



PROYECTO 2

LITTLE CAD 2D

INTEGRANTES

. FRANCISCO RAFAEL ESPINOZA SAAVEDRA

. ADRIAN DANIEL POMA TORRES

- LAB

DESCRIPCIÓN DEL PROGRAMA



python™

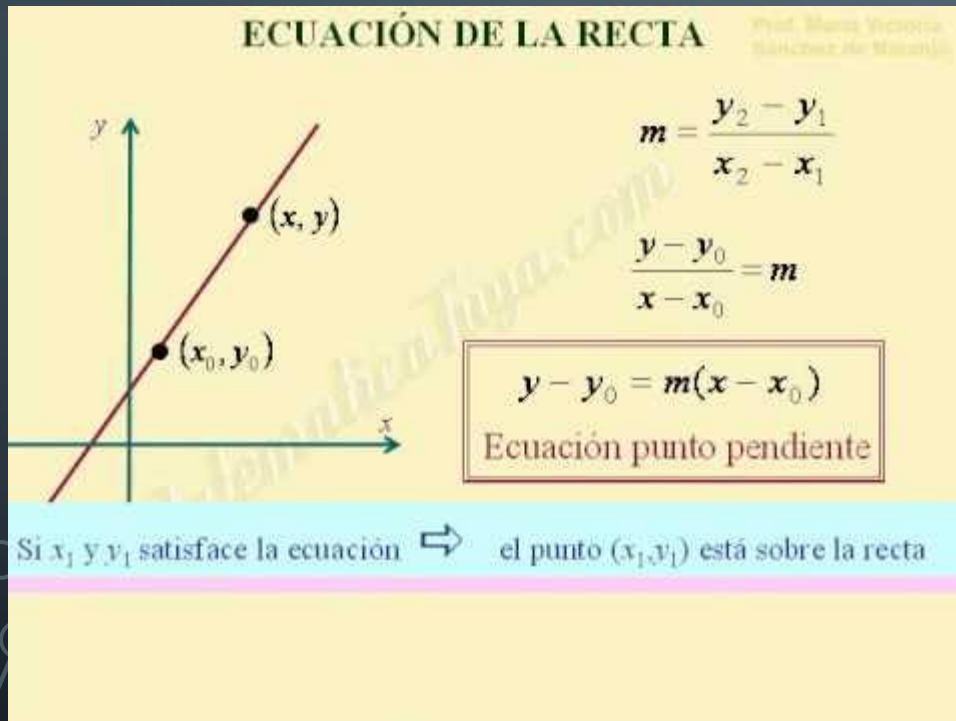
CONSTRUCCIÓN DE LA MATRIZ

```
def crear_lista(largo):  
    lista = []  
    for i in range(largo):  
        lista.append(" ")  
    return lista  
  
def crear_matriz(largo, alto):  
    matriz = []  
    for i in range(alto):  
        matriz.append(crear_lista(largo))  
    return matriz  
  
def lienzo(largo, alto):  
    matriz_lienzo = crear_matriz(largo, alto)  
    for i in range(alto):  
        for j in range(largo):  
            if j == 0 or j == largo - 1 or i == 0 or i == alto - 1:  
                matriz_lienzo[i][j] = "."  
    return matriz_lienzo
```

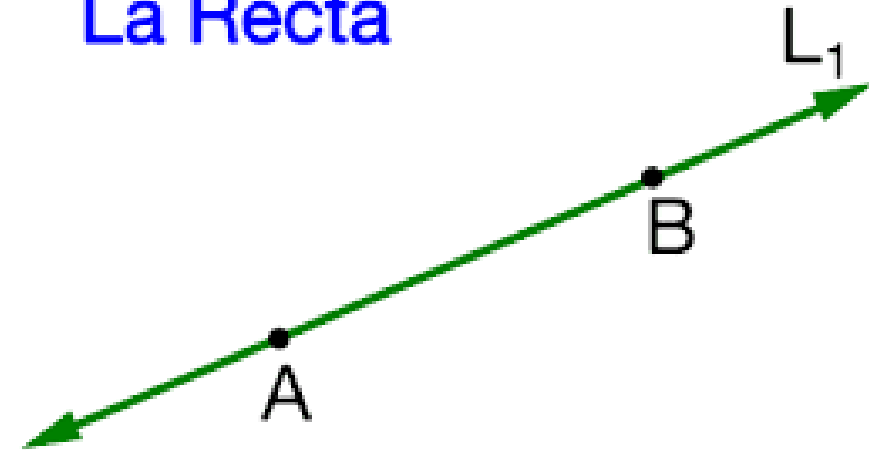
A decorative graphic on the left side of the slide, consisting of a network of thin, light-blue lines and small circles, resembling a circuit board or a neural network, extending vertically from the top to the bottom.

