

---

GPIC. Generador gráfico de comandos *pic*

GSyC

*rsevilla@gsync.escet.urjc.es*



2004

---

## Plan de la exposición

- Introducción
- Objetivos de *Gpic*
- Funcionalidad
- Metodología empleada
- Desarrollo
- Conclusión

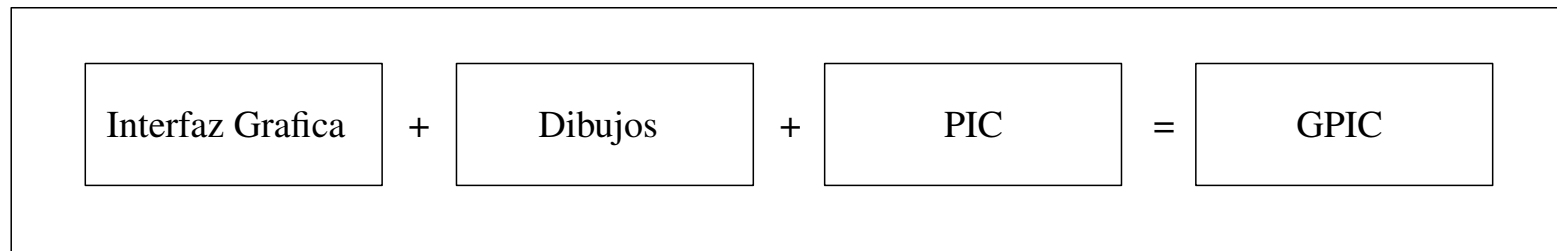
# Introducción

- Programas de dibujo
- Formatos:
  1. Mapas de bits
  2. Dibujo Vectorial o Geométrico
- Plan 9 y pic

