

# VC

# GDV == Visual Computing

## Objetos gráficos y su programación

Universidad de Darmstadt

Björn Frömmer

Prof. Dr. Elke

Hergenröther Prof. Dr.

Benjamin Meyer

1.

## Notas importantes

---

Asista a la conferencia

¿Por qué?

- El guión y las diapositivas son sólo un resumen
- La pizarra y los ejercicios de clase son ayudas importantes para el examen
- Las preguntas, también las de otros estudiantes, le ayudarán a comprender
- Contenido interesante y motivador :-)

Prepara o repasa el material

¿Por qué?

- El examen no es "memorístico"
- Es realmente necesario dominar el material para poder aplicarlo correctamente en el examen.

1.

## Contenido de la conferencia VC

---

### 1. Gráficos por ordenador (OpenGL y C++)

- Programación gráfica (OpenGL v3.3+)
- Sistemas de coordenadas
- Técnicas de visualización
- Transformaciones geométricas

### 2. Tratamiento de imágenes (OpenCV en Python)

- Edición de imágenes (dar más brillo a la imagen, etc.)
- Tratamiento de imágenes (por ejemplo, extracción de información de una imagen)
- Método de compresión (por ejemplo, JPEG)
- Modelos en color

# ¿De qué se trata?

## ... siempre sobre píxeles

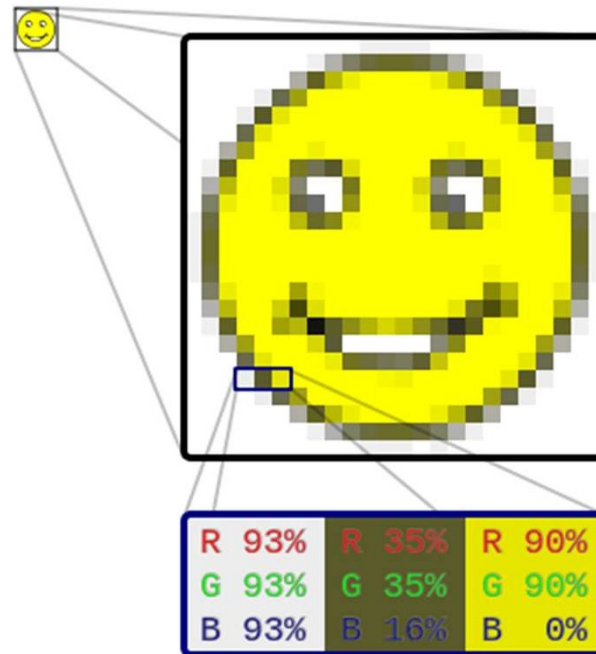
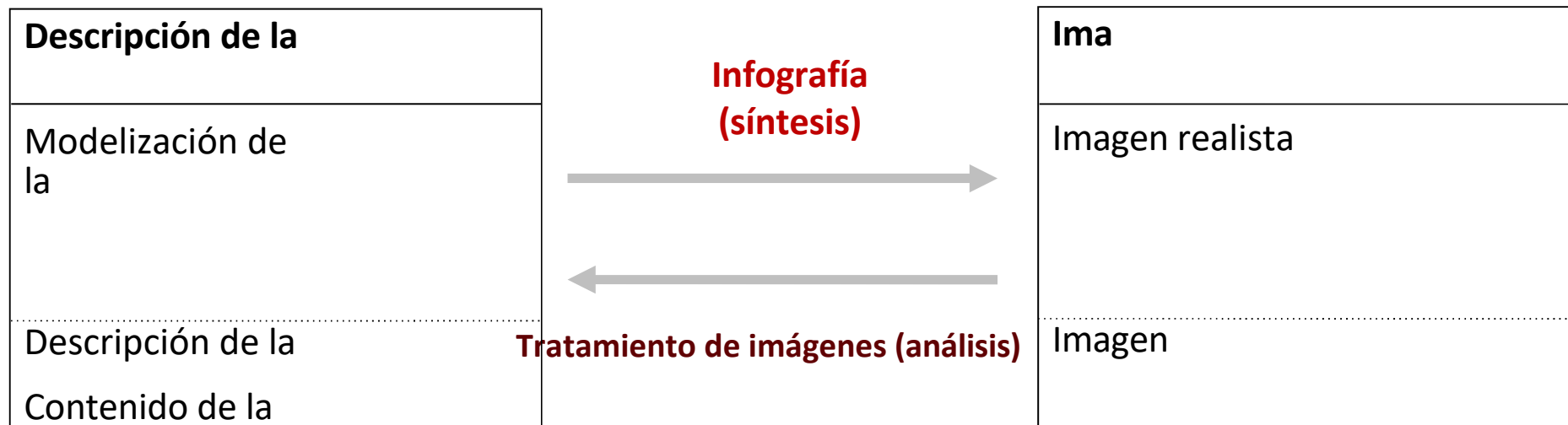


Imagen de: <https://developer.tizen.org/forums/native-application-development/what-rasterisation-opengl-graphic-pipeline>

