



ugr

Universidad
de Granada

PRÁCTICA 4

EL LENGUAJE POSTSCRIPT

Periféricos y Dispositivos de Interfaz Humana

Autores:
Clara Sola Ruiz
Mario Casas Pérez

Índice

1. Ejercicio 1 mínimo	3
2. Ejercicio 2 mínimo	6
3. Ejercicio 3 mínimo	10
4. Ejercicio 1 opcional	12
5. Ejercicio 2 opcional	15
Primera página: Tabla	15
Segunda página: Gráfica	18
Ejercicio en dos páginas	20
6. Ejercicios adicionales	21
Dibujo de Super Mario (ejercicio adicional 1).	21
Tarjeta de “El Juego del Calamar” (ejercicio adicional 2).	25
Primera página: Parte frontal	25
Segunda página: Parte trasera	27
Tarjeta en dos páginas	29
Flor (ejercicio adicional 3).	30
Tarjeta Feria de Empleo (ejercicio adicional 4)	34
7. Bibliografía	37

1.Ejercicio 1 mínimo

Para este ejercicio se pide que se muestre el dibujo de unos árboles. El código sigue los siguientes pasos:

- Tronco del árbol grande

El tronco del árbol es un rectángulo, por lo que primero se inicia un nuevo camino (*newpath*) y se mueve el lápiz al punto de inicio del tronco en la coordenada (120, 140).

A partir de ahí, trazamos tres líneas: una hacia arriba (300 unidades), una a la derecha (70 unidades), y otra hacia abajo (300 unidades), formando tres lados del rectángulo. Con *closepath* se cierra el rectángulo automáticamente. Luego establecemos un color marrón mediante *setrgbcolor* y se rellena el tronco con *fill*.

```
% Tronco

newpath
120 140 moveto
0 300 rlineto
70 0 rlineto
0 -300 rlineto
closepath

0.4 0.2 0 setrgbcolor % marrón
fill
```

- Copa del árbol grande

La copa del árbol grande la vamos a hacer con tres óvalos. Cada uno se crea a partir de un círculo al que se le aplica una transformación de escala para alargarlo verticalmente, pareciéndose más al del dibujo que si lo hiciéramos con círculos.

Para ello, se usa *gsave* y *grestore*, que permiten modificar el sistema de coordenadas de forma temporal. Dentro de cada bloque de los óvalos:

- Se traslada el origen al punto donde estará el centro del óvalo (*translate*).
- Se escala el espacio verticalmente (*scale*) para transformar el círculo en óvalo.
- Se dibuja un círculo con *arc*, que aparece como un óvalo debido a la escala.

Finalmente, se rellena con color verde. Al usar *grestore*, el sistema vuelve a la normalidad y el siguiente óvalo no se ve afectado por las transformaciones anteriores.

```
% Copa con 3 círculos verdes

0 0.6 0 setrgbcolor % verde

gsave
```

```

155 500 translate
0.8 1.5 scale
0 0 100 0 360 arc
0 0.6 0 setrgbcolor
fill
grestore          % Restaurar sistema de coordenadas original

gsave
215 550 translate
0.7 1.5 scale
0 0 100 0 360 arc
0 0.6 0 setrgbcolor
fill
grestore

gsave
120 600 translate
0.8 1.5 scale
0 0 100 0 360 arc
0 0.6 0 setrgbcolor
fill
grestore

```

- Tronco del árbol pequeño

El tronco pequeño sigue los mismos pasos de construcción que el del tronco grande, pero en este caso lo hemos movido a la derecha y lo hemos hecho más estrecho cambiando las coordenadas.

```

% Tronco

newpath
400 140 moveto
0 180 rlineto
40 0 rlineto
0 -180 rlineto
closepath

0.4 0.2 0 setrgbcolor
fill

```

- Copa del árbol pequeño

La copa del árbol se forma utilizando tres círculos superpuestos, cada uno dibujado con la instrucción `arc`, que genera un círculo completo (de 0° a 360°). Antes de dibujarlos, establecemos el color verde con `setrgbcolor`. Cada círculo se sitúa en una posición distinta mediante sus coordenadas (x, y) y se define su tamaño con el radio.

Tras definir cada camino con `newpath`, se usa `fill` para rellenarlos con el color actual.

A diferencia de los óvalos del árbol grande, aquí no aplicamos ninguna transformación de escala, por lo que las formas se mantienen como círculos.

```
% Copa con 3 círculos verdes
```

```
0 0.6 0 setrgbcolor
```

```
newpath  
440 340 80 0 360 arc  
fill
```

```
newpath  
375 380 75 0 360 arc  
fill
```

```
newpath  
435 450 75 0 360 arc  
fill
```

- Círculo

Dibujamos un círculo decorativo formado por dos pasos: un relleno verde claro y un borde rojo. Creamos un nuevo camino con *newpath* y se utiliza *arc* para dibujar un círculo centrado en (295, 165) con un radio de 25. Establecemos el color de relleno con *setrgbcolor* en tono verde claro y se aplica *fill*.

Posteriormente hacemos el mismo círculo pero en rojo y con un grosor de línea definido por *setlinewidth*. Finalmente, se dibuja solo el contorno con *stroke*, creando un círculo con borde de color diferente.