FUNCIONES MATEMÁTICAS UTILIZADAS

ECUACIÓN DE LA RECTA



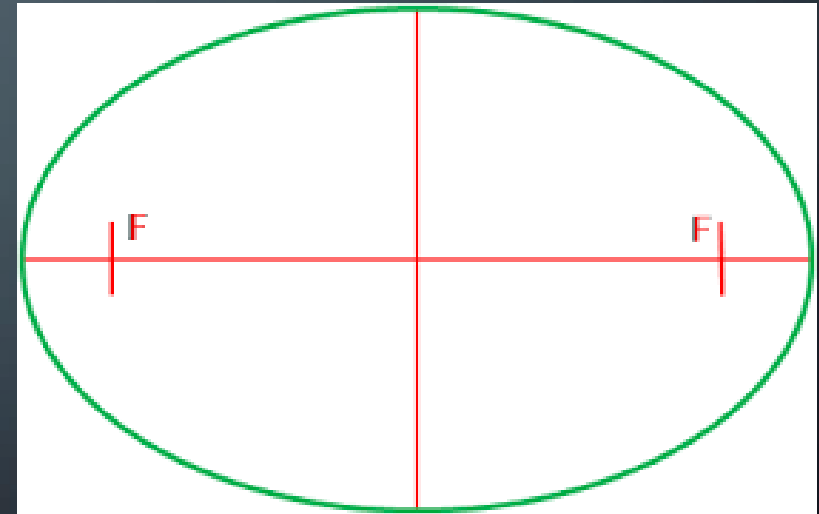
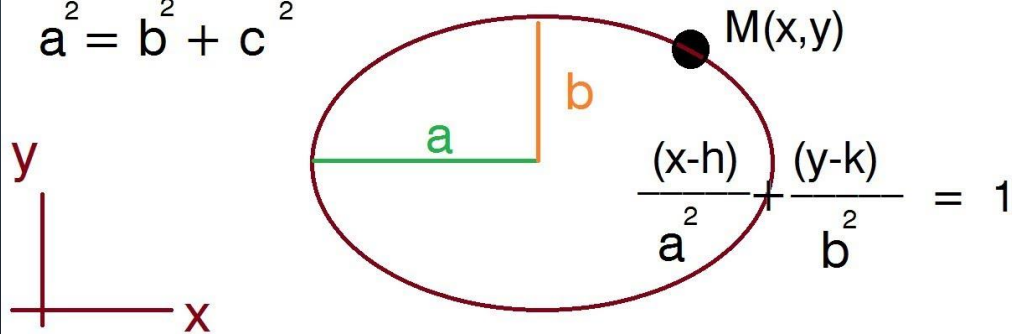
La Recta



