The background is a complex, abstract network of thin red lines connecting various points. Scattered throughout the scene are numerous 3D cubes of different sizes and orientations. Some cubes are dark grey or black, while others are white or light grey. The overall lighting is soft, with a slight gradient from top to bottom, and some blurred light spots in the lower right corner.

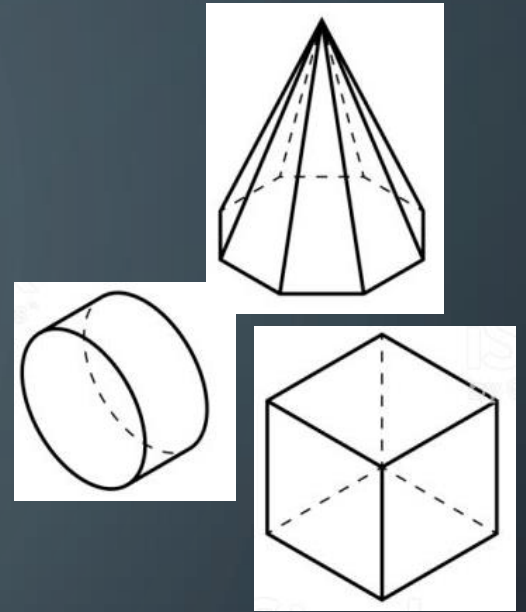
# PROYECTO 1 COUTOUR FOLLOWING

DENNIS CHAVARRÍA SOTO B82097

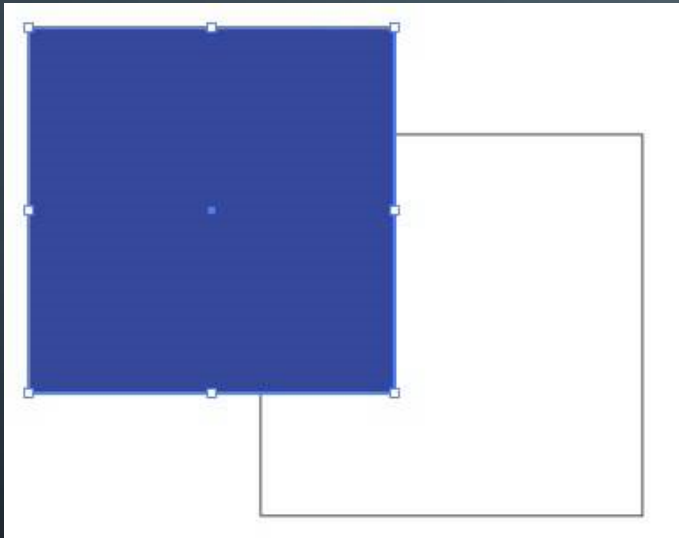
JESÚS ZÚÑIGA MÉNDEZ B59084

# ¿QUÉ ES CONTOUR FOLLOWING?

- Algoritmos que siguen el borde de una figura
- Determinan los contornos a partir de variaciones de color
- Considera el contorno como el borde, o límite, de una figura en una matriz de píxeles



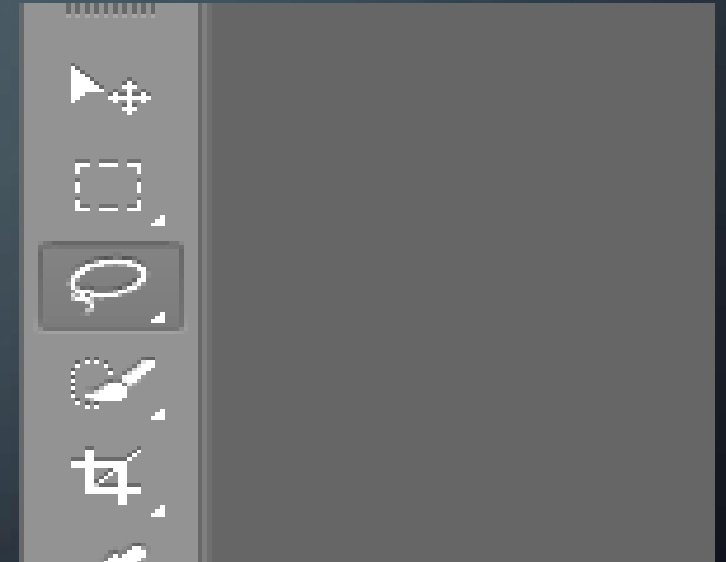
# IMPORTANCIA Y APLICACIONES



Coloreado  
limitado por  
contornos



Reconocimiento de patrones  
(Robots seguidores y láseres  
automatizados)



Herramientas  
de edición de  
fotos

# FUNCIONAMIENTO DEL ALGORITMO

- Se utiliza el lenguaje C++
- Implementación de lectura y escritura de imágenes con la biblioteca OpenCV

# FUNCIONAMIENTO DEL ALGORITMO

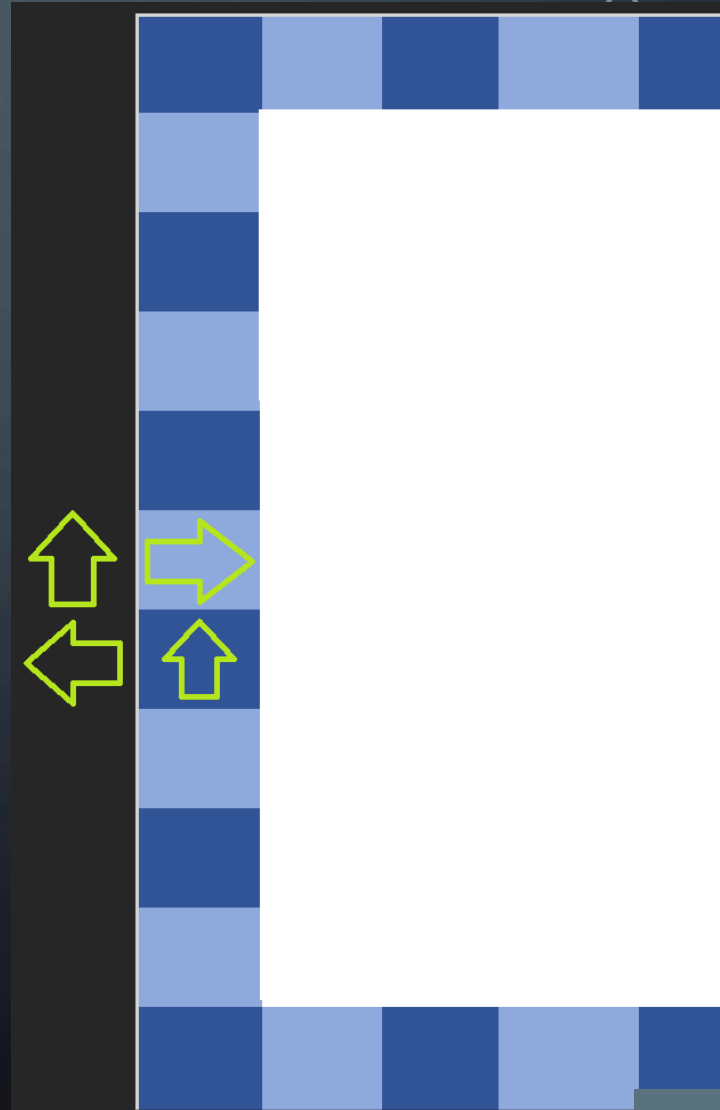
- Cuatro métodos fundamentales, cuatro algoritmos de seguimiento
  - Algoritmo de trazado en cuadros
  - Algoritmo de Pavlidi
  - Algoritmo de Vecindario Moore
  - Algoritmo Radial

# FUNCIONAMIENTO DEL ALGORITMO

## Algoritmo de trazado en cuadros

Con base en las direcciones.

