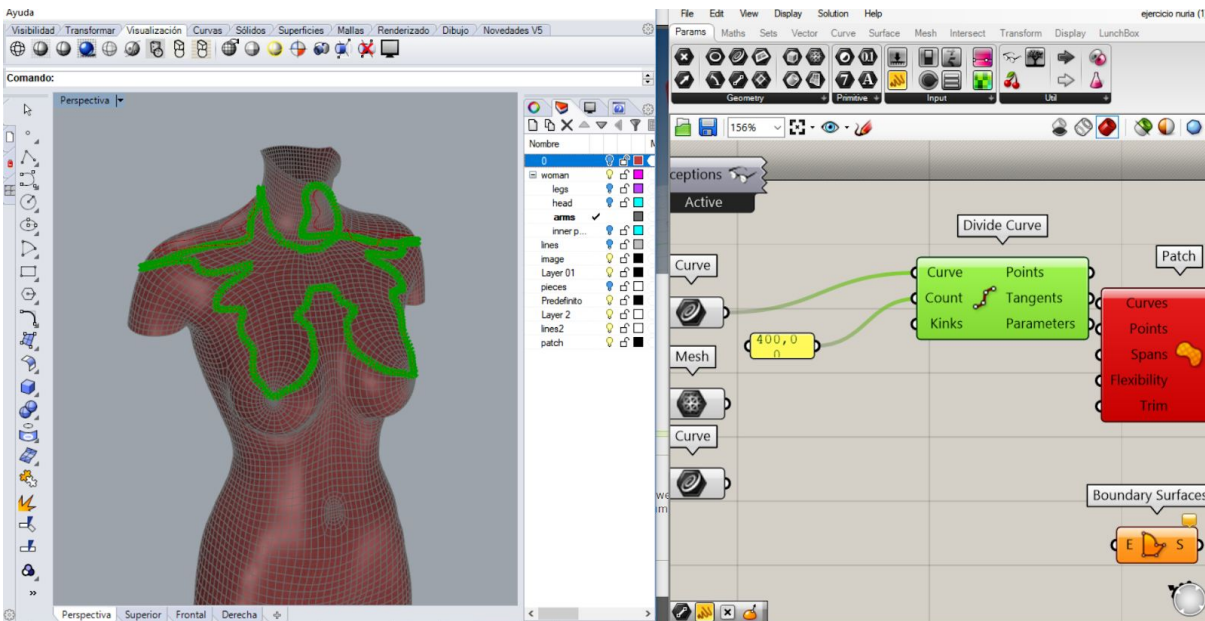
Once we place this component (we have to place it once!!) the appearance of the script will be with tags (yupi!)



In defect, the curve is divided into 10 points, and it is not enough, so, we need to subdivide the curve that will form the patch into more points



