

# Visión Artificial

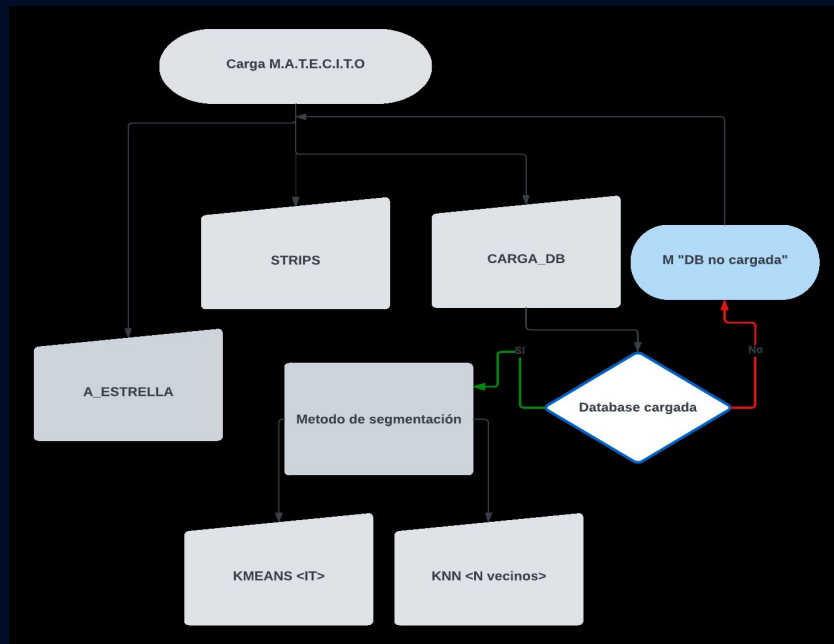
Inteligencia Artificial I  
Juan Ignacio Luna Casabene



# M.A.T.E.C.I.T.O



El software M.A.T.E.C.I.T.O fue diseñado para ser cargado directamente a la memoria del Robot IA. Se conecta por puerto serie con dicho robot y por medio de una interfaz gráfica o consola (GUI) puede ser controlado por el usuario, manejando las tareas a realizar y pasándole los parámetros correspondientes.



# M.A.T.E.C.I.T.O

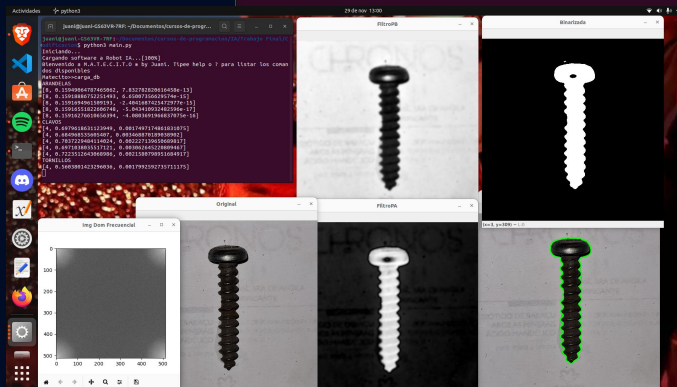
La forma en la que se controla dicha GUI es simple e intuitiva:

1. Se carga el software correspondiente a la memoria del Robot.
2. Se ingresa "HELP" para visualizar los comandos disponibles.
3. Se tipea el comando que dará lugar al algoritmo a utilizar.
4. En el caso de querer acceder a los algoritmos de agrupamiento con KMEANS y KNN es necesario primero cargar la base de datos de 20 imágenes, 5 de cada pieza.
5. Una vez cargada la base de datos, podemos acceder al método KMEANS pasándole el número de iteraciones de cálculo de los centroides, o en el caso de KNN el número de los K vecinos con los que se pretende trabajar.
6. Una vez ingresados los comandos, el programa accede y analiza las imágenes presentes en la carpeta "INPUT", las procesa, las clasifica y las guarda en las rutas relativas "Output/Kmeans" o "Output/Knn" según el método que se elija.

```
juani@juani-GS63VR-7RF: ~/Documentos/cursos-de-p...
juani@juani-GS63VR-7RF:~/Documentos/cursos-de-programacion/IA/Trabajo Final/Codificacion$ python3 main.py
Iniciando...
Cargando software a Robot IA..[100%]
Bienvenido a M.A.T.E.C.I.T.O © by Juani. Tipee help o ? para listar los comandos disponibles
Matecito>|
```

```
juani@juani-GS63VR-7RF:~/Documentos/cursos-de-p...
juani@juani-GS63VR-7RF:~/Documentos/cursos-de-programacion/IA/Trabajo Final/Codificacion$ python3 main.py
Iniciando...
Cargando software a Robot IA..[100%]
Bienvenido a M.A.T.E.C.I.T.O © by Juani. Tipee help o ? para listar los comandos disponibles
Matecito>help

Lista de comandos disponibles:
=====
a_estrella carga_db exit help kmeans knn strips
Matecito>|
```



# Etapapas



# El proceso....



## Adquisición

Obtención y Digitalización

01

02

## Segmentación

Muestras dotadas de significado. Pueden ser temporales o espaciales

03

04

## Clasificación

Algoritmos de clustering. Clasificación en grupos caract. similares

05

06

## Preprocesado

Eliminar ruido. Afinar bordes. Transform. a otro espacio

## Extracción

Extracción de propiedades o atributos cualitativos

## Interpretación

Interpretación de la información. Puede ser humana o manual o forma automática

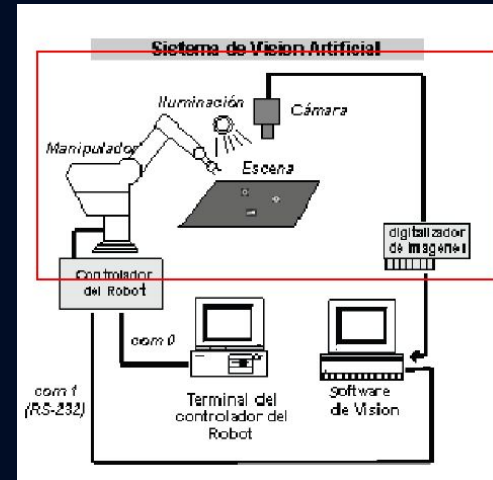


01

# Adquisición

# Una imagen dice más que mil palabras

Las magnitudes físicas (sonidos, imágenes, etc.) se transforman mediante sensores en señales eléctricas que una vez filtradas, amplificadas y digitalizadas pueden procesarse computacionalmente.