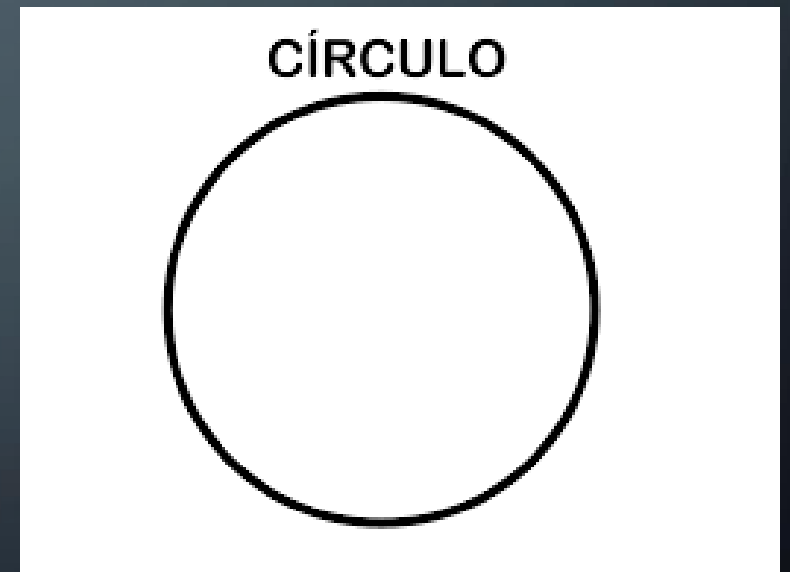
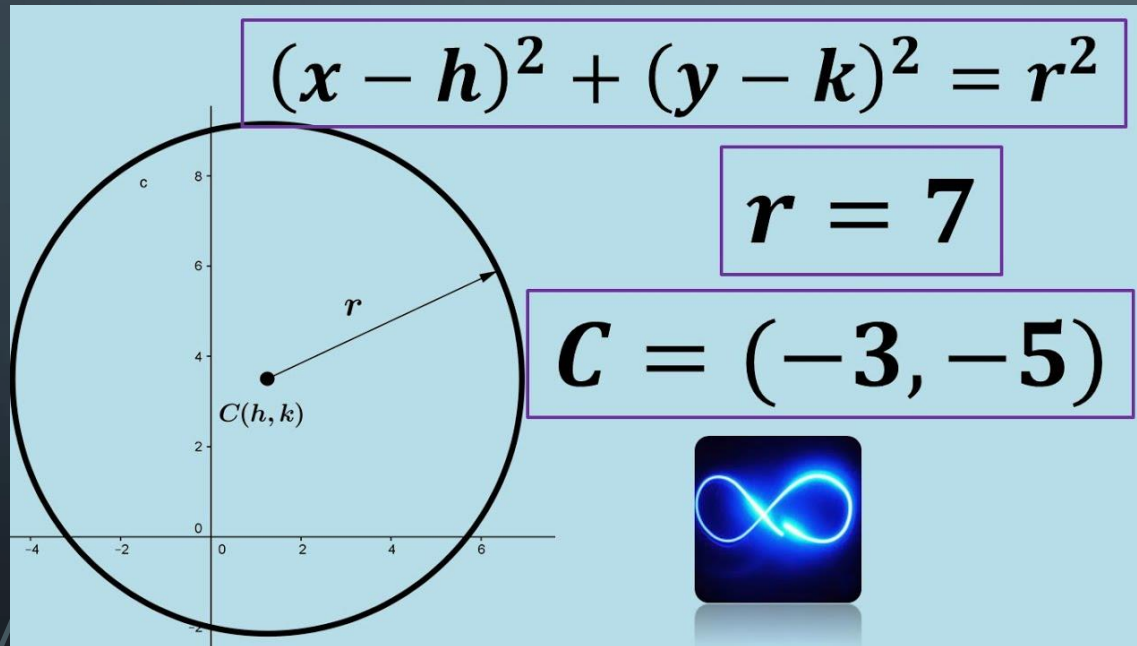
ECUACIÓN DE LA ELIPSE