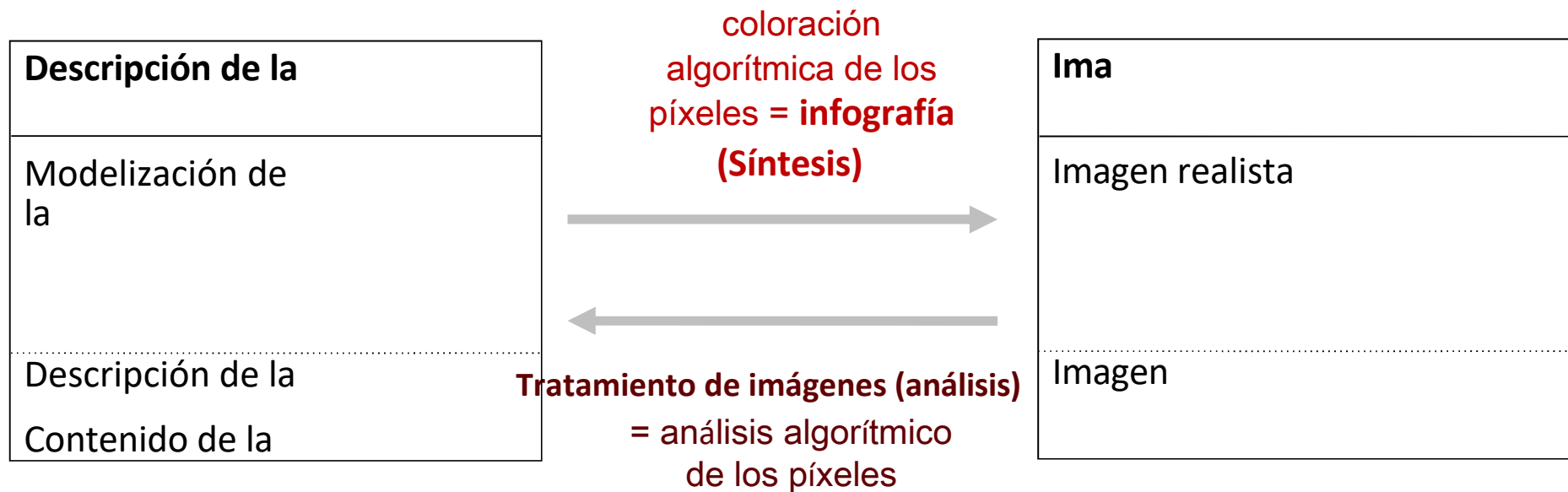
## 1. introducción

# División entre infografía y tratamiento de imágenes



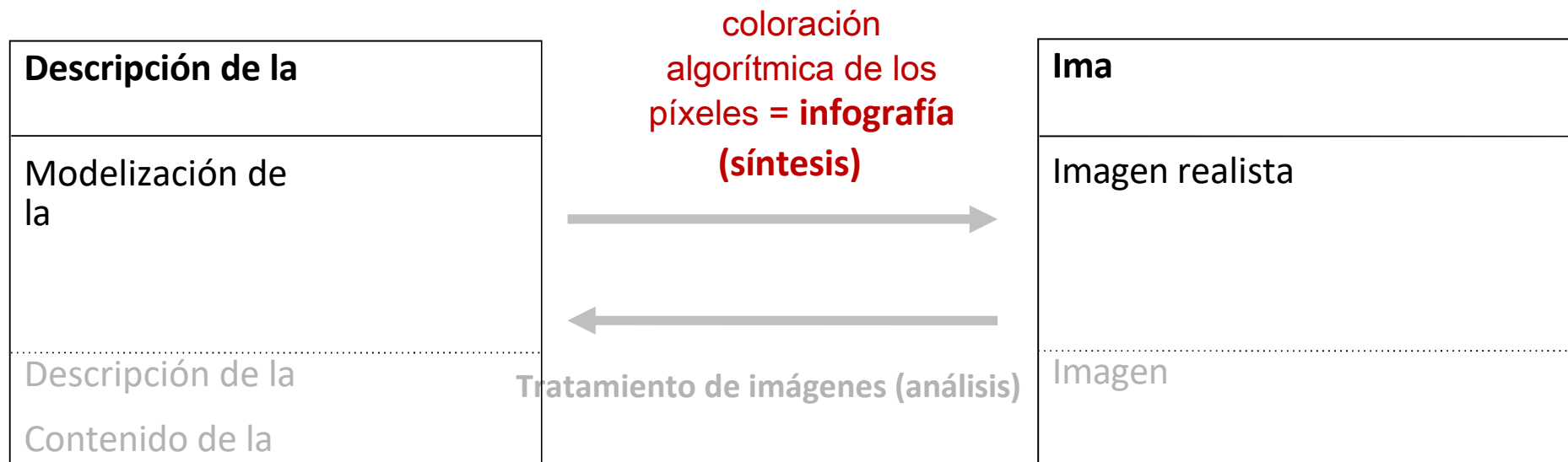
## 1. introducción

# División entre infografía y tratamiento de imágenes



## 1. introducción

# División entre infografía y tratamiento de imágenes

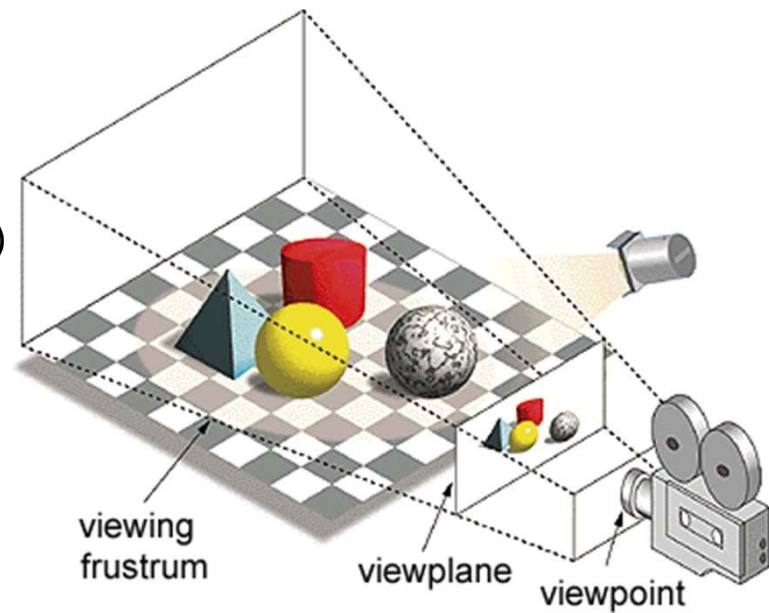


## 1. introducción

# Infografía

## Modelización del contenido de la imagen

- Geometrías tridimensionales
- Fuentes de luz
- Cámara virtual (incl. punto de vista para perspectiva)
- Área visible (frustum de visión)
- Posición del plano de la imagen (viewplane)



Objetivo: colorear los píxeles del plano de visión según la escena grabada.



## 1. introducción

# Infografía

Las formas de los objetos de la escena se representan mediante cuadrículas triangulares tridimensionales. se acercó.

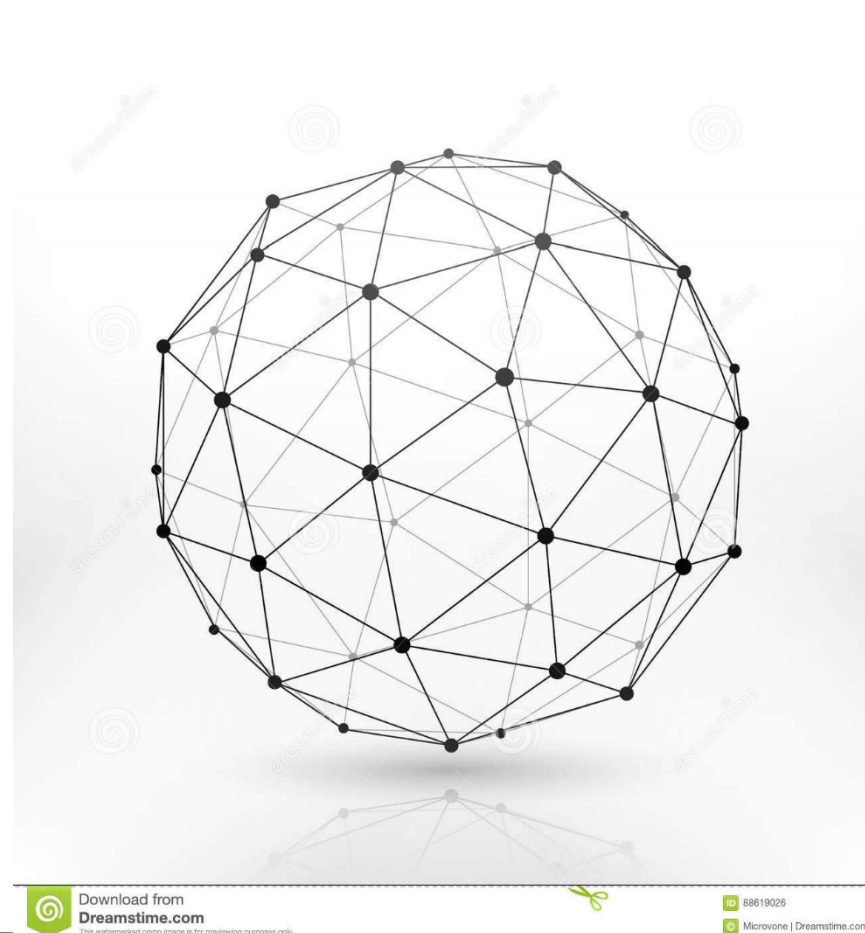


Imagen de: [https://de.dreamstime.com/stock-abbildung-wireframe-sphere-area-connection-network-technology-connection-vector-concept-image88619026#\\_](https://de.dreamstime.com/stock-abbildung-wireframe-sphere-area-connection-network-technology-connection-vector-concept-image88619026#_)

## Infografía

- Gráficos de trama
- Raytracing



Informática visual  
Semestre de verano de  
2023

## 1. introducción

# Infografía

Generar un **gráfico rasterizado** es una forma de pasar de la cuadrícula triangular a la cuadrícula.

1) vértices del triángulo en píxeles del plano de la imagen

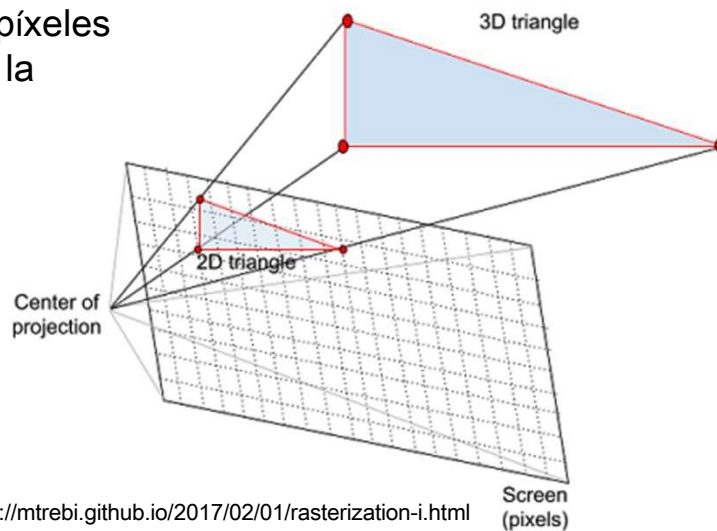


Imagen de: <https://mtrebi.github.io/2017/02/01/rasterization-i.html>

2) Utilizando los píxeles de los vértices del triángulo, encuentre los píxeles restantes del triángulo en el plano de la imagen.

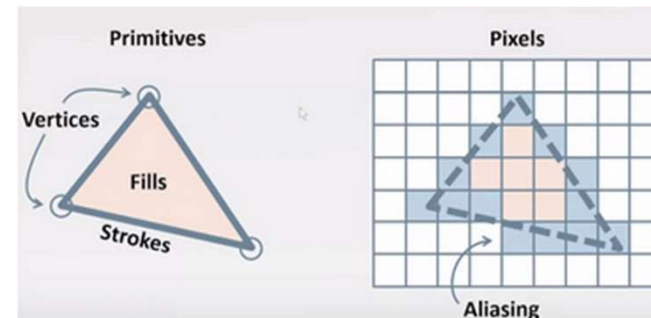


Imagen de: <https://developer.tizen.org/forums/native-application-development/what-rasterisation-opengl-graphic->

## 1. introducción

# Infografía

Generar una imagen con **trazado de rayos** es otra forma de pasar de la rejilla triangular a la representación renderizada.

En lugar de generar el renderizado de las geometrías a partir de los puntos de esquina, se intenta renderizar la escena con "rayos de visión" que emanan de la cámara virtual.

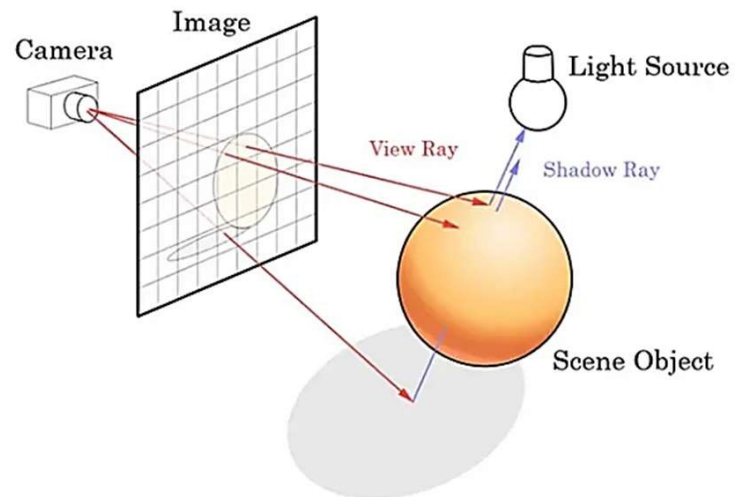


Imagen de: <https://medium.com/@junyingw/future-of-gaming-rasterisation-vs-ray-tracing-vs-path-tracing-32b334510f1f>

1.