02

Preprocesado



# Preprocesado



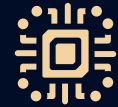
Reajuste de  
imagen



Escala de  
grises

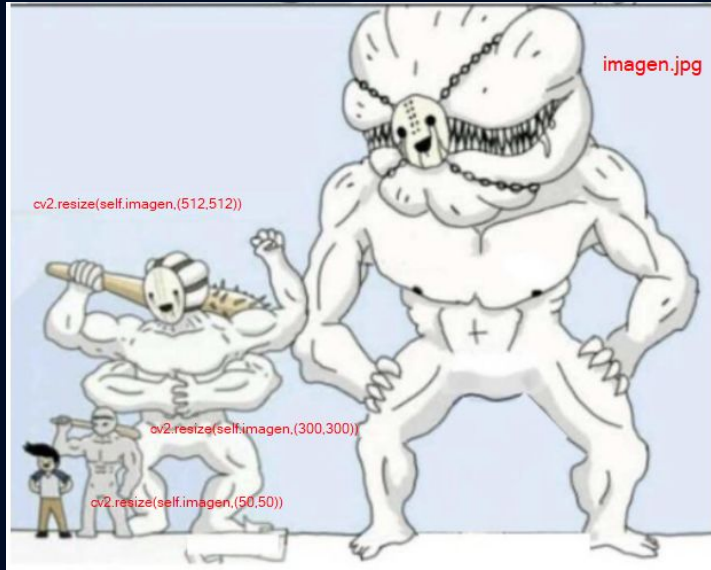


Filtrado



Binarización

# Reajuste de imagen



La primera etapa de nuestro preprocesamiento es reajustar la imagen a un tamaño fijo (para que cada imagen no se muestre en pantalla en su tamaño por defecto)



# Escala de grises



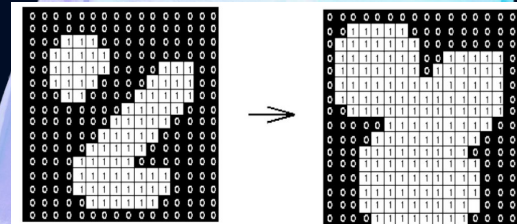
Todo pixel de una imagen a color tiene por defecto “3 canales”, un canal R[red], un canal G[Green] y un canal B[Blue], lo que en conjunto se conoce como RGB. RGB es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores de luz primarios. Cada canal está codificado con 1 byte (8 bits). Se admiten en escala decimal valores de 0 a 255

3 canales de 1 byte

1 canal de 1 byte

A modo de optimización se pueden usar métodos de:

- Erosión
- Dilatación



Effect of dilation using a 3x3 square structuring element



# Filtrado



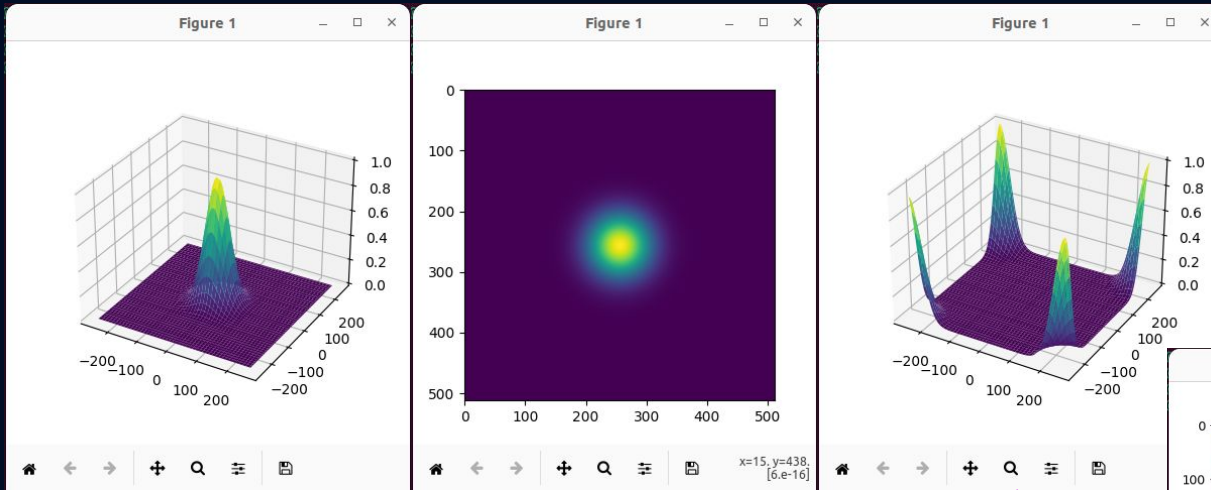
Filtros  
frecuenciales



Filtros  
espaciales

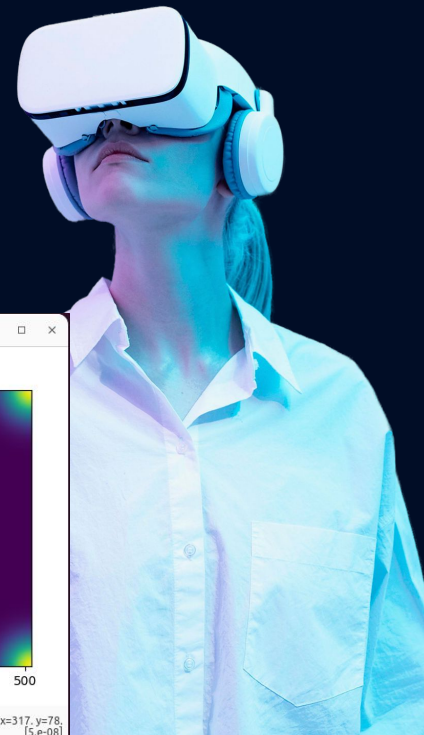
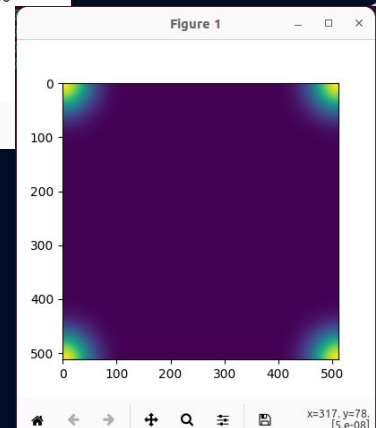
# Filtrado