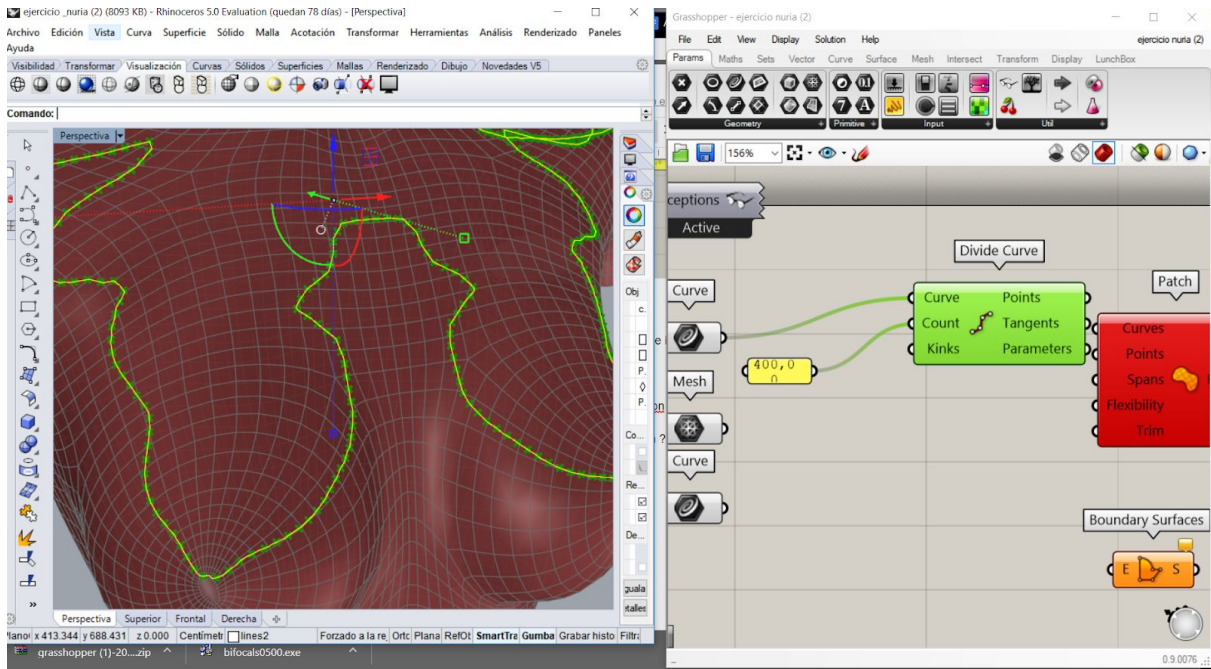
For doing this, we will add a panel and put here 400 (the number of divisions). Note: if we get an error here like: *"Data conversion failed from text to integer"* we have to add a coma and put 400,00 so that it is considered as a number.



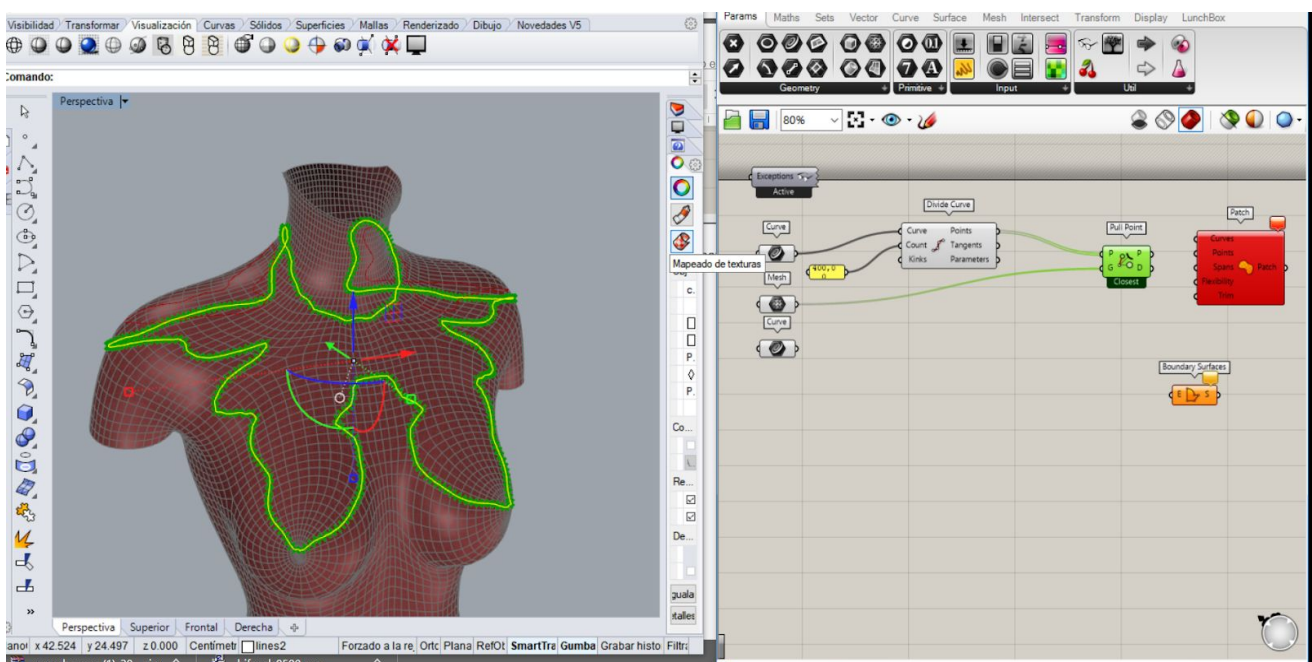
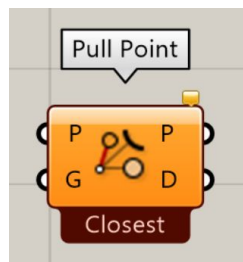
With these number of points (400) we are able to create the patch. The inputs parameters on the patch component are:

- Curve to patch
- Points to patch
- Number of Spans: number of spans
- Flexibility of the patch
- Attempt to trim (it will cut the things out of our system )

The thing we have now is that the points are not over my mesh and we need to project them over the mesh

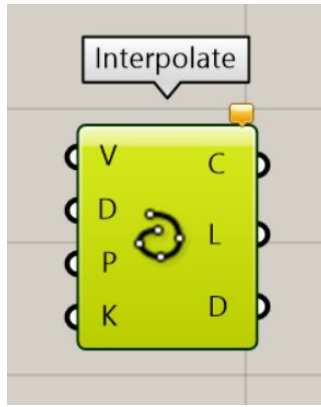


And we need that the input curve that we introduce into the patch is over the mesh  
 With the pull point component, here the input are the divided pints and the initial mesh:

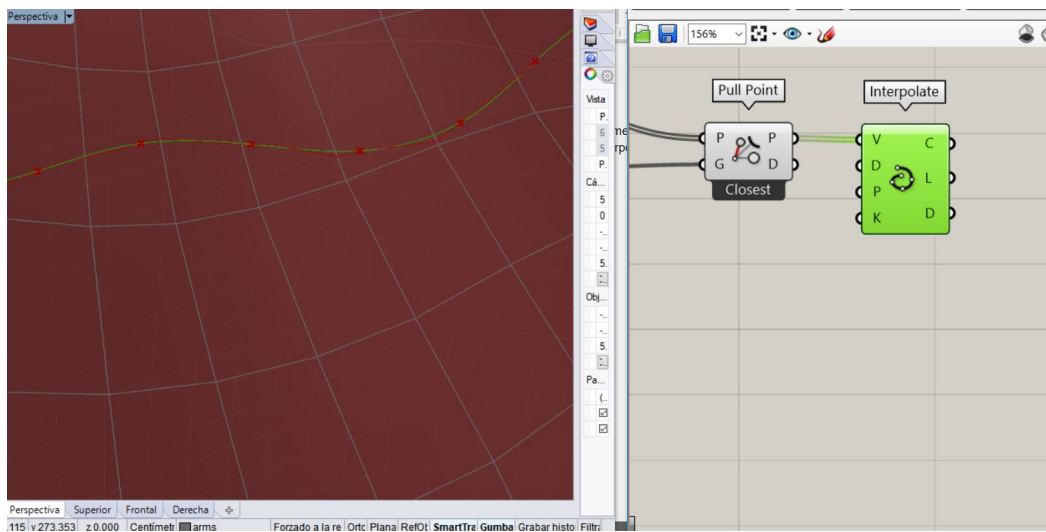


And with this operation we are sure that the points are on the mesh

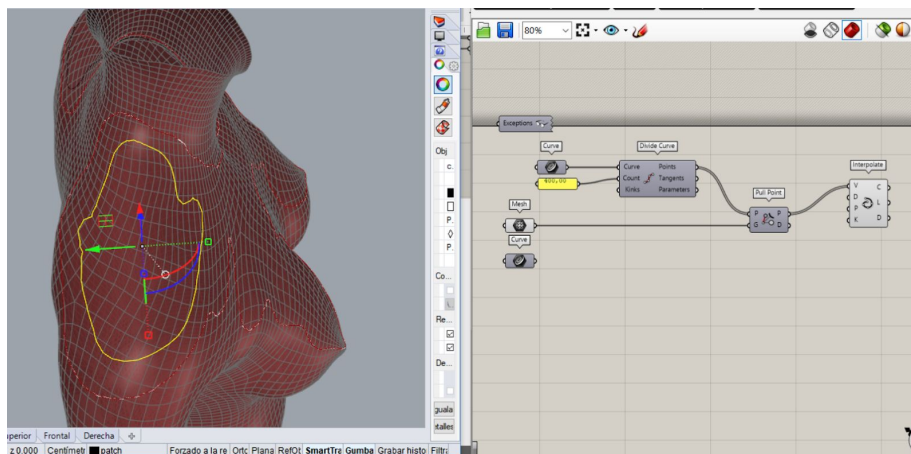
To be sure of this, we can draw a curve:(interpolate curve):