## Ejemplo para gráficos por ordenador: Trazado de rayos Variante: Trazado de trayectorias



### Ventaja:

- Reflexiones
- Iluminación indirecta
- ...

### Desventaja:

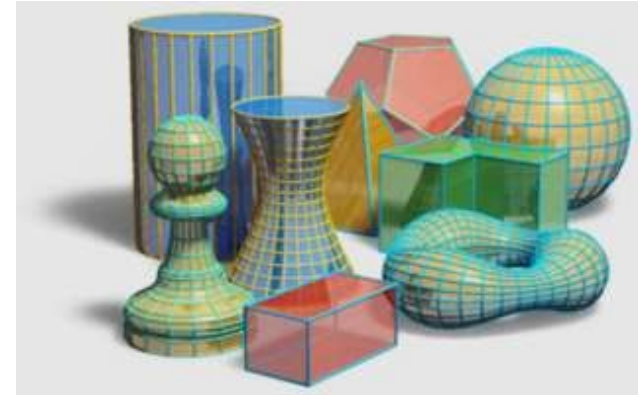
- Ligero desenfoque, se justifica en el procedimiento

<https://youtu.be/aoW4HRi8VKg>

1.

# Infografía

---



- Algoritmos de generación de geometrías
- Algoritmos de "interacción" con geometrías
- Algoritmos para la visualización de geometrías (incl. cálculo de iluminación, visualización estereoscópica, ... )
- Tecnología de sombreado
- ...



Imágenes tomadas de: [http://www.pytha.de/produkt/modeler\\_1.de.php](http://www.pytha.de/produkt/modeler_1.de.php)

1.

## División entre infografía y tratamiento de imágenes












1.

## Tratamiento de imágenes

La posibilidad más sencilla es analizar las imágenes mediante "matrices de convolución":



Imágenes de: <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

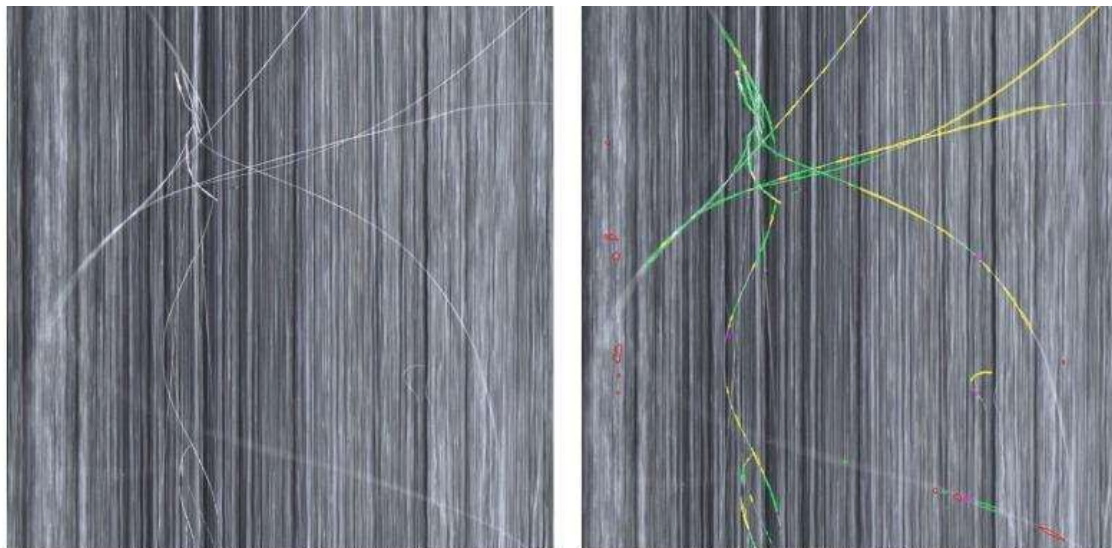


## 1. introducción

# Tratamiento de imágenes

Las matrices de convolución pueden utilizarse para analizar imágenes:

Ejemplo izquierda Detección de fibras de carbono, ejemplo derecha Recuento de cepas bacterianas



© Fraunhofer IGC  
Aufnahme eines Faserbündels auf Carbonfaserteppich  
Links: Originalaufnahme  
Rechts: Segmentierter Defekt



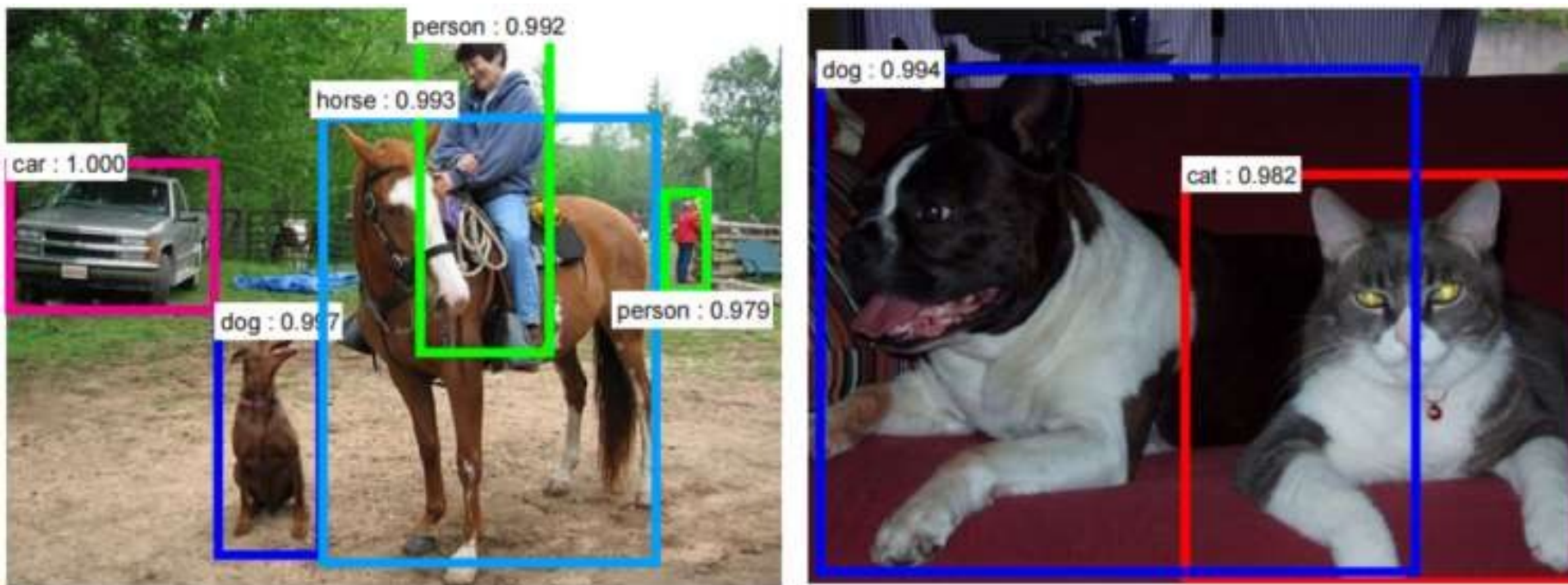
Imagen de: [Ilchev Trendafil - I+D - Beissbarth GmbH | LinkedIn](#)

Imagen de: [Reconocimiento de imágenes - GedonSoft GmbH](#)

## 1. introducción

# Tratamiento de imágenes

Con una red de matrices convolucionales, las redes neuronales convolucionales, se pueden reconocer objetos complejos en la imagen o el vídeo.



Imágenes de: <https://uijwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

1.