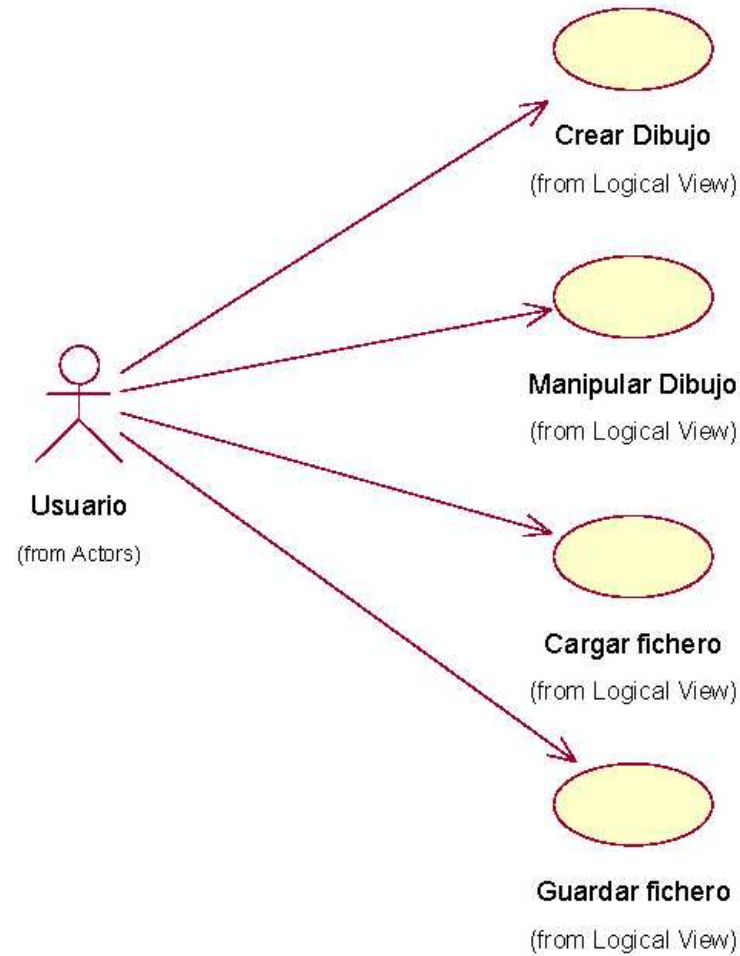
## Objetivos

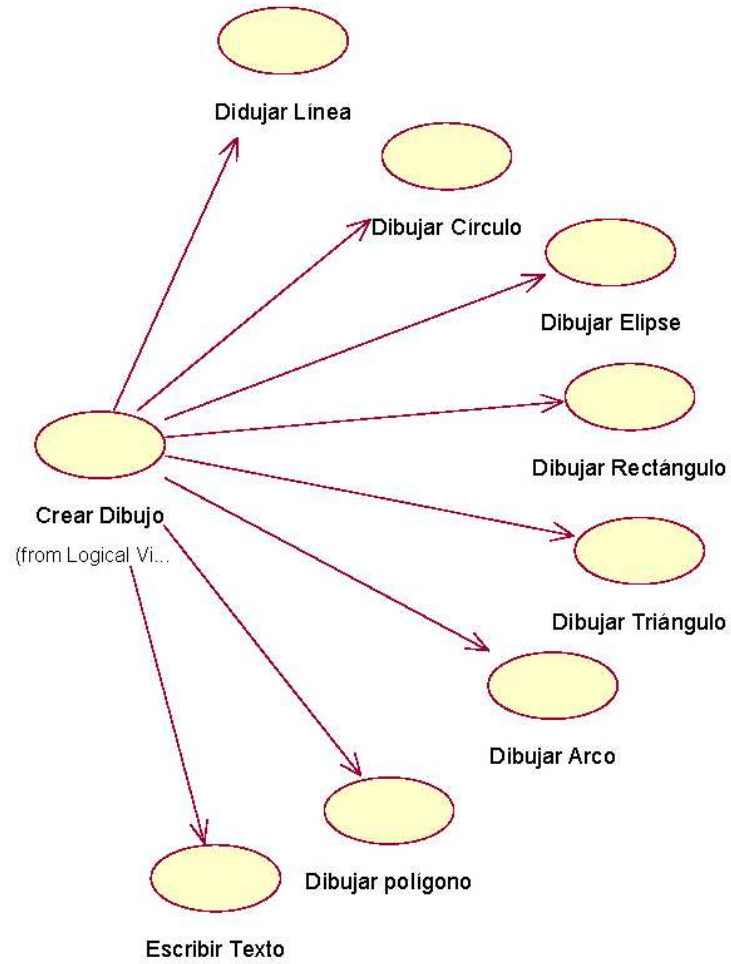
- Creación de una aplicación gráfica
- Simple e intuitiva
- Proporcione una instantánea del dibujo en creación
- Genere código *pic*