Deducción de la ecuación de la elipse

$$a^2 = b^2 + c^2$$



ECUACIÓN DE LA CIRCUNFERENCIA

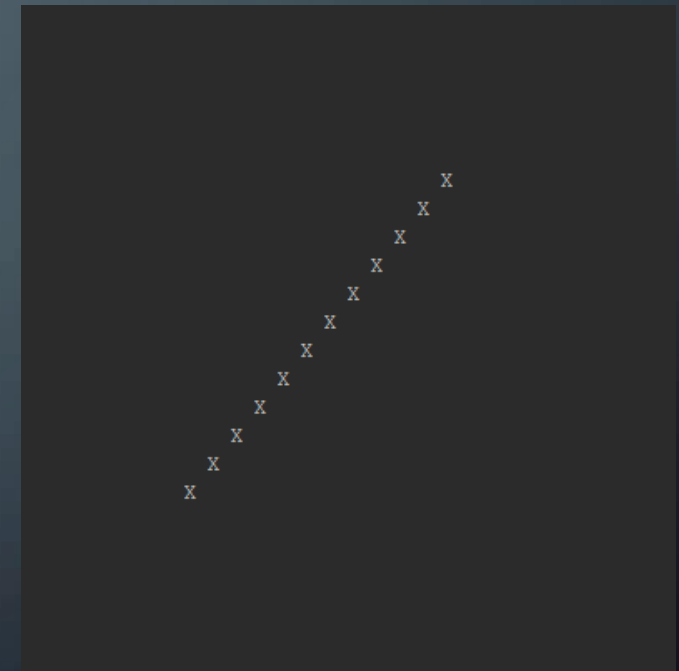




DIBUJANDO LAS FIGURAS

RECTA

```
def recta(*lista):
    global lienzoNuevo
    try:
        if lista.__len__() < 1:
            input_string = input("'Recta'\nIntroduzca los puntos P0(X0,Y0) y P1(X1,Y1): ")
            lista = input_string.split()
        if lista.__len__() < 5:
            a=1
        else:
            lienzoNuevo = lista[4]
            X0 = int(lista[0])
            Y0 = int(lista[1])
            X1 = int(lista[2])
            Y1 = int(lista[3])
            if Y1 == Y0:
                for x in range(min(X0, X1), max(X0, X1)+1):
                    lienzoNuevo = plotear(x, Y1, lienzoNuevo)
            if X1 != X0:
                m = ((Y1-Y0)/(X1-X0))
                for y in range(min(Y0, Y1), max(Y0, Y1)+1):
                    for x in range(min(X0, X1), max(X0, X1)+1):
                        if (y - Y1) == int(m * (x-X1)):
                            lienzoNuevo = plotear(x, y, lienzoNuevo)
            else:
                for y in range(min(Y0, Y1), max(Y0, Y1)+1):
                    lienzoNuevo = plotear(X1, y, lienzoNuevo)
            return lienzoNuevo
    except:
        print("\nIntroduzca valores válidos!\n")
```



CUADRADO

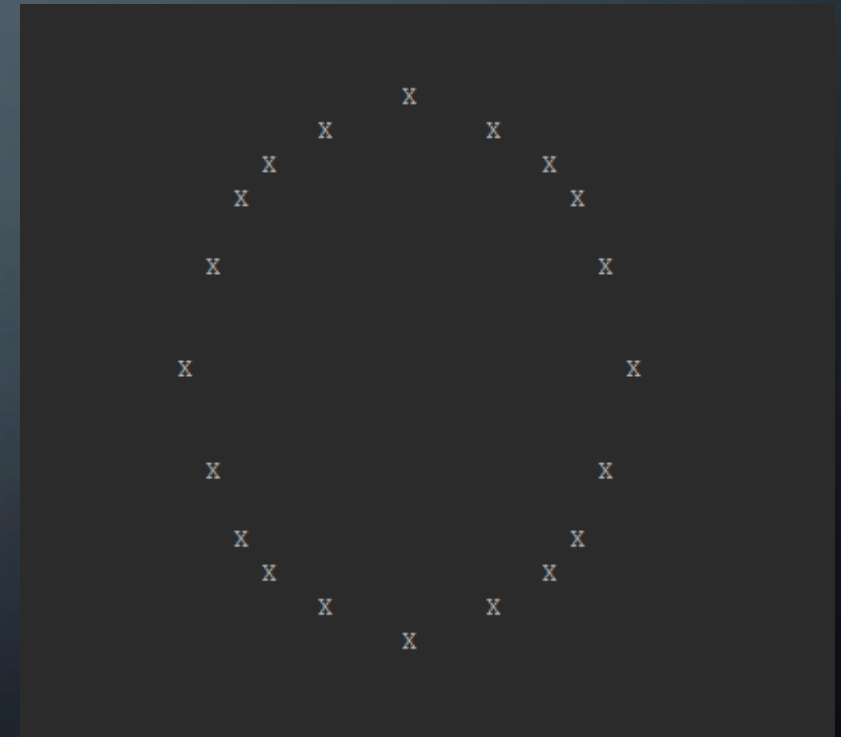
[illegible]

RECTANGULO

TRIANGULO

CIRCULO

```
def circulo():  
    global lienzoNuevo  
    try:  
        input_string = input("'Circulo'\nIntroduzca las coordenadas del centro P0(X0,Y0) y el radio r:  
        lista = input_string.split()  
        X0 = int(lista[0])  
        Y0 = int(lista[1])  
        r = int(lista[2])  
        for y in range(42):  
            for x in range(82):  
                if r**2-r < (x-X0)**2+(y-Y0)**2 <= r**2:  
                    lienzoNuevo = plotear(x, y, lienzoNuevo)  
        return lienzoNuevo  
    except:  
        print("\n;Introduzca valores válidos!\n")
```

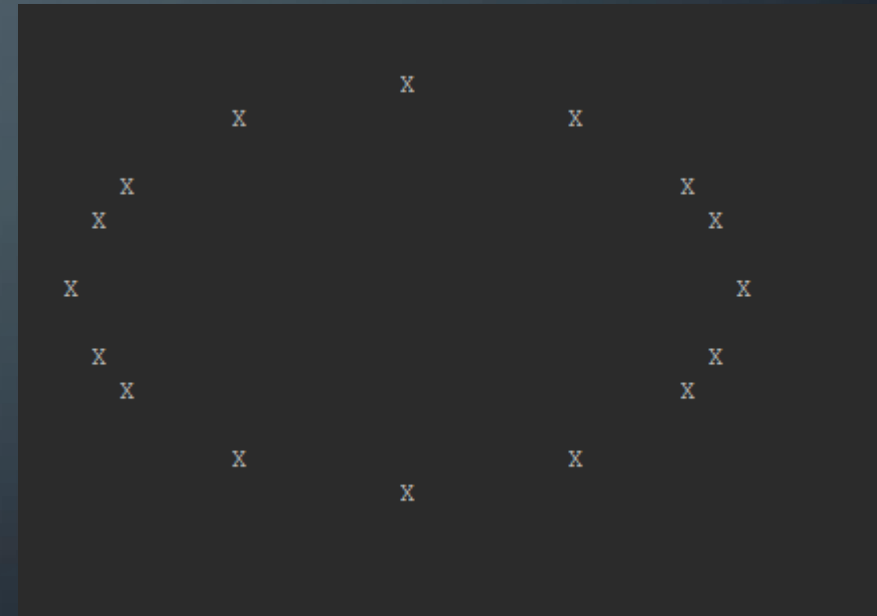



ELIPSE

```
def elipse():
    global lienzoNuevo
    try:
        input_string = input("'Elipse'\nIntroduzca las coordenadas del centro P0(X0,Y0) y \n"
                               "luego la medida del semieje 'X' y la medida del semieje 'Y', \n"
                               "separados por espacios: \n\n")

        lista = input_string.split()
        X0 = int(lista[0])
        Y0 = int(lista[1])
        a = int(lista[2])
        b = int(lista[3])
        for y in range(42):
            for x in range(82):
                if 0.92 < ((x-X0)/a)**2+((y-Y0)/b)**2 <= 1:
                    lienzoNuevo = plotear(x, y, lienzoNuevo)

        return lienzoNuevo
    except:
        print("\n;Introduzca valores válidos!\n")
```





MOSTRAR,
GUARDAR Y
LEER UN
DIBUJO



MOSTRAR UN DIBUJO

```
def mostrar(dibujofinal):  
    imprimir(dibujofinal)
```


GUARDAR UN DIBUJO

```
def guardar(dibujofinal):  
    nombrearchivo = input("Introduzca el nombre con que desea guardar el dibujo: \n") + ".cad"  
    with open(nombrearchivo, "a+") as archivoNuevo:  
        archivoNuevo.seek(0, 0)  
        for j in range(42):  
            for i in dibujofinal[41-j]:  
                archivoNuevo.write(i+",")  
            archivoNuevo.write("\n")  
    print("\n.\n.\n.\n.\n Archivo guardado satisfactoriamente.\n")
```

LEER UN DIBUJO

```
def leer():
    global lienzoNuevo
    global dibujo
    nombrearchivo = input("Introduzca el nombre del dibujo que desea abrir: \n") + ".cad"
    print('\n.\n.\n.\n Leyendo archivo: "', nombrearchivo, '"\n')
    try:
        with open(nombrearchivo, "a+") as abrirarchivo:
            abrirarchivo = open(nombrearchivo, "r")
            listalienzo = abrirarchivo.read()
            filas = listalienzo.splitlines()
            for y in range(42):
                temp = filas[y].split(",")
                if temp.__contains__("\\n"):
                    temp.remove("\\n")
                temp.pop(82)
                lienzoNuevo[41-y] = temp
            dibujo = lienzoNuevo
    except:
        print("No se encontro el archivo: '", nombrearchivo, "'")
```

A decorative graphic on the left side of the slide, consisting of a network of thin, light blue lines and small circles, resembling a circuit board or a stylized tree structure.

¡GRACIAS!