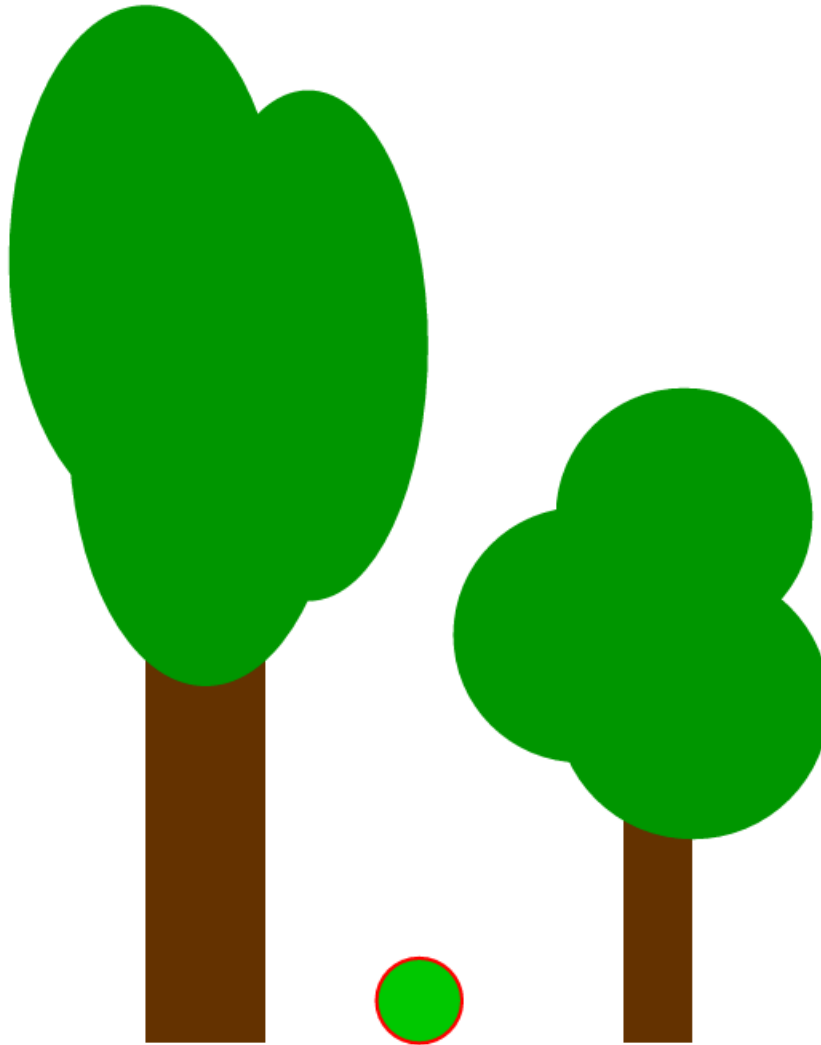
```
% --- Círculo ---
```

```
newpath  
295 165 25 0 360 arc    % 165 = 140 + 25  
0 0.8 0 setrgbcolor  
fill
```

```
newpath  
295 165 25 0 360 arc  
1 0 0 setrgbcolor  
2 setlinewidth  
stroke
```

```
showpage
```

Al final usamos el comando *showpage*, esto es para indicar al intérprete PostScript que se debe imprimir la página, finalizando toda la construcción de los árboles y el círculo. Nos queda la siguiente figura:



2.Ejercicio 2 mínimo

Para este ejercicio se ha realizado un diseño de una tarjeta de visita basada en la organización de juegos *Play Time*. El código lo hemos descompuesto de la siguiente manera:

- **Recuadro grande de fondo.**

Usamos el comando *newpath* para iniciar una nueva ruta gráfica y *moveto* nos situamos en una coordenada concreta para empezar a hacer el recuadro. Luego, usaremos *rlineto* para dibujar líneas, haciendo el contorno del rectángulo. Por último, establecemos el color del recuadro grande a gris mediante *setrgbcolor* y rellenamos la forma con *fill*.

```

newpath          %Recuadro grande de fondo
168 560 moveto
280 0 rlineto
0 -200 rlineto
-280 0 rlineto
0.7 0.7 0.7 setrgbcolor
fill
closepath
5 setlinewidth
stroke

```

- Recuadro grande interno.

Aquí seguiremos una estructura similar a la descrita previamente, donde cambiamos las coordenadas y dimensiones ligeramente para crear un recuadro interno. Además, ponemos un color gris más claro para generar un contraste visual con el fondo, haciendo más “realista” la tarjeta de invitación.

```

newpath          %Recuadro grande
175 550 moveto
265 0 rlineto
0 -180 rlineto
-265 0 rlineto
0.82 0.82 0.82 setrgbcolor
fill
closepath
5 setlinewidth
stroke

```

- Círculo del logo.

Iniciamos una nueva ruta estableciendo el color en escala de grises para la línea. Luego, hemos procedido a trazar un círculo con radio 45, teniendo un ángulo completo (0 a 360 grados). Además, usaremos *gsave* para guardar el estado actual de la gráfica y *grestore* para restaurar el estado gráfico anterior, de esta forma los cambios que se hagan dentro de este bloque no afectarán al resto del código. Luego, pasamos a redefinir el ancho de la traza a 1 y dibujamos el contorno con *stroke*.

```

newpath          %Círculo del logo de nuestra tarjeta
0 setgray
0.5 setlinewidth
235 475 45 0 360 arc closepath
gsave
0.87 0.87 0.87 setrgbcolor fill
grestore

```

```
1 setlinewidth
stroke
closepath
```

- Texto escrito.

El resto de código lo hemos hecho para posicionar varios bloques de texto en la tarjeta. Hemos estado viendo diversas posiciones para ver cual quedaba más visual la tarjeta. La estructura de todos los bloques de texto son similares (a excepción de las posiciones de estos), en donde escribiremos lo siguiente:

- **Siglas dentro del círculo:** definimos nuestro logo de la tarjeta de empresa.
- **Nombre y objetivo de la empresa:** en la parte superior derecha de la tarjeta indicaremos el nombre de la empresa y a qué se encarga.
- **Datos de la empresa:** dejaremos un espacio para indicar los datos de la empresa en nuestra tarjeta. Aquí se indicará la dirección, el código postal, el número del local y edificio, dos números de teléfono y el correo electrónico de la empresa

```
/Times-Roman findfont
40 scalefont
setfont
newpath
210 462 moveto
(PT) show
stroke

/Times-Roman findfont
15 scalefont
setfont
newpath
300 520 moveto
(Play Time) show
stroke

/Times-Roman findfont
10 scalefont
setfont
newpath
300 500 moveto
(Organizador de juegos) show
stroke

/Times-Roman findfont
```



```
10 scalefont
setfont
newpath
300 460 moveto
(Paseo de la Avenida Madrid, 2) show
stroke

/Times-Roman findfont
10 scalefont
setfont
newpath
300 440 moveto
(459023, Madrid) show
stroke

/Times-Roman findfont
10 scalefont
setfont
newpath
300 420 moveto
(Local 7: Edificio FAI) show
stroke

/Times-Roman findfont
10 scalefont
setfont
newpath
300 400 moveto
(Tlfs: 233 555 092 / 256 778 222) show
stroke

/Times-Roman findfont
10 scalefont
setfont
newpath
300 380 moveto
(casas10@gmail.com) show
stroke

showpage
```

Vemos que al final usamos el comando *showpage*, esto es para indicar al intérprete PostScript que se debe imprimir la página, finalizando toda la construcción de nuestra tarjeta. Finalmente, esta queda de la siguiente forma.