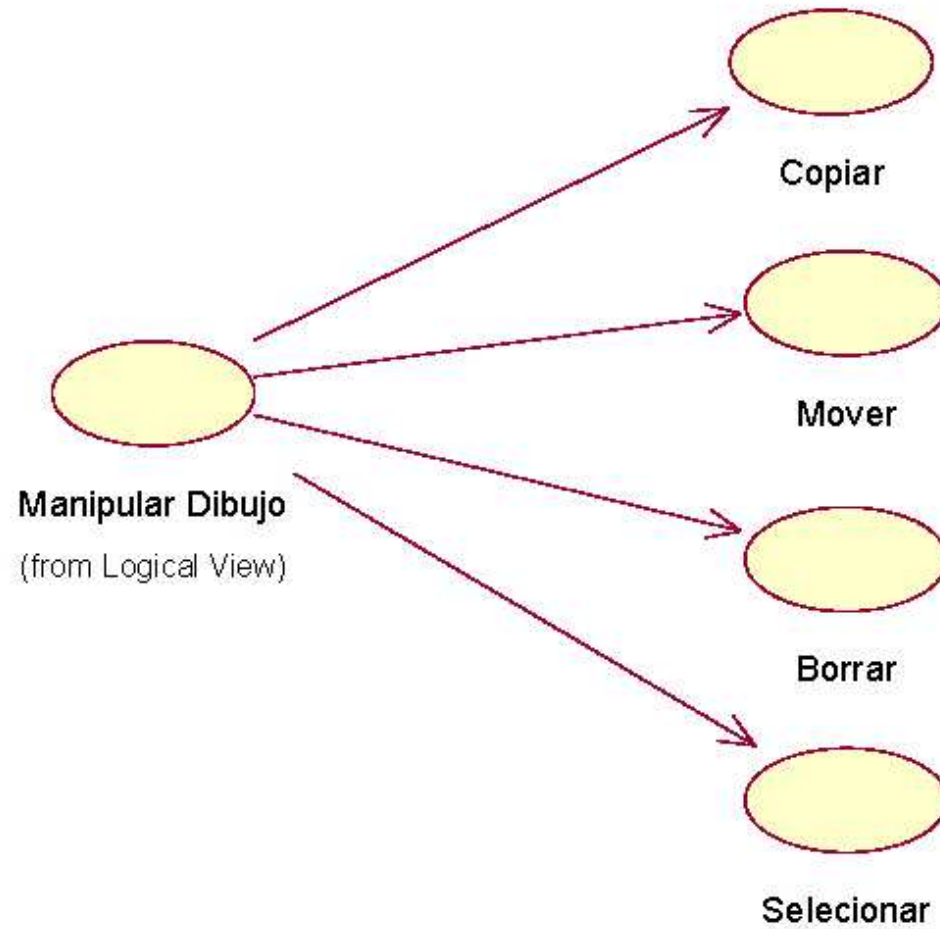
## Componentes



# Funcionalidad



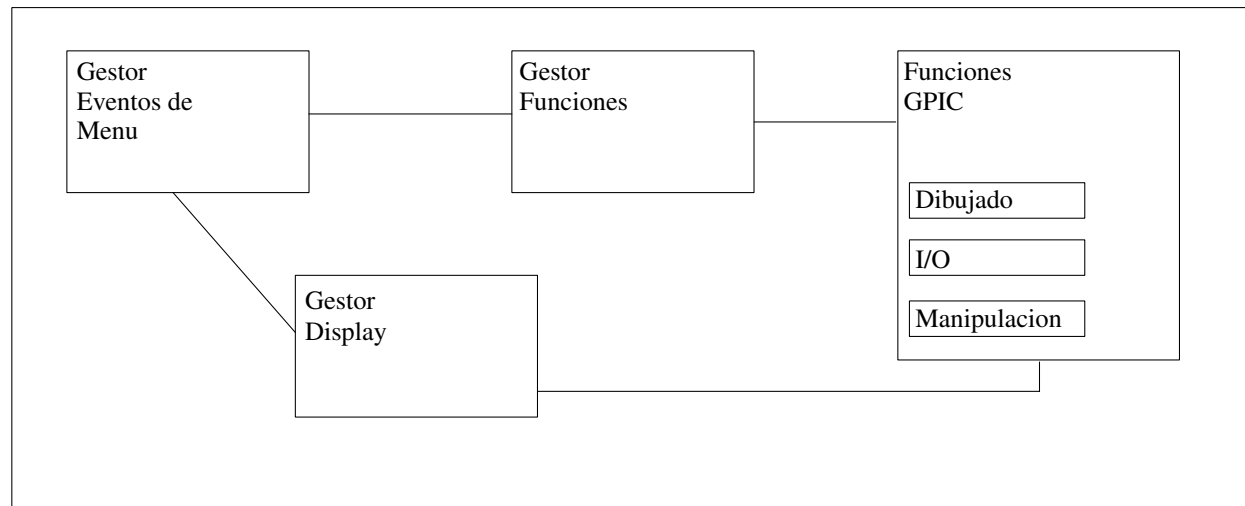




## Metodología

Se ha usado un modelo de desarrollo en espiral basado en prototipos.  
5 iteraciones. 5 prototipos.