Filtro frecuencial pasa bajo Gaussiano



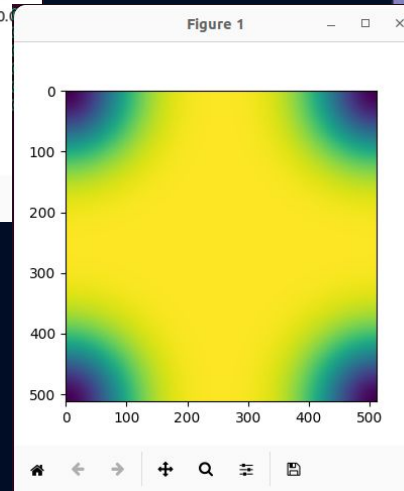
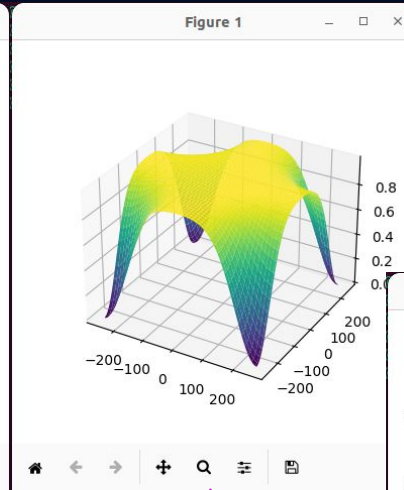
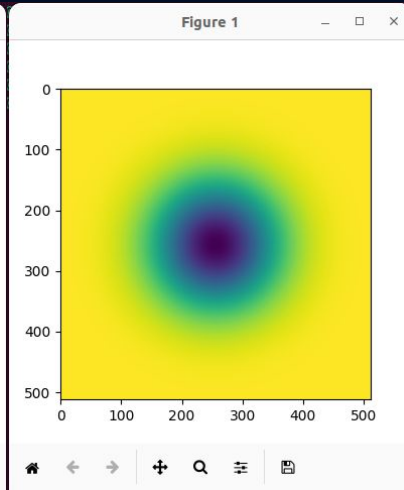
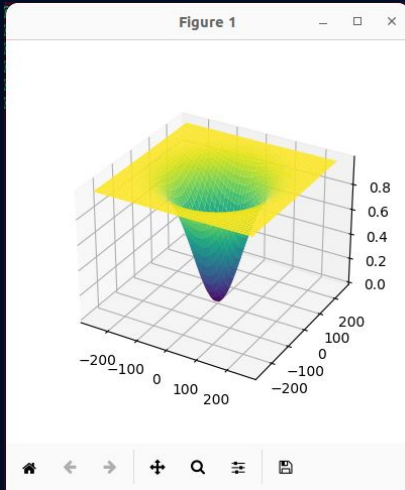
Fftshift →

$$G = e^{-\frac{1}{2}\left(\frac{R^2}{\sigma^2}\right)} \quad G = ke^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)} \quad R = \sqrt{x^2 + y^2} \quad \sigma_x = \sigma_y = \sigma$$



# Filtrado

Filtro frecuencial pasa alto Gaussiano



Ftshift

$$G = e^{-\frac{1}{2}\left(\frac{R^2}{\sigma^2}\right)}$$

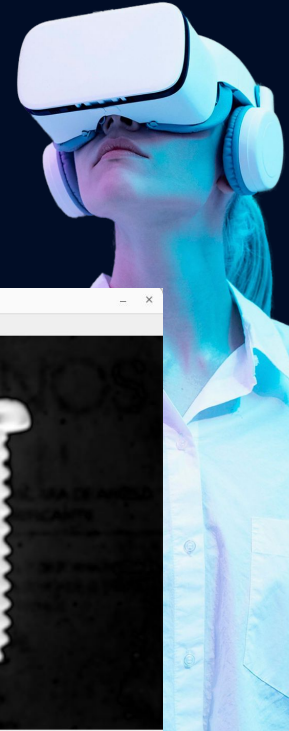
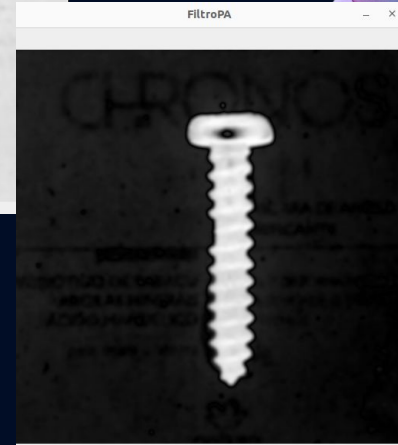
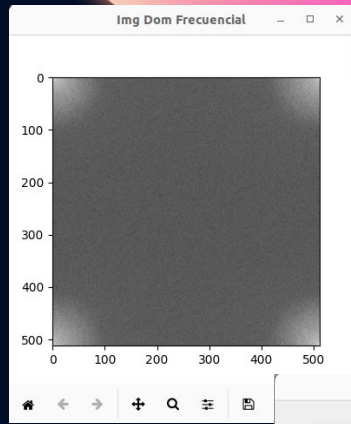
$$G = ke^{-\frac{1}{2}\left(\frac{x^2+y^2}{\sigma_x^2+\sigma_y^2}\right)}$$

$$R = \sqrt{x^2 + y^2}$$

$$\sigma_x = \sigma_y = \sigma$$

# Filtrado

Luego mediante el producto de la función filtro en el dominio frecuencial (habiendo aplicado primero la función `fftshift` para reacomodar el origen del centro a las esquinas) con la transformada de Fourier en 2D de la imagen, lo cual es equivalente a realizar la convolución entre ambas funciones en el dominio espacial, y posteriormente aplicando la antitransformada de Fourier en 2D a dicho producto, obtenemos las imágenes filtradas.



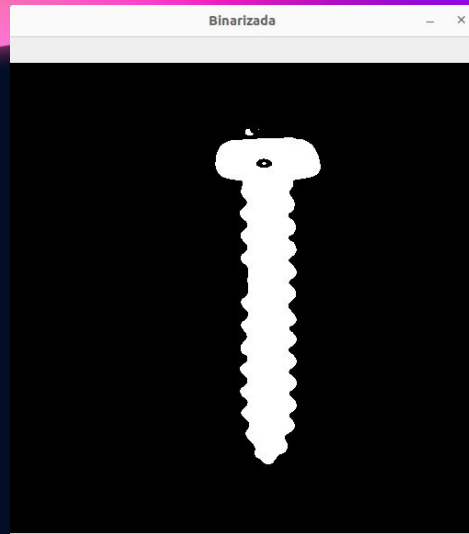
# Binarización

Se pasa la imagen de una escala de grises, con píxeles de 1 solo canal de intensidad que varía entre 0 a 255, a píxeles cuyos únicos valores posibles de intensidad sean 0 [negro] o 1 [blanco].