3. Ejercicio 3 mínimo

Para este ejercicio vamos a realizar un dibujo de un corazón con una poesía al lado. Para ello hemos descompuesto el código de la siguiente manera:

- Dibujo de la forma de un corazón

Hemos creado varios componentes gráficos, para construir la figura de un corazón mediante **curvas Bézier**. Para ello, iniciamos con *newpath* y nos posicionamos para dibujar en la parte superior central del corazón con *moveto*. Luego, pasamos a dibujar las curvas. Esto lo haremos con el comando *curveto*, en donde especificaremos coordenadas tanto para el lóbulo izquierdo como el derecho, las cuales tienen que ser iguales para que el corazón tenga simetría en ambos lóbulos. Finalmente, con el comando *closepath* cerramos la ruta conectando el último punto con el del inicio para rellenar la figura después de color rojo.

```
%Dibujo de forma de corazón
newpath
%Empezamos en la parte superior central del corazón
160 520 moveto
%Parte izquierda
160 600 40 600 40 520 curveto
40 440 160 400 160 360 curveto
%Parte derecha
160 400 280 440 280 520 curveto
280 600 160 600 160 520 curveto
closepath

%Color rojo de relleno y trazado del contorno
1 0 0 setrgbcolor
fill
```

```
0 setgray
stroke
```

- Texto poético

Vamos a añadir una poesía en la parte central derecha, justo al lado del corazón. Para ello, seleccionamos la fuente y escalamos el tamaño base del texto. Cada línea escrita de la poesía se ha escrito en posiciones diferentes, en donde se ha usado *moveto* para esto. Además, para cada línea, hemos ajustado el color en escala de grises, por lo que antes de cada *show* modificamos el nivel de gris. La primera línea aparece con un valor relativamente alto y lo vamos reduciendo progresivamente, tal y como viene en el guión de la práctica, haciendo un sutil degradado en la intensidad del texto. Finalmente, cada línea de texto se muestra con el comando *show* y usamos el comando *stroke* para terminar el trazado del texto.

```
%Texto poético
/Times-Roman findfont
15 scalefont
setfont

300 480 moveto
0.8 setgray
(Como recordar tu cabello dorado) show
stroke

300 460 moveto
0.7 setgray
(con solo verte me dejas encantado) show
stroke

300 440 moveto
0.5 setgray
(los animales en tu jardin te sonrien) show
stroke

300 420 moveto
0.3 setgray
(contemplando como ella se peina) show
stroke

300 400 moveto
0.1 setgray
(quiero que a mi lado seas la reina) show
stroke

showpage
```

El corazón y el texto poético han quedado de la siguiente manera.



Como recordar tu cabello dorado
con solo verte me dejas encantado
los animales en tu jardin te sonrien
contemplando como ella se peina
quiero que a mi lado seas la reina

4. Ejercicio 1 opcional

En este ejercicio se nos pide que se muestre la imagen de una sonrisa en apaisado, dentro de una serie de círculos concéntricos de diferentes colores. Para ello, hemos seguido la siguiente lógica en el código:

- Configuración del tamaño de la página

Tal y como se indica en el ejercicio, la imagen debe de estar configurada, por lo que definimos el tamaño de la página.

```
<< /PageSize [792 612] >> setpagedevice
```

- Círculos concéntricos

En un bloque *gsave* y *grestore* nos encargaremos de dibujar varios círculos concéntricos de distintos tamaños y colores. Lo primero de todo será trasladar el origen al centro para ayudarnos a la creación de estos círculos. Además, para la creación de cada uno de ellos usaremos en comando *newpath* para iniciar la definición de la figura, *arc* para trazar la curva del círculo, y *fill* para rellenar el área con el color que se ha especificado. Como ponemos el comando *gsave* al principio y al final *grestore* nos aseguramos que este bloque de transformación y estilo se limite al fondo.

```
gsave
  %Nos situamos en el centro
  396 306 translate

  %Primer circulo de radio 260, un poco más grande que el resto, color
```

```

magenta
  newpath
    0 0 260 0 360 arc
  closepath
  1 0 1 setrgbcolor
  fill

%Segundo círculo de radio 200, color rojo
  newpath
    0 0 200 0 360 arc
  closepath
  1 0 0 setrgbcolor
  fill

%Tercer círculo de radio 160, color naranja
  newpath
    0 0 160 0 360 arc
  closepath
  1 0.5 0 setrgbcolor
  fill

%Cuarto círculo de radio 120, color verde
  newpath
    0 0 120 0 360 arc
  closepath
  0 1 0 setrgbcolor
  fill

%Quinto círculo de radio 80, color azul
  newpath
    0 0 80 0 360 arc
  closepath
  0 0 1 setrgbcolor
  fill
grestore

```

- Imagen de la cara sonriente

En esta parte del código vamos a superponer una imagen bit-mapped de una cara sonriente en el centro de la página. Usaremos un bloque *gsave* y *grestore* y, lo primero que haremos será usar el comando *initmatrix* para restablecer la matriz de transformación y eliminar cualquier alteración que hayamos hecho antes. Luego, trasladamos el origen nuevamente al centro de la página con el comando *translate* para luego aplicar un escalado a la imagen (*scale*) y un ajuste fino de la traslación para que quede centrado en la página.

Finalmente, usaremos el operador *image* para dibujar la imagen bit-mapped. Los parámetros de la imagen serán las dimensiones, la profundidad de bits y la matriz de transformación (usado para mapear la cuadrícula de la imagen al espacio definido). La