# Diseño general



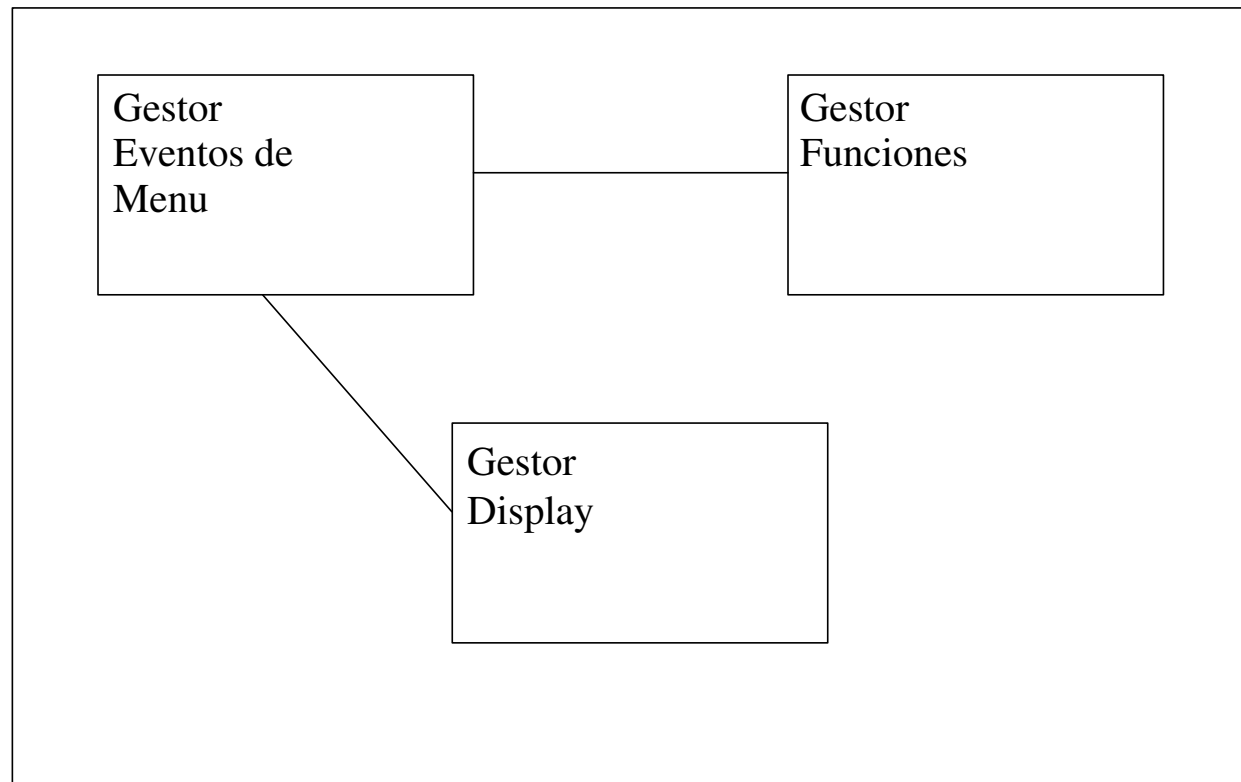
## Prototipo 1. Interfaz Gráfica

Objetivos  $\Rightarrow$

1. Creación de la interfaz de la aplicación.
2. Atender eventos del menú.
3. Lanzar las funciones correspondientes.

Requisitos  $\Rightarrow$  Estudio librería *control*, y comprensión de la comunicación entre sus elementos.

## Prototipo 1. Diseño



## Prototipo 1. Implementación

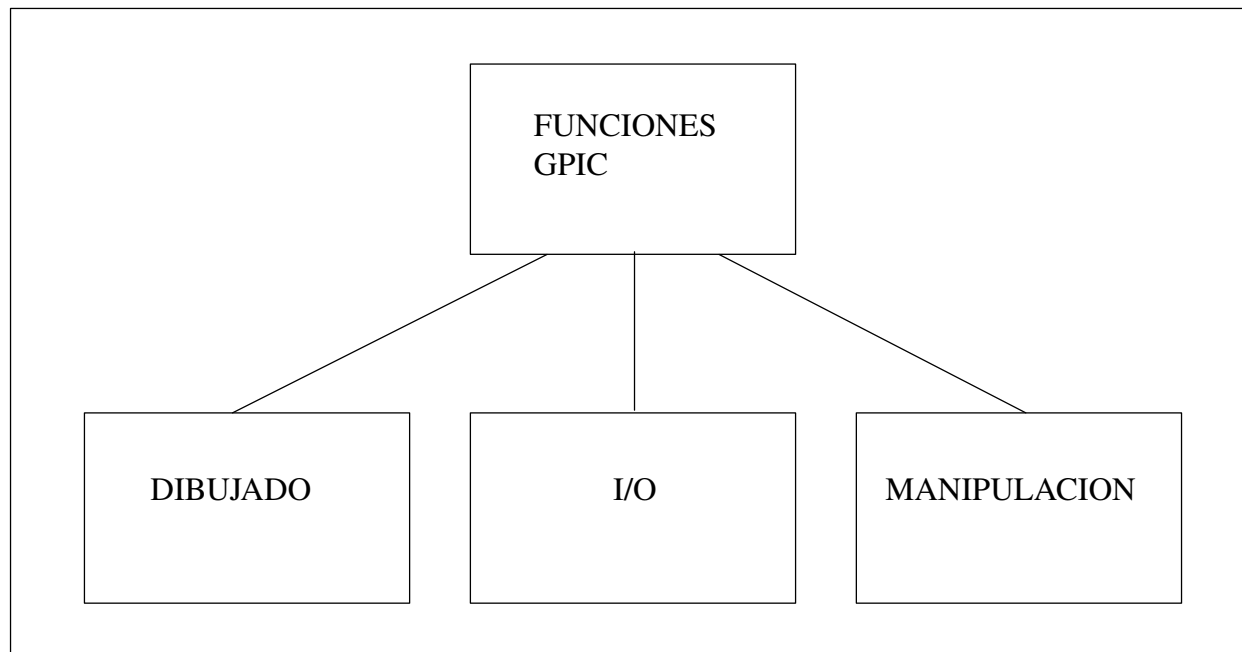
- Gestor del display: Creación de las estructuras necesarias para la construcción de la interfaz, y creación de las funciones que dibujan dicha interfaz.
- Gestor eventos de menú: Thread que escucha eventos del menú y asigna función a ejecutar.
- Gestor de funciones: Thread dedicado a lanzar las funciones que el gestor de eventos le indica.