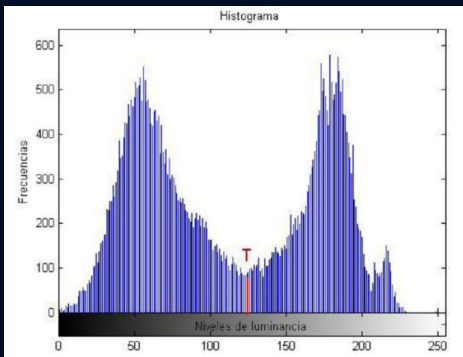
Método elegido:

Thresholding [método del valor umbral]

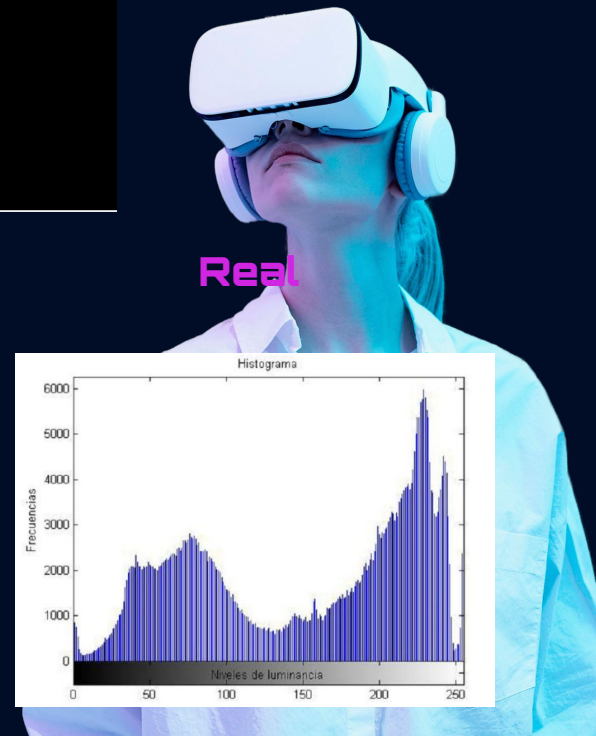
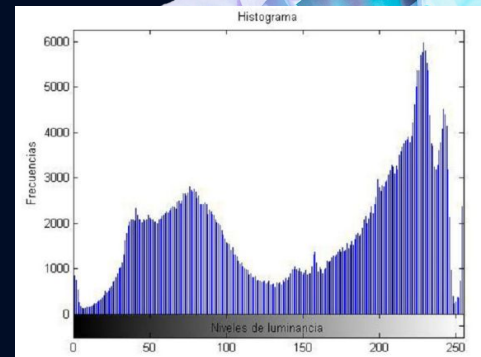
Se debe seleccionar un Valor umbral entre 0 y 255



Ideal



Real







03

# Segmentación

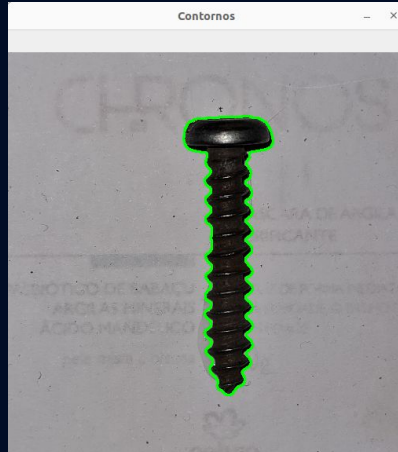


# Segmentación



Sirve para encontrar dentro de la información los elementos individuales que parecen estar dotados de significado. Pueden clasificarse en:

- Temporales: se aíslan fragmentos de la secuencia de datos recibidos a lo largo del tiempo (ej: micro fonemas en voz).
- Espaciales: el conjunto de datos en un instante contiene varias subunidades relevantes (varios objetos en una escena)



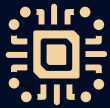
M.A.T.E.C.I.T.O utiliza algunos métodos y módulos de programación , que a partir de una imagen binarizada me generan los puntos de contorno, los cuales luego de una interpolación se convierten en el contorno mismo de la imagen, permitiendo calcular el perímetro de la pieza, el área, los momentos invariantes de Hu,etc.

The background is a dark, almost black, space filled with vibrant, glowing shapes. In the top left, there's a large, soft-edged circle with a gradient from purple to pink. In the top right, there are several smaller, glowing blue and cyan circles of varying sizes, some with a slight glow. On the right side, there's a large, irregular shape with a gradient from purple to pink, resembling a planet or a nebula. The overall aesthetic is futuristic and digital.

04

Extracción

# Parámetros utilizados por M.A.T.E.C.I.T.O



Aproximación  
polinomial



1er Momento  
Invariante de  
Hu



6to Momento  
Invariante de  
Hu

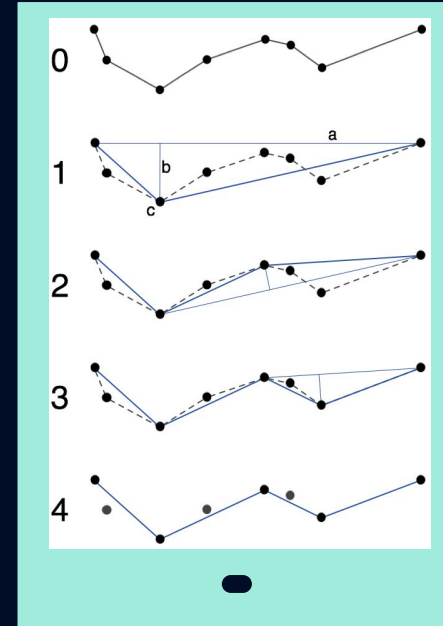
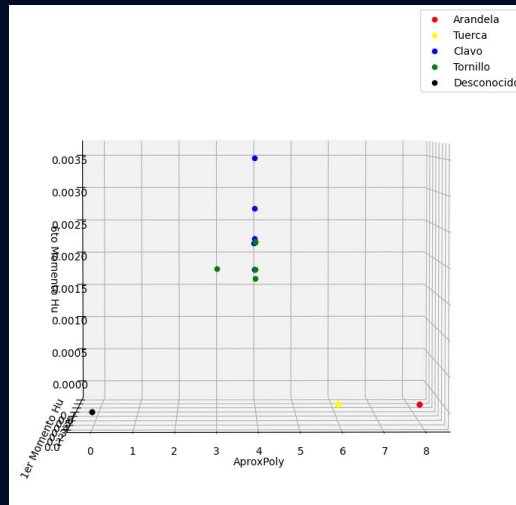
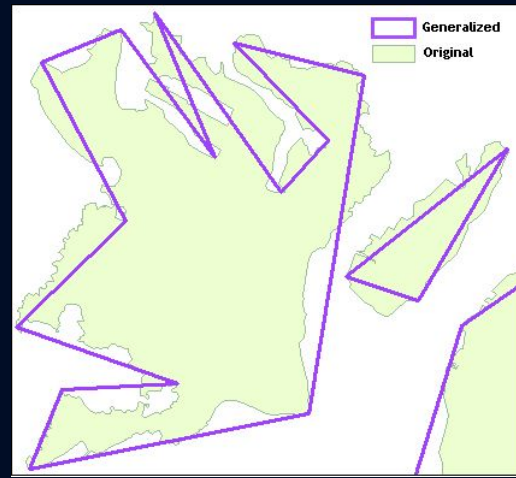
# Aproximación polinomial

La aproximación polinomial es un método que pretende aproximar el contorno de una figura mediante la interconexión recursiva de rectas.

