

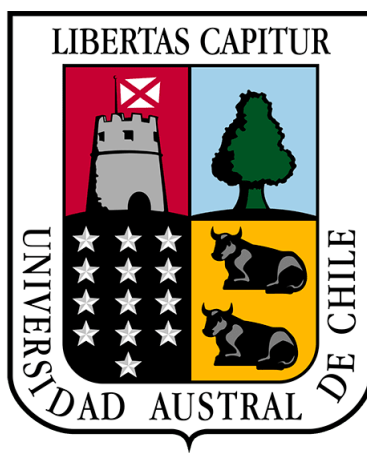
---

**UNIVERSIDAD AUSTRAL DE CHILE**

**FACULTAD DE CIENCIAS DE LA INGENIERÍA**

**INGENIERÍA CIVIL ELECTRÓNICA**

---



## **ELEP 233 - VISIÓN ARTIFICIAL Y REDES NEURONALES**

DESARROLLO DE TAREA N°3

**PROFESOR:**

Gustavo Schleyer.

**INTEGRANTES:**

Ángel Andrade

Jorge Palavecino.

30 de mayo de 2023

# Índice general

Pregunta 1	3
Pregunta 2	4
Pregunta 3	4
Pregunta 4	6
Pregunta 5	7
Pregunta 6	8

El presente documento desarrolla la tarea 4 del ramo ELEL233. El repositorio en GitHub de la actividad puede encontrarse en <https://github.com/Atrabilis/UACH/tree/main/Trabajos/Vision%20artificial/Tarea%204>.

Las imágenes originales utilizadas para el desarrollo de la tarea fueron las siguientes:

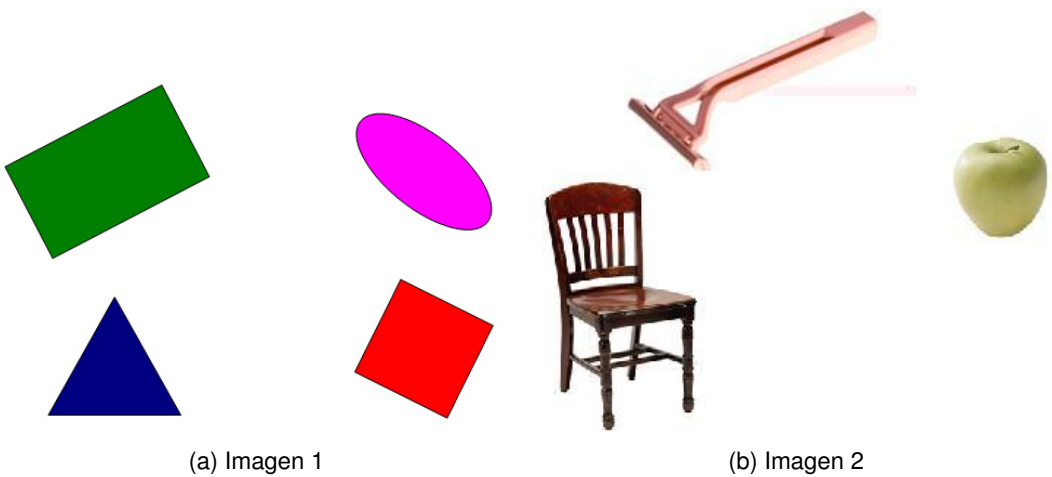


Figura 1: Imágenes de prueba utilizadas en el trabajo.

## Pregunta 1

El objetivo de este apartado es codificar cada uno de los objetos de la imagen 1a utilizando código spline. Los resultados son los siguientes:

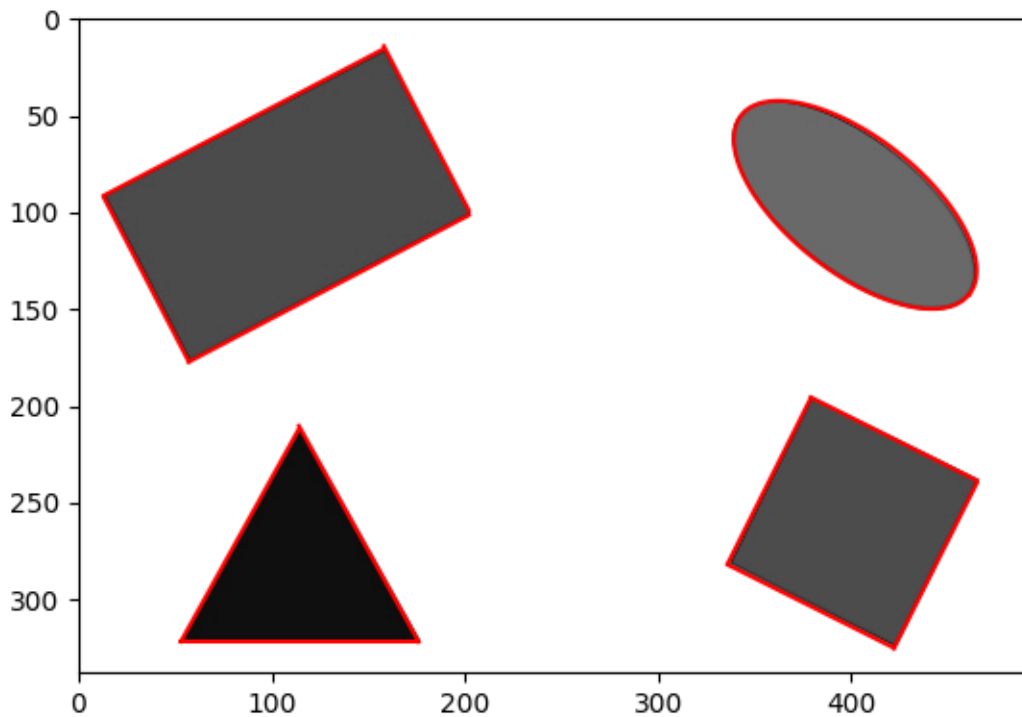


Figura 2: codificación utilizando código spline

La matriz de parámetros para el triángulo es:

$$\begin{bmatrix} 0,0 & 322,0 & 53,0 & 176,0 \\ -1,8 & 417,0 & 53,0 & 114,0 \\ 1,8 & 5,0 & 114,0 & 176,0 \end{bmatrix}$$

La matriz de parámetros para el rectángulo es:

$$\begin{bmatrix} -0,523 & 98,0 & 13,0 & 158,0 \\ -0,523 & 207,0 & 57,0 & 202,0 \\ 1,93 & 67,0 & 13,0 & 57,0 \\ 1,93 & -291,0 & 158,0 & 202,0 \end{bmatrix}$$

La matriz de parámetros para el cuadrado rotado es:

$$\begin{bmatrix} 5,000e-01 & 6,000e+00 & 3,790e+02 & 4,650e+02 \\ 5,000e-01 & 1,140e+02 & 3,360e+02 & 4,220e+02 \\ -1,990e+00 & 9,500e+02 & 3,360e+02 & 3,790e+02 \\ -1,990e+00 & 1,165e+03 & 4,220e+02 & 4,650e+02 \end{bmatrix}$$

No se muestran los parámetros de la elipse debido a que la ecuación utilizada para representarla no es polinómica.

## Pregunta 2

El objetivo de este apartado es codificar cada uno de los objetos de la imagen “objetos.jpg” utilizando código cadena y luego crear una nueva imagen en donde los objetos son generados a partir del código cadena obtenido anteriormente.

Los resultados son los siguientes:



Figura 3: código cadena implementado para codificar objetos

## Pregunta 3

El objetivo de este apartado es elegir uno de los objetos de la imagen 1b y crear un programa en Python que, utilizando el código cadena, le permita al usuario mover el objeto en 4 direcciones. Los resultados son los siguientes:

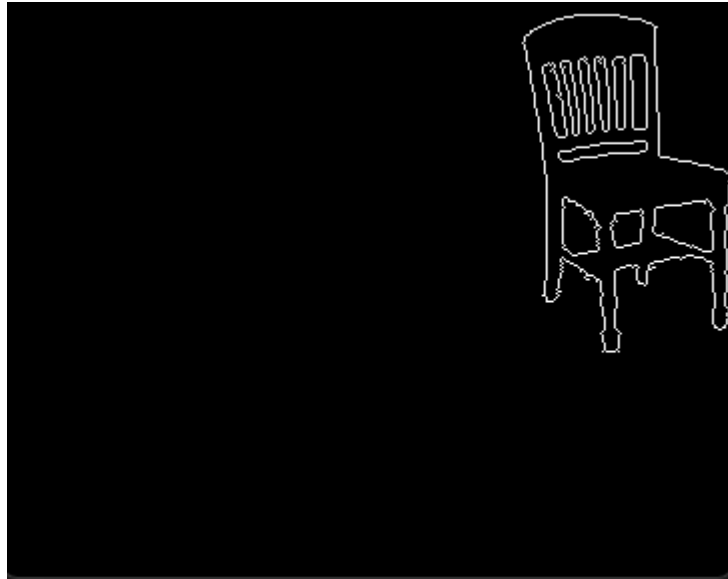


Figura 4: Código cadena utilizado para seguir los contornos de un objeto y moverlo

## Pregunta 4

El objetivo de este apartado es codificar la imagen 1b utilizando codificación Run Length, los resultados son los siguientes:



Figura 5: Código cadena utilizado para seguir los contornos de un objeto y moverlo

Los resultados son idénticos a lo utilizado con código cadena, esto se debe a que los bordes leídos son los mismos, solo que la técnica de codificación cambia.

