Tecnologías informáticas para el diseño y la construcción de objetos

Rodrigo Vargas Peña

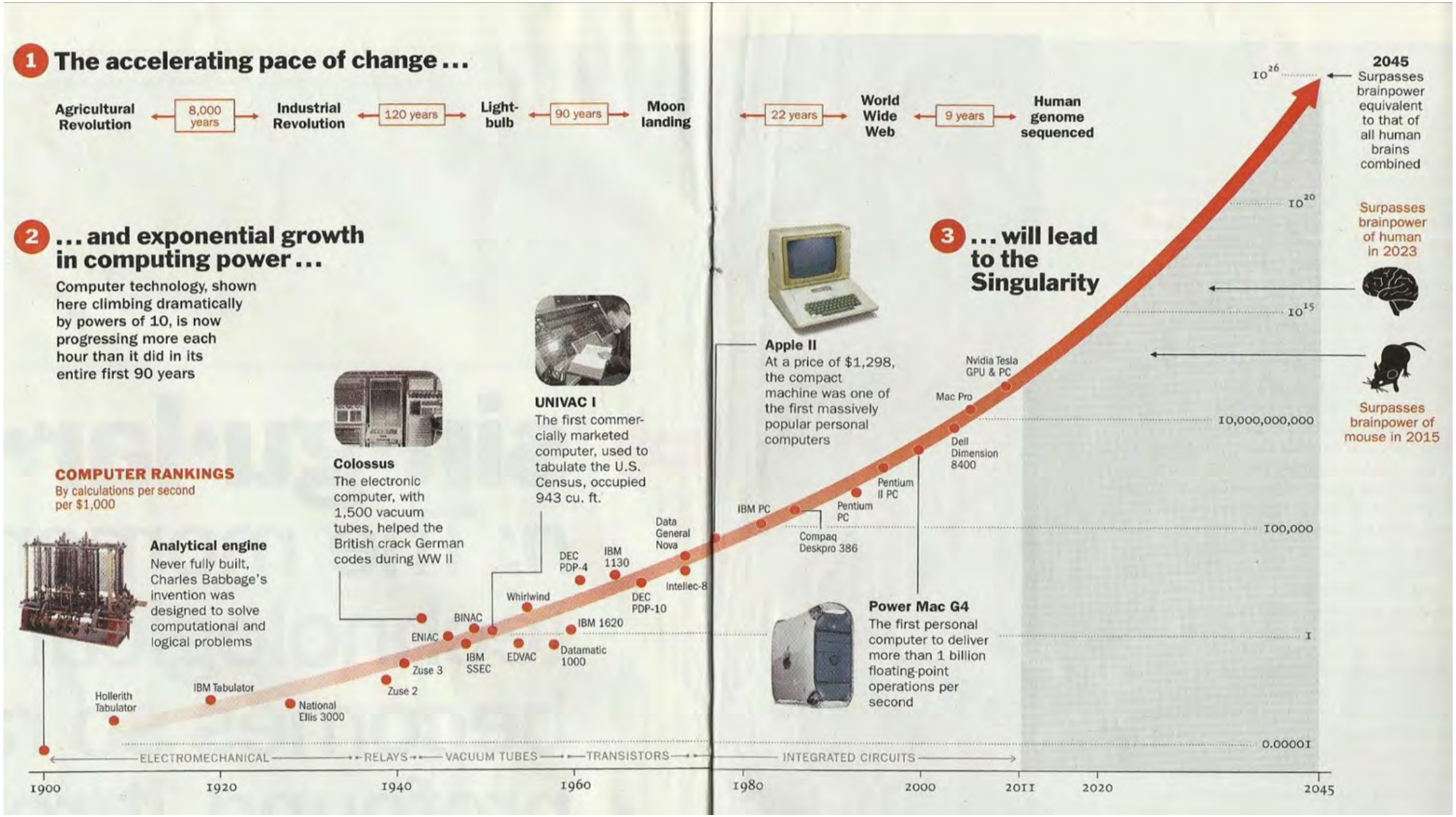
Arquitecto

Docente Escuela de Arquitectura - Universidad del Valle

Universidad del Valle

CALI PROGRESA contigo

ALCALDÍA DE SANTIAGO DE CALI



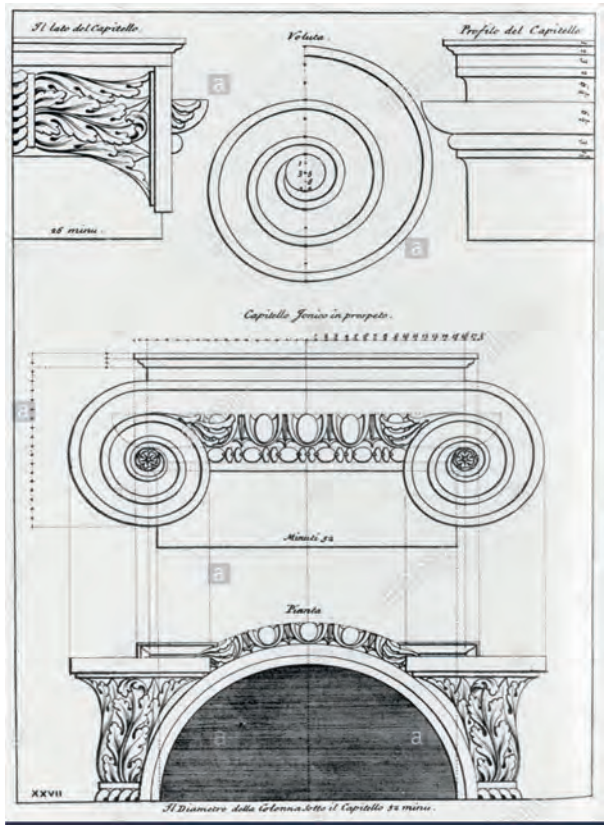
Revista Time. 2011. Raymond Kurzweil

* Todos los objetos con los que convivimos han sido previamente diseñados, y por ende, dibujados.

* El papel más frecuente de los computadores, en el campo del diseño, es el de un “esclavo” superdotado

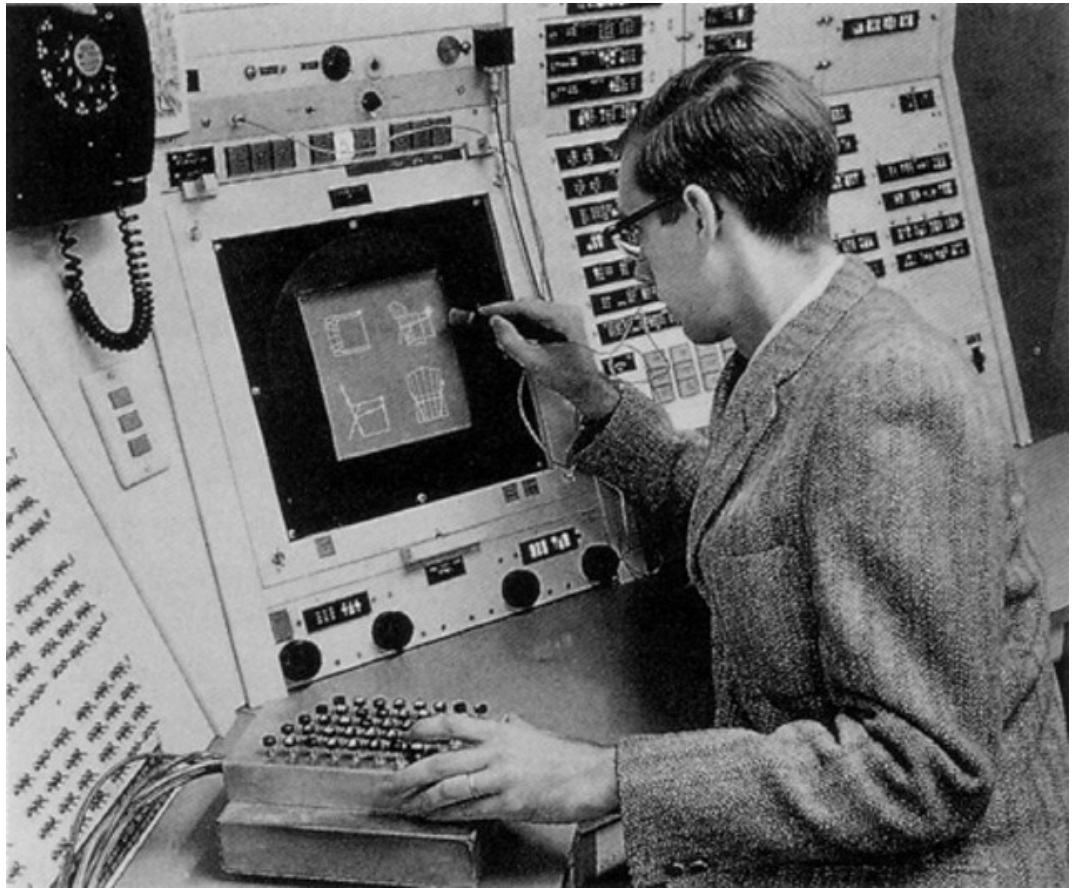
* El uso de la informática implica la aparición de nuevos tipos de “creatividad”. El computador es un co-diseñador

* La informática aplicada al diseño y producción de objetos implica una nueva aproximación al espacio de aprendizaje

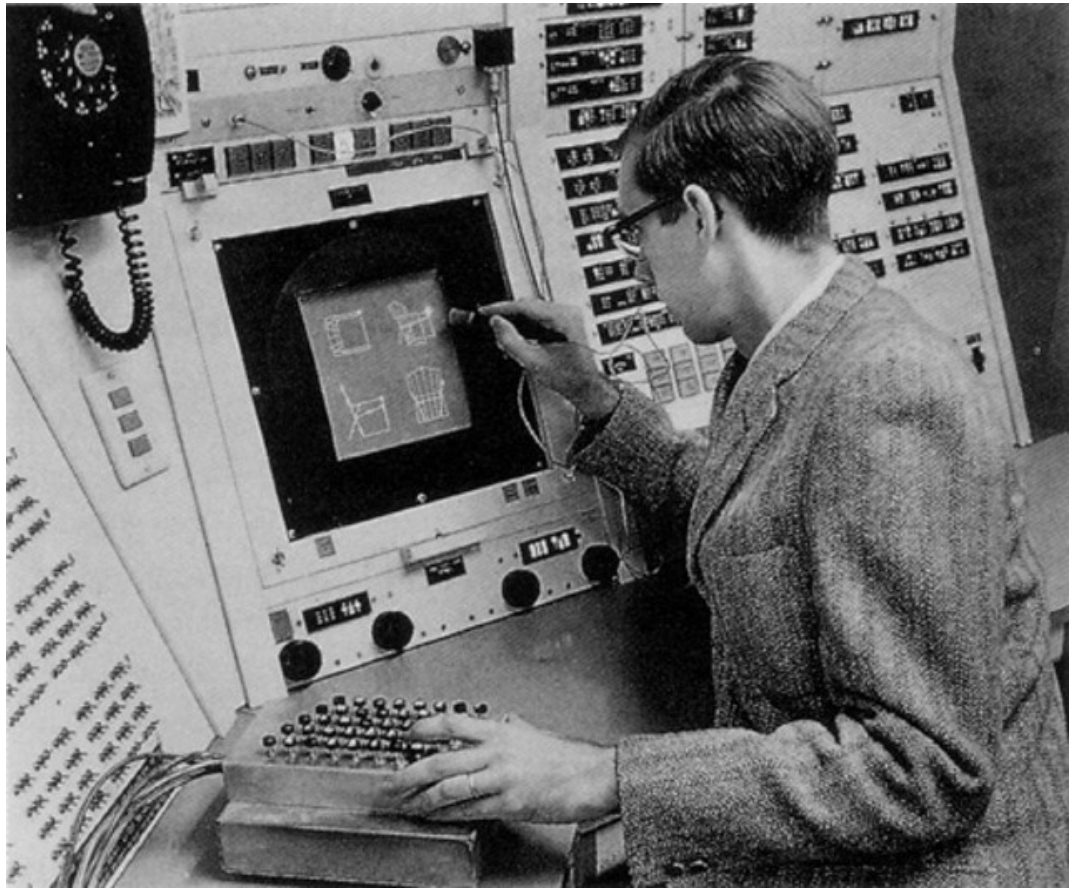


León Battista Alberti

La Escuela de Atenas. Rafael. 1510

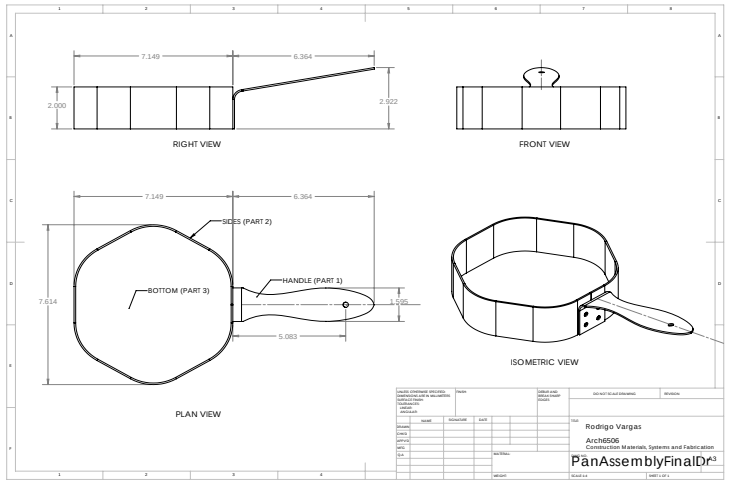


Ivan Sutherland. Sketchpad. MIT. 1963.
Memoria RAM: 0.3 Gb
Display: 7"
Tamaño: 100 m2

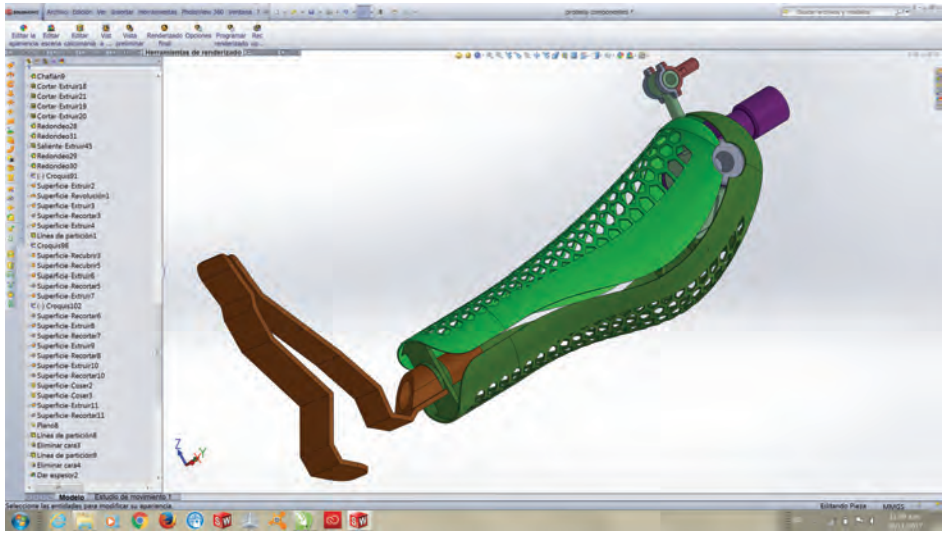


Ivan Sutherland. Sketchpad. MIT. 1963.
Memoria RAM: 0.3 Gb
Display: 7"
Tamaño: 100 m2

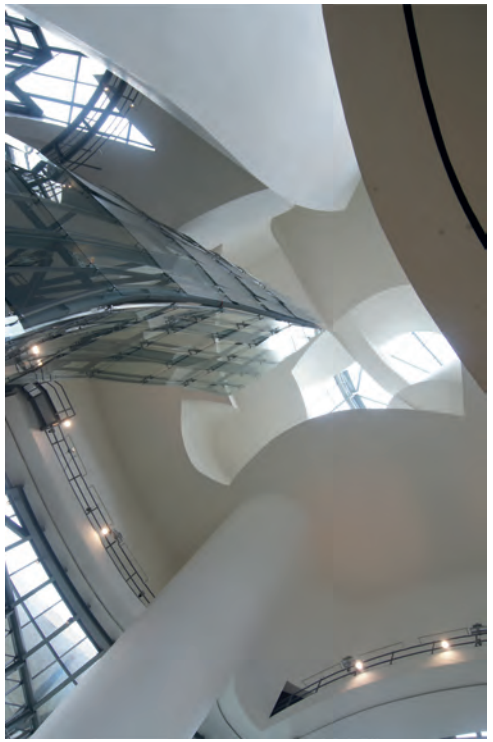
Un celular actual cuesta aproximadamente la millonésima parte, es un millón de veces mas pequeño y es mil veces mas potente que los computadores en MIT hace 40 años.
(Grossman, 2011)

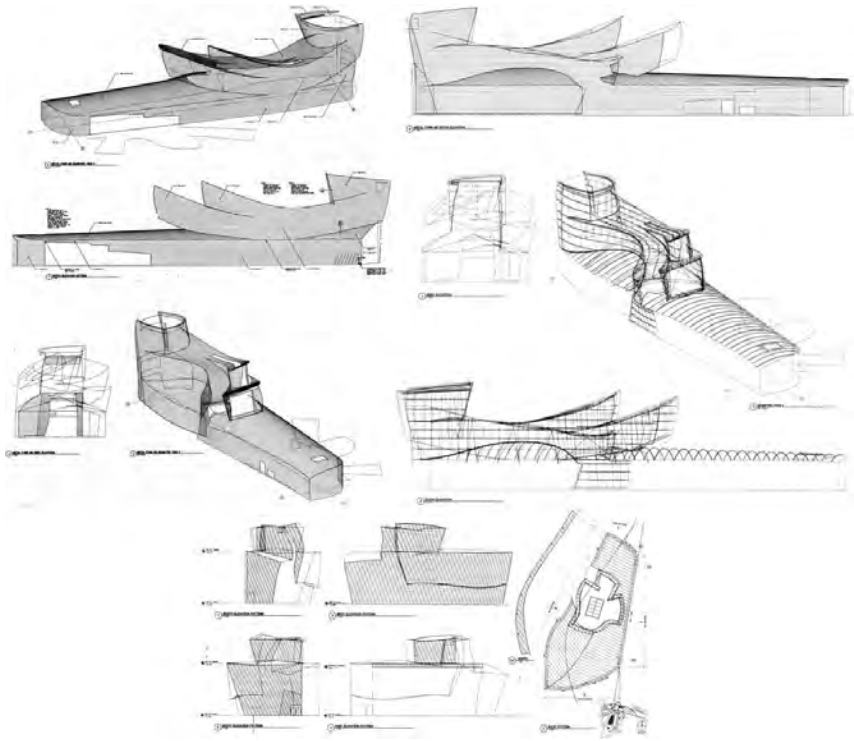


Planos de fabricación de una sartén



Prototipo de Prótesis para amputados. Patente D.I. Miguel Uribe. Univalle. 2016





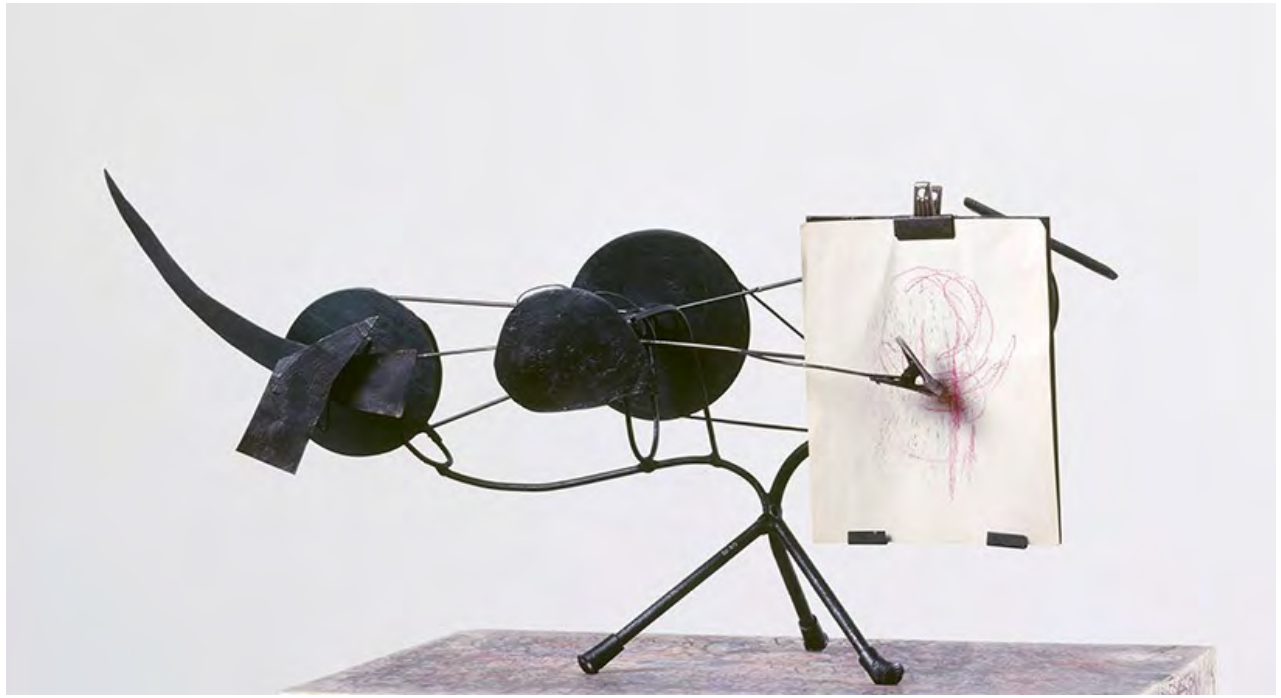
Museo Guggenheim. Bilbao. 1997

La tecnología es la respuesta, pero
¿cuál era la pregunta?

Cedric Price

Nuevas creatividades

Los recursos digitales permiten nuevas aproximaciones al diseño, no solo como herramienta de **representación gráfica**, sino como determinante conceptual en la generación de la forma.



Jean Tinguely. méta-matic no. 10, 1959
https://www.youtube.com/watch?v=GOo5uq2fH6g

Raymond Kurzweil. I've got a secret. 1965
https://www.youtube.com/watch?v=X4Neivqp2K4

Building Information Modeling (BIM)

Modelado paramétrico

Algoritmos

Sistemas de Información Geográfica (SIG)

Topología

Algoritmos de generación

Fabricación digital (CAM)

Diseño basado en desempeño

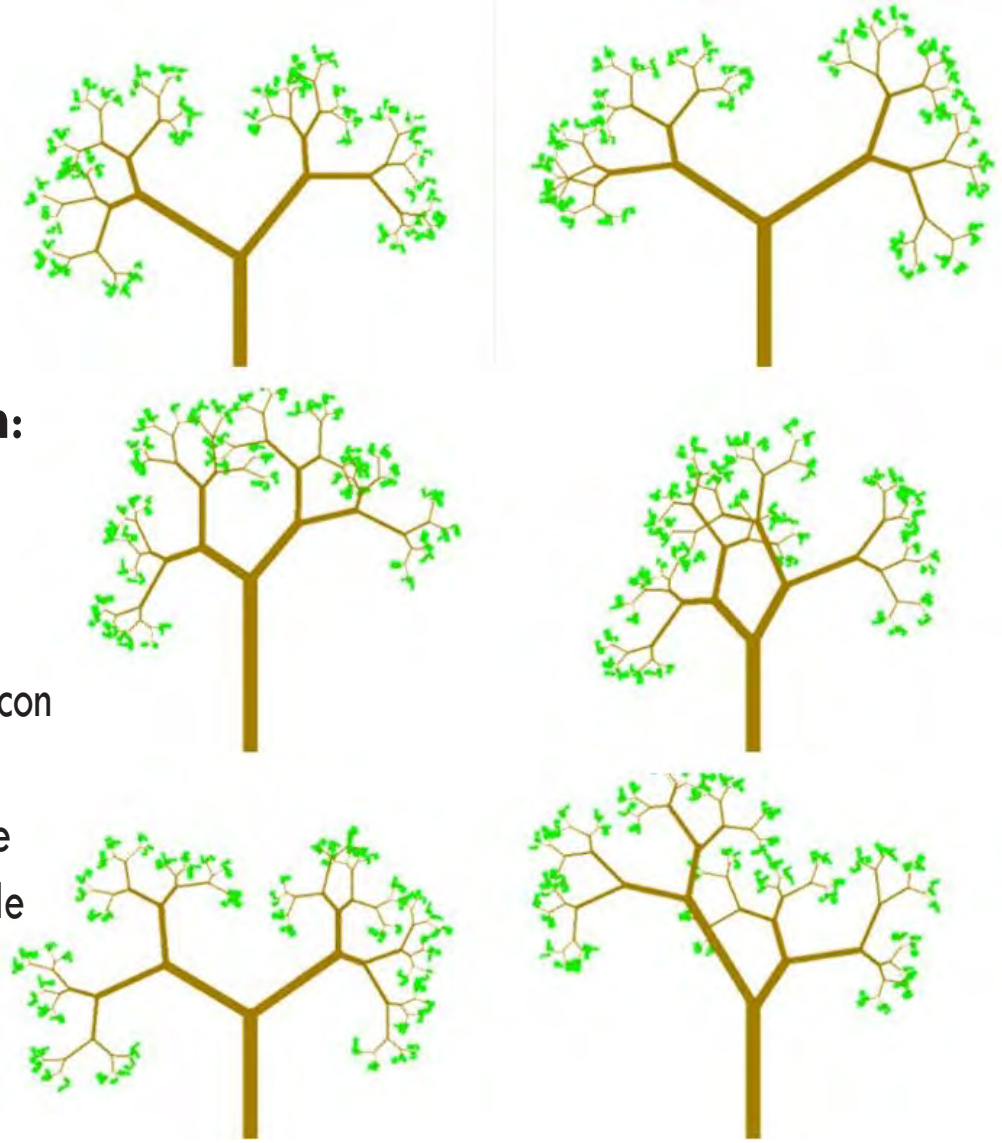
Prototipado rápido

Gramáticas formales

Diseño computacional

Algoritmo de generación:

Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema asociado con la generación de forma.
Se presenta como una secuencia de comandos escritos en un lenguaje de programación determinado, almacenados en un archivo.



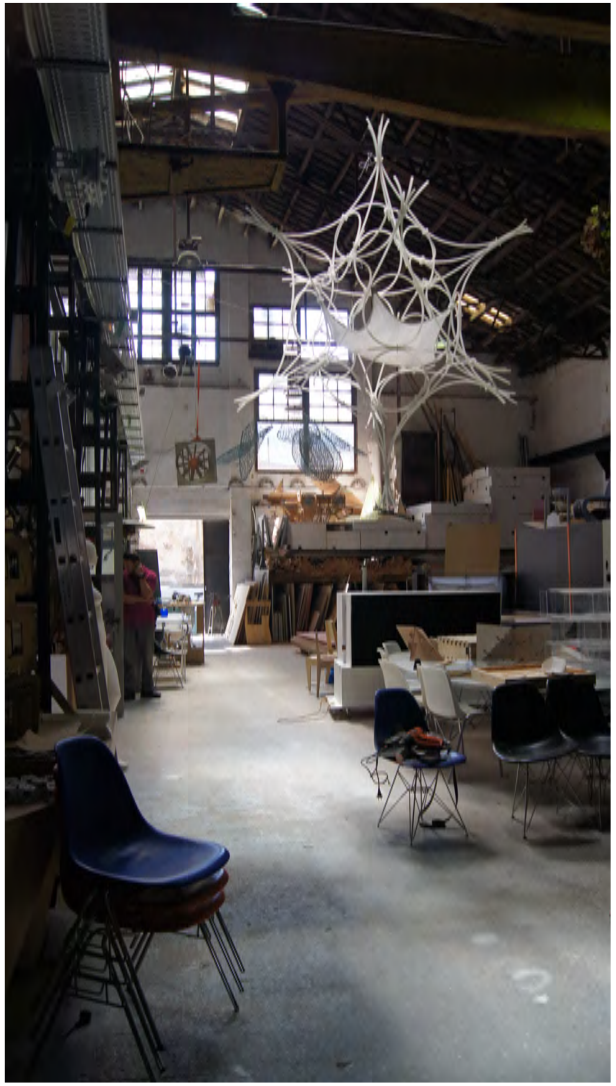
```
def simpleSquare():
  w = makeWorld(400,400)
  t = makeTurtle(w)
  for m in range(4):
    forward(t,120)
    turn(t,90)
    sleep(0.5)
 from time import sleep
from random import randrange
def test_tree():
w = 640
  h = 480
  orchard = makeWorld(w, h)
  bud = makeTurtle(orchard)
  bud.penUp()
  bud.moveTo(w/2, h)
bud.setPenColor(makeColor(160, 127, 0)) # brown
  bud.setPenWidth(2)
  bud.penDown()
sleep(1.0)
  drawTree(bud, 8, 150)
bud.hide()
  orchard.repaint()
def drawTree(bud, numBranches, branchLength):
sleep(0.03)
  if numBranches < 1:
    drawLeaf(bud)
else:
perturbedLength = branchLength*randrange(5,16)/10.0
    bud.setPenWidth(branchLength/8)
    bud.forward(int(perturbedLength))
leftAngle = -45 + randrange(-15, 15)
    bud.turn(leftAngle)
    drawTree(bud, numBranches-1, 2*branchLength/3)
rightAngle = 45 + randrange(-15, 15)
    bud.turn(rightAngle - leftAngle)
    drawTree(bud, numBranches-1, 2*branchLength/3)
       # To finish drawing a tree, we have to return bud to his
    # starting point FOR THAT TREE.
    bud.turn(-rightAngle)
    bud.backward(int(perturbedLength))
def drawLeaf(turtle):
```

Taller de objetos algorítmicos y diseño basado en reglas. Escuela de Arquitectura. Univalle. Octubre de 2017

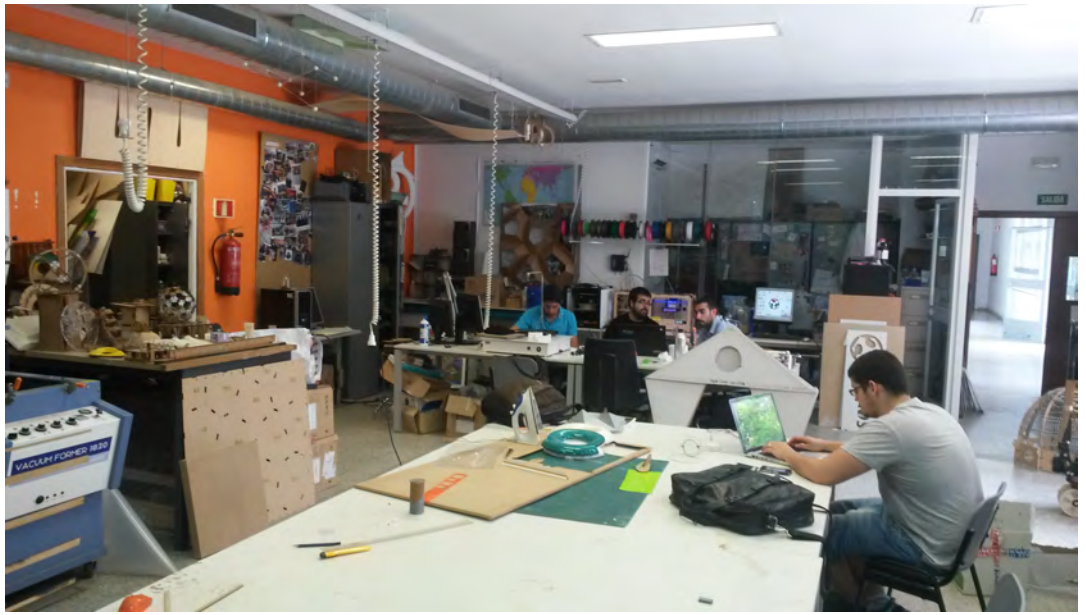
Nuevas experiencias espaciales

La tecnología de **fabricación digital** implica cambios fundamentales en la relación entre el diseñador, el objeto producido y el espacio

FAB-LAB. IaaC. Barcelona



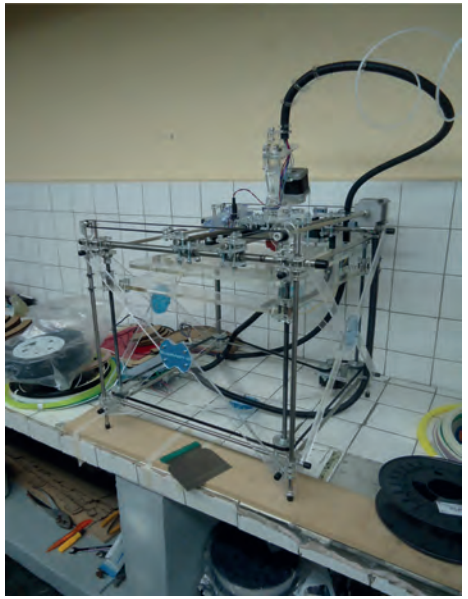
FAB-LAB. IaaC. Barcelona

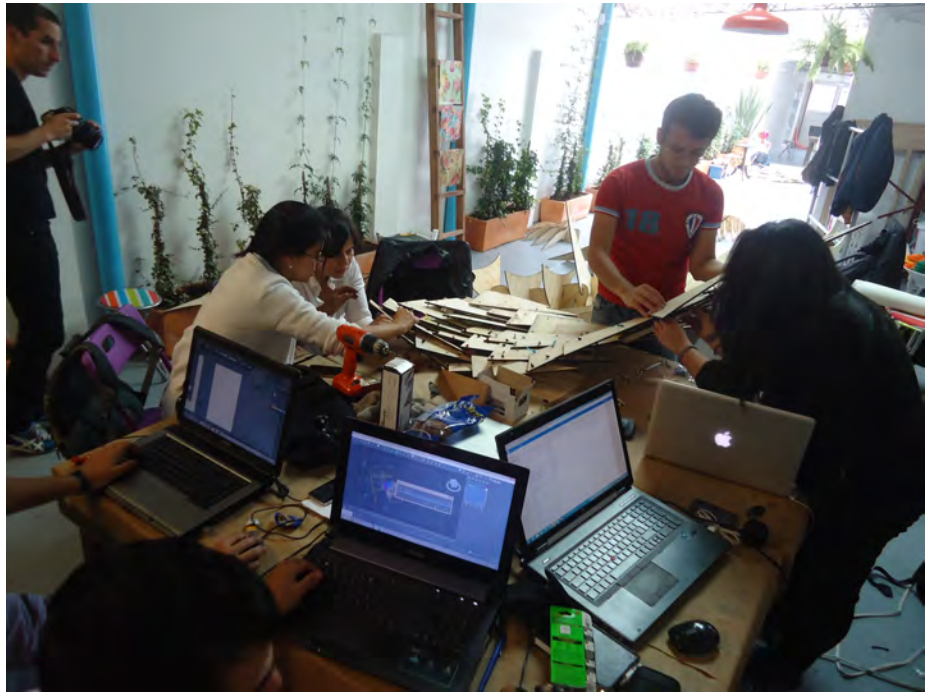


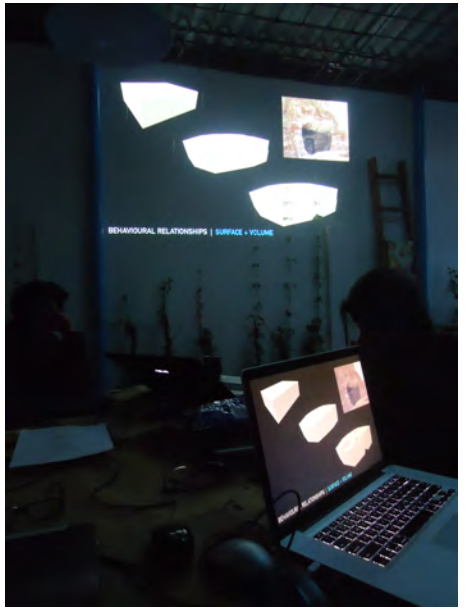
FAB-LAB Valencia. Universidad Politécnica de Valencia

FAB-LAB Sevilla. Universidad de Sevilla

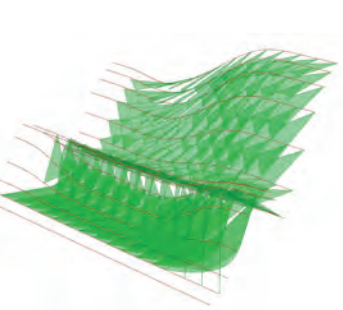
Molusco Pavilion - workshop sobre fabricación digital - Seminario la Representación del Proyecto 2016. Fablabsevillla + FablabUN + Fablabuninorte

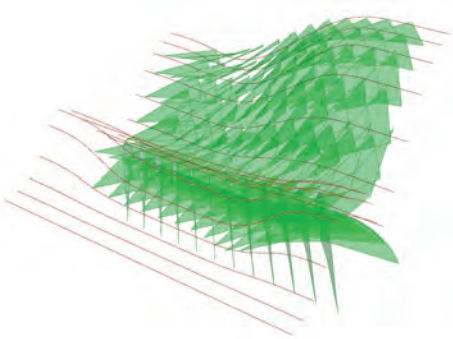


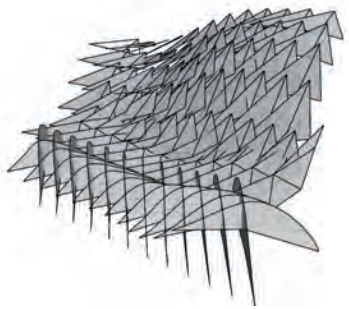


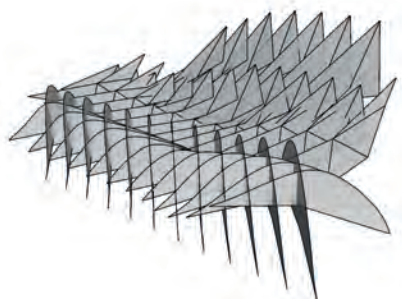


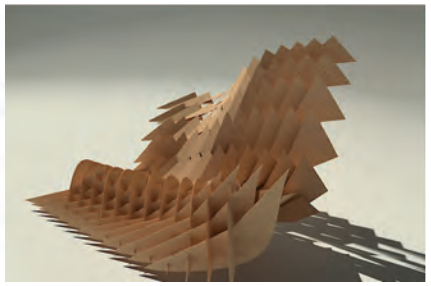
Taller AIA Visiting School. Up the River - Up the Mountain. Bogotá 2013

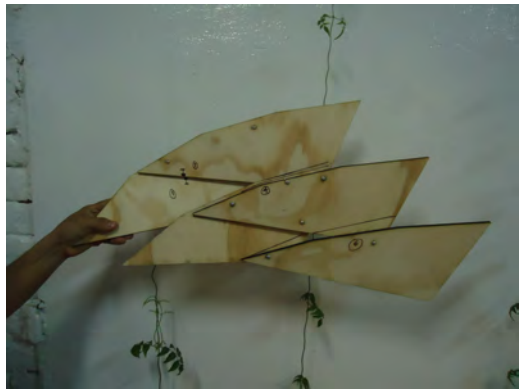












Taller AIA Visiting School. Up the River - Up the Mountain. Bogotá 2013

```python
import rhinoscriptsyntax as rs

divPTCoor = [[0 for i in range(100)] for i in range(100)]
divPT = [[0 for i in range(100)] for i in range(100)]

allSrf1 =[]
allSrf2 =[]
allSrf3 =[]
allSrf4 =[]
allSrf5 =[]
allSrf6 =[]

for i in range(len(curves)):
    div = rs.DivideCurve (curves[i], dblDIV)
    for j in range(len(div)):
        divPTCoor[i][j] = div[j]
        divPT = rs.AddPoint (divPTCoor[i][j])

for i in range(9, len(curves),3):
    for j in range(0, len(div)-5):
        if i>12 :
            crv1Temp = rs.AddInterpCurve ((divPTCoor[i][j], divPTCoor[i-1][j], divPTCoor[i-2][j+1], divPTCoor[i-6][j+2], divPTCoor[i-9][j+4]))
            plane1 = rs.AddSrfPt ((divPTCoor[i][j], divPTCoor[i-2][j+1], divPTCoor[i-9][j+4]))
            centroid = rs.SurfaceAreaCentroid (plane1)
            rs.ScaleObject (plane1, centroid[0], (20,20,20))
            param = rs.SurfaceClosestPoint(plane1, divPTCoor[i][j])
            data = rs.SurfaceCurvature(plane1, param)
            crv1 = rs.ProjectCurveToSurface(crv1Temp, plane1, data[1])
            srf1 = rs.ExtrudeCurvePoint  (crv1, centroid[0])
            allSrf1.append (srf1)

        if i>12 and i<1:
            crv2Temp = rs.AddInterpCurve ((divPTCoor[i][j+4], divPTCoor[i-1][j+1], divPTCoor[i-2][j],divPTCoor[i-3][j],divPTCoor[i-4][j]))
            plane2 = rs.AddSrfPt ((divPTCoor[i][j+4], divPTCoor[i-2][j], divPTCoor[i-4][j]))

for i in range(0, len(curves)-8):
    for j in range(0, len(div)-5):

        if i<12 :
            crv3Temp = rs.AddInterpCurve (((divPTCoor[i][j], divPTCoor[i+2][j+2], divPTCoor[i+1][j+3])),1)
            plane3 = rs.AddSrfPt ((divPTCoor[i][j], divPTCoor[i][j+2], divPTCoor[i+2][j+4]))
            centroid = rs.SurfaceAreaCentroid (plane3)
            rs.ScaleObject (plane3, centroid[0], (20,20,20))
            param = rs.SurfaceClosestPoint(plane3, divPTCoor[i][j])
            data = rs.SurfaceCurvature(plane3, param)
            crv3 = rs.ProjectCurveToSurface(crv3Temp, plane3, data[1])
            srf3 = rs.ExtrudeCurvePoint  (crv3, centroid[0])
            allSrf3.append (srf3)
        if i==5:
            crv4Temp = rs.AddInterpCurve (((divPTCoor[i][j], divPTCoor[i+1][j],divPTCoor[i+2][j],divPTCoor[i-4][j],divPTCoor[i+5][j])),1)
            plane4 = rs.AddSrfPt ((divPTCoor[i+4][j], divPTCoor[i][j+2], divPTCoor[i+2][j+3]))
            centroid = rs.SurfaceAreaCentroid (plane4)
            rs.ScaleObject (plane4, centroid[0], (20,20,20))
            param = rs.SurfaceClosestPoint(plane4, divPTCoor[i][j])
            data = rs.SurfaceCurvature(plane4, param)
            crv4 = rs.ProjectCurveToSurface(crv4Temp, plane4, data[1])
            srf4 = rs.ExtrudeCurvePoint  (crv4, centroid[0])
            allSrf4.append (srf4)

srfs1 = allSrf1
srfs2 = allSrf2
srfs3 = allSrf3
srfs4 = allSrf4
srfs5 = allSrf5
srfs6 = allSrf6
```