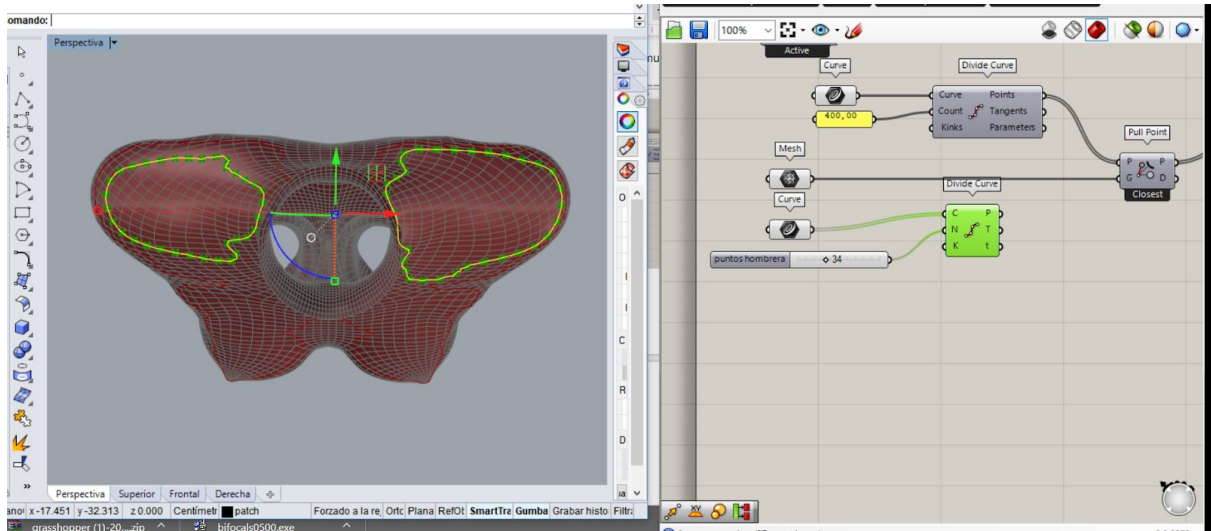
The green curve is very similar to the starting curve (red) but it is quite different



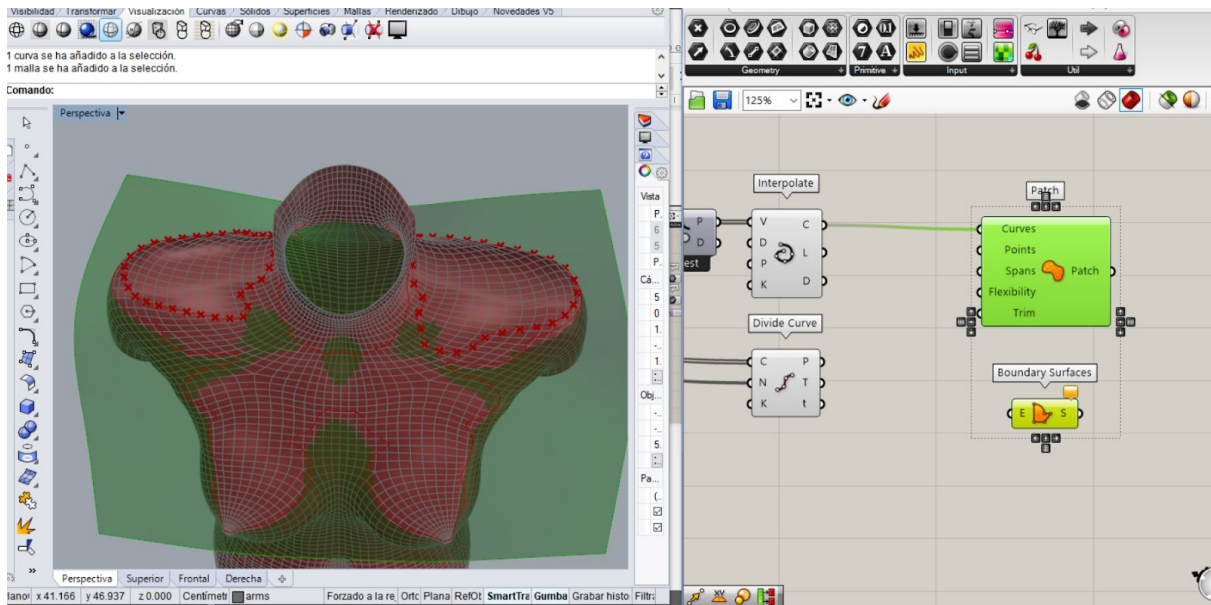
We select the curve over the shoulder and will divide it into multiple points