## Pregunta 5

El objetivo de este apartado es elegir uno de los objetos de la imagen 1b y crear un programa en Python que, utilizando la código Run Length, le permita al usuario mover el objeto en 4 direcciones. Los resultados son los siguientes:

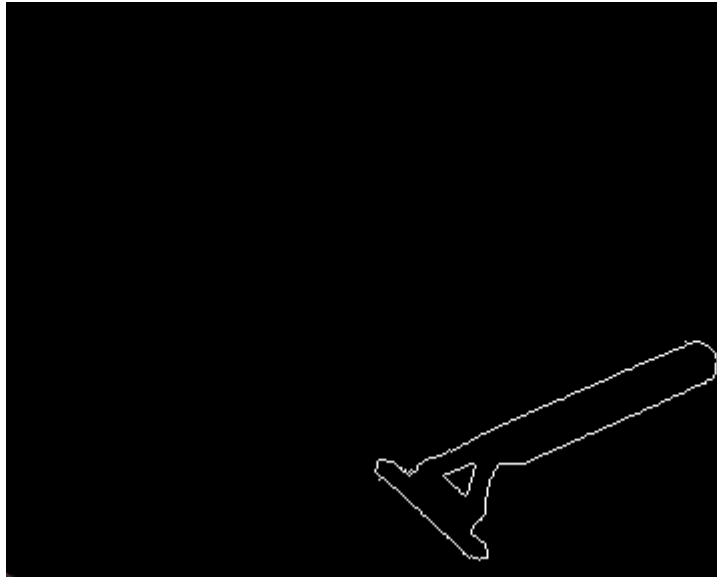


Figura 6: Código Run Length utilizado para seguir los contornos de un objeto y moverlo

## Pregunta 6

A continuación se muestra un cuadro comparativo entre las 3 codificaciones.

	<b>Spline</b>	<b>Run-Length (RLE)</b>	<b>Codificación de Cadenas (Chain Coding)</b>
<b>Descripción</b>	Representación suave y continua de los datos.	Almacena el valor de píxel y su longitud de ejecución.	Utiliza una secuencia de direcciones de vecinos.
<b>Aplicación</b>	Curvas y contornos suaves.	Imágenes con áreas repetitivas o regiones uniformes.	Formas con bordes bien definidos y estructura jerárquica.
<b>Precisión</b>	Alta fidelidad en la reconstrucción de la forma original.	Pérdida de precisión debido a la compresión de datos.	Pérdida de precisión debido a la simplificación del contorno.
<b>Eficiencia</b>	Puede requerir más espacio de almacenamiento debido a la representación continua.	Eficiente en términos de almacenamiento debido a la compresión de datos.	Eficiente en términos de almacenamiento debido a la representación de direcciones de vecinos.
<b>Compresión</b>	No está diseñada específicamente para compresión de datos.	Ofrece una compresión efectiva para imágenes con áreas repetitivas.	Ofrece cierta compresión al representar el contorno con direcciones de vecinos.
<b>Complejidad</b>	Requiere cálculos y algoritmos complejos para la representación suave.	Algoritmo simple y fácil de implementar.	Algoritmo relativamente simple y fácil de implementar.
<b>Reconstrucción</b>	Permite una reconstrucción precisa de la forma original.	Reconstrucción precisa del contorno original, aunque con menor detalle.	Reconstrucción aproximada del contorno original debido a la simplificación.
<b>Aplicaciones</b>	Gráficos, animación, modelado 3D.	Compresión de imágenes, transmisión de datos eficiente.	Reconocimiento de patrones, seguimiento de contornos.

Cuadro 1: Comparación de las codificaciones: Spline, Run-Length (RLE) y Codificación de Cadenas (Chain Coding)

Dependiendo de la imagen a procesar uno u otro método puede ser mas adecuado, por ejemplo, para 1a el código Spline puede interpretar con mayor facilidad los bordes. Sin embargo, su implementación es engorrosa y compleja. El código Run length y el código Cadena son adecuados para 1a en este caso en específico, aunque si una imagen posee multiples objetos sería eficiente el uso de Run Length.