imagen se define mediante una secuencia hexadecimal que representa los valores de píxel en la cuadrícula. Luego, una vez dibujada la imagen, se finaliza el bloque con *grestore* para restaurar la configuración previa y, por último, finalizamos la página con el comando *showpage*.

```
%Dibujamos la cara en el centro
gsave
  %Reestablecemos la CTM y trasladamos
  initmatrix
  396 306 translate

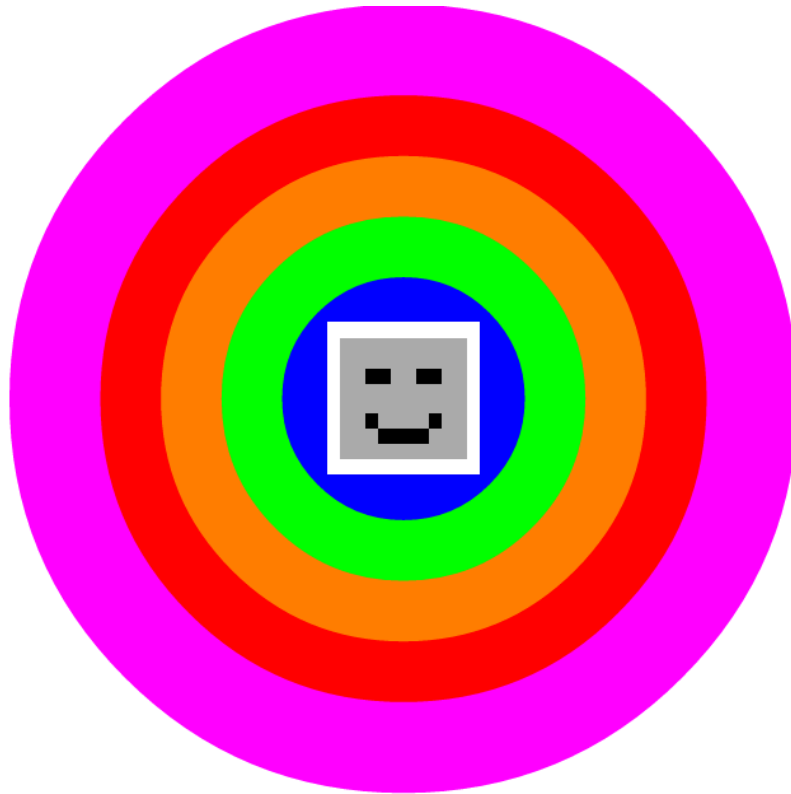
  %Escalamos la imagen
  100 100 scale

  -0.5 -0.5 translate

  %Dibujamos la imagen
  12 10 8 [12 0 0 -10 0 10]
  {<
    ffffffffffffffffffffffffffff
    ffaaaaaaaaaaaaaaaaaaaaaaff
    ffaaaaaaaaaaaaaaaaaaaaaaff
    ffaaaa0000aaaa0000aaaaaff
    ffaaaaaaaaaaaaaaaaaaaaaaff
    ffaaaaaaaaaaaaaaaaaaaaaaff
    ffaaaa00aaaaaaaa00aaaaaff
    ffaaaaaa00000000aaaaaaff
    ffaaaaaaaaaaaaaaaaaaaaaaff
    ffffffffffffffffffffffffffff
  >}
  image
grestore

showpage
```

La página queda de la siguiente manera.



5. Ejercicio 2 opcional

En este ejercicio se nos pide que creamos dos páginas, una donde se muestra una tabla y en la otra una imagen como un gráfico. Se han seguido los siguientes pasos para construir la figura:

Primera página: Tabla

- Configuración del texto

Seleccionamos la fuente Times-Roman y ajustamos su tamaño con *scalefont*. También establecemos el tamaño de la letra y lo colocamos con *moveto* en la posición (72, 750), utilizando el comando *show* para mostrar el texto "Mi tabla de datos".

```
%!PS
% -----
% PÁGINA 1 - TABLA

/Times-Roman findfont
26 scalefont
setfont

newpath
72 750 moveto
```

```
(Mi tabla de datos) show
```

```
/Times-Roman findfont  
20 scalefont  
setfont
```

- Dibujo de la estructura de la tabla

Las líneas horizontales de la tabla se trazan con *moveto* y *lineto*, definiendo las tres líneas que separan el encabezado, la fila de datos y los bordes superior e inferior. Las líneas verticales delimitan el borde izquierdo, el borde derecho y la separación entre las dos columnas. Estas también se trazan con *moveto* y *lineto*.

Con el comando *stroke* se imprimen todas las líneas trazadas anteriormente, haciendo visible la tabla.

```
% Líneas horizontales  
newpath  
100 680 moveto 500 680 lineto  
100 620 moveto 500 620 lineto  
100 560 moveto 500 560 lineto  
  
% Líneas verticales  
100 680 moveto 100 560 lineto  
300 680 moveto 300 560 lineto  
500 680 moveto 500 560 lineto  
  
stroke
```

- Texto

Ya que tenemos construida la estructura podemos poner el texto:

Los encabezados de columna ("Columna-1" y "Columna-2") se sitúan dentro de la primera fila y los valores de datos ("42" y "21") se posicionan de manera similar en la fila inferior, en coordenadas (130, 585) y (330, 585). Para ello utilizamos comandos *newpath* y *moveto*.

```
% Texto encabezado  
  
newpath  
130 645 moveto  
(Columna-1) show  
  
newpath  
330 645 moveto  
(Columna-2) show  
  
% Texto datos  
newpath  
130 585 moveto
```


(42) show

newpath

330 585 moveto

(21) show

showpage

% FINAL PÁGINA 1

Podemos ver el siguiente resultado:

Mi tabla de datos

Columna-1	Columna-2
42	21

Segunda página: Gráfica

En esta parte del código se construye una gráfica en la segunda página del documento.

- Configuración de texto

De nuevo seleccionamos la fuente Times-Roman y se escala a 26 puntos mediante *scalefont*. Luego, se posiciona el título "La imagen" utilizando *moveto* y *show*, lo que sirve como encabezado de la gráfica.

```
% Página 2 - GRÁFICA  
  
/Times-Roman findfont  
26 scalefont  
setfont  
newpath  
72 700 moveto  
(La imagen) show
```

- Dibujo de los ejes

Para construir el sistema de coordenadas, se trazan los ejes utilizando *moveto* y *lineto*, y luego se imprimen con *stroke*.

```
% Ejes  
newpath  
100 400 moveto 100 600 lineto % Eje Y  
100 400 moveto 300 400 lineto % Eje X  
0 setgray  
stroke
```

- Trazado de la gráfica en cresta

Dibujamos una línea poligonal que representa la evolución de los valores.

Se utiliza *moveto* para posicionarse en el primer punto y luego se van trazando líneas sucesivas a los siguientes puntos mediante *lineto*, creando así una forma de cresta irregular

Finalmente, *stroke* se encarga de dibujar la línea conectando todos esos puntos.

```
% Curva cresta  
  
newpath  
100 450 moveto  
130 430 lineto
```

```
160 480 lineto
190 420 lineto
220 500 lineto
stroke
```

- Dibujo de puntos sobre la curva

Para resaltar los vértices de la línea poligonal, se dibujan pequeños círculos centrados en cada uno de los puntos mencionados con un color rojo oscuro con *setrgbcolor*. Definimos el grosor del borde con *setlinewidth*.

Para cada punto, se utiliza *arc* para dibujar un círculo de radio 2, seguido de *fill* para rellenarlo con el color rojo.