Para ello M.A.T.E.C.I.T.O utiliza el algoritmo Ramer–Douglas–Peucker (RDP), utilizado para reducir el número de puntos utilizados en la aproximación de una curva.

Ventajas:

- Diferenciación tornillos/clavos de arandelas/tuercas
- Diferenciación tuercas de arandelas (más importante)



# Algunos rasgos geométricos....

## Circularidad

¿Qué tanto se parece a un círculo perfecto?

$$\text{roundness} = \frac{4\pi \cdot \text{area}}{(\text{convex perimeter})^2}$$

## Convexidad

Medición de la aspereza del perímetro de la imagen bidimensional tomada

$$\text{convexity} = \frac{\text{convex perimeter}}{\text{perimeter}}$$

## Elongación

Se define como relación de aspecto o ancho/longitud

$$\text{elongation} = \frac{\text{width}_{\text{bounding-box}}}{\text{length}_{\text{bounding-box}}}$$



# Momentos de imagen

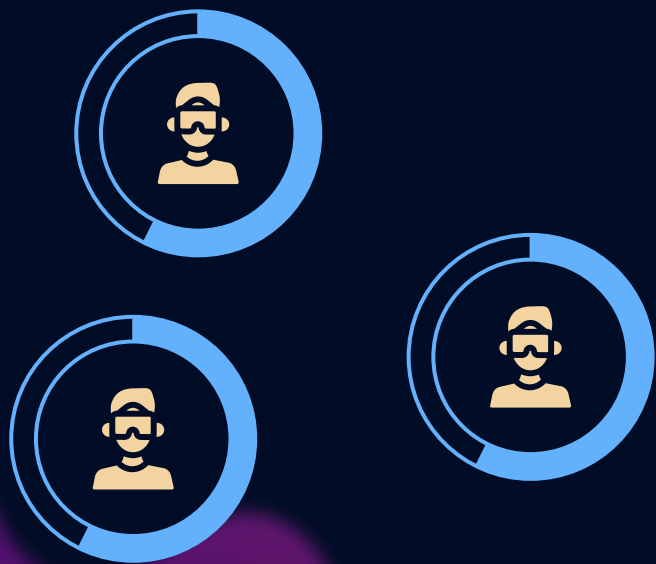
Un momento de imagen es cierto promedio ponderado particular de las intensidades de los píxeles de una imagen. El dominio de una imagen es discreto, y está dado por las coordenadas de sus píxeles. Tales coordenadas se expresan con números enteros.

Interpretando las intensidades de los píxeles como  $I(x,y)$ , los momentos de una imagen se calculan de la siguiente manera:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y)$$

# Momentos de imagen

## Momentos centrales



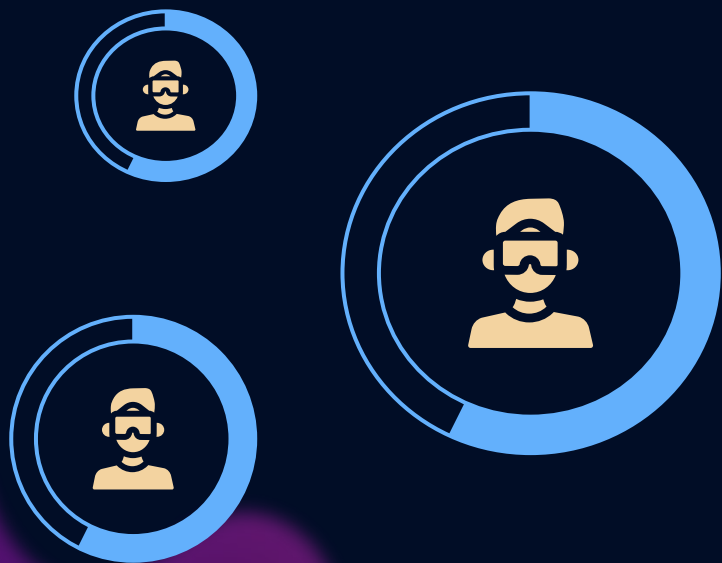
## Invariencias





# Momentos de imagen

## Momentos normales



## Invariencias



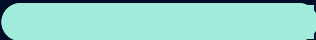
# Momentos de imagen

## Momentos invariantes de Hu



## Invariencias

100%  Invariante a la traslación

100%  Invariante a la escala

100%  Invariante a la rotación

Invariantes a la traslación,  
escala y rotación



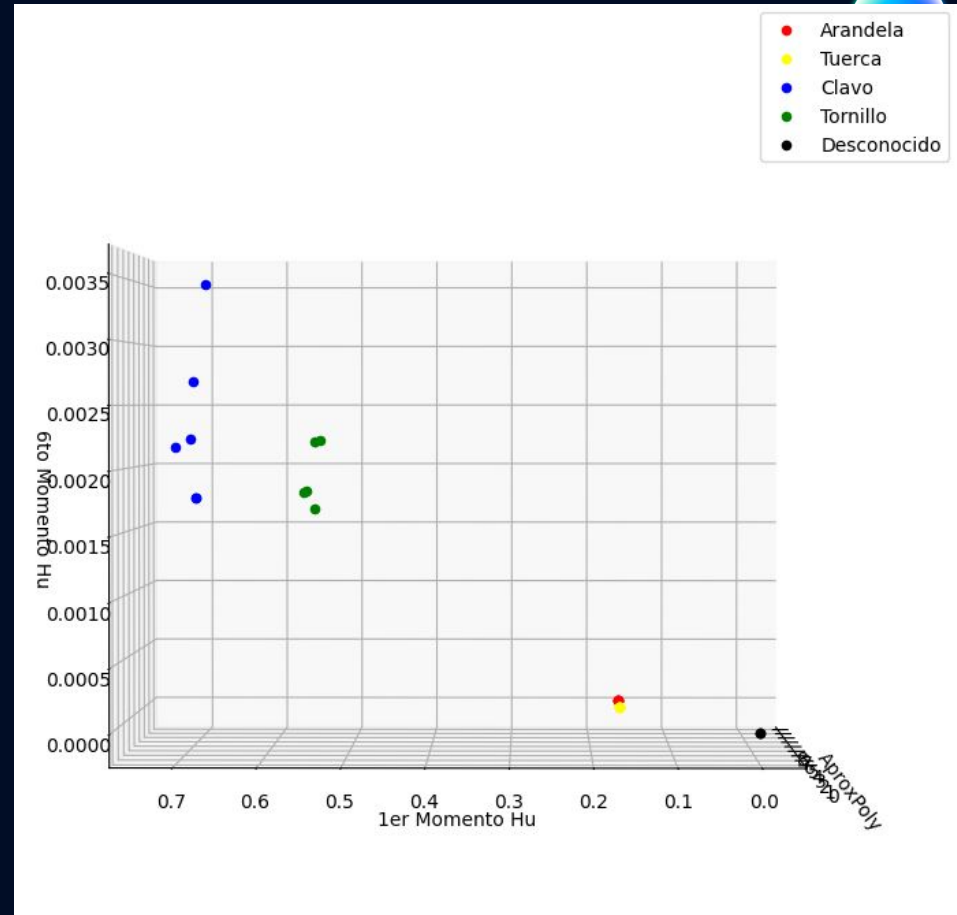
# 1er y 6to Momento Invariante de Hu

Ventajas del 1er momento de Hu:

- Buena diferenciación de lo que son tornillos y clavos por un lado, y arandelas y tuercas por otro
- Diferenciación entre los clusters de clavos y tornillos [importante]

Ventajas del 6to momento de Hu:

- Contribuye a la diferenciación entre los puntos que pertenecen al mismo cluster, evitando así el overfitting [en el caso de arandelas y tuercas no es apreciable debido a la escala]





05

# Clasificación

# Algoritmos de clustering



Supervisado

Knn



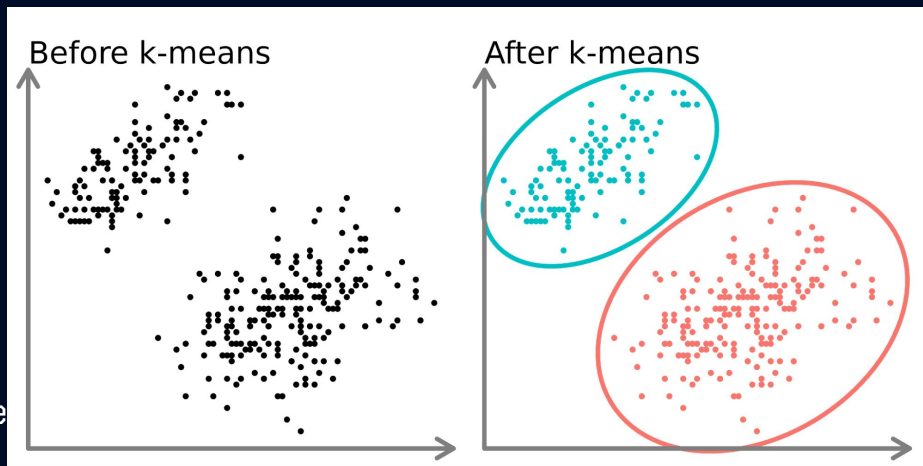
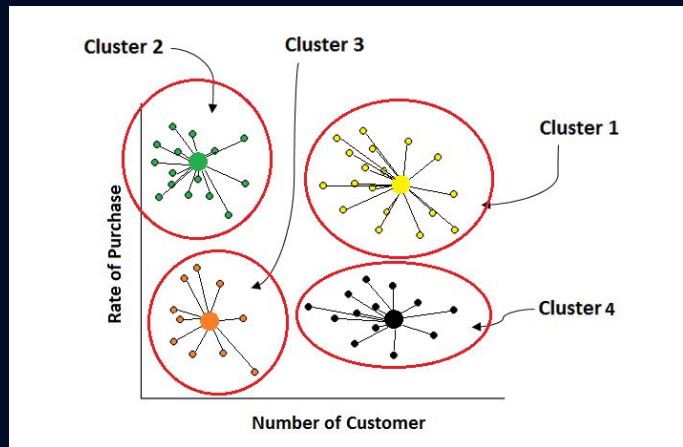
No supervisado

K-means

# K-means

K-means es un método de agrupamiento, que tiene como objetivo la partición de un conjunto de  $n$  observaciones en  $k$  grupos o “clusters” en el que cada observación pertenece al grupo cuyo valor medio es más cercano. El nombre K-means proviene del hecho de que este método representa cada uno de los “clusters” por la media [o media ponderada] de sus puntos, es decir, por su centroide.

- K centroides que determinan los clusters
- No supervisado. Información presente en la base de datos no se encuentra previamente clasificada.
- Centroides determinados de manera aleatoria
- Recalculo iterativo de centroides mediante la media aritmética de las distancias euclidianas del centroide a cada nodo

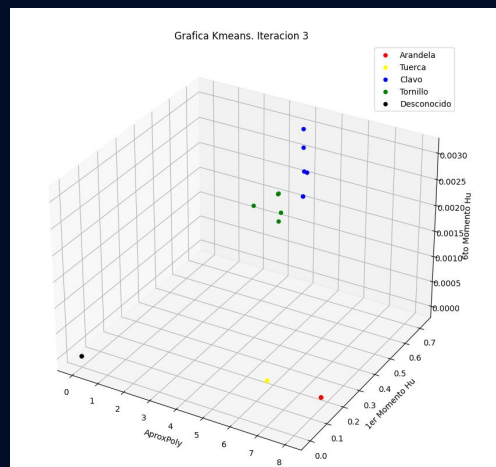
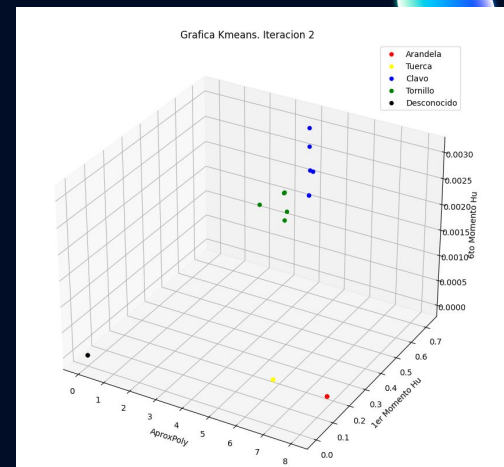
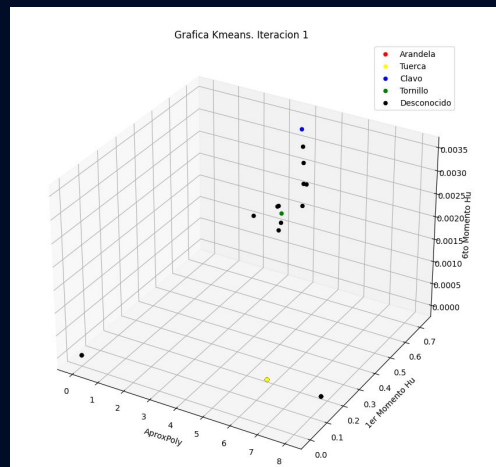