- Compara si se encuentra en un pixel de borde o en el fondo
- Revisa su dirección actual (Con la cuál entró al pixel)
- Cambia la dirección a la que apunta
- Entra al pixel, sea de borde o de fondo, según corresponda.
- Repite hasta llegar al pixel inicial

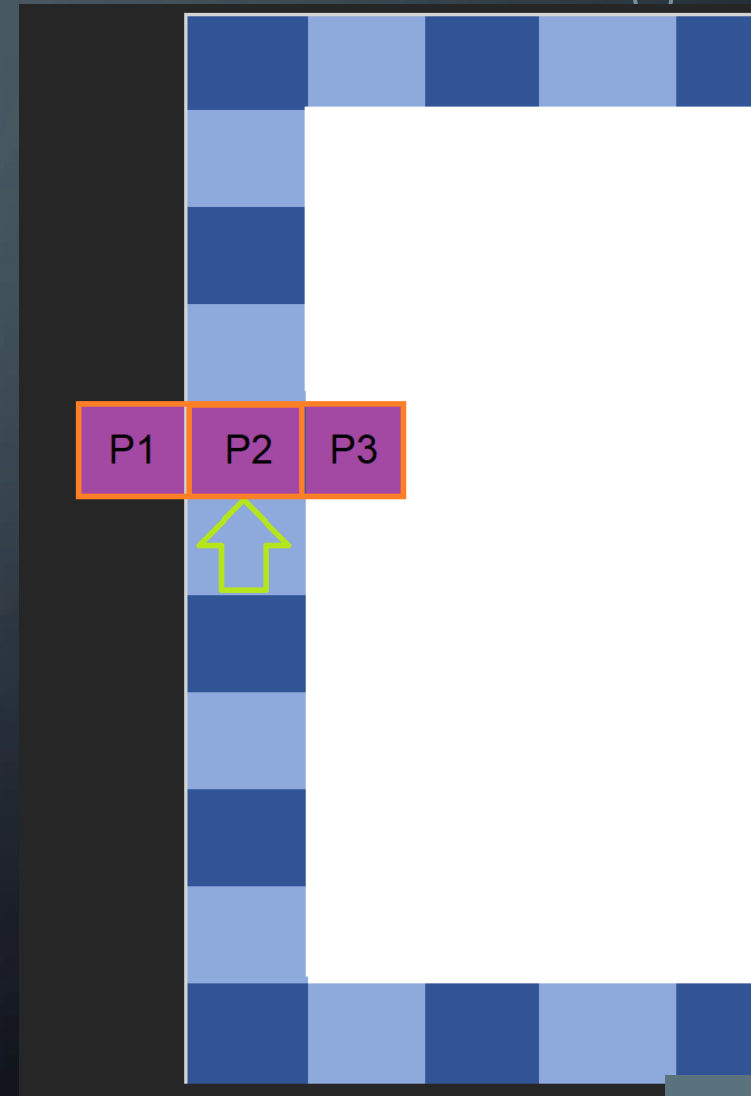


# FUNCIONAMIENTO DEL ALGORITMO

## Algoritmo de Pavlidi

Con base en las direcciones.

- Compara si se encuentra en un pixel de borde o en el fondo
- Establece los pixeles P1,P2,P3
- Determina el pixel que compone el borde y se traslada a él.
- Si es P1, la dirección es izquierda con respecto al frente actual.
- Si es P2 o P3, la dirección se mantiene.

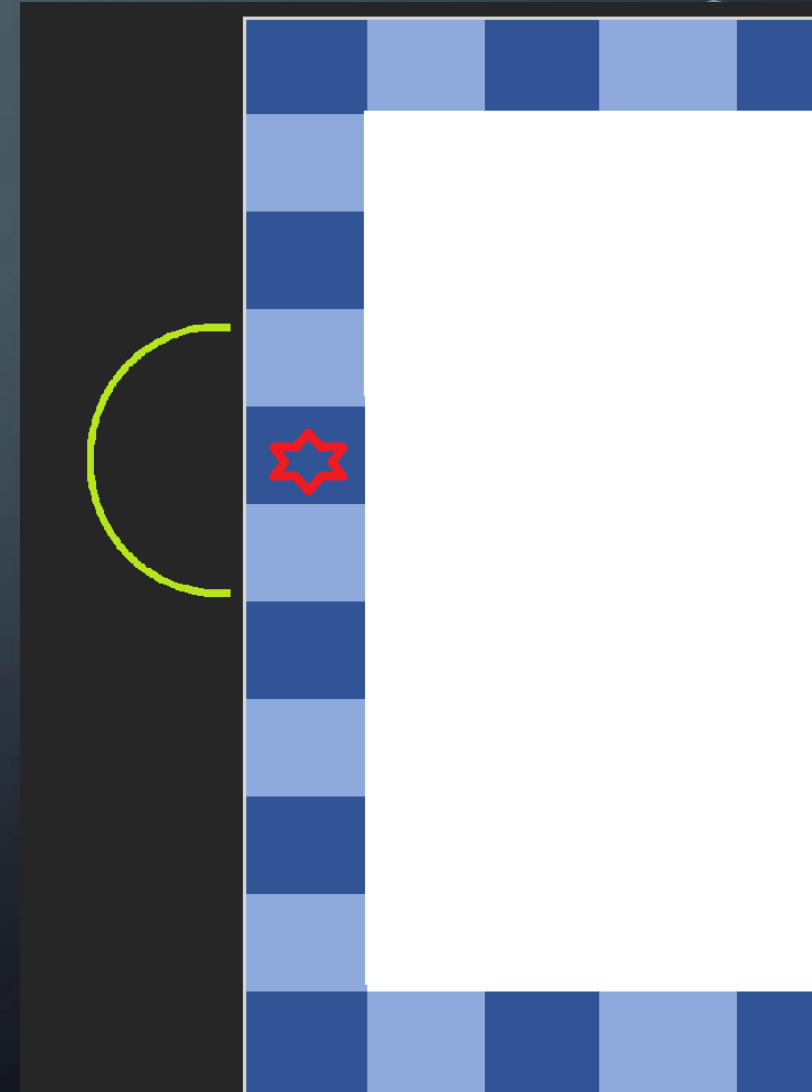




# FUNCIONAMIENTO DEL ALGORITMO

## Algoritmo de Moore

- Establece un punto inicial en la matriz de pixeles
- Estudia un vecindario de 8 pixeles
- Se traslada a donde se encuentra el pixel de borde y vuelve a muestrear

