

Ej_003

DivideSrf

```
import Rhino.Geometry as rg
from Grasshopper import DataTree as Tree
from Grasshopper.Kernel.Data import GH_Path as Path

ptList = []
ptTree = Tree[object]()
pl = []

for i in range(uDiv+1):
    ptListTemp = []
    for j in range(vDiv+1):

        tempPt = srf.Evaluate(i/uDiv,j/vDiv,2)[1]
        ptTree.Add(tempPt,Path(i))
        ptListTemp.append(tempPt)
    ptList.append(ptListTemp)

for i in range(uDiv):
    for j in range(vDiv):

        pt1 = ptList[i][j]
        pt2 = ptList[i+1][j]
        pt3 = ptList[i][j+1]
        pt4 = ptList[i+1][j+1]

        pl.append(rg.Polyline([pt1,pt2,pt4,pt3,pt1]))
```

Planarize

```
import Rhino.Geometry as rg
import rhinoscriptsyntax as rs

#Creamos un plano con tres puntos cualesquiera
plane = rg.Plane(pts[0],pts[1],pts[2])

#Proyectamos el punto 3 sobre el plano calculado
newPt3 = plane.ClosestPoint(pts[3])

#Calculamos el desplazamiento de cada punto
dev = pts[3].DistanceTo(newPt3)
```

```
#Sustituimos el punto 3 por su nuevo valor
```

```
pts[3] = newPt3
```

```
lines = []
```

```
#Creamos una marca en cada panel fuera de tolerancia
```

```
if dev>T:
```

```
    l0 = rg.Line(pts[0],pts[2])
```

```
    l1 = rg.Line(pts[1],pts[3])
```

```
    lines.extend([l0,l1])
```

```
#Creamos la polilinea de cada panel y calculamos su area
```

```
pol = rg.PolylineCurve((pts[0],pts[1],pts[2],pts[3],pts[0]))
```

```
areaObj = rg.AreaMassProperties.Compute(pol)
```

```
area = areaObj.Area
```

```
a = pts
```

```
w = lines
```