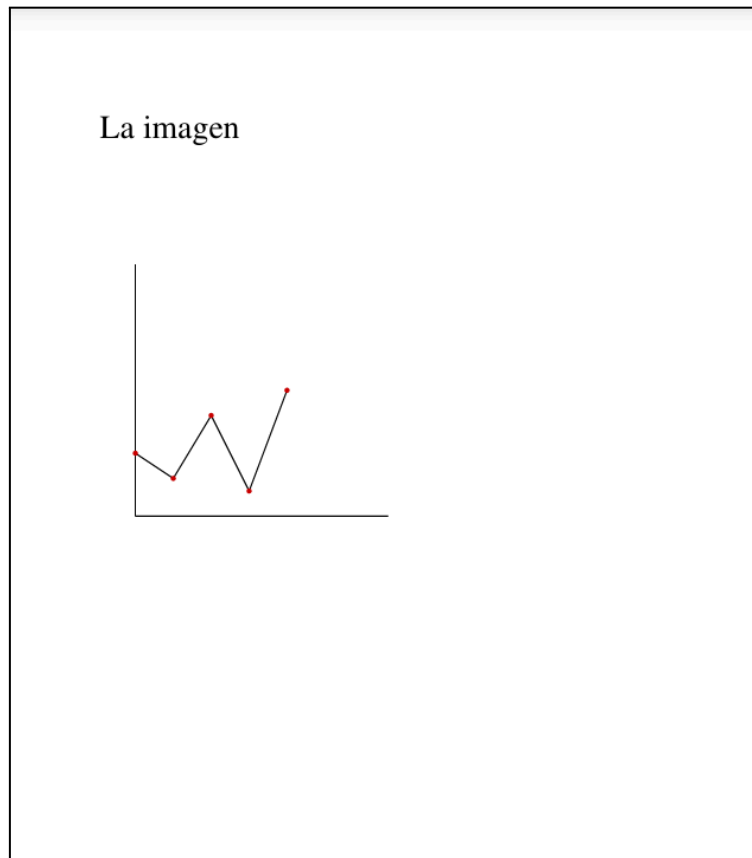
Finalmente el comando *showpage* se utiliza para indicar que esta es la segunda página del documento y debe mostrarse o imprimirse por separado.

```
% Puntos en cada vértice
0.8 0 0 setrgbcolor
5 setlinewidth

newpath 100 450 2 0 360 arc fill
newpath 130 430 2 0 360 arc fill
newpath 160 480 2 0 360 arc fill
newpath 190 420 2 0 360 arc fill
newpath 220 500 2 0 360 arc fill

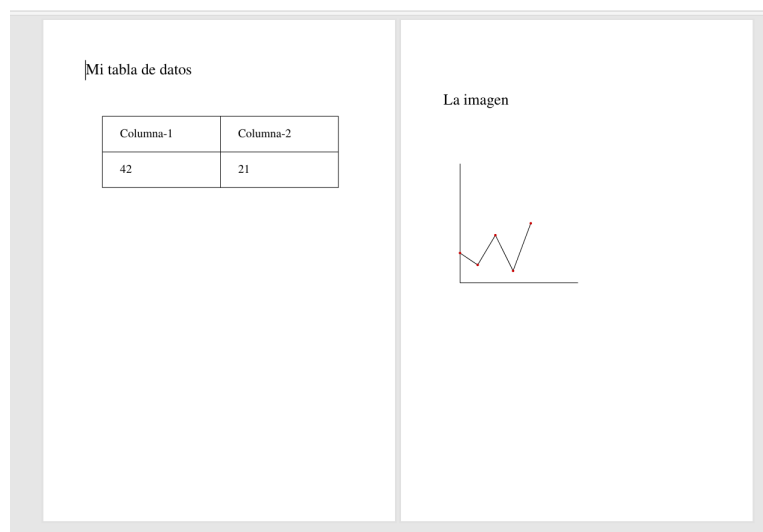
showpage
```

El resultado es el siguiente:



Ejercicio en dos páginas

Finalmente podemos ver que están en dos páginas diferentes:



6. Ejercicios adicionales

Dibujo de Super Mario (ejercicio adicional 1).

Este ejercicio consistirá en dibujar el personaje de videojuegos Super Mario en 8 bits en la parte inferior izquierda. Para ello, hemos seguido la siguiente lógica en el código:

- Definición de la unidad base (*tamCelda*)

Vamos a definir una constante con valor 10 para que cada píxel tenga 10 puntos de ancho y alto.

```
/tamCelda 10 def
```

- Creación del fondo

Vamos a asegurarnos de que el fondo de la imagen sea blanco configurando el color y dibujando y rellenando el rectángulo de fondo.

```
%Relleno el fondo de blanco
1 1 1 setrgbcolor
newpath
0 0 moveto
160 0 lineto
160 160 lineto
0 160 lineto
closepath
fill
```

- Definición del método *dibujaCelda*

Crearemos una función que nos permita dibujar celdas individuales en la cuadrícula. Cada celda la hemos definido por su posición *x* e *y* además de su color en *r g b*. Lo que haremos será extraer estos parámetros, calcular la posición de la celda multiplicando la coordenada de la cuadrícula por *tamCelda* usando *rlineto* para dibujar la celda con dimensiones iguales al *tamCelda*. Una vez cerrada la ruta con *closepath*, estableceremos el color mediante *setrgbcolor* y rellenamos la celda.

```
%Usamos dibujaCelda para dibujar un píxel en la cuadrícula
/dibujaCelda {
  /b exch def
  /g exch def
  /r exch def
  /y exch def
  /x exch def
```

```

newpath
x tamCelda mul y tamCelda mul moveto
tamCelda 0 rlineto
0 tamCelda rlineto
tamCelda neg 0 rlineto
closepath
r g b setrgbcolor
fill
} def

```

- Construcción de la figura de Super Mario

Procederemos a la creación de nuestra figura. Para ello, con el método `dibujaCelda` vamos creando todas las partes del cuerpo de Super Mario, desde la gorra hasta los zapatos. Lo hemos realizado dibujando fila a fila, trazándolo de la siguiente manera.

- **Gorra:** filas 15,14 y 13. En la fila 12 dibujamos dos celdas en rojo junto con el tono de piel para suavizar la transición entre la gorra y la cara.
- **Cara y bigote:** usaremos la fila 11 y 9 para dibujar la cara mientras que la fila 10 la usaremos para dibujar el bigote.
- **Cuerpo y tirantes:** usamos la fila 8 para dibujar el torso y la fila 7 los tirantes.
- **Piernas y zapatos:** la fila 6 y 5 dibujamos las piernas y en la fila 4 los zapatos.

```
%Construcción de Super Mario
```

```
%Gorra
```

```

6 15 1 0 0 dibujaCelda
7 15 1 0 0 dibujaCelda
8 15 1 0 0 dibujaCelda
9 15 1 0 0 dibujaCelda

```

```

5 14 1 0 0 dibujaCelda
6 14 1 0 0 dibujaCelda
7 14 1 0 0 dibujaCelda
8 14 1 0 0 dibujaCelda
9 14 1 0 0 dibujaCelda
10 14 1 0 0 dibujaCelda

```

```

4 13 1 0 0 dibujaCelda
5 13 1 0 0 dibujaCelda
6 13 1 0 0 dibujaCelda
7 13 1 0 0 dibujaCelda
8 13 1 0 0 dibujaCelda
9 13 1 0 0 dibujaCelda
10 13 1 0 0 dibujaCelda
11 13 1 0 0 dibujaCelda

```

```
4 12 1 0 0 dibujaCelda
```

```
5 12 1 0.8 0.6 dibujaCelda
6 12 1 0.8 0.6 dibujaCelda
7 12 1 0.8 0.6 dibujaCelda
8 12 1 0.8 0.6 dibujaCelda
9 12 1 0.8 0.6 dibujaCelda
10 12 1 0.8 0.6 dibujaCelda
11 12 1 0 0 dibujaCelda
```

%Cara

```
4 11 1 0.8 0.6 dibujaCelda
5 11 1 0.8 0.6 dibujaCelda
6 11 1 0.8 0.6 dibujaCelda
7 11 1 0.8 0.6 dibujaCelda
8 11 1 0.8 0.6 dibujaCelda
9 11 1 0.8 0.6 dibujaCelda
10 11 1 0.8 0.6 dibujaCelda
11 11 1 0.8 0.6 dibujaCelda
```

%Zona de bigote y parte inferior de la cara

```
4 10 1 0.8 0.6 dibujaCelda
5 10 1 0.8 0.6 dibujaCelda
6 10 0.4 0.2 0 dibujaCelda
7 10 0.4 0.2 0 dibujaCelda
8 10 0.4 0.2 0 dibujaCelda
9 10 0.4 0.2 0 dibujaCelda
10 10 1 0.8 0.6 dibujaCelda
11 10 1 0.8 0.6 dibujaCelda
```

%Cara

```
4 9 1 0.8 0.6 dibujaCelda
5 9 1 0.8 0.6 dibujaCelda
6 9 1 0.8 0.6 dibujaCelda
7 9 1 0.8 0.6 dibujaCelda
8 9 1 0.8 0.6 dibujaCelda
9 9 1 0.8 0.6 dibujaCelda
10 9 1 0.8 0.6 dibujaCelda
11 9 1 0.8 0.6 dibujaCelda
```

%Cuerpo

```
4 8 1 0 0 dibujaCelda
5 8 1 0 0 dibujaCelda
6 8 1 0 0 dibujaCelda
7 8 1 0 0 dibujaCelda
8 8 1 0 0 dibujaCelda
9 8 1 0 0 dibujaCelda
10 8 1 0 0 dibujaCelda
11 8 1 0 0 dibujaCelda
```

%Pantalones

```
4 7 0 0 1 dibujaCelda
5 7 0 0 1 dibujaCelda
6 7 0 0 1 dibujaCelda
7 7 0 0 1 dibujaCelda
8 7 0 0 1 dibujaCelda
9 7 0 0 1 dibujaCelda
```

```

10 7 0 0 1 dibujaCelda
11 7 0 0 1 dibujaCelda

%Piernas y zapatos
4 6 0 0 1 dibujaCelda
5 6 0 0 1 dibujaCelda
6 6 0 0 1 dibujaCelda
7 6 0 0 1 dibujaCelda
8 6 0.4 0.2 0 dibujaCelda
9 6 0.4 0.2 0 dibujaCelda
10 6 0.4 0.2 0 dibujaCelda
11 6 0.4 0.2 0 dibujaCelda

%Piernas
4 5 0 0 1 dibujaCelda
5 5 0 0 1 dibujaCelda
6 5 0 0 1 dibujaCelda
7 5 0 0 1 dibujaCelda
8 5 0.4 0.2 0 dibujaCelda
9 5 0.4 0.2 0 dibujaCelda
10 5 0.4 0.2 0 dibujaCelda
11 5 0.4 0.2 0 dibujaCelda

%Zapatos
4 4 0.4 0.2 0 dibujaCelda
5 4 0.4 0.2 0 dibujaCelda
6 4 0.4 0.2 0 dibujaCelda
7 4 0.4 0.2 0 dibujaCelda
8 4 0.4 0.2 0 dibujaCelda
9 4 0.4 0.2 0 dibujaCelda
10 4 0.4 0.2 0 dibujaCelda
11 4 0.4 0.2 0 dibujaCelda

```

- Detalles adicionales

Finalmente, vamos a añadir varios detalles, en donde dibujaremos la letra “M” justo en medio de la gorra, los ojos y la boca. Una vez dibujado, terminamos con el comando *showpage*.

```

%Dibujamos la M en la gorra
/Courier-Bold findfont 10 scalefont setfont
1 1 1 setrgbcolor
76 144 moveto
(M) show

%Dibujamos los ojos
6 12 0 0 0 dibujaCelda
9 12 0 0 0 dibujaCelda

%Dibujamos la boca
newpath
80 95 3 0 360 arc
closepath

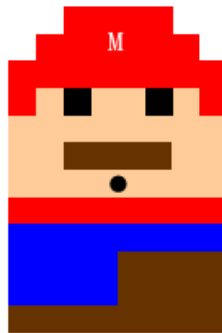
```



```
0 0 0 setrgbcolor
fill

showpage
```

El dibujo de Super Mario queda de la siguiente forma.



Tarjeta de “El Juego del Calamar” (ejercicio adicional 2).

Este ejercicio consistirá en realizar la parte frontal y trasera de la tarjeta de la famosa serie de Netflix “El juego del calamar” con el lenguaje PostScript. Para ello, vamos a realizar la parte frontal de la tarjeta en la primera página y la parte trasera en la segunda página. Recordemos que la tarjeta de la serie es la siguiente.



Primera página: Parte frontal

Indicaremos el fondo de la página, en donde definimos la ruta que abarca el tamaño de la página. Haremos que el diseño de la tarjeta sea rectangular (usaremos `rlineto` para construir líneas para hacer el rectángulo), además de poner un color marrón cartón para hacer más realista la tarjeta. Rellenamos y pintamos el contorno mediante *fill* y *stroke*.