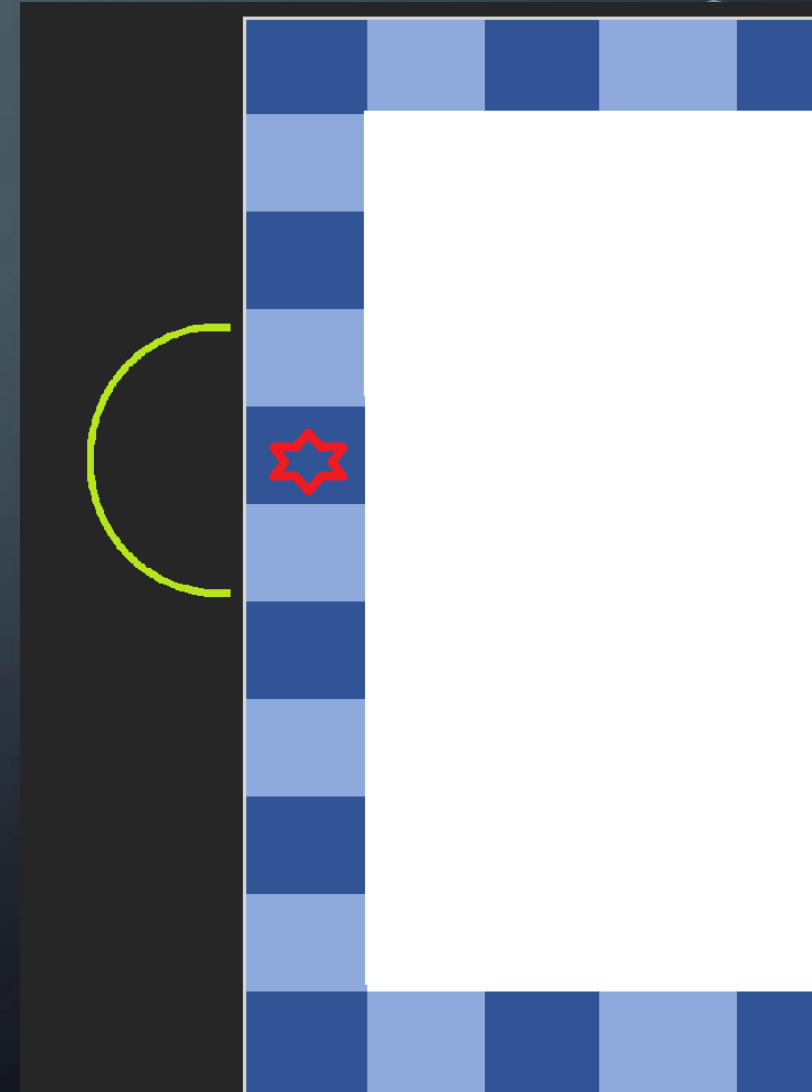




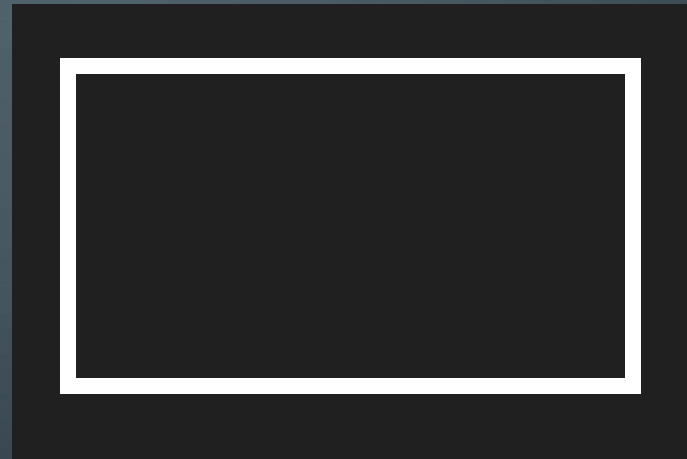
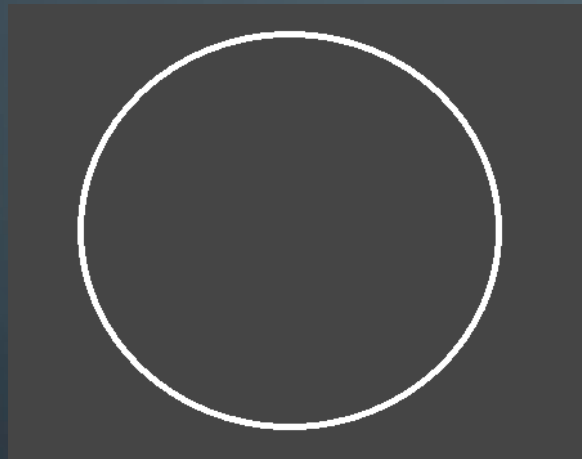
# FUNCIONAMIENTO DEL ALGORITMO