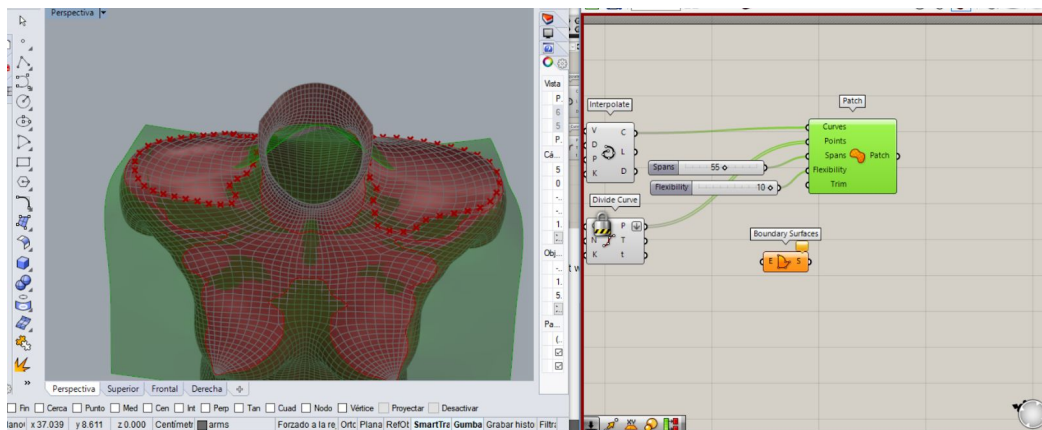
[divide curve + slider]



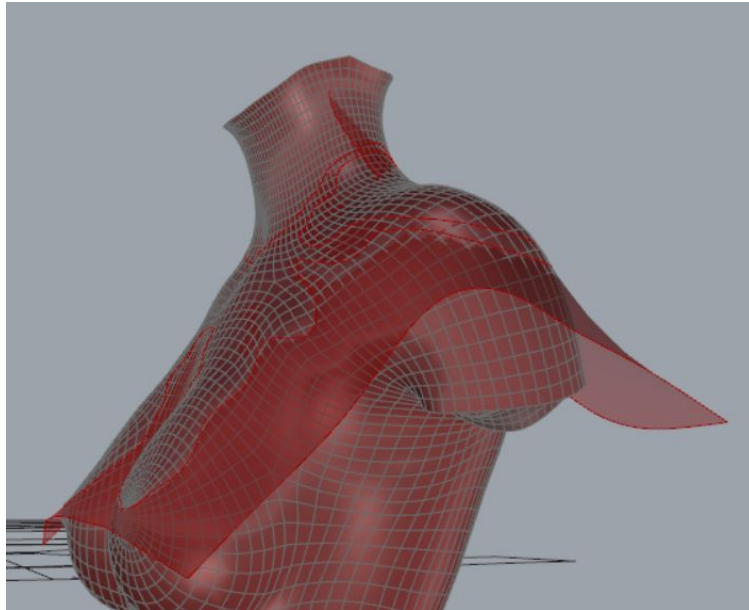
Now we need to connect the interpolate curve and the curve [input] of the patch component.



After this: flatten the points from divide curve: this means that we will put all these points in a one unique list



We can see that the surface is over the mesh, including the shoulders



And after project and split the surface:

