

# PoliedreEnUnHiperplaDeR4

## Objectiu

L'objectiu d'aquest seminari és aplicar els coneixements adquirits de geometria euclidiana afi per arribar a construir amb cartró un model acurat d'un cert políedre que anomenarem  $P$  del qual coneixem les coordenades dels seus vèrtexs.

## Motivació

$P$  és un poliedre convex de  $\mathbf{R}^3$  que té 14 vèrtexs i 9 cares. Coneixem les coordenades dels vèrtexs com a punts d'un hiperplà de  $\mathbf{R}^4$ . El motiu de donar les coordenades a  $\mathbf{R}^4$  és que, com passa moltes vegades (pensem, per exemple, en el triangle equilàter o el tetràedre regular), podem trobar coordenades més senzilles en una dimensió superior a la necessària. En aquest cas concret, és possible donar coordenades enteres per als vèrtexs de  $P$ .

## Dades del poliedre

### Hiperplans de les cares

H_i	Equacions dels hiperplans de $\mathbf{R}^4$
H_1	$3x + y + z + t + 6 = 0$
H_2	$-x + 3y + z + t + 6 = 0$
H_3	$-x + y + 3z + t + 6 = 0$
H_4	$-x + y + z + 3t + 6 = 0$
H_5	$x + y + z + t + 4 = 0$
H_6	$-x + y + z + t + 4 = 0$
H_7	$-x + y + z + t + 4 = 0$
H_8	$x + y + z + 3t + 6 = 0$
H_9	$-3x + y + z + t + 6 = 0$

### Vèrtexs

V_i	coordenades a $\mathbf{R}^4$
V_1	(4,1,2,3)
V_2	(3,1,2,4)
V_3	(3,2,1,4)
V_4	(4,2,1,3)
V_5	(1,4,1,4)
V_6	(1,6,1,2)
V_7	(1,6,2,1)
V_8	(1,2,6,1)
V_9	(2,1,6,1)
V_10	(4,1,4,1)
V_11	(2,1,3,4)
V_12	(1,2,3,4)
V_13	(4,3,2,1)
V_14	(4,3,1,2)

## Equació del Hiperplà

$$\underline{\underline{H \quad x+y+z+t=10}}$$

## Codi

### Programa 1

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

#####
#  10/12/2019                                Geometria lineal #
#                                                    #
#      Programa_1 per determinar quins vèrtexs      #
#      pertanyen a cada cara del poliedre.          #
#                                                    #
#  *Els primers dos comentaris són necessaris per imprimir accents #
#                                                    #
#####

def caretal_programa1():
    print("#"*40)
    print("#" + " " *1 + "10/12/2019" + " " *46 + "Geometria lineal " + "#")
    print("#" + " " *38 + "#")
    print("#" + " " *5 + "Programa per determinar quins vèrtexs" + " " *29 + "#")
    print("#" + " " *5 + "pertanyen a cada cara del poliedre." + " " *30 + "#")
    print("#" + " " *38 + "#")
    print("#"*40)

caretal_programa1()
#Llista amb les coordenades dels vèrtex del poliedre com a punts de R^4
vertex=[(4,1,2,3),
        (3,1,2,4),
        (3,2,1,4),
        (4,2,1,3),

        (1,4,1,4),
        (1,6,1,2),
        (1,6,2,1),
        (1,2,6,1),

        (2,1,6,1),
        (4,1,4,1),
        (2,1,3,4),
        (1,2,3,4),

        (4,3,2,1),
        (4,3,1,2)]
#Llista amb les equacions del hiperplans com a "vectors" (c_x,c_y,c_z,c_t)
hipers=[(3,-1,-1,-1,6),
```

```

        (-1,3,-1,-1,6),
        (-1,-1,3,-1,6),

        (-1,-1,-1,3,6),
        (1,1,-1,-1,4),
        (-1,1,1,-1,4),

        (-1,-1,1,1,4),
        (1,1,1,-3,6),
        (-3,1,1,1,6),]
#En aquesta llista guardarem els vèrtex de cada cara (com a nom):
relacions=[]
#En aquesta llista guardarem els vèrtex de cada cara (de forma explícita):
relacionsExplicit=[]
# 9 cares del poliedre
for i in range(9):
    nameOfList="H_"+str(i+1)+": "
    relacions.append([nameOfList,])
    relacionsExplicit.append([nameOfList,])
# 14 cares del poliedre
for j in range(14):
    sum=0
    for k in range(4):
        sum+=vertex[j][k]*hipers[i][k]
    sum=sum+hipers[i][4]
    if(sum==0):
        (relacions[i]).append("V_"+str(j+1)+"")
        (relacionsExplicit[i]).append(vertex[j])
print("\n")
# Solucions Implícites
print("Solucions en format implícit: ")
for epsilon in range(9):
    print(relacions[epsilon])
print("\n")
#Solucions Explícites
print("Solucions en format explícits: ")
for epsilon in range(9):
    print(relacionsExplicit[epsilon])
print("\n")

```

## Programa 2

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-

#####
#+ 10/12/2019 Geometria lineal #+
#+
#+ Programa_2 per generar les cares del poliedre P, #+
#+ com ha arxius ./cara(NombreDelHiperpla).png en #+
#+ el directori de treball. #+
#+ #+
#+ El codi de colors ajuda a distingir els vèrtexs. #+

```

```

#+      També s'imprimeixen les coordenades dels punts de
#+      cada base. Aquests polígons són els que després
#+      dibuixarem a Geogebra per imprimir i muntar.
#+      -----
#+
#+      [ Nota teòrica del codi ]
#+
#+
#+      1. Tenim un poliedre a un hiperplà de  $R^4$  del qual sabem les
#+      coordenades de tots els seus vèrtexs i també les equacions
#+      dels hiperplans que contenen cadascuna.
#+
#+      2. Determinem quins vèrtexs pertanyen a cada cara comprovant
#+      si els punts compleixen o no les equacions dels hiperplans.
#+      (Programa 1)
#+
#+      3. Una vegada classificats els vèrtexs, com sabem que els
#+      punts que pertanyen a una mateixa cara del poliedre generen
#+      un pla, poden ser "retornats" al pla agafant un dels vèrtexs
#+      i considerant dos vèrtexs (més) per formar dos vectors
#+      continguts en el "pla de la cara".
#+
#+      4. Trobem una base ortonormal ( $e_1, e_2$ ) a partir del mètode de
#+      valors  $\lambda$  i  $\mu$  tals que:
#+      $$ \text{vector}\{V_{OV_i}\} = \mu \cdot e_1 + \lambda \cdot e_2. $$
#+
#+      5. Llavors les coordenades del Vèrtex són  $(\mu, \lambda)$ .
#+      Fem els mateix per la resta de vèrtexs que considerarem
#+      com a vectors al punt a partir del qual hem construït la
#+      base ortonormal.
#+
#+
#++
#####
##      Per poder executar s'ha d'haver fet          ##
##      "pip install" + els paquets elementals usats  ##
##                                                    ##
#####

import sympy #Per resoldre sistemes lineals ràpid
from math import sqrt #Per fer el modul
import matplotlib.pyplot as plt #Per fer els dibuixets

#Funció que imprimeix recuadre informatiu
def caretal_programa2():
    print("#+"*40)
    print("#+ " + " *1 + "10/12/2019" + " " *46 + "Geometria lineal " + "#+")
    print("#+ " + " *38 + "#+")
    print("#+ " + " *5 + "Programa per generar les cares del poliedre P," + " " *20 + "#+")
    print("#+ " + " *5 + "com ha arxius ./cara(NombreDelHiperpla).png en" + " " *20 + "#+")
    print("#+ " + " *5 + "el directori de treball. " + " " *20 + "#+")
    print("#+ " + " *38 + "#+")

```

```

print("#" + " " * 5 + "El codi de colors ajuda a distingir els vèrtexs. " + " " * 16 + "#")
print("#" + " " * 5 + "També s'imprimeixen les coordenades dels punts de " + " " * 16 + "#")
print("#" + " " * 5 + "cada base. Aquests polígons són els que després " + " " * 16 + "#")
print("#" + " " * 5 + "dibuixarem a Geogebra per imprimir i muntar. " + " " * 16 + "#")
print("#" + " " * 38 + "#")
print("#" * 40)

#Funció per obtenir els vectors a partir del vèrtex "Punts" \neg "Vectors"
def vectors_inicials(v_i,v_ii,v_iii):
    A=[0,0,0,0]
    B=[0,0,0,0]
    for i in range(4):
        A[i]=v_ii[i]-v_i[i]
        B[i]=v_iii[i]-v_i[i]
    return [A,B]

#Producte escalar de vectors
def vectors_producte_escalar(u,v):
    sum=0
    for i in range(4):
        sum+=u[i]*v[i]
    return sum;

#Funció modul d'un vector
def vector_modul(u):
    sum=0
    for i in range(4):
        sum+=u[i]**2
    return sqrt(sum)

#Funció que troba la base ortonormal de cada hiperplà
def base_pla_ortonormal(u,v):
    modul=vector_modul(u)
    e_1=[0,0,0,0]
    e_2=[0,0,0,0]
    for i in range(len(u)):
        e_1[i]=u[i]*(1/modul)
    vectProd=vectors_producte_escalar(v,u)
    for i in range(len(v)):
        e_2[i]=v[i]-vectProd*e_1[i]*(1./vector_modul(u))
    modul=vector_modul(e_2)
    for i in range(4):
        e_2[i]=e_2[i]*(1./modul)
    return [e_1,e_2]

#Funció que retorna el vector del punt origen del hiperplà
#escollit per obtenir la base ortonormal al vèrtex normal
def vector_rellevant(v_i,v_m):
    v=[0,0,0,0]
    for i in range(4):
        v[i]=v_m[i]-v_i[i]
    return v

```

```
#####
#
#          Funció Principal
#
#####
caretal_programa2()
# Llista amb les llistes dels vèrtex de cada cara, obtingut en programa 1
llistaDeVectors=[['H_1:', (1, 4, 1, 4), (1, 6, 1, 2), (1, 6, 2, 1), (1, 2, 6, 1), (1, 2, 3, 4)],
                  ['H_2:', (4, 1, 2, 3), (3, 1, 2, 4), (2, 1, 6, 1), (4, 1, 4, 1), (2, 1, 3, 4)],
                  ['H_3:', (3, 2, 1, 4), (4, 2, 1, 3), (1, 4, 1, 4), (1, 6, 1, 2), (4, 3, 1, 2)],
                  ['H_4:', (1, 6, 2, 1), (1, 2, 6, 1), (2, 1, 6, 1), (4, 1, 4, 1), (4, 3, 2, 1)],
                  ['H_5:', (1, 2, 6, 1), (2, 1, 6, 1), (2, 1, 3, 4), (1, 2, 3, 4)],
                  ['H_6:', (4, 1, 2, 3), (3, 1, 2, 4), (3, 2, 1, 4), (4, 2, 1, 3)],
                  ['H_7:', (1, 6, 1, 2), (1, 6, 2, 1), (4, 3, 2, 1), (4, 3, 1, 2)],
                  ['H_8:', (3, 1, 2, 4), (3, 2, 1, 4), (1, 4, 1, 4), (2, 1, 3, 4), (1, 2, 3, 4)],
                  ['H_9:', (4, 1, 2, 3), (4, 2, 1, 3), (4, 1, 4, 1), (4, 3, 2, 1), (4, 3, 1, 2)]]

# Per ficar lletres estil \LaTeX{} al gràfic
llistaDeVectorsNoms=[['H_1:', 'V_{5}', 'V_{6}', 'V_{7}', 'V_{8}', 'V_{12}'],
                     ['H_2:', 'V_{1}', 'V_{2}', 'V_{9}', 'V_{10}', 'V_{11}'],
                     ['H_3:', 'V_{3}', 'V_{4}', 'V_{5}', 'V_{6}', 'V_{14}'],
                     ['H_4:', 'V_{7}', 'V_{8}', 'V_{9}', 'V_{10}', 'V_{13}'],
                     ['H_5:', 'V_{8}', 'V_{9}', 'V_{11}', 'V_{12}'],
                     ['H_6:', 'V_{1}', 'V_{2}', 'V_{3}', 'V_{4}'],
                     ['H_7:', 'V_{6}', 'V_{7}', 'V_{13}', 'V_{14}'],
                     ['H_8:', 'V_{2}', 'V_{3}', 'V_{5}', 'V_{11}', 'V_{12}'],
                     ['H_9:', 'V_{1}', 'V_{4}', 'V_{10}', 'V_{13}', 'V_{14}']]

#Bucle per cada cara del poliedre
for i in range(9):
    auxiliarList=[]
    #Els dos vector que farem servir per trobar la base ortonormal
    AB=vectors_inicials(llistaDeVectors[i][1],llistaDeVectors[i][2],llistaDeVectors[i][3])
    #Trobem la base ortonormal
    e=base_pla_ortonormal(AB[0],AB[1])
#Bucle per cada vertex de la cara 5 o 4 (depèn de la cara)
    for j in range(len(llistaDeVectors[i])-1):
        sympy.init_printing()
        x,y=sympy.symbols('x,y')
        equations=[]
        vectorDirector=vector_rellevant(llistaDeVectors[i][1],llistaDeVectors[i][j+1])
        for m in range(4):
            equations.append(sympy.Eq(e[0][m]*x+e[1][m]*y-vectorDirector[m],0.0))
        solucio=sympy.solve(equations,(x,y))
        auxiliarList.append(solucio)
        #print(solucio) útil per debugger si es vol modificar codi

#Part del codi dedicada a dibuixar els punts en el pla
    list_xy=[]
    for gfr in range(len(auxiliarList)):
        list_xy.append(list(auxiliarList[gfr].values())) #llista amb tot
    xlist=[] #Llista amb coordenada x
    ylist=[] #Llista amb coordenada y
    xylist=[]
    noms=[] #Llista amb noms dels Vèrtexs
```

```

for m in range(0,len(auxiliarList)):
    noms.append(llistaDeVectorsNoms[i][m+1])
    xlist.append(list_xy[m][0])
    ylist.append(list_xy[m][1])
    xylist.append((list_xy[m][0],list_xy[m][1]))
print("\nCara corresponent al Hiperplà "+str(i+1))
print(noms)
print(xylist)
plt.axis('equal') # Volem que sigui x/y=1 escala
plt.title(r'Cara de  $H_{\%d}$ ' %(i+1)) #LaTeX{}
colors= ["blue","red","green","orange","violet"]
for dibuix in range(len(xlist)):
    plt.scatter(xlist[dibuix], ylist[dibuix],c=colors[dibuix],label=r'$%s$'%noms[dibuix])
#plt.show() útil per debugger si es vol modificar codi
plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.05),
           fancybox=True, shadow=True, ncol=5)
plt.savefig('./cara'+str(i+1)+'.png')
plt.close()

```

## Resultats

### Outputs del codi

#### Programa 1

```

#####
#+ 10/12/2019                                     Geometria lineal  +#+
#+
#+          Programa per determinar quins vèrtexs                +#+
#+          pertanyen a cada cara del poliedre.                  +#+
#+                                                                +#+
#####

```

#### Solucions en format implícit:

```

['H_1:', 'V_{5}', 'V_{6}', 'V_{7}', 'V_{8}', 'V_{12}']
['H_2:', 'V_{1}', 'V_{2}', 'V_{9}', 'V_{10}', 'V_{11}']
['H_3:', 'V_{3}', 'V_{4}', 'V_{5}', 'V_{6}', 'V_{14}']
['H_4:', 'V_{7}', 'V_{8}', 'V_{9}', 'V_{10}', 'V_{13}']
['H_5:', 'V_{8}', 'V_{9}', 'V_{11}', 'V_{12}']
['H_6:', 'V_{1}', 'V_{2}', 'V_{3}', 'V_{4}']
['H_7:', 'V_{6}', 'V_{7}', 'V_{13}', 'V_{14}']
['H_8:', 'V_{2}', 'V_{3}', 'V_{5}', 'V_{11}', 'V_{12}']
['H_9:', 'V_{1}', 'V_{4}', 'V_{10}', 'V_{13}', 'V_{14}']

```

#### Solucions en format explícits:

```

['H_1:', (1, 4, 1, 4), (1, 6, 1, 2), (1, 6, 2, 1), (1, 2, 6, 1), (1, 2, 3, 4)]
['H_2:', (4, 1, 2, 3), (3, 1, 2, 4), (2, 1, 6, 1), (4, 1, 4, 1), (2, 1, 3, 4)]
['H_3:', (3, 2, 1, 4), (4, 2, 1, 3), (1, 4, 1, 4), (1, 6, 1, 2), (4, 3, 1, 2)]
['H_4:', (1, 6, 2, 1), (1, 2, 6, 1), (2, 1, 6, 1), (4, 1, 4, 1), (4, 3, 2, 1)]
['H_5:', (1, 2, 6, 1), (2, 1, 6, 1), (2, 1, 3, 4), (1, 2, 3, 4)]
['H_6:', (4, 1, 2, 3), (3, 1, 2, 4), (3, 2, 1, 4), (4, 2, 1, 3)]
['H_7:', (1, 6, 1, 2), (1, 6, 2, 1), (4, 3, 2, 1), (4, 3, 1, 2)]

```

```

['H_8:', (3, 1, 2, 4), (3, 2, 1, 4), (1, 4, 1, 4), (2, 1, 3, 4), (1, 2, 3, 4)]
['H_9:', (4, 1, 2, 3), (4, 2, 1, 3), (4, 1, 4, 1), (4, 3, 2, 1), (4, 3, 1, 2)]

```

## Programa 2

```

#####
# 10/12/2019                                Geometria lineal #
#
# Programa per generar les cares del poliedre P,          #
# com ha arxius ./cara(NombreDelHiperpla).png en         #
# el directori de treball.                                #
#
# El codi de colors ajuda a distingir els vèrtexs.        #
# També s'imprimeixen les coordenades dels punts de     #
# cada base. Aquests polígons són els que després       #
# dibuixarem a Geogebra per imprimir i muntar.          #
#
#####

```

Cara corresponent al Hiperplà 1

```

['V_{5}', 'V_{6}', 'V_{7}', 'V_{8}', 'V_{12}']
[(0.0, 0.0), (2.82842712474619, 0.0),
(3.53553390593274, 1.22474487139159),
(0.707106781186548, 6.12372435695795),
(-1.41421356237310, 2.44948974278318)]

```

Cara corresponent al Hiperplà 2

```

['V_{1}', 'V_{2}', 'V_{9}', 'V_{10}', 'V_{11}']
[(0.0, 0.0), (1.41421356237310, 0.0),
(0.0, 4.89897948556636),
(-1.41421356237310, 2.44948974278318),
(2.12132034355964, 1.22474487139159)]

```

Cara corresponent al Hiperplà 3

```

['V_{3}', 'V_{4}', 'V_{5}', 'V_{6}', 'V_{14}']
[(0.0, 0.0), (1.41421356237310, 0.0),
(-1.41421356237310, 2.44948974278318),
(0.0, 4.89897948556636),
(2.12132034355964, 1.22474487139159)]

```

Cara corresponent al Hiperplà 4

```

['V_{7}', 'V_{8}', 'V_{9}', 'V_{10}', 'V_{13}']
[(0.0, 0.0), (5.65685424949238, 0.0),
(6.36396103067893, 1.22474487139159),
(4.94974746830583, 3.67423461417477),
(2.12132034355964, 3.67423461417477)]

```

Cara corresponent al Hiperplà 5

```

['V_{8}', 'V_{9}', 'V_{11}', 'V_{12}']
[(0.0, 0.0), (1.41421356237310, 0.0),
(1.41421356237310, 4.24264068711928),
(-3.14018491736755e-16, 4.24264068711928)]

```

Cara corresponent al Hiperplà 6



```
['V_{1}', 'V_{2}', 'V_{3}', 'V_{4}']  
[(0.0, 0.0), (1.41421356237310, 0.0),  
(1.41421356237310, 1.41421356237310),  
(-3.14018491736754e-16, 1.41421356237310)]
```

Cara corresponent al Hiperplà 7

```
['V_{6}', 'V_{7}', 'V_{13}', 'V_{14}']  
[(0.0, 0.0), (1.41421356237310, 0.0),  
(1.41421356237310, 4.24264068711928),  
(-3.14018491736755e-16, 4.24264068711928)]
```

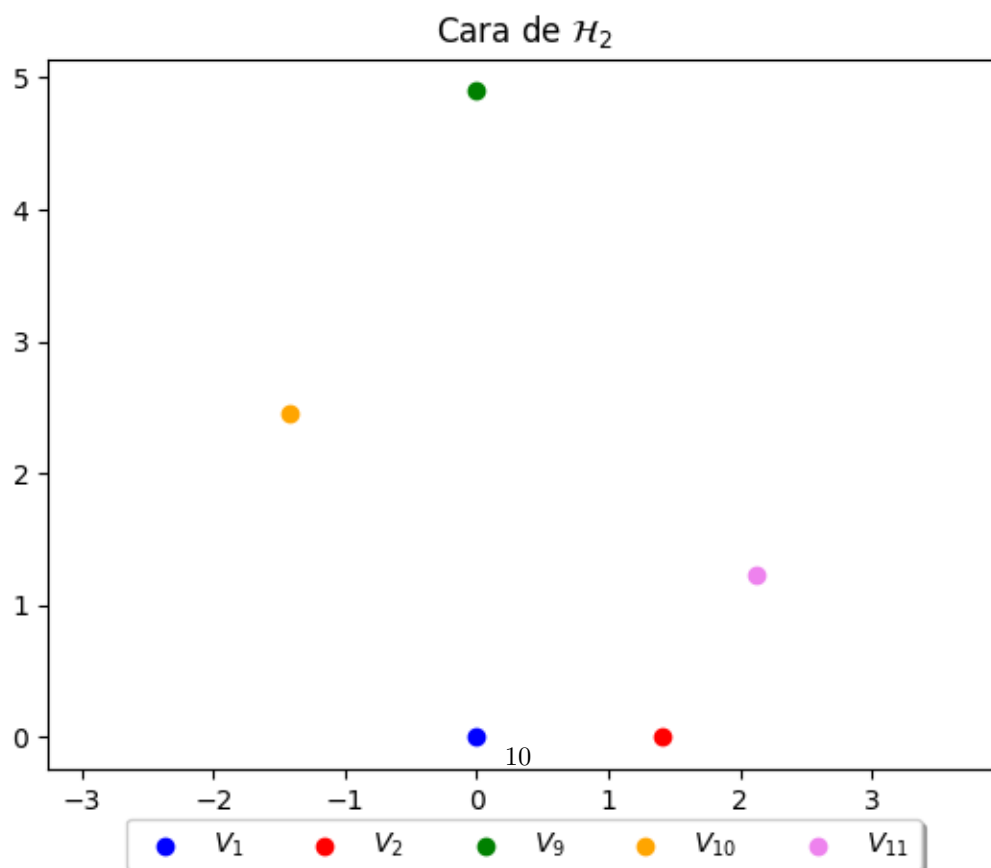
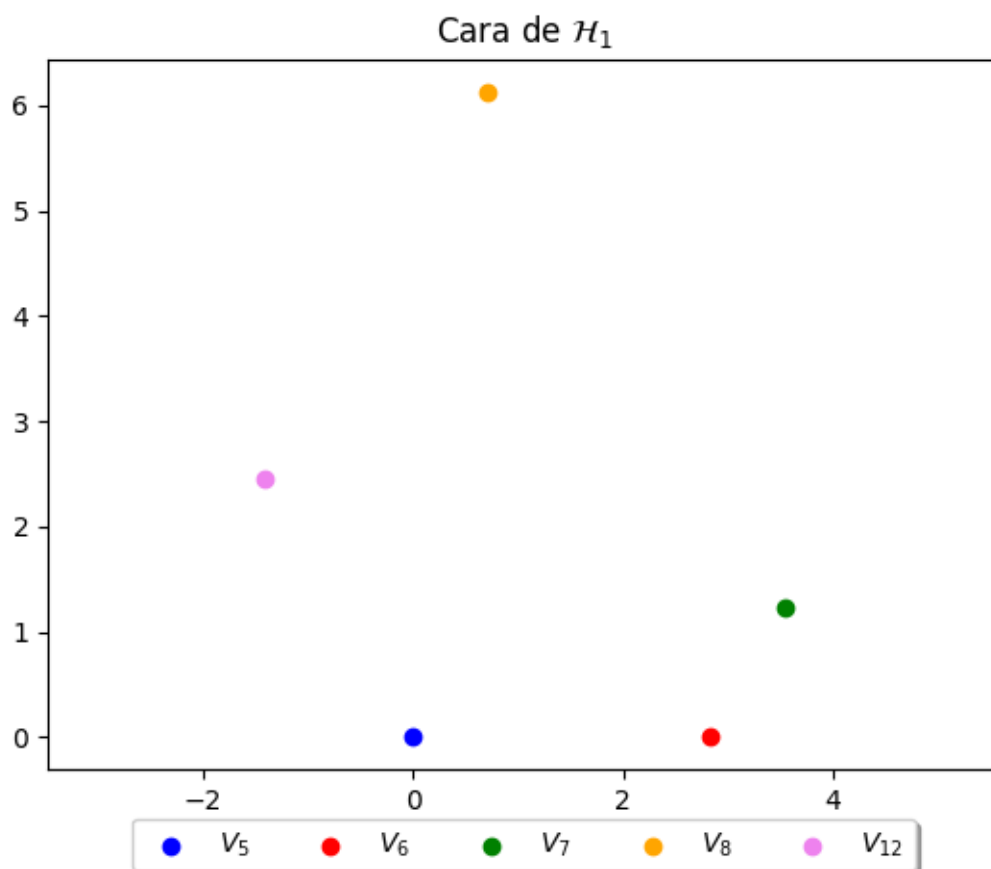
Cara corresponent al Hiperplà 8

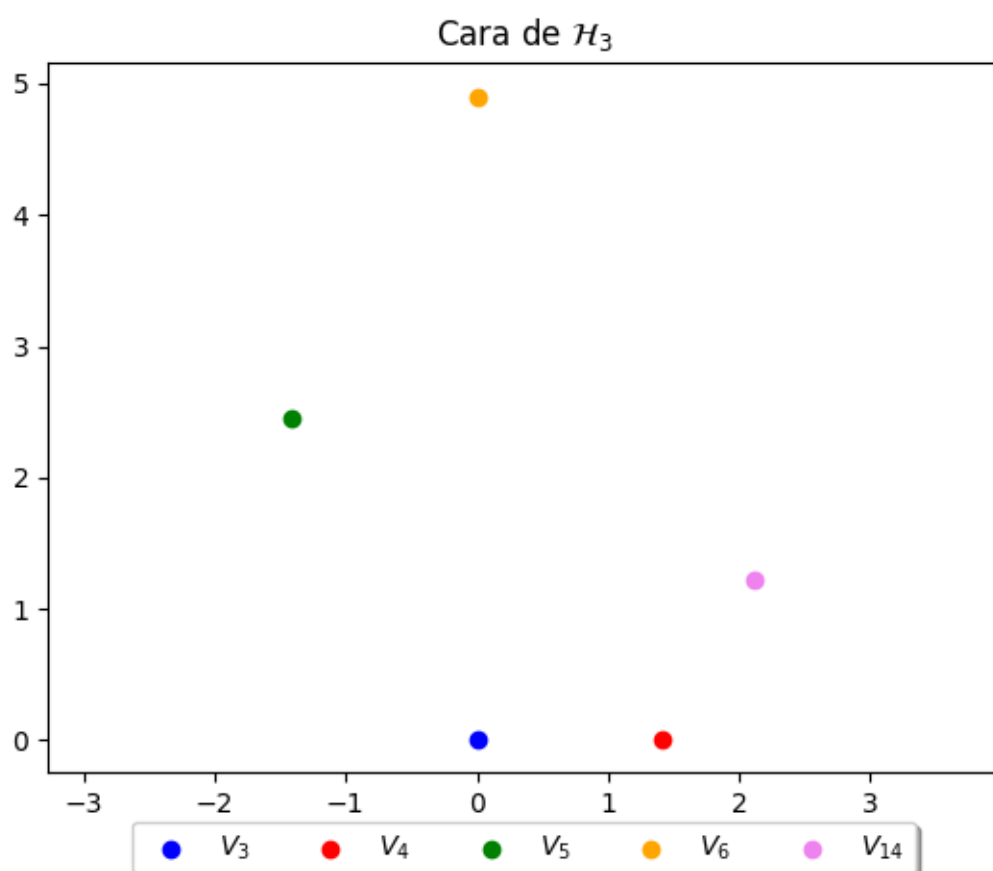
```
['V_{2}', 'V_{3}', 'V_{5}', 'V_{11}', 'V_{12}']  
[(0.0, 0.0), (1.41421356237310, 0.0),  
(2.82842712474619, 2.44948974278318),  
(-0.707106781186548, 1.22474487139159),  
(0.0, 2.44948974278318)]
```

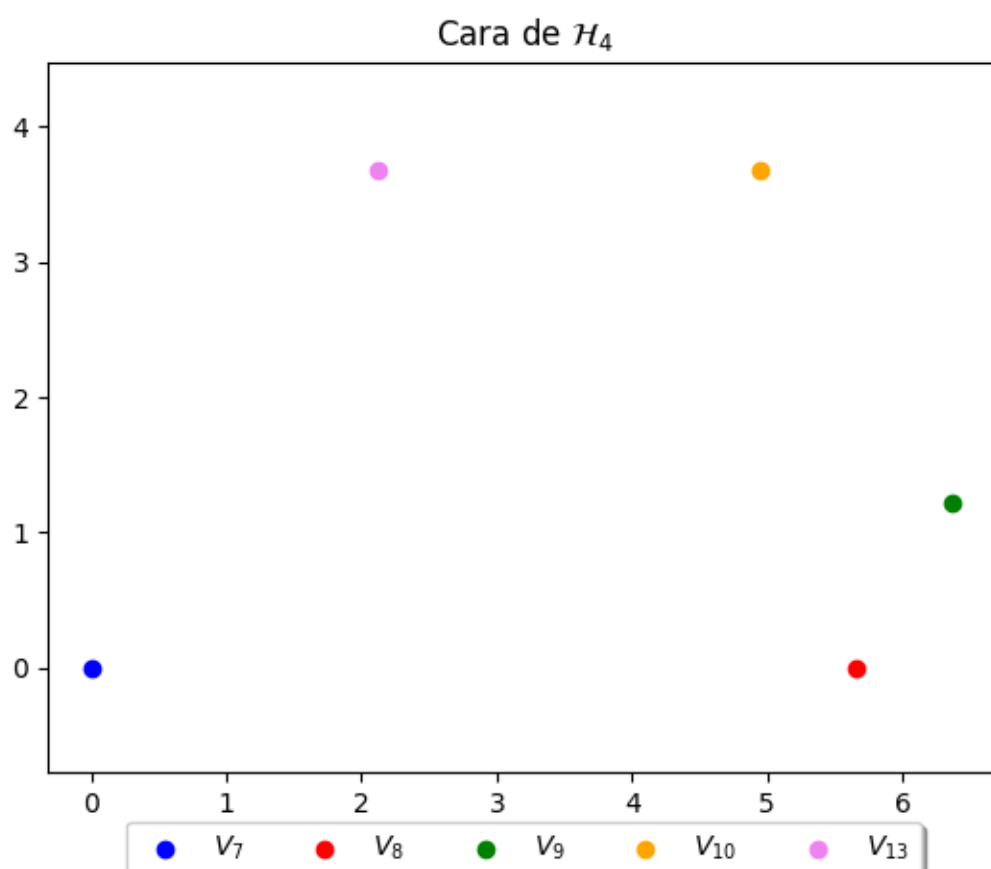
Cara corresponent al Hiperplà 9

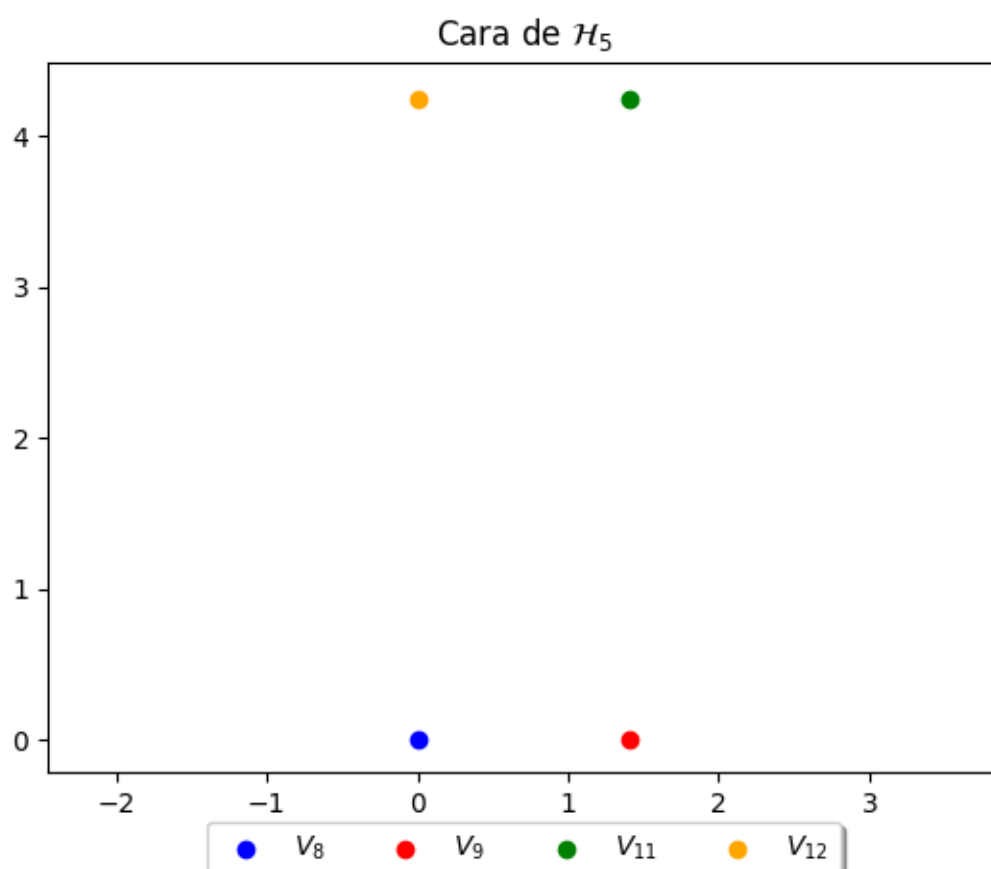
```
['V_{1}', 'V_{4}', 'V_{10}', 'V_{13}', 'V_{14}']  
[(0.0, 0.0), (1.41421356237310, 0.0),  
(-1.41421356237310, 2.44948974278318),  
(1.41421356237310, 2.44948974278318),  
(2.12132034355964, 1.22474487139159)]
```

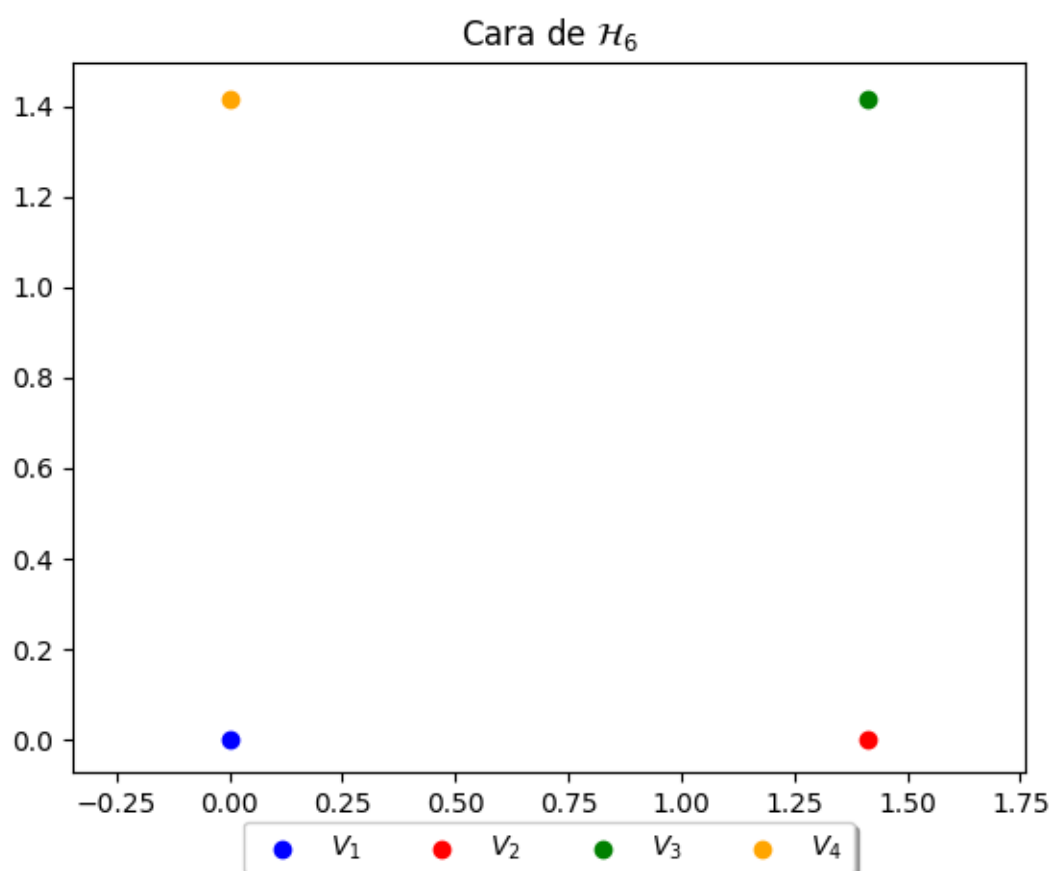
Imatges de les cares del poliedre

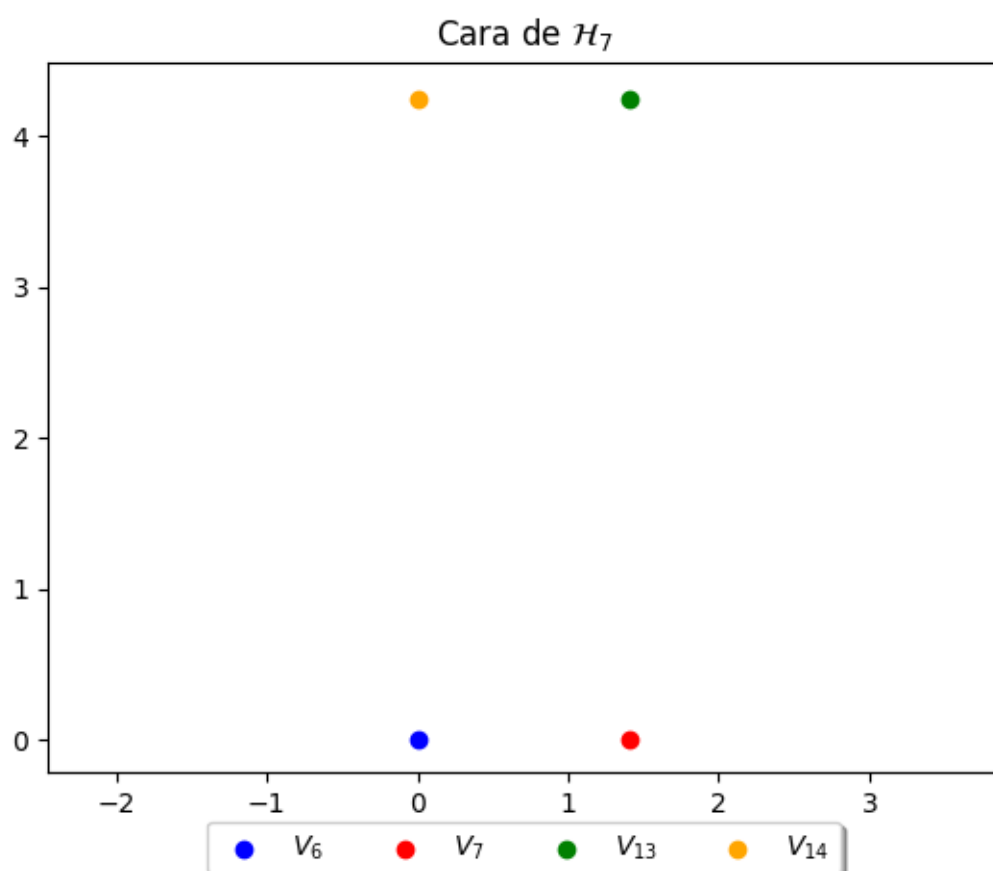


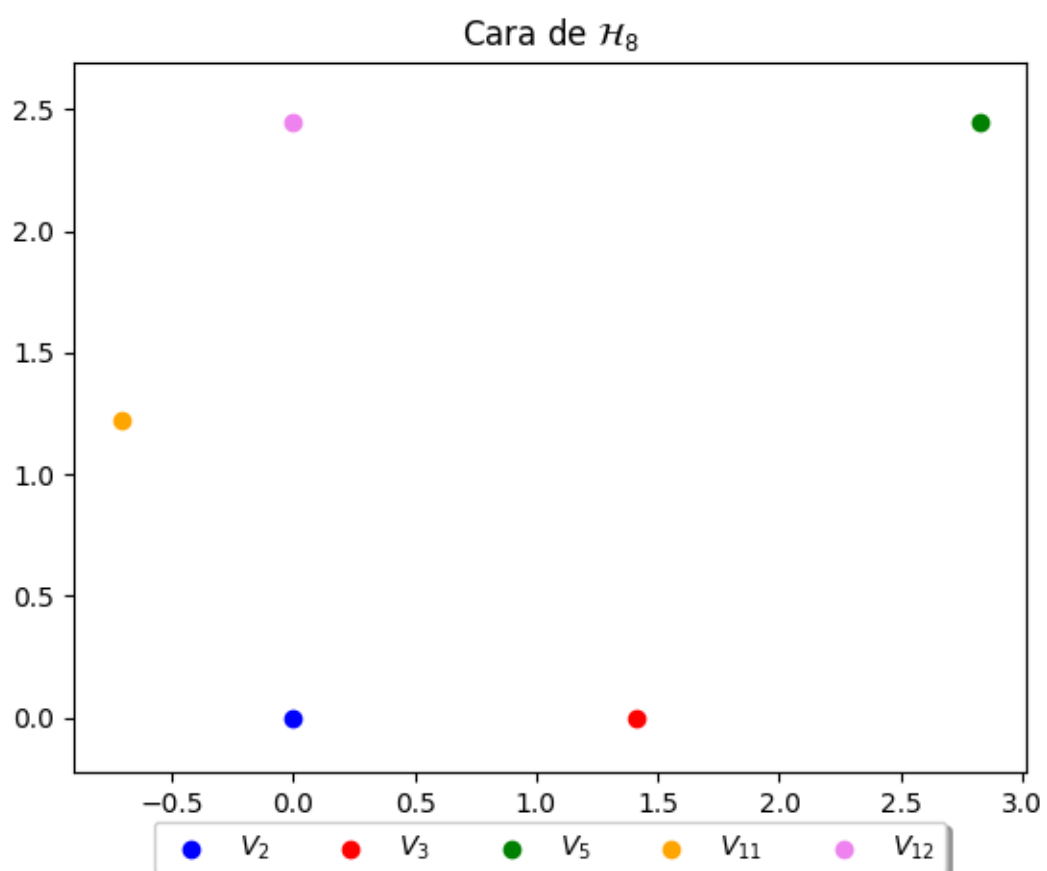




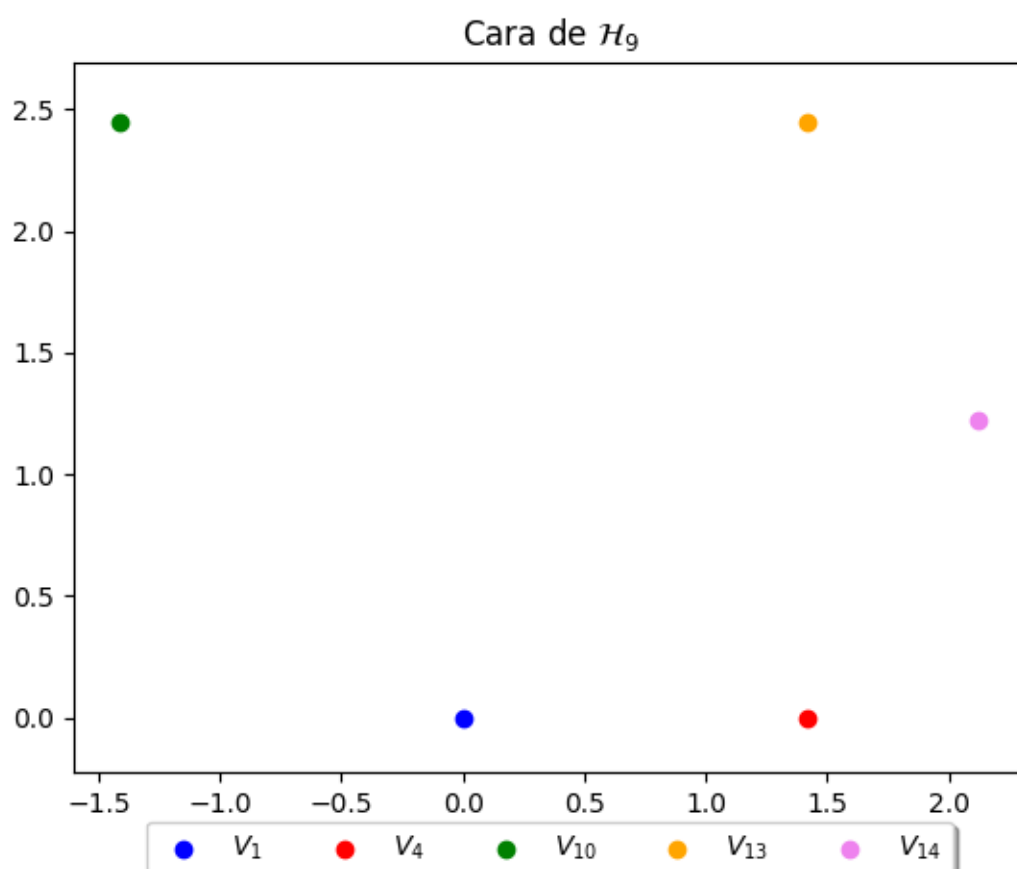












Imatge per imprimir

Imatge Construcció Final

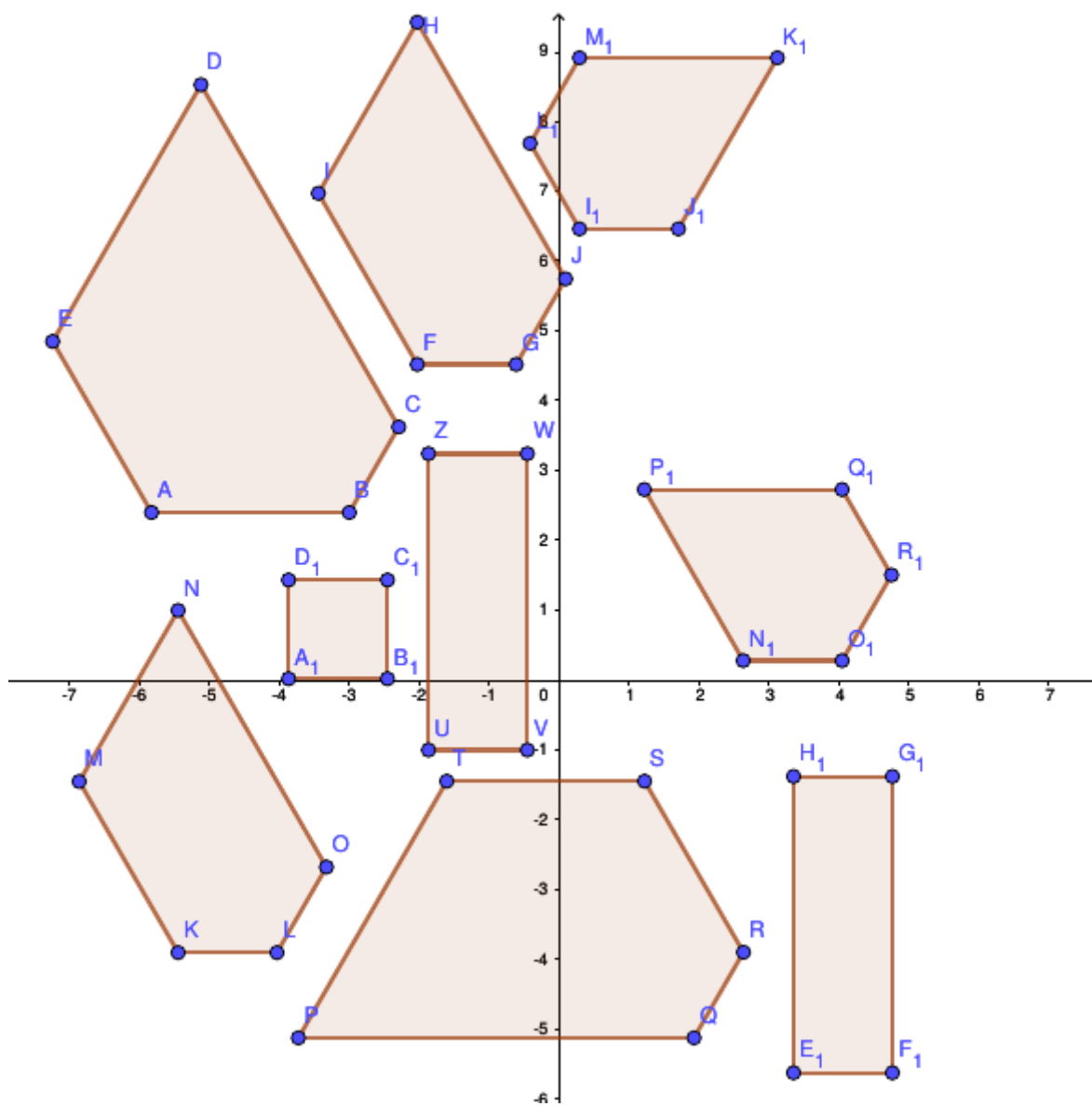


Figure 1: Imatge per imprimir el poliedre

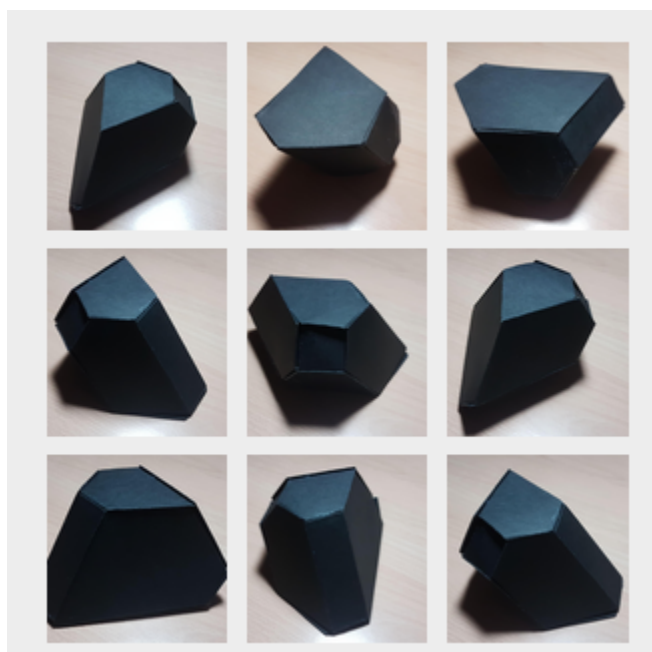


Figure 2: Colaix de la construcció en 3D