# K-means

M.A.T.E.C.I.T.O da la opción de trabajar con el algoritmo K-means como estrategia para clasificar las piezas. Para este software se plantea una pequeña modificación que automatiza la posterior interpretación

1. Trabajamos con K=4
2. Elegimos aleatoriamente como centroides a los puntos correspondientes a 1 imagen de cada grupo de piezas de la base de datos (20 imágenes en total, 5 imágenes de cada pieza)
3. Generamos los clusters
4. “Propagamos” la etiqueta correspondiente
5. Iteramos, recalculando los centroides

Si bien no es intrínseco del algoritmo Kmeans ya que se supone no hay una preclasificación dado que es “no supervisado”, esto nos sirve simplemente para que, una vez realizada la agrupación por medio del algoritmo, se etiquete al cluster con el nombre de la pieza que le corresponde. Podría parecer que esto corrompe la esencia misma del algoritmo, sin embargo esto no es así, porque a la hora de comenzar con el agrupamiento una vez calculados los centroides, el algoritmo no sabe a qué categoría corresponde cada punto, simplemente verifica la distancia euclidiana de cada punto a cada uno de los 4 centroides y según cual haya sido la distancia mínima de cada uno de los puntos respecto a cada centroide, coloca al punto en el cluster que le corresponde, y recién ahí lo etiqueta, según cual haya sido la etiqueta del centroide del cluster correspondiente.

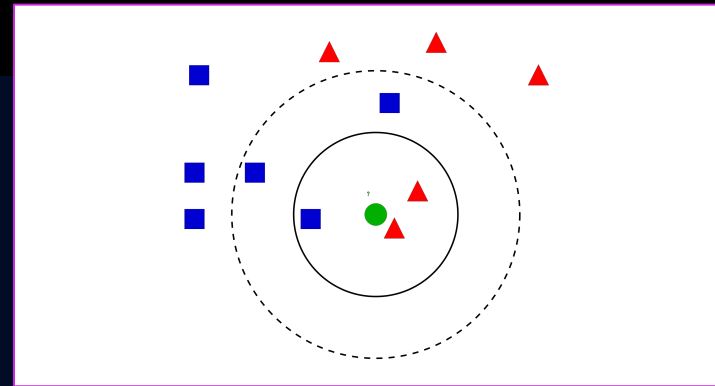
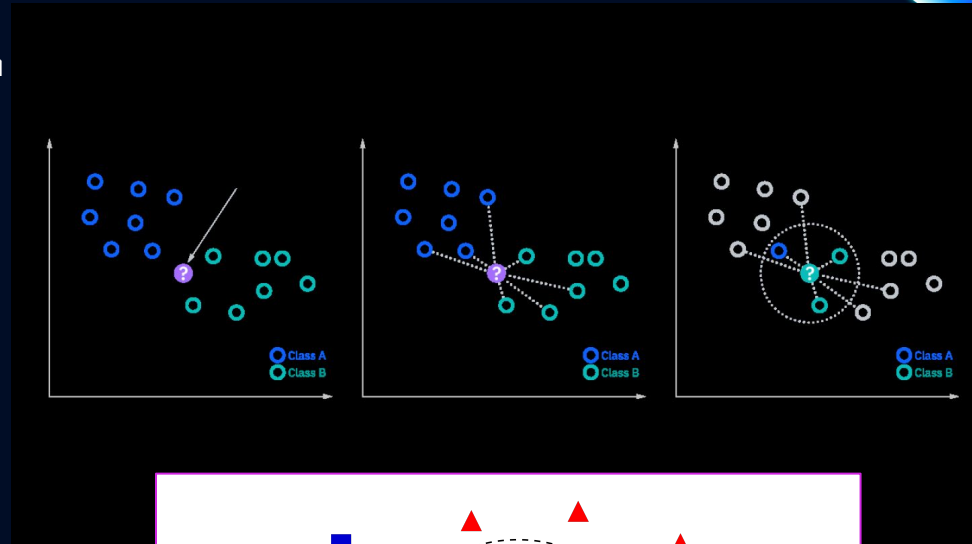


```
juani@juani-GS63VR-TRF...  
[4, 0.5466837141754873, 0.0016643971024612079]  
Iteracion 1  
Centroides:  
[[8, 0.15919184966750968, -3.341475015987438e-14], [6, 0.1597059724802038, -1.1282678306706147e-11], [4, 0.684968535605407, 0.003468870189038902], [4, 0.5658011821220297, 0.001988189966990194]]  
Iteracion 2  
Centroides:  
[[8.0, 0.15916240934684786, -1.5139714591282156e-15], [6.0, 0.1597867001706275, 7.072321592985247e-15], [4.0, 0.697961863112395, 0.0017497174861831075], [4.0, 0.5603801423296036, 0.0017992592735711175]]  
Iteracion 3  
Centroides:  
[[8.0, 0.15916240934684786, -1.5139714591282156e-15], [6.0, 0.1597867001706275, 7.072321592985247e-15], [4.0, 0.697961863112395, 0.0017497174861831075], [4.0, 0.5603801423296036, 0.0017992592735711175]]  
Imagenes guardadas en la carpeta Output/Kmeans  
Matecito>>]
```

# Knn

El algoritmo de k vecinos más cercanos, también conocido como KNN o k-NN, es un clasificador de aprendizaje supervisado no paramétrico, que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual. Si bien se puede usar para problemas de regresión o clasificación, generalmente se usa como un algoritmo de clasificación, partiendo de la suposición de que se pueden encontrar puntos similares cerca uno del otro.

- K vecinos
- Entrenamiento de IA. 3 conjuntos de datos: Train, Validation y Test
- Valores grandes de k reducen efecto del ruido en la clasificación, pero crean límites entre clases parecidas.
- Supervisado. Elementos de la base de datos clasificados a priori