## Algoritmo Radial

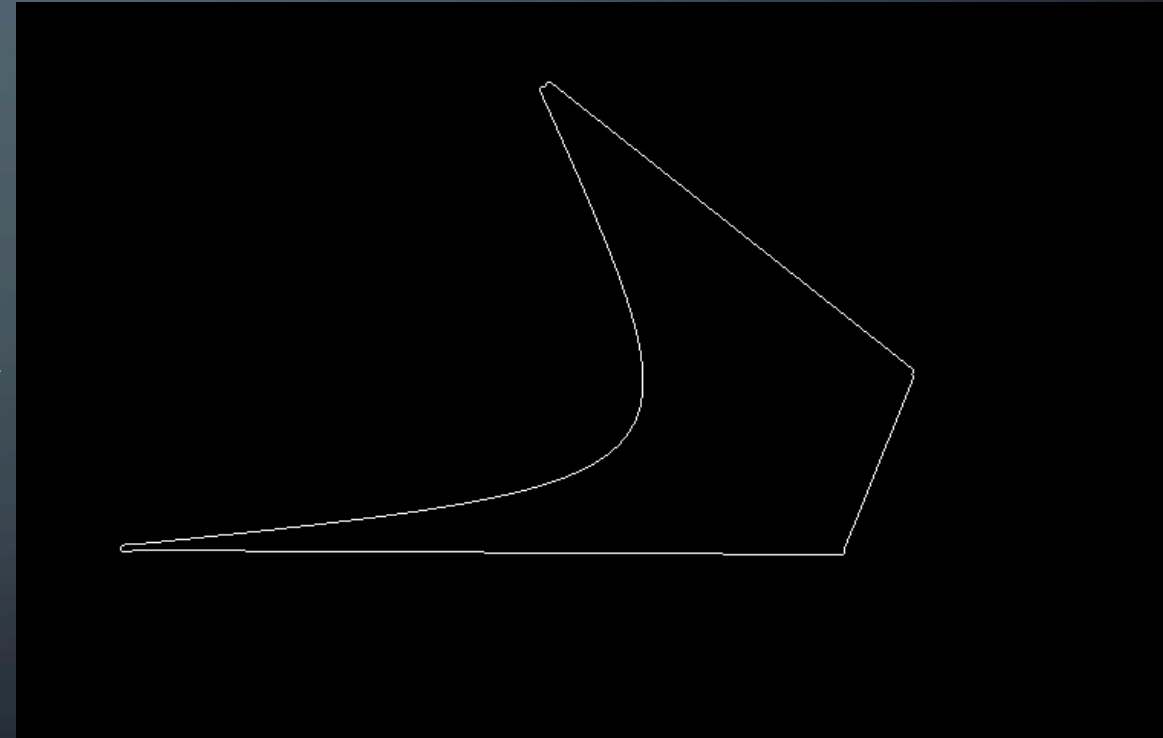
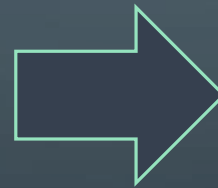
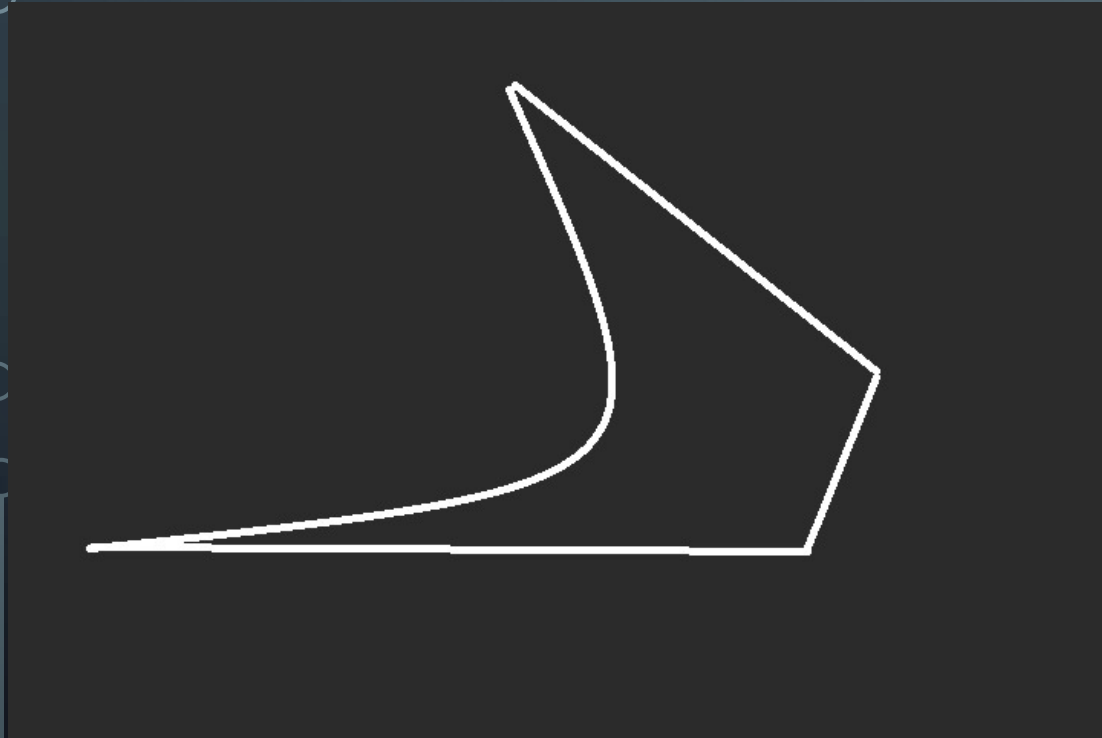
- Establece un punto inicial en la matriz de pixeles
- Estudia un vecindario de 8 pixeles
- Establece un “lazo” con el pixel anterior, y desde él hace el sondeo
- Se traslada a donde se encuentra el pixel de borde y vuelve a muestrear