## Prototipo 2. Dibujado de figuras

Objetivo  $\Rightarrow$  Creación de las funciones de dibujado de figuras.

Requisitos  $\Rightarrow$  Estudio de de las funciones de *draw*

## Prototipo 2. Diseño



## Prototipo 2. Implementación

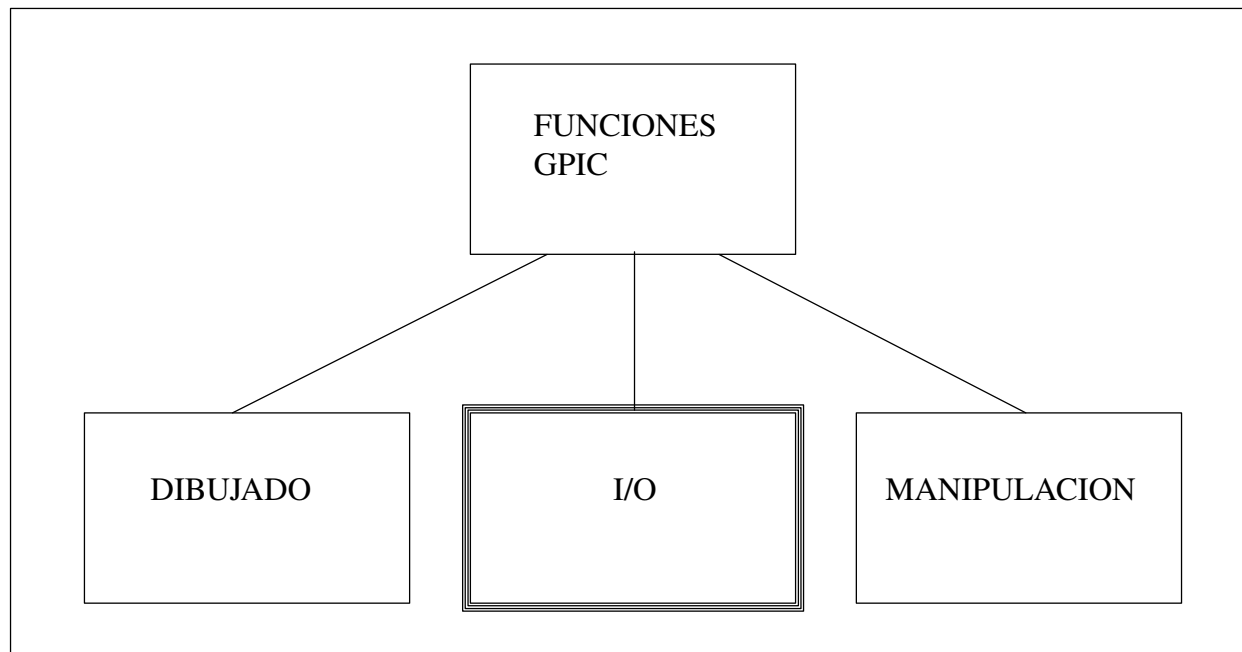
- Se crea la estructura global que alberga las figuras.
- Las funciones recogen las pulsaciones del ratón.
- Se Hacen las llamadas a la librería *Draw*.