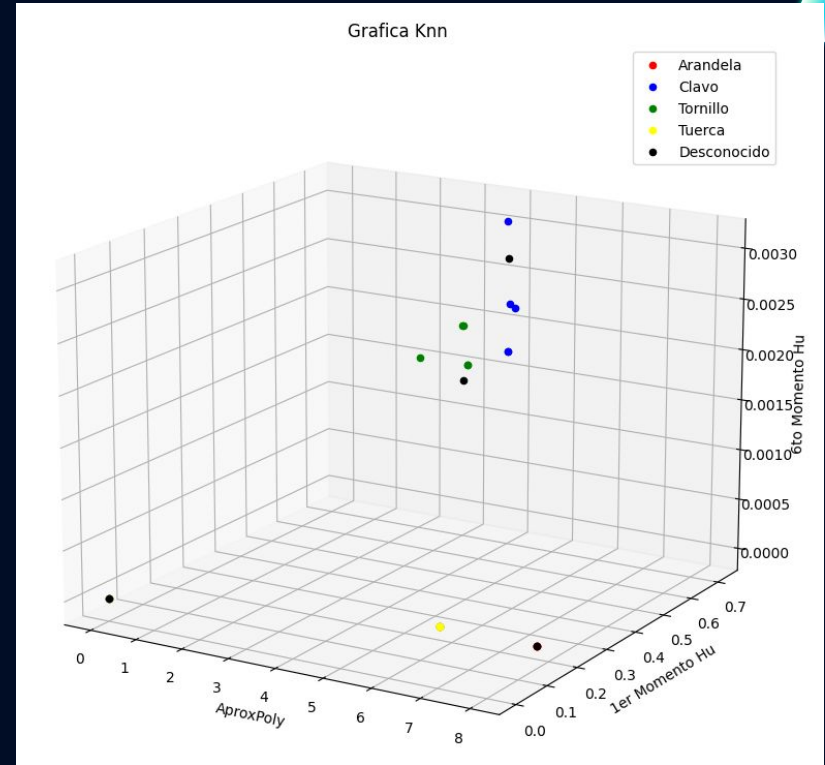




# Knn

M.A.T.E.C.I.T.O da la opción de trabajar con el algoritmo Knn como estrategia para clasificar las piezas.

1. Se cargan los elementos preclasificados de la base de datos (el software solo trabaja con un conjunto de train)
2. Se le pasa el parámetro de los K vecinos a analizar
3. Para cada punto se calcula la distancia euclidiana a todos los demás puntos (imaginarse propagación de esfera), y solo nos quedamos con los K puntos que presenten menor distancia
4. En cada iteración, una vez seleccionado el punto con la distancia más pequeña, se actualiza a un valor grande (1000 en mi caso), para que de esta forma al realizar una nueva iteración, el programa pueda tomar como punto con la mínima distancia, al siguiente punto con la distancia mínima.
5. Finalmente se evalúa la etiqueta de los K vecinos más cercanos a cada punto y se analiza la frecuencia de los datos. La etiqueta que más veces se repita será la que utilizemos para etiquetar a mi pieza desconocida.



Podría pasar que el análisis no sea concluyente debido a que no predomina ninguna etiqueta frente a otra. En este caso lo que se hace es aumentar en 1 el valor de K y ejecutar nuevamente el algoritmo, entendiendo que, en algún momento, uno de los valores predominan frente a los demás. El software se programó para hacerlo automáticamente.



06



Interpretación

# Interpretación

La interpretación de la información obtenida una vez clasificadas las imágenes puede ser en forma humana o manual o forma automática, M.A.T.E.C.I.T.O posee una interpretación del tipo automática dado que es el robot el que toma las imágenes de el directorio "Output" presente en su memoria interna, y luego a partir de allí interpreta el orden de las cajas y las reordena a partir de STRIPS





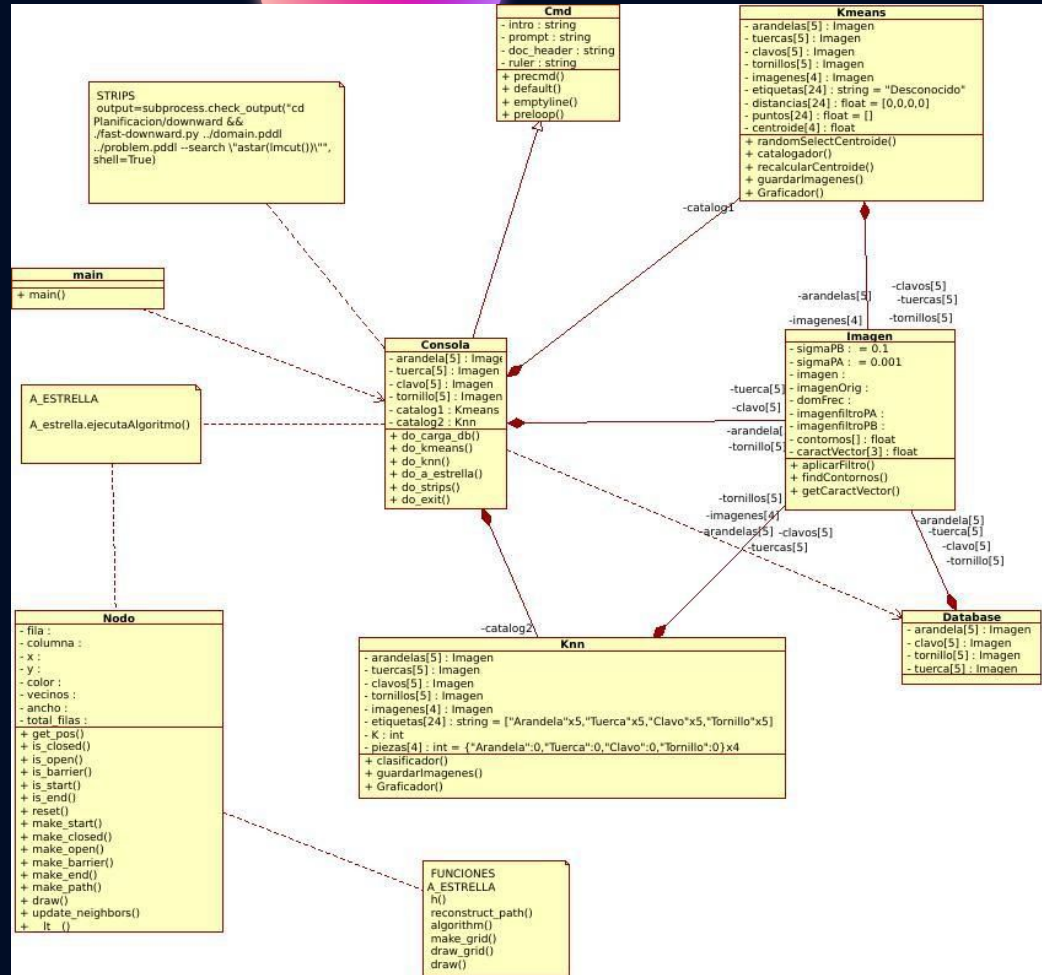
**Implementación**

# Implementación



es un elixir

# Implementación



The background is a dark, almost black, space filled with several glowing, ethereal shapes. On the left, a large, irregular shape transitions from a bright pink to a deep purple. In the top center, a glowing blue arc is visible. In the top right, a soft purple glow is present. In the bottom left, there are three small, glowing blue spheres of varying sizes. In the bottom right, a large, glowing pink and purple sphere is partially visible. The overall aesthetic is futuristic and digital.

**Gracias!**