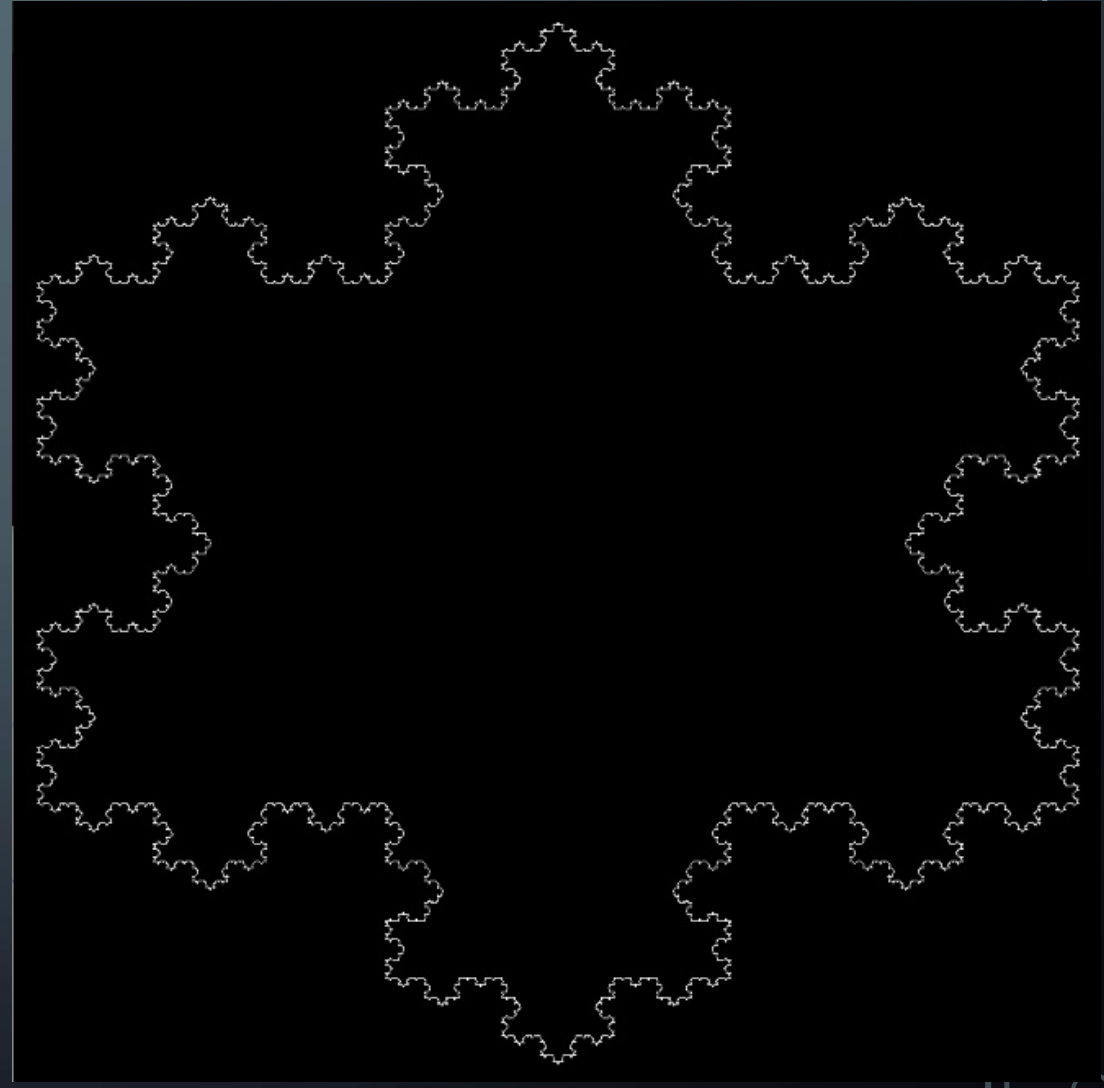
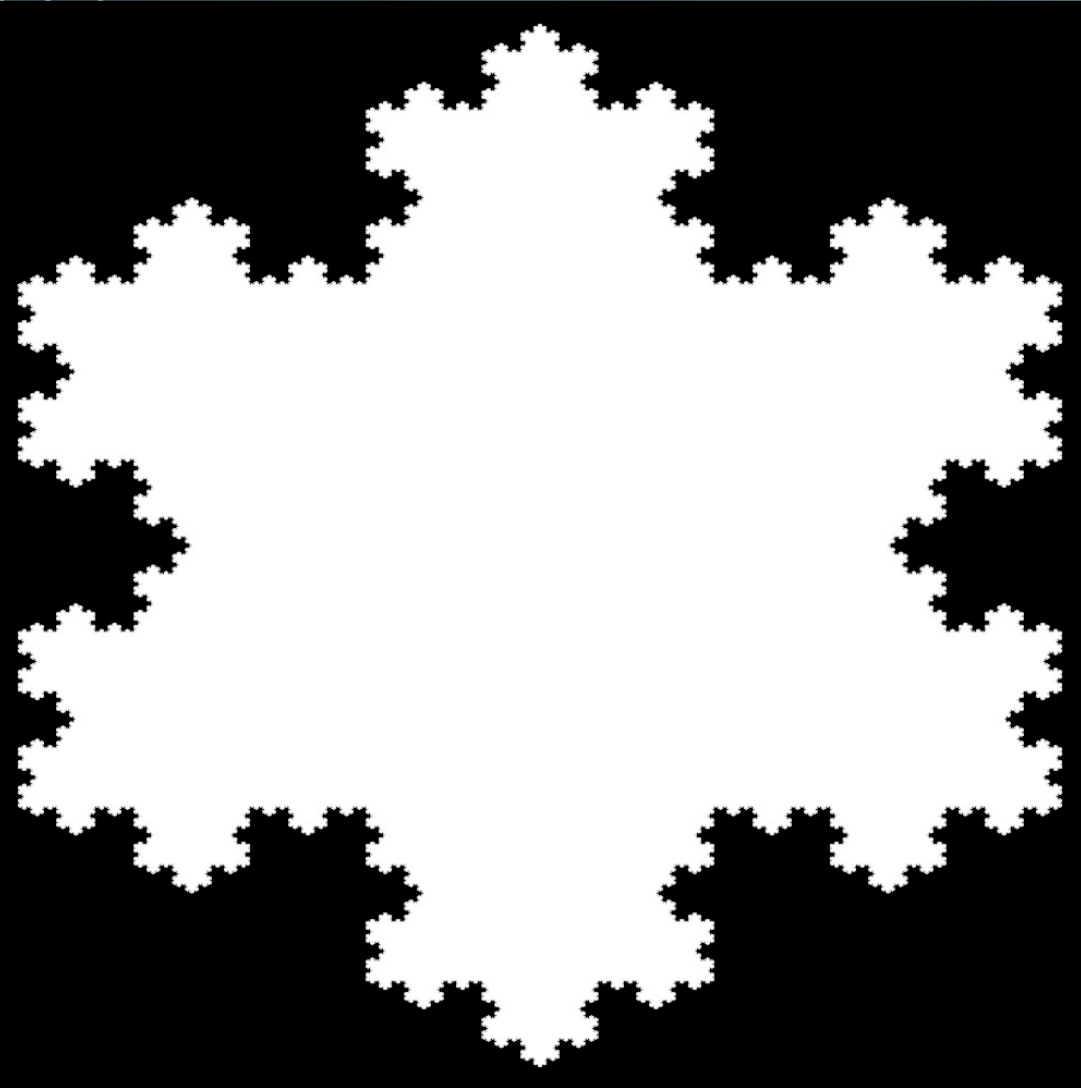
# EXPERIMENTOS REALIZADOS