More surfaces added to have new sets of pieces.

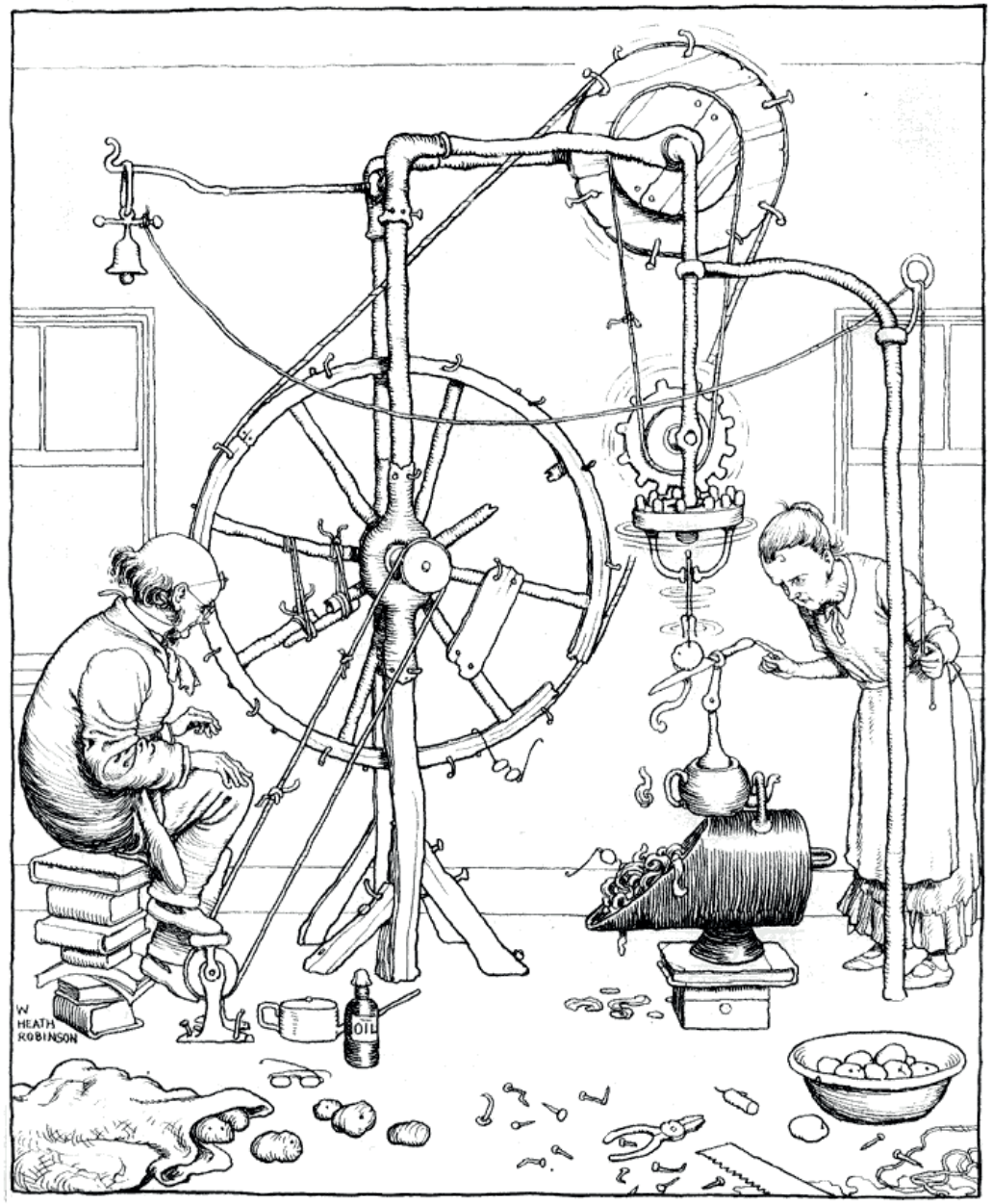
Inverting the order for curve reading.

Coordinates moddified to change the shape of individual pieces.

Controling surface sections sepparately

New loops in the code to generate additional sets of parts
Curves degree changed to generate rough surfaces

Curve generated on i=5 to créate the joining piece between the lower and upper parts



The Professor's invention for peeling potatoes.

William Heath Robinson

Gracias

Rodrigo Vargas Peña
rodrigo.vargas@correounivalle.edu.co