## Red neuronal convolucional (CNN o ConvNet)

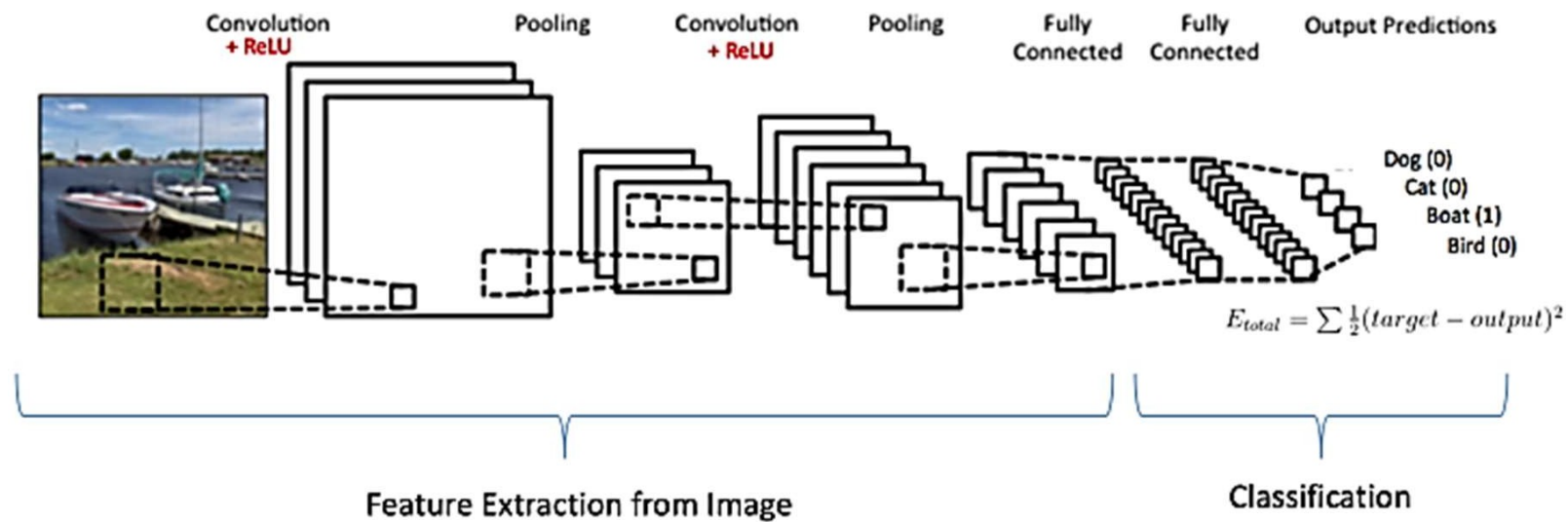
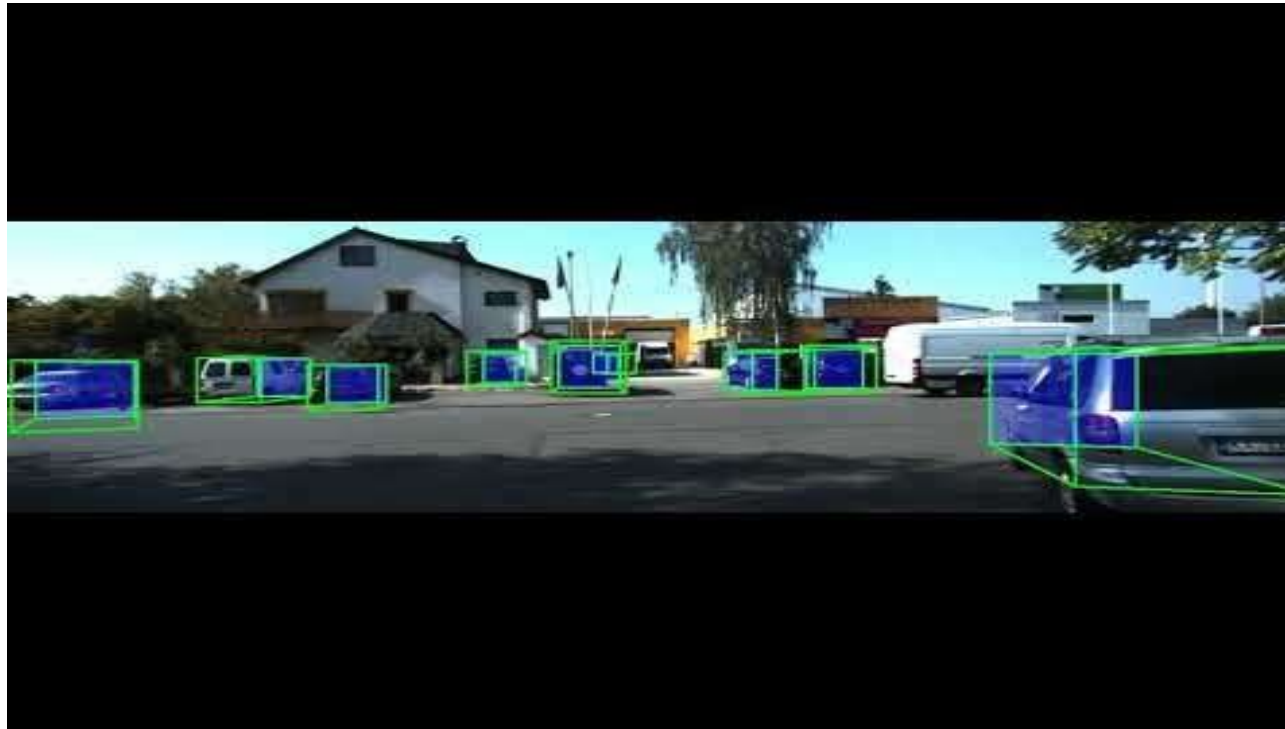


Imagen de: <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

1.

## Tratamiento de imágenes

Esto también funciona (parcialmente) para objetos 3D ☐



### YOLO3D: Detección monocular de objetos en 3D

1.

## Visualización del funcionamiento de una CNN

A través de las capas de convolución, las imágenes de entrada se transforman gradualmente en una descripción abstracta. Las imágenes similares tienen descripciones abstractas similares.

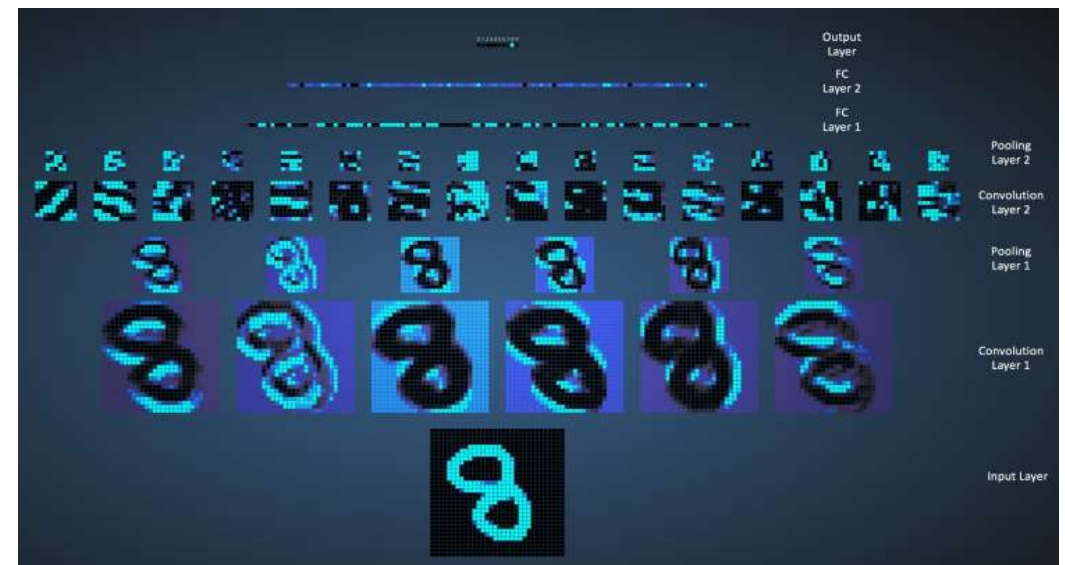
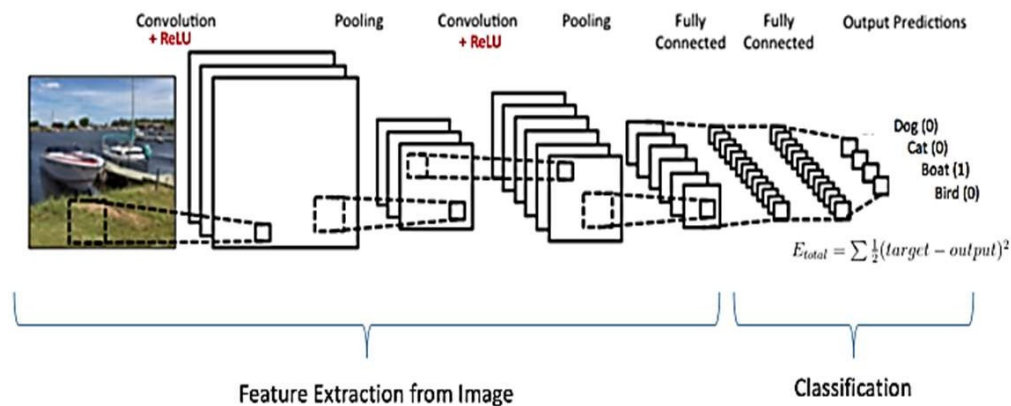


Imagen de: <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

Imagen de: <http://cs231n.github.io/convolutional-networks/>

pruébelo usted mismo: [https://adamharley.com/nn\\_vis/cnn/3d.html](https://adamharley.com/nn_vis/cnn/3d.html)



1.

## Ejemplo de infografía: Trazado de rayos



Destacados 1-4

min.:

- Reflejo automático (seguimiento de ruta)
- Generación intuitiva del terreno
- Voz a voz

<https://www.youtube.com/watch?v=aoW4HRi8VKg>

---

# CAPÍTULO 2

## Objetos gráficos

## 2. objetos gráficos

# Creación de objetos gráficos



**Todos los objetos gráficos están formados, en última instancia, por puntos que se conectan con aristas para formar triángulos.**

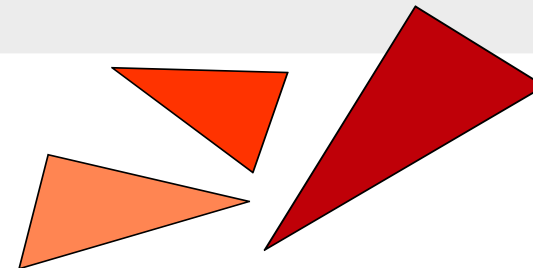
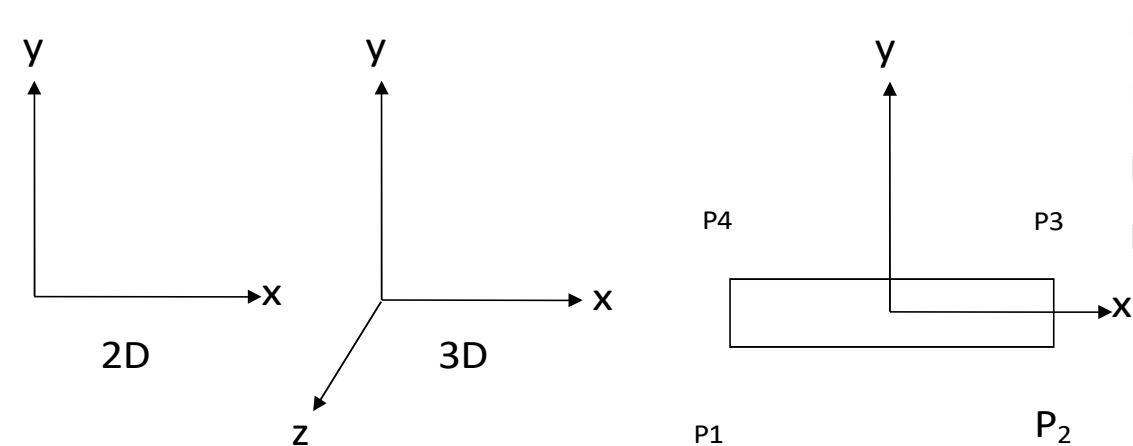


