

# **Visión Artificial**



**Título: Practica 9**

**Nombre: Mario Gerardo Casas Miramontes**

**Registro: 22310165**

## Objetivo

El objetivo de esta practica es el de realizar la comparación de dos imágenes en busca de similitudes entre las mismas, y siendo además de poder controlar el nivel de parentesco para la detección de las dos imágenes.

## Funcionamiento

El funcionamiento del código se basa en el usar dos imágenes principalmente, la primera siendo con la cual vamos a buscar coincidencias y la otra siendo la imagen que buscamos encontrar en la primera.

Para ello usamos la función de match template con la cual buscamos similitudes entre la imagen principal y la búsqueda, en base a eso generamos una variable threshold con la cual marcaremos un nivel de parentesco con el cual se pueda decir que si tiene un porcentaje de similitud se pueda dibujar un rectángulo que delimite el ROI

## Código Fuente

#Mario Gerardo Casas Miramontes 22310165

```
import cv2
```

```
import numpy as np
```

```
img_rgb = cv2.imread('Principal.jpg')
```

```
img_gray2 = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)
```

```
template = cv2.imread('Busqueda.jpg',0)
```

```
w, h = template.shape[::-1] #El tamaño del parentesco
```

```
res = cv2.matchTemplate(img_gray2,template,cv2.TM_CCOEFF_NORMED) #Funcion para  
buscar similitudes entre las imagenes
```

```
threshold = 0.85 #Grado de parentesco
```

```
loc = np.where( res >= threshold)
```

```
for pt in zip(*loc[::-1]): #Genero bloques par marcar la zona de interes
```

```
    cv2.rectangle(img_rgb, pt, (pt[0] + w, pt[1] + h), (0,255,255), 2)
```

```
cv2.imshow('Detected',img_rgb)
```

## Resultado



<https://github.com/M-C117/VA/tree/main/Practica%209>