A continuación, dibujaremos las formas geométricas de la tarjeta. Debe de ser un círculo en la parte izquierda, un triángulo en la parte central, y un cuadrado en la parte derecha. Para ello, dentro de un bloque *gsave* y *grestore* trasladamos las coordenadas para centrar las figuras en el interior de la tarjeta. Dibujaremos el círculo con el comando *arc*, el triángulo con el comando *lineto* y el cuadrado con *rlineto*. En todas las figuras deberemos de poner la coordenada de donde queremos dibujar con *moveto*. Finalmente, con *grestore* se restaura el estado gráfico original, terminando de esta manera las formas geométricas hechas dentro de la tarjeta.

```
%Parte frontal de la tarjeta en la primera página
/tarjeta {
  newpath
    0 0 moveto
    612 0 lineto
    612 792 lineto
    0 792 lineto
  closepath
  1 setgray fill

  %La tarjeta será rectangular
  newpath
    106 271 moveto
    400 0 rlineto
    0 250 rlineto
    -400 0 rlineto
  closepath
  0.8 0.63 0.21 setrgbcolor    %Vamos a usar un color marrón
  fill
  stroke

  %Formas geométricas
  gsave
    306 396 translate
    5 setlinewidth
    0 0 0 setrgbcolor

    %Círculo en la parte izquierda
    newpath
      -90 0 35 0 360 arc
    closepath
    stroke

    %Triángulo en la parte central
    newpath
      0 35 moveto
      -43.3 -35 lineto
      43.3 -35 lineto
    closepath
    stroke

    %Cuadrado en la parte derecha
    newpath
```

```

        65 -35 moveto
        70 0 rlineto
        0 70 rlineto
        -70 0 rlineto
    closepath
    stroke
    grestore
} def

```

Nuestra primera página con la parte frontal de la tarjeta quedaría de la siguiente forma.



Segunda página: Parte trasera

En la segunda página haremos la parte trasera de la tarjeta, en donde haremos en la parte izquierda un círculo que engloba una casa, donde el techo es un triángulo y la base un cuadrado, mientras que en la parte derecha pondremos el número de teléfono que sale en la serie. Debemos de tener en cuenta que seguiremos la estructura de fondo y tarjeta rectangular tal y como lo hemos hecho en la primera página con la parte frontal de la tarjeta.

Una vez dibujamos el círculo, dentro de un bloque *gsave* y *grestore*, ajustamos la posición de la casa para meterla justo en el círculo. Con *moveto* ajustamos la posición y creamos un cuadrado simulando la base de (con el comando *rlineto*) y un triángulo simulando el techo (con el comando *lineto*) de la casa. Luego, con *grestore* terminamos el bloque interno de la casa y regresamos la configuración de transformación de antes. Finalmente, en la parte derecha escribimos el número de la tarjeta.

```

%Parte trasera de la tarjeta en la segunda página
/tarjeta2 {

```

```

newpath
  0 0 moveto
  612 0 lineto
  612 792 lineto
  0 792 lineto
closepath
1 setgray fill

%La tarjeta será rectangular
newpath
  106 271 moveto
  400 0 rlineto
  0 250 rlineto
  -400 0 rlineto
closepath
0.8 0.63 0.21 setrgbcolor  %Igual que antes, usamos un color marrón
fill
stroke

gsave
  306 396 translate
  -80 0 translate

%Dibujo del círculo que envolverá la casa
newpath
  0 0 18.93 0 360 arc
closepath
2 setlinewidth  % Línea fina
0 setgray
stroke

%Dibujo de la casa
gsave
  0 -6.07 translate  % Ajusta la posición de la casa
  2 setlinewidth  % Línea fina para la casa
  0 setgray

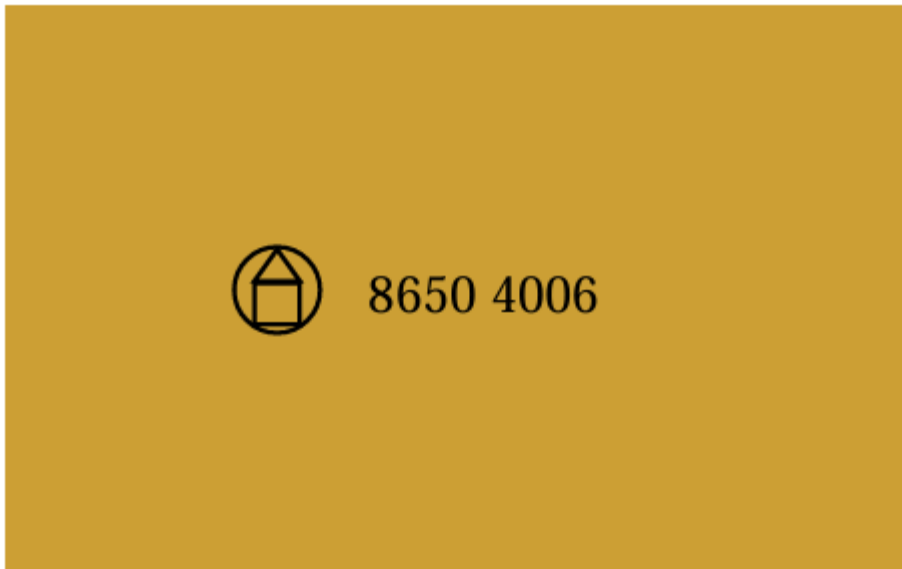
%Cuadrado
newpath
  -10 -9 moveto
  20 0 rlineto
  0 18 rlineto
  -20 0 rlineto
closepath
stroke

%Triángulo
newpath
  -10 10 moveto
  10 10 lineto
  0 24 lineto
closepath
stroke
grestore