Imagen tomada de: [http://www.pytha.de/produkt/modeler\\_1.de.php](http://www.pytha.de/produkt/modeler_1.de.php)



## 2. objetos gráficos

# Sistemas de coordenadas



Cartesiano  
Sistemas de

Definición de rectángulo en  
Coordenadas cartesianas

$$P_1 = \{-2, 4, -1, 2\}$$

$$P_2 = \{2, 4, -1, 2\}$$

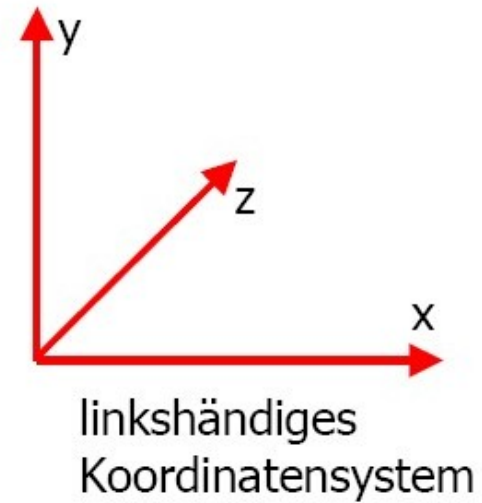
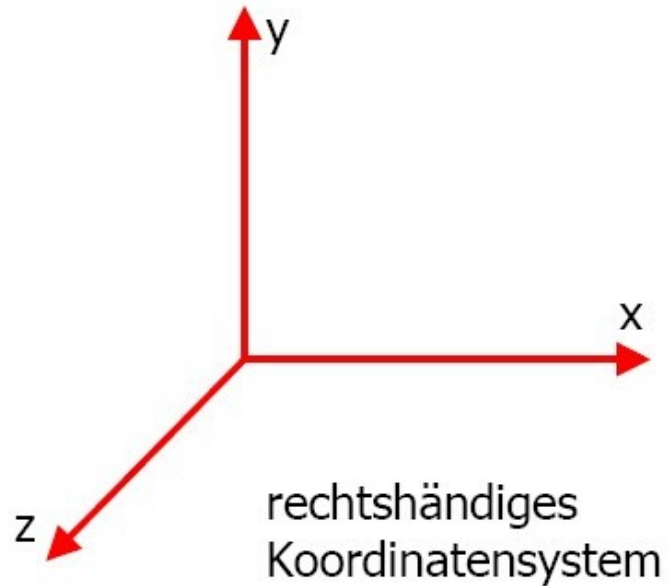
$$P_3 = \{2, 4, 1, 2\}$$

$$P_4 = \{-2, 4, 1, 2\}$$

**Cada geometría está  
definida por  
Puntos, aristas,  
superficies en un sistema  
de coordenadas**

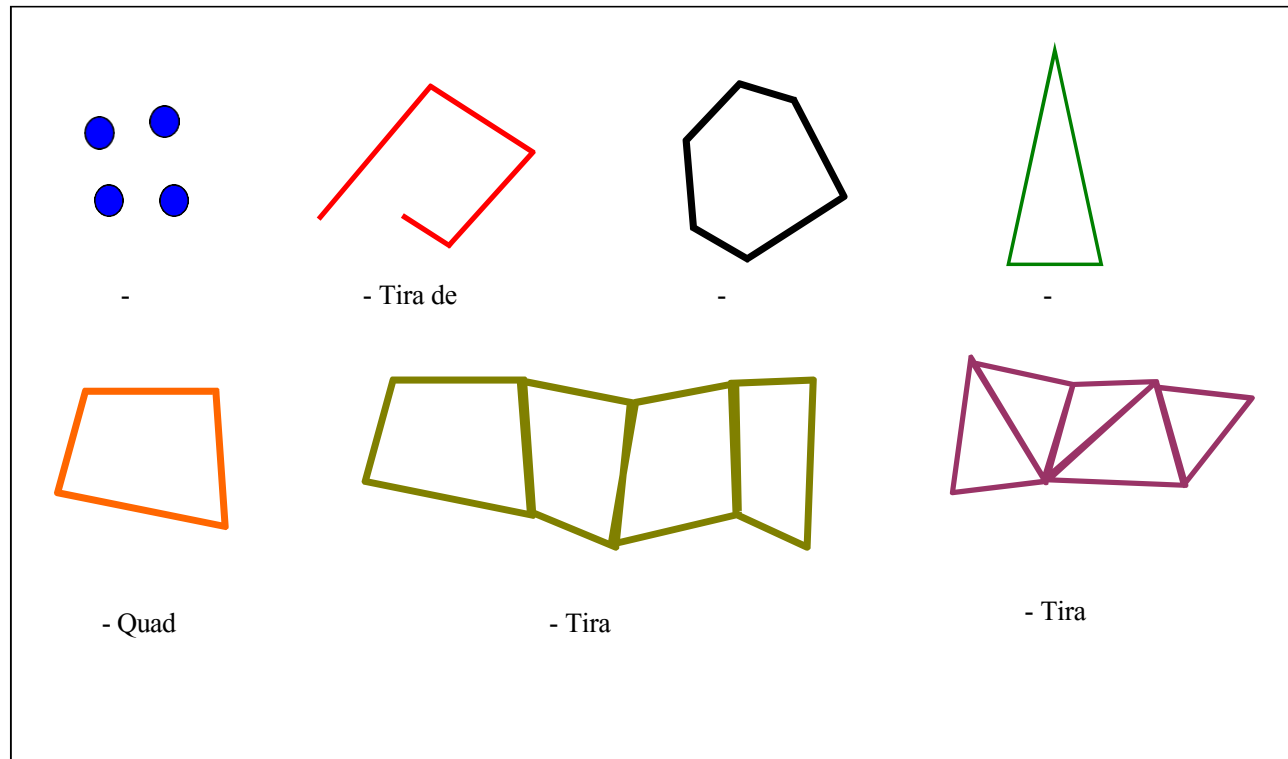
## 2. objetos gráficos

# Sistemas de coordenadas cartesianas



## 2. objetos gráficos

# Primitivas y objetos

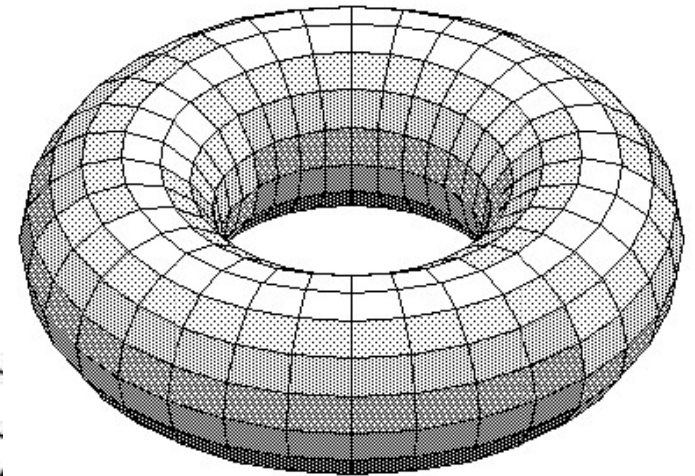
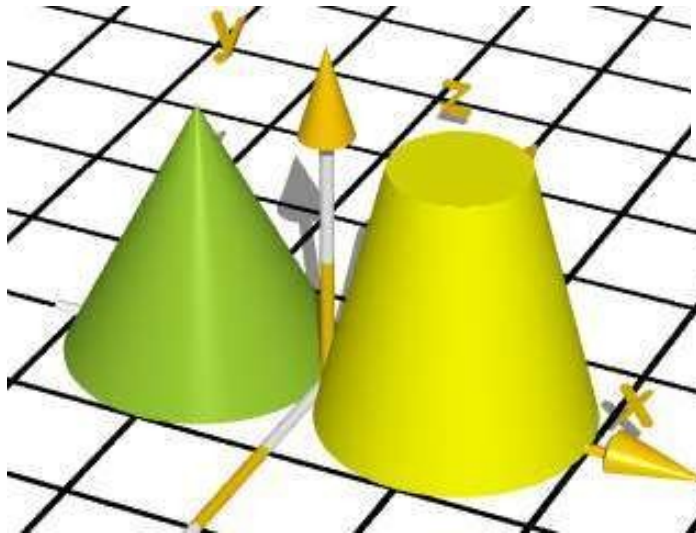
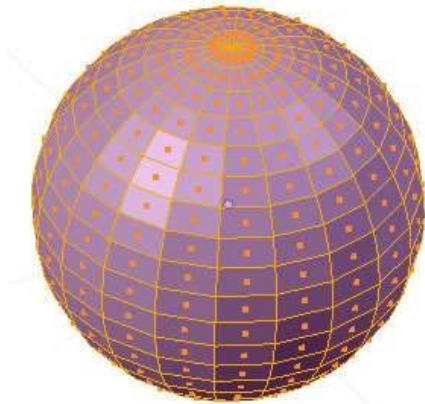