# RESULTADOS OBTENIDOS



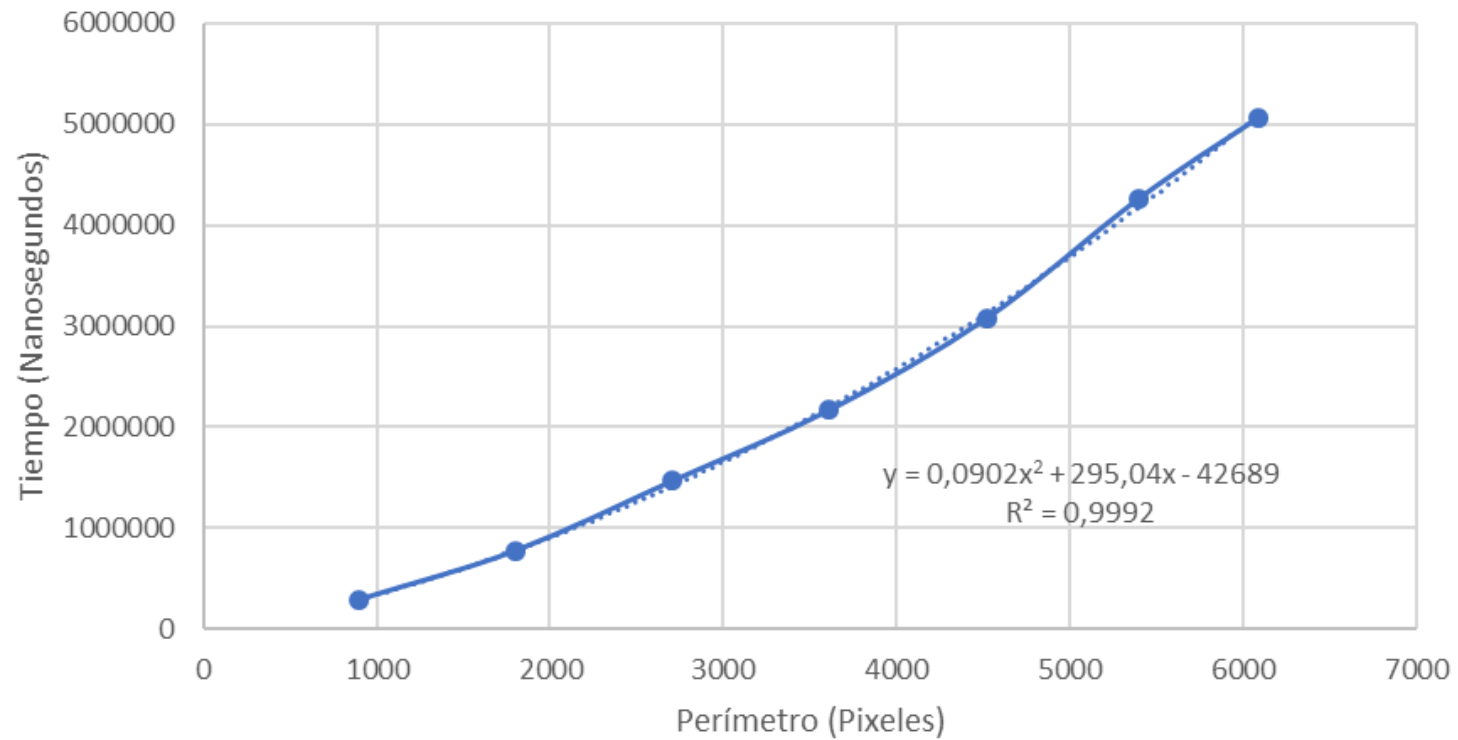
# RESULTADOS OBTENIDOS



# ANÁLISIS DE COMPLEJIDAD

## Algoritmo de trazado en cuadros

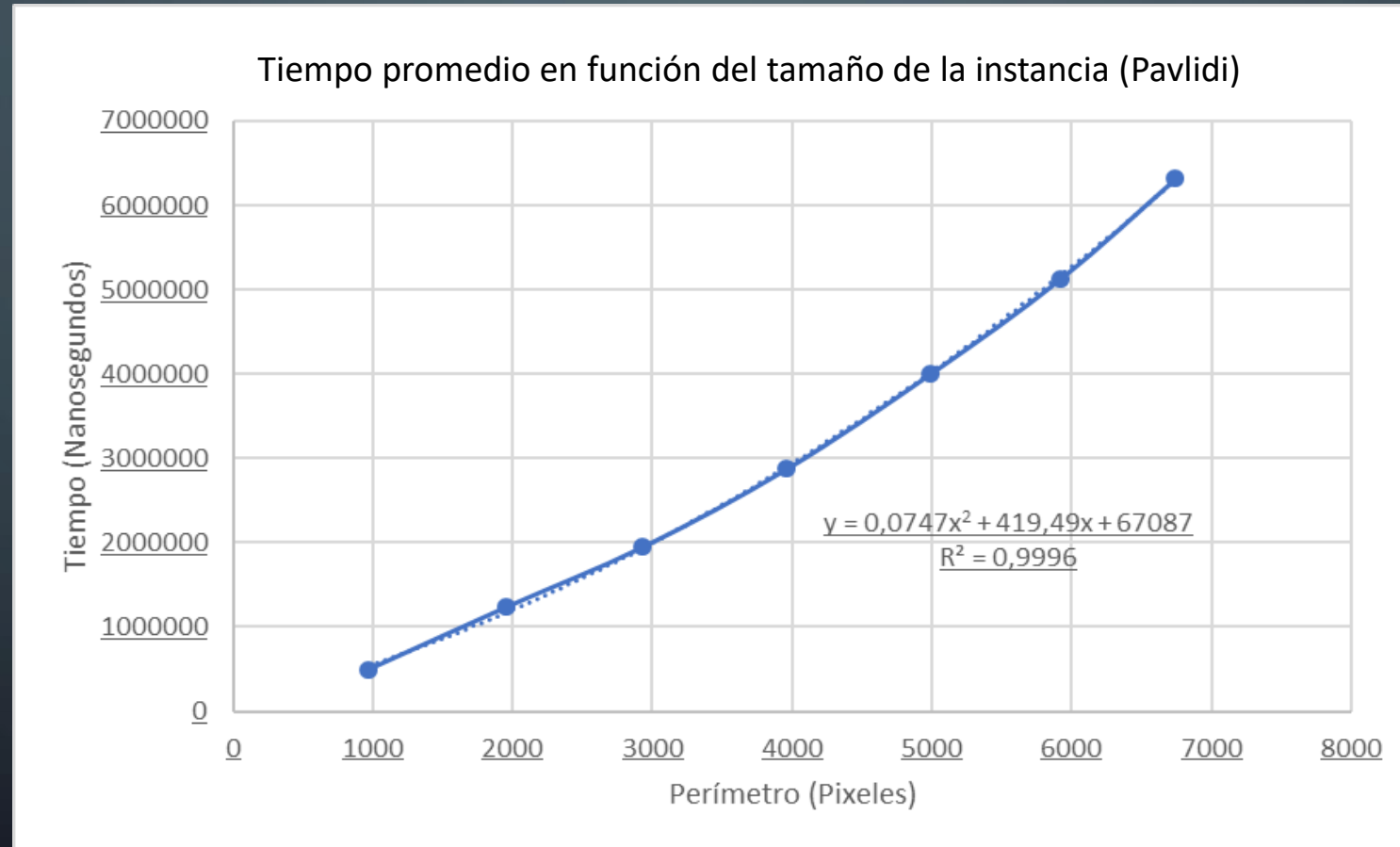
Tiempo promedio en función del tamaño de la instancia (Square)



Tiempo promedio en función del tamaño de la instancia (Square)