## Prototipo 3. I/O. Generación código *pic*

Objetivo  $\Rightarrow$  Generar ficheros con el código *pic* correspondiente al dibujo.

Requisitos  $\Rightarrow$  Estudio del language *pic*.

## Prototipo 3. Diseño



## Prototipo 3. Implementación

Se crean las funciones de escritura de código *pic* y lectura del mismo.

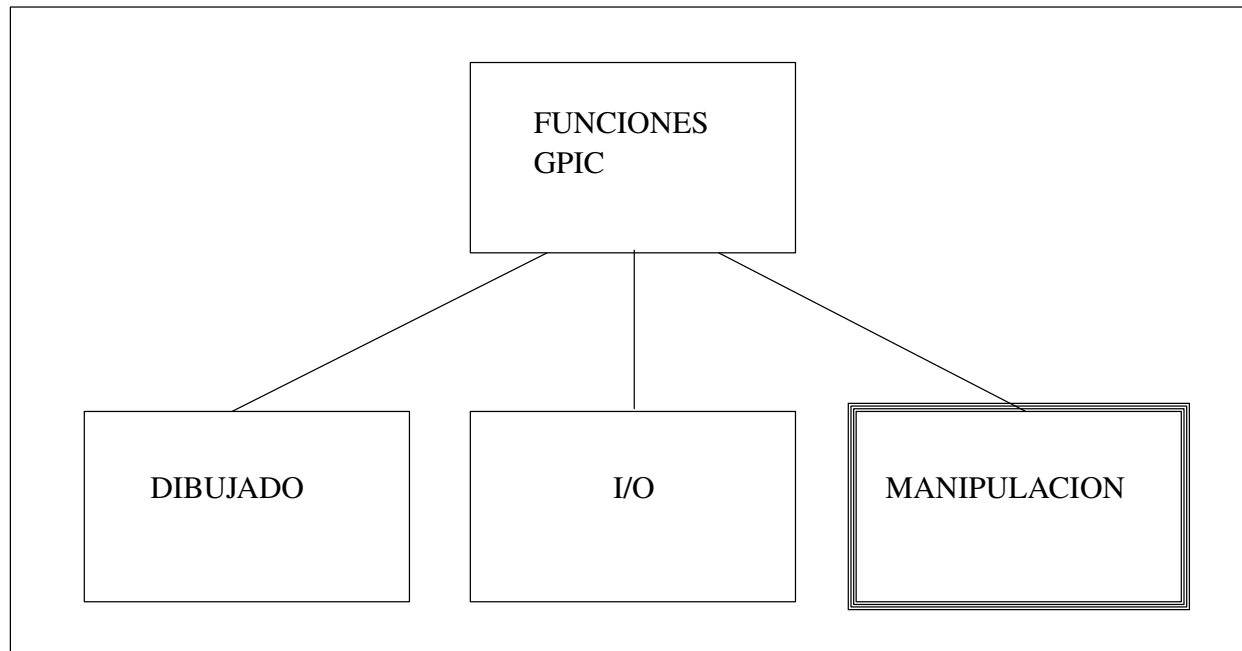
Ejemplo de línea en *pic*:

*“ellipse ht 00 wid 00 with .c at 00,00”*

## Prototipo 4. Manipulación de figuras

Objetivos  $\Rightarrow$  Creación de las funciones de manipulación de figuras.

## Prototipo 4. Diseño



## Prototipo 4. Implementación

Funciones:

- Seleccionar figuras.
- Borrar las figuras seleccionadas.
- Mover o Copiar las figuras seleccionadas.

## Prototipo 5. Optimización de la aplicación

Objetivo  $\Rightarrow$  Mejorar algunas facetas de la aplicación.

Mejoras efectuadas a la aplicación:

- Efecto goma
- Carga de fichero desde la línea de comando.

## Conclusiones

Se ha creado una aplicación gráfica para dibujar, que genera código *pic*, donde se visualiza en todo momento el dibujo creado.

Líneas futura:

- Ampliación de funcionalidad
  1. Opción deshacer
  2. Agrupación de figuras
- Reconocimiento completo de código *pic*