OpenGL conecta los puntos determinados a primitivas denominadas

## 2. objetos gráficos

# Primitivas y objetos

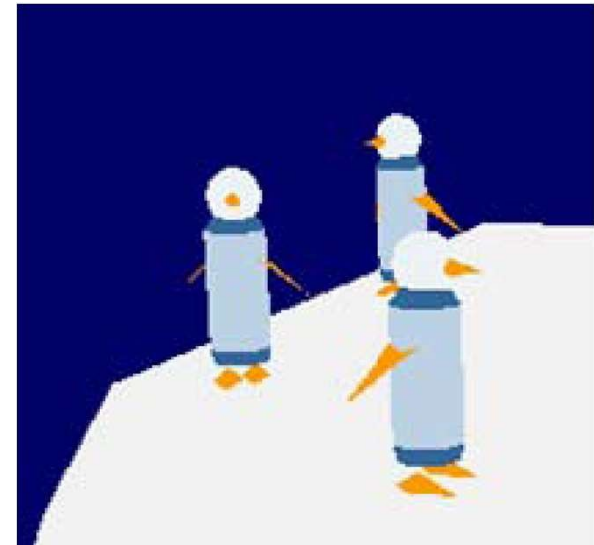
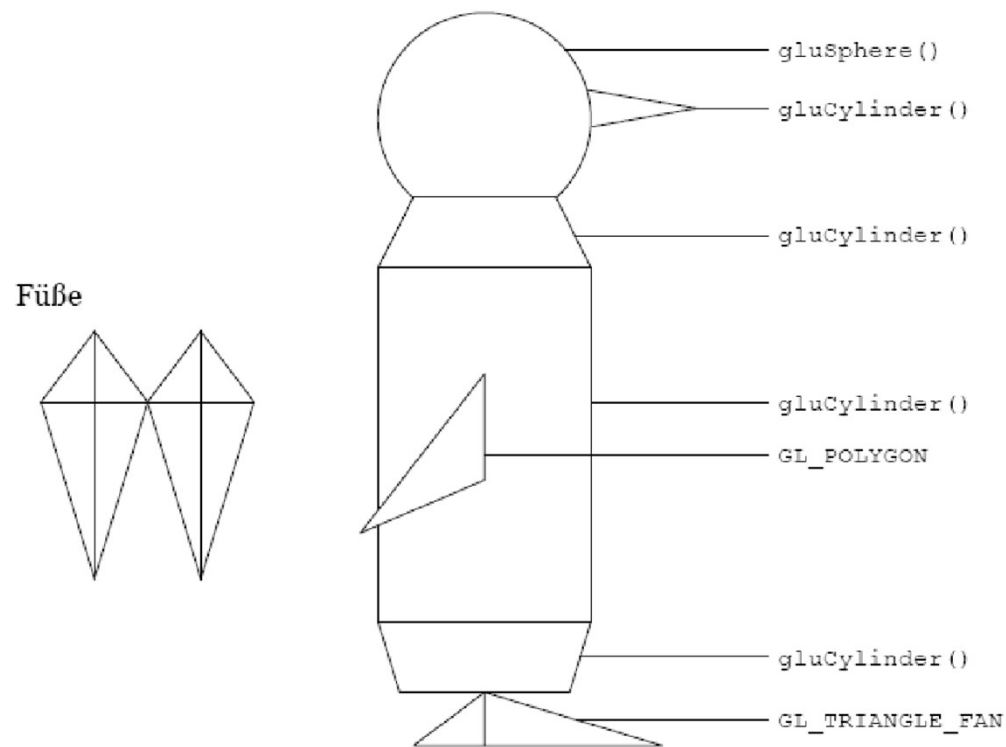
Las bibliotecas auxiliares proporcionan geometrías más complejas



## 2. objetos gráficos

# Creación de una geometría utilizando el ejemplo de

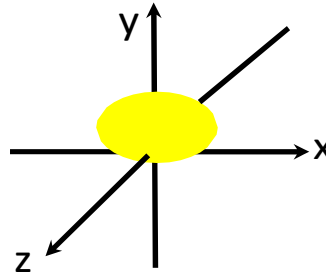
### Pinguin - Aufbau



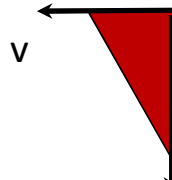
## 2. objetos gráficos

# Creación de una geometría utilizando el ejemplo de

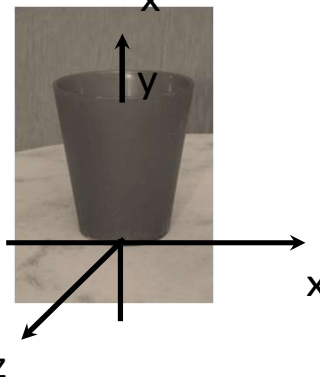
Cabeza de pingüino en local  
Sistema de coordenadas



Triángulo para los  
brazos y los pies en el sistema de  
coordenadas local



Copa en el sistema  
de coordenadas  
local



Pingüino en el mundo

## 2.1.2. Objetos gráficos

ma

de

coord

enad

as

## Creación de una geometría utilizando el ejemplo de

## 2. objetos gráficos

# Creación de una geometría utilizando el ejemplo de

¿Qué falta para completar el modelado del pingüino?

- Además de las magnitudes geométricas, también deben asignarse los **atributos gráficos**
- Los atributos gráficos se refieren a la "**apariencia**" de los objetos
- Ejemplo de atributos gráficos:
  - Color
  - Textura sintética mapeada (= "pegada") en la superficie
  - Imagen JPEG asignada a la superficie
  - ...



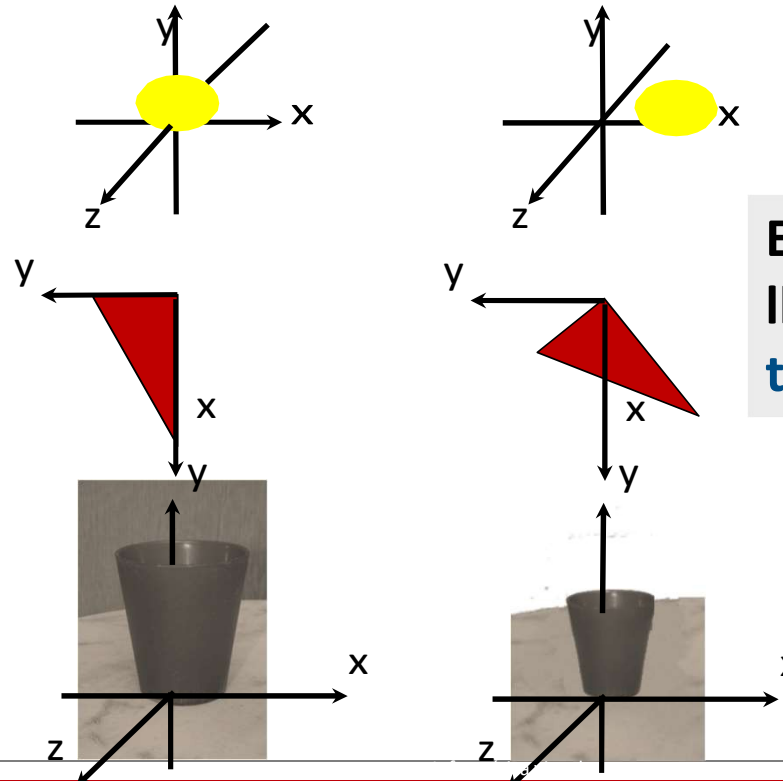
## 2. objetos gráficos

# Creación de una geometría utilizando el ejemplo de

¿Cómo pueden las primitivas "manipular" en el sistema de coordenadas local para construir un pingüino?

Tú puedes:

- mover (**traducir**)
- girar (**rotate**)
- ampliar y reducir (**escala**)