# ANÁLISIS DE COMPLEJIDAD

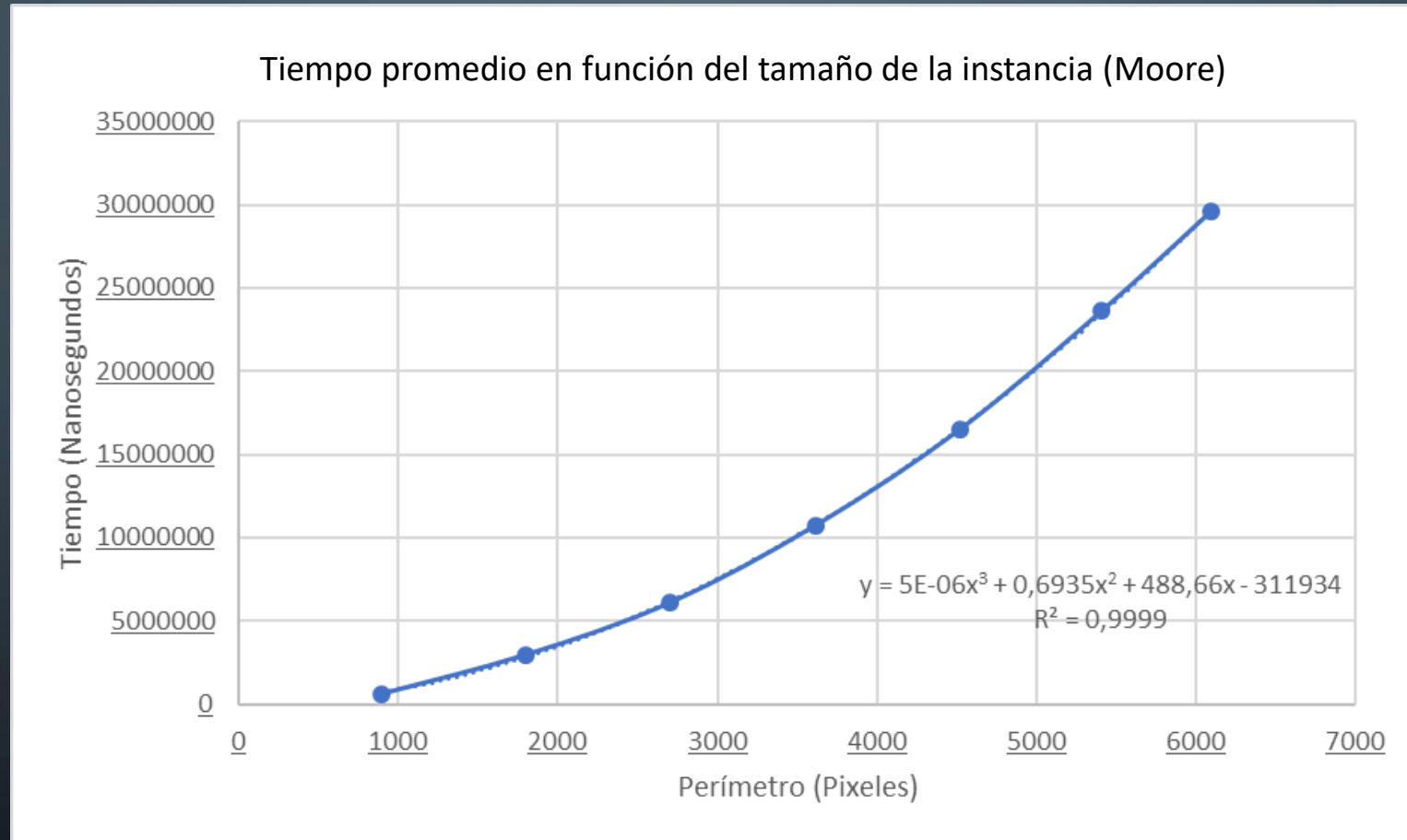
## Algoritmo de Pavlidi



Tiempo promedio en función del tamaño de la instancia (Pavlidi)