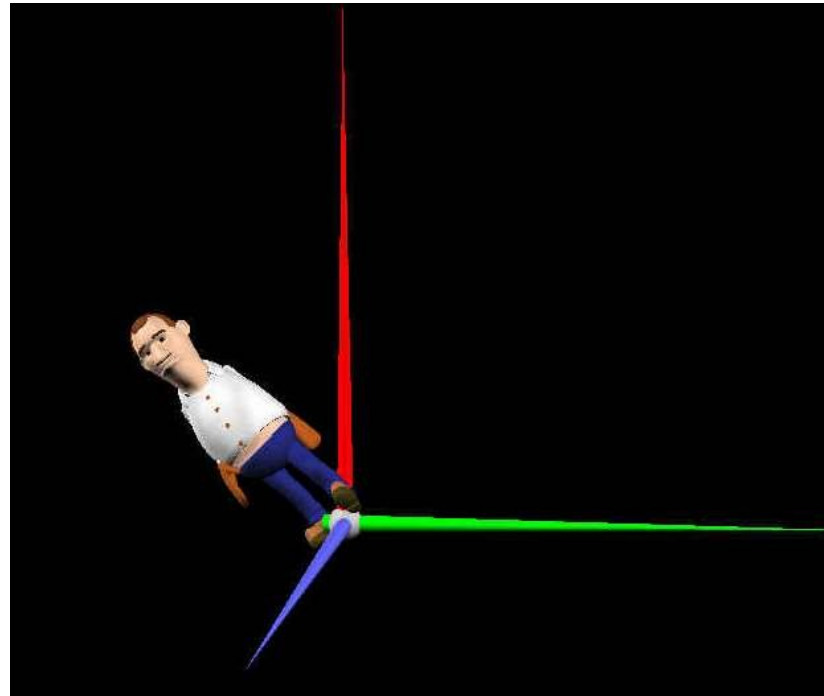
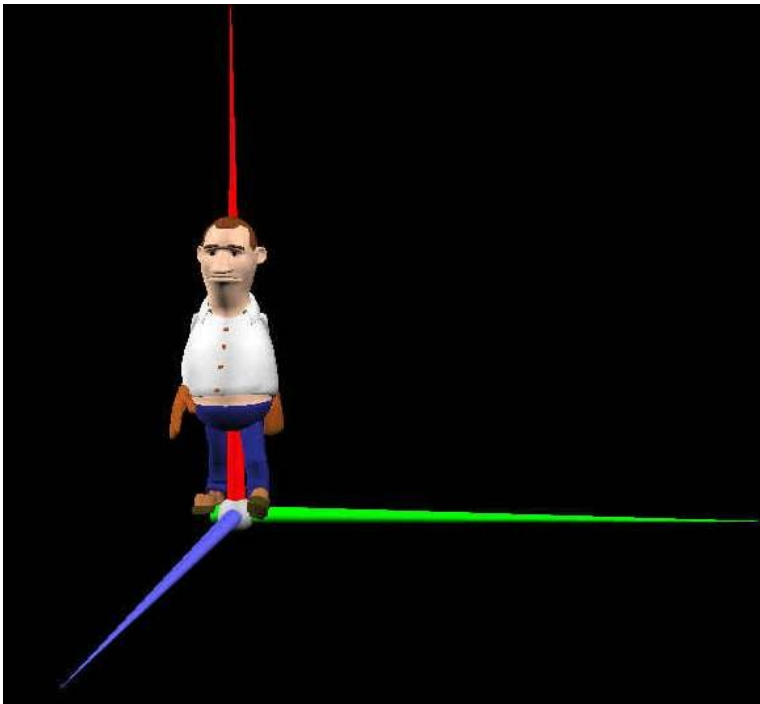


Estas manipulaciones se llaman **transformaciones**.

## 2. objetos gráficos

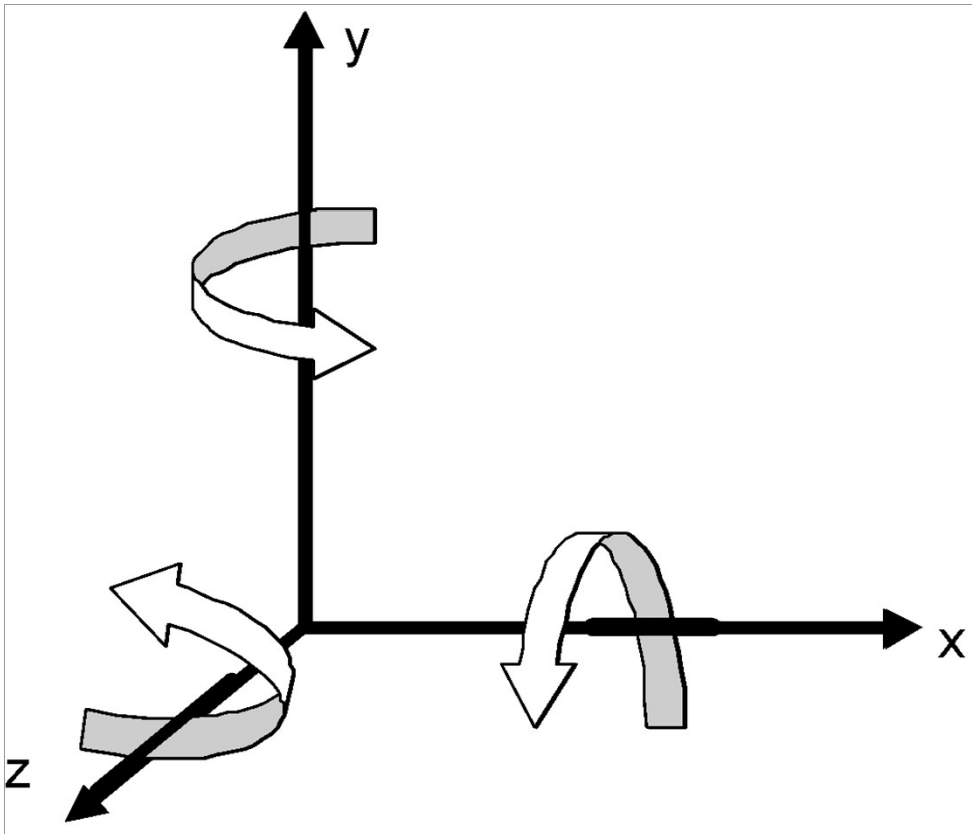
# Transformaciones: Rotación

¿En torno a qué eje gira la geometría durante la rotación?



## 2. objetos gráficos

### Demostración del sentido matemático positivo de la rotación

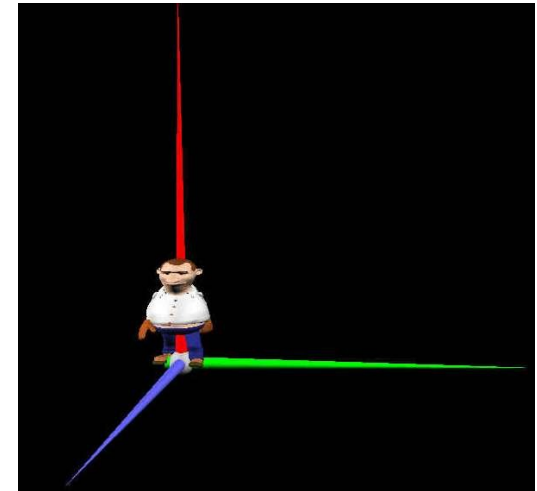
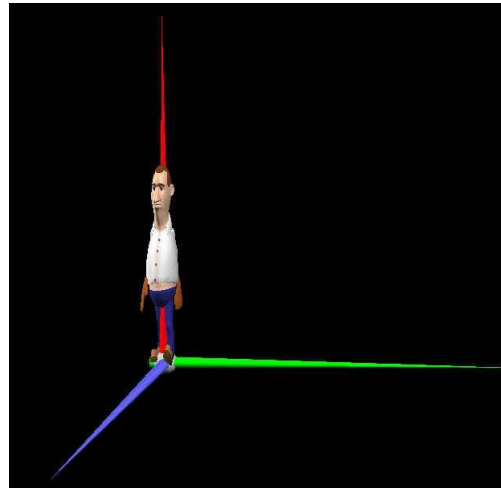
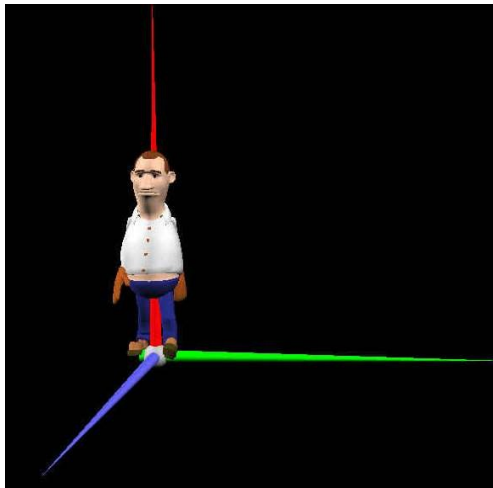


Cuando el pulgar de la mano derecha forma el eje de coordenadas, die angewinkelten indican el senhrichtung an.