# ANÁLISIS DE COMPLEJIDAD

## Algoritmo de Moore

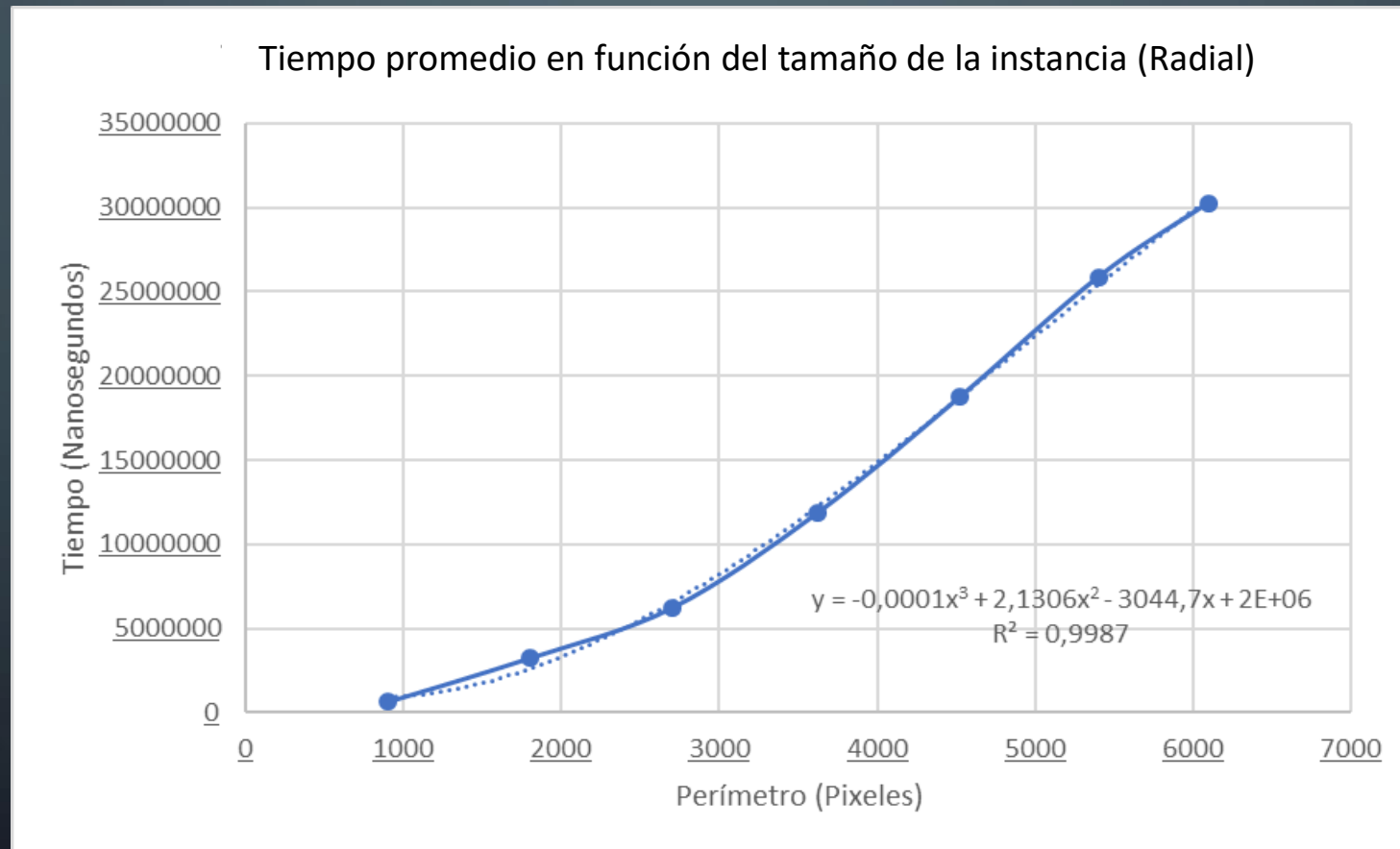


Tiempo promedio en función del tamaño de la instancia (Moore)



# ANÁLISIS DE COMPLEJIDAD

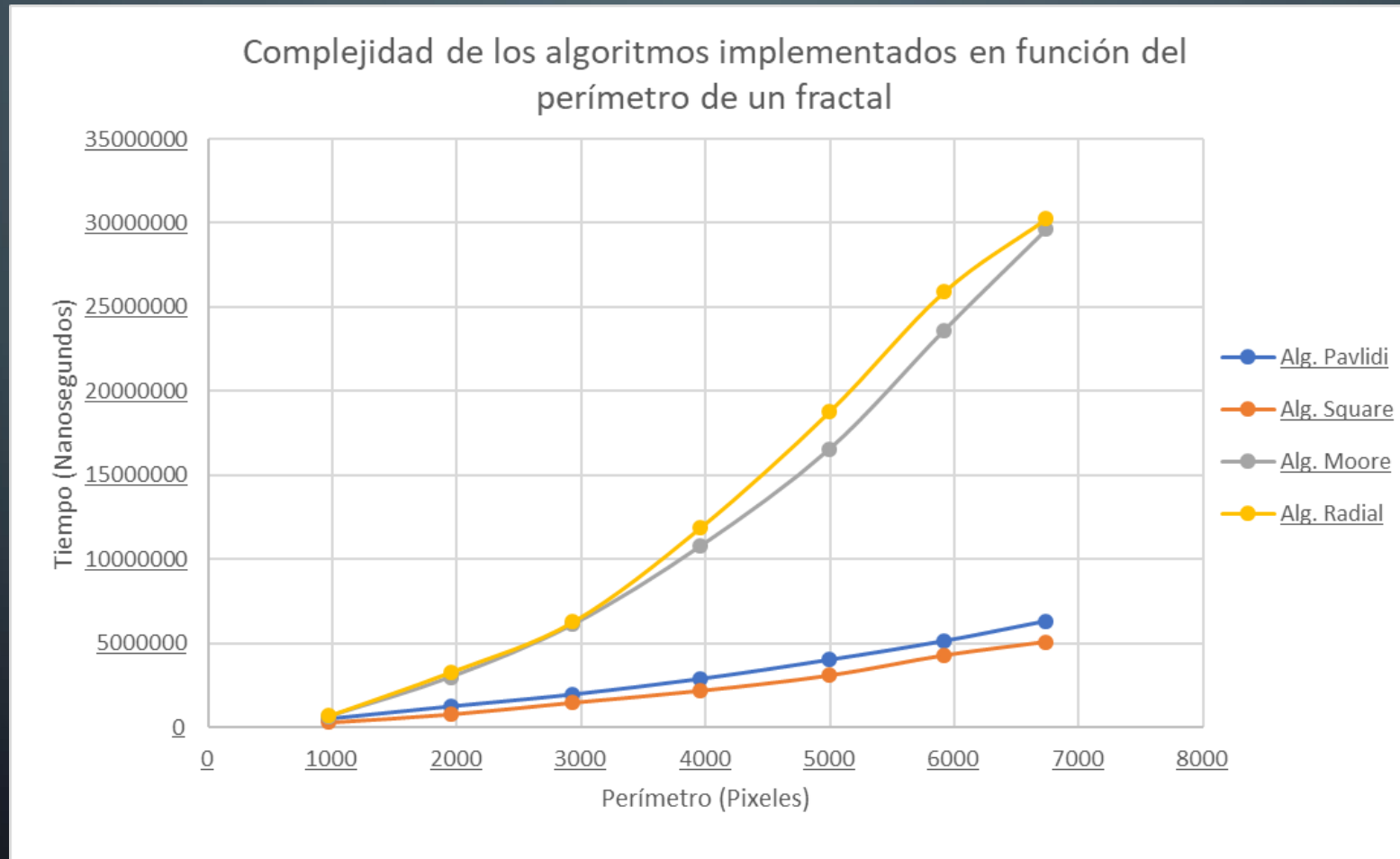
## Algoritmo Radial



Tiempo promedio en función del tamaño de la instancia (Radial)

# ANÁLISIS DE COMPLEJIDAD

## Comparación entre los cuatro algoritmos



Comparación de tiempo, de los cuatro algoritmos, en función de la complejidad

# CONCLUSIONES

- Se implementó los algoritmos de seguimiento de bordes.
- Se logró detectar el contorno de figuras, como fractales, de forma satisfactoria.
- El algoritmo de trazado por cuadros resulta ser más efectivo ante complejidades con instancias superiores a 2000 pixeles.

# CONCLUSIONES

- El algoritmo radial resultó ser el menos eficiente conforme se aumentó el tamaño de la instancia.
- Fue posible verificar el comportamiento del tiempo en función de la complejidad de cada algoritmo.

# REFERENCIAS BIBLIOGRÁFICAS

- Ghuneim, A.G. (2000) What is Contour Tracing? Recuperado de: <http://www.imageprocessingplace.com/downloads-V3/root-downloads/tutorials/contour-tracing-Abeer-George-Ghuneim/intro.html>
- Jonghoon, S.; Seungho, C.; et al. (2016) Fast Contour-Tracing Algorithm Based on a Pixel-Following Method for Image Sensors. Recuperado de: <https://www.mdpi.com/1424-8220/16/3/353/pdf>
- Universidad Don Bosco (s.f) Tema: Reglas para el cálculo de complejidad en los métodos de Ordenamiento. Recuperado de [http://www.udb.edu.sv/udb\\_files/recursos\\_guias/informatica-ingenieria/programacion-iv/2019/ii/guia-6.pdf](http://www.udb.edu.sv/udb_files/recursos_guias/informatica-ingenieria/programacion-iv/2019/ii/guia-6.pdf)
- OpenCV (2019) About. Recuperado de <https://opencv.org/about/>