## 2. objetos gráficos

# Transformaciones: Escala

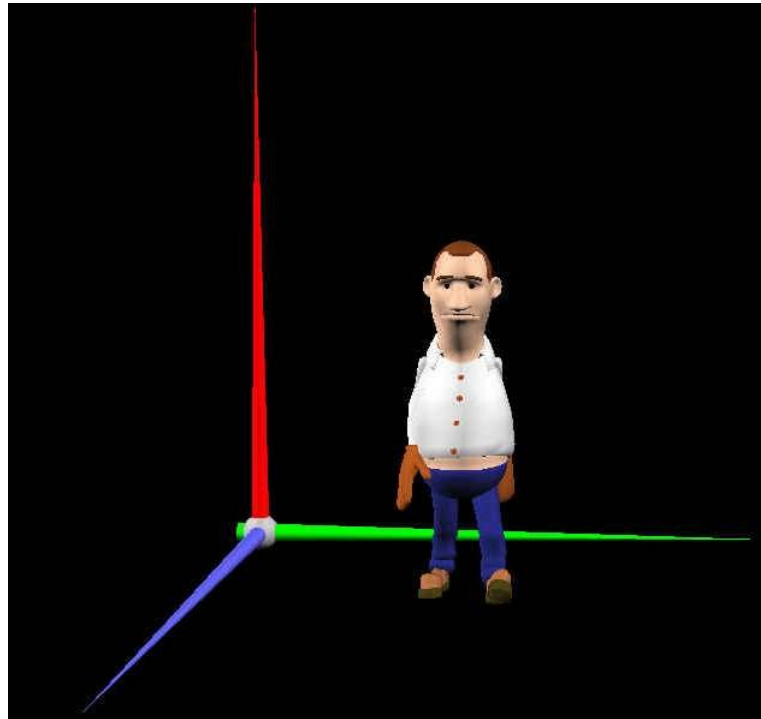


**¿Cómo se comporta la figura al escalarla si no está definida en el punto cero?**

## 2. objetos gráficos

# Transformaciones: Traducción

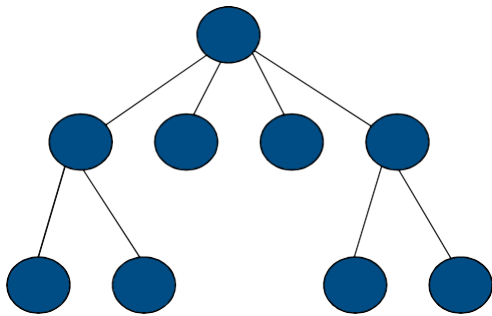
---



## 2. objetos gráficos

# Transformaciones: el gráfico de la escena

Diferencia entre árbol y gráfico



Árbol

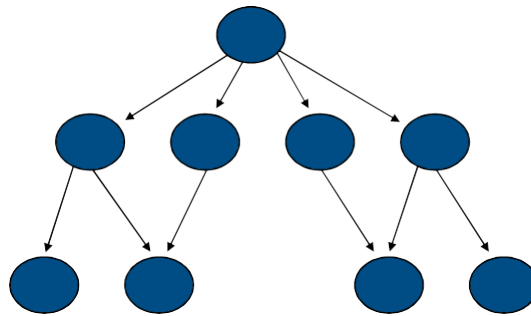
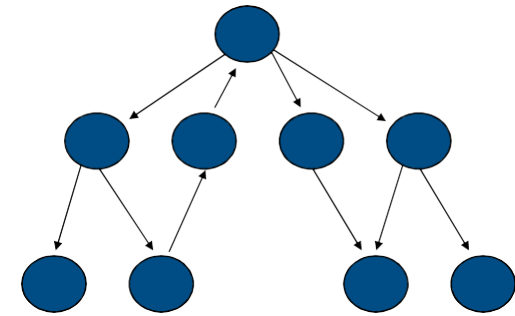


Gráfico  
direccional sin ciclos



dirigido  
grafo no cíclico

## 2. objetos gráficos

# Transformaciones: el gráfico de la escena

El grafo de escena consta de al menos 3 tipos de nodos:



Grupos



Geometrías (incl. propiedades de los materiales)



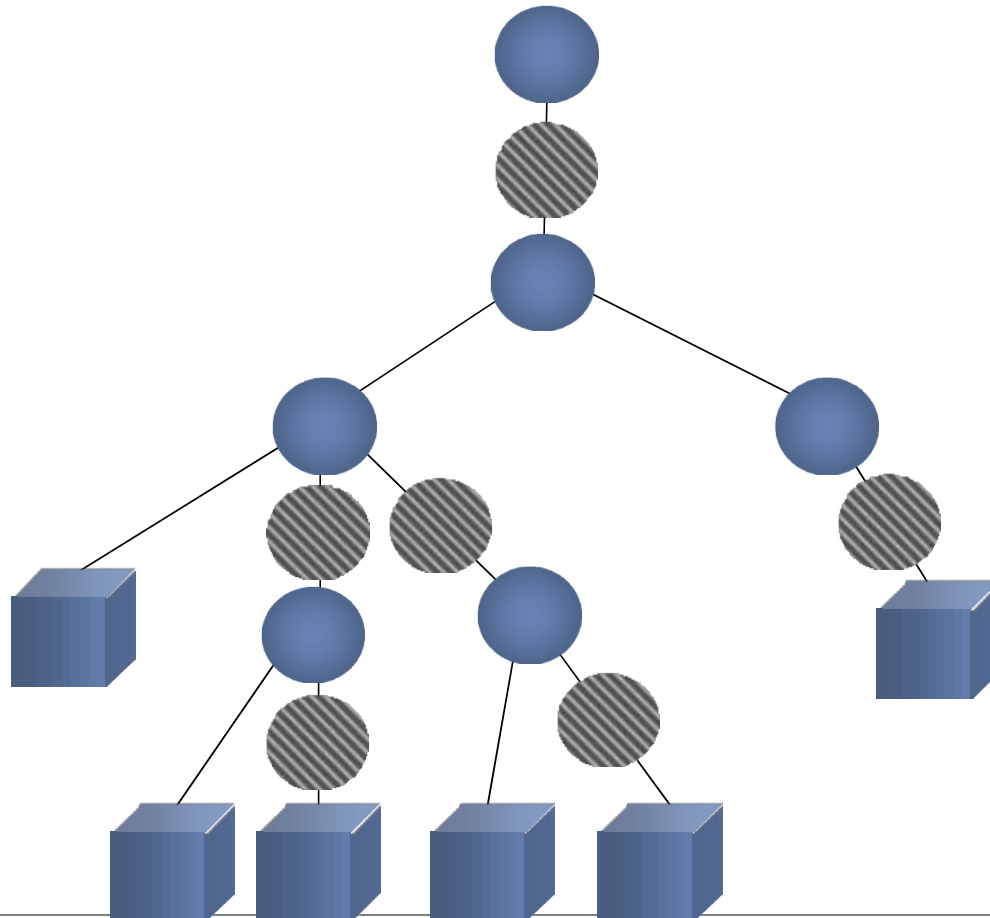
Transformaciones

El gráfico de escena se utiliza para gestionar una escena compleja:

- Agrupación de geometrías en grupos
- Agrupación en grupos
- Agrupación de grupos en una escena

## 2. objetos gráficos

# Estructura del gráfico de escenas



- dirigido
- acíclico
- heterogéneo
- Geometría / primitivas representables en las hojas

¿Qué ventaja tiene un grafo de escena acíclico dirigido sobre los grafos de escena realizados como árboles?

(Veamos entonces al pingüino)



## 2. objetos gráficos

# Creación del pingüino con ayuda de un gráfico de

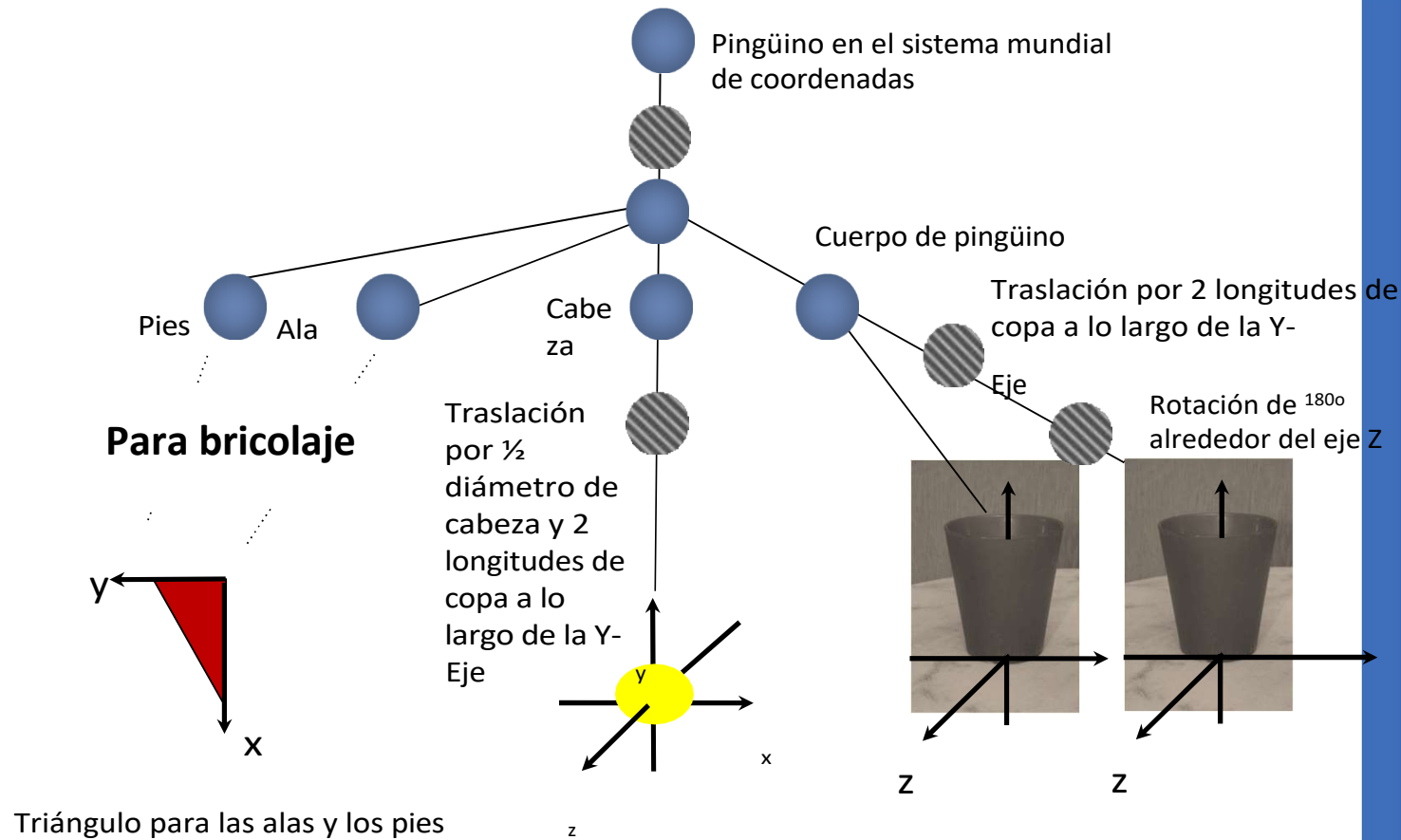


Están disponibles las siguientes primitivas:

- Taza con la abertura hacia arriba
- Pelota (ya con ojos y nariz)
- Un triángulo

## 2. objetos gráficos

# Creación del pingüino con ayuda de un gráfico de



## 2. objetos gráficos

# Creación del pingüino con ayuda de un gráfico de