```

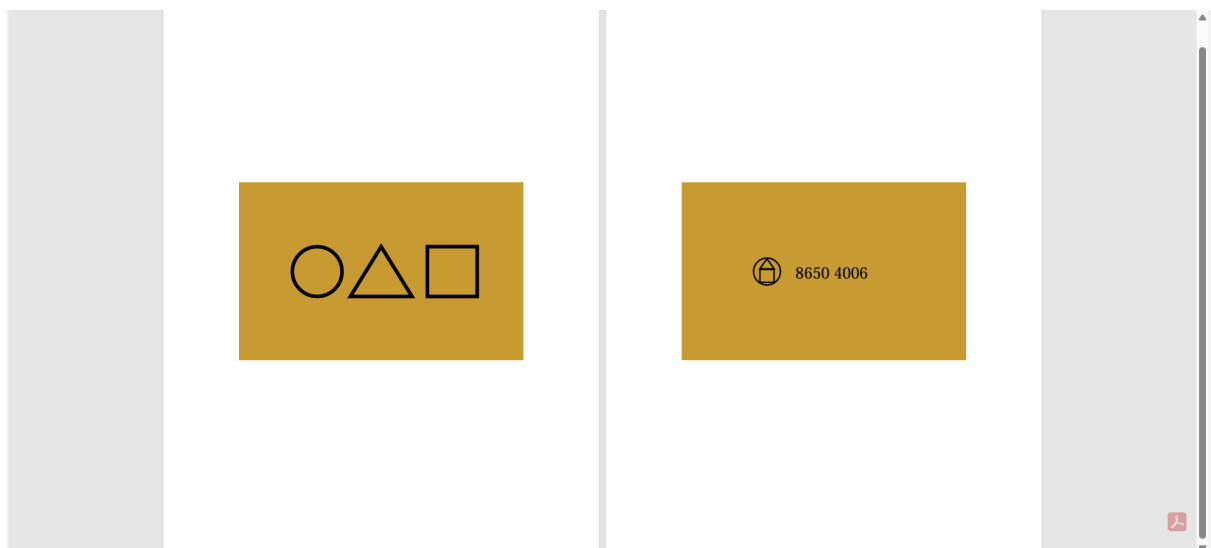
```
%Escribimos los números de la tarjeta
/Times-Roman findfont 24 scalefont setfont
newpath
  40 -10 moveto
  (8650 4006) show
stroke
grestore
} def
```

Nuestra segunda página con la parte trasera de la tarjeta quedaría de la siguiente forma.



Tarjeta en dos páginas

Finalmente podemos ver que están en dos páginas diferentes, viéndose de manera muy similar a la tarjeta de la serie.



Flor (ejercicio adicional 3).

Para este ejercicio adicional, vamos a representar una flor cuyos pétalos son ovalados.

- Título

Generamos el título con los comandos *findfont*, *scalefont* y *setfont* para configurar el tipo y tamaño de letra. Luego, con *moveto* se posiciona el texto en la parte superior de la página y se muestra con *show*.

```
%!PS
% Flor con 8 pétalos y centro

/Times-Roman findfont
20 scalefont
setfont
newpath
72 750 moveto
(Flor) show
```

- Pétalos

Para dibujar los pétalos, definimos el color mediante *setrgbcolor*. Cada pétalo se crea dentro de un bloque *gsave* - *grestore* para que podamos aislar las transformaciones aplicadas a ese pétalo del resto del dibujo.

Dentro de cada bloque, se traslada el sistema de coordenadas al centro de la flor con *translate*, se rota con *rotate* para orientar el pétalo en la dirección deseada, y se traslada un poco más hacia afuera con otro *translate*. Utilizamos posteriormente *scale* para poder transformar el círculo en una elipse (igual que en el primer ejercicio). El pétalo se dibuja con *newpath*, *arc* y se rellena con *fill*.

Este proceso se repite ocho veces, cambiando el ángulo de rotación en cada caso para distribuir los pétalos uniformemente alrededor del centro (0°, 45°, 90°, etc.).

```
% Pétalo 1
gsave
300 400 translate
0 rotate
0 50 translate
0.5 1.5 scale
newpath 0 0 30 0 360 arc fill
grestore

% Pétalo 2
gsave
300 400 translate
```

```
45 rotate
0 50 translate
0.5 1.5 scale
newpath 0 0 30 0 360 arc fill
grestore
```

```
% Pétalo 3
gsave
300 400 translate
90 rotate
0 50 translate
0.5 1.5 scale
newpath 0 0 30 0 360 arc fill
grestore
```

```
% Pétalo 4
gsave
300 400 translate
135 rotate
0 50 translate
0.5 1.5 scale
newpath 0 0 30 0 360 arc fill
grestore
```

```
% Pétalo 5
gsave
300 400 translate
180 rotate
0 50 translate
0.5 1.5 scale
newpath 0 0 30 0 360 arc fill
grestore
```

```
% Pétalo 6
gsave
300 400 translate
225 rotate
0 50 translate
0.5 1.5 scale
newpath 0 0 30 0 360 arc fill
grestore
```

```
% Pétalo 7
gsave
300 400 translate
270 rotate
0 50 translate
0.5 1.5 scale
newpath 0 0 30 0 360 arc fill
grestore
```

```
% Pétalo 8
gsave
300 400 translate
315 rotate
```

```
0 50 translate
0.5 1.5 scale
newpath 0 0 30 0 360 arc fill
grestore
```

- Centro de la flor

El centro de la flor se dibuja con un círculo sencillo utilizando *newpath*, *arc* y *fill*, esta vez con un color verde definido con *setrgbcolor*.

```
% Centro de la flor
newpath
300 400 25 0 360 arc
0 0.5 0 setrgbcolor
fill

showpage
```

Finalmente, utilizamos *showpage* para mostrar el dibujo como se puede ver a continuación:

Flor



Tarjeta Feria de Empleo (ejercicio adicional 4)

En este ejercicio vamos a realizar una tarjeta que promociona la Feria Internacional de Empleo de Granada, y que indica el lugar y fecha.

- Fondo de la Tarjeta

Para realizar el fondo de la tarjeta, dibujamos un rectángulo de fondo en la parte superior de la página, con el comando *moveto* y lo rellenamos con un color azul claro que se pinta con *fill*.

```
%!PS

% Fondo de la Tarjeta
newpath
180 560 moveto
380 0 rlineto
0 160 rlineto
-380 0 rlineto
closepath
0.6 0.85 1.0 setrgbcolor    % Azul Claro
fill
```

- Logo de la Tarjeta

Para realizar el logo de la tarjeta, se dibuja el logotipo de la tarjeta a la izquierda, formado por un círculo blanco de fondo y dos formas en arco superpuestas (*arc*) con diferentes tonos de azul que se establecen con *setrgbcolor*.

```
% Logo de la Tarjeta
newpath
220 660 30 0 360 arc
1 1 1 setrgbcolor    % Círculo blanco
fill

% Diferentes tipos de azul
newpath
220 660 25 0 180 arc closepath
0.2 0.5 0.9 setrgbcolor
fill

newpath
220 660 20 180 360 arc closepath
0.0 0.7 1.0 setrgbcolor
fill
```

- Texto de la Tarjeta

Aquí nos encargamos de mostrar el texto principal de la tarjeta, como el título del evento, el lugar y la fecha. Usamos el comando *show* para escribir cada línea de texto en pantalla, y antes de eso, *moveto* posiciona el punto de inicio donde aparecerá esa línea. Además, se cambia el tipo y tamaño de letra con *findfont* y *scalefont*, y se establece el color del texto en negro con *setrgbcolor*.

```
% Texto de la Tarjeta
/Helvetica findfont 24 scalefont setfont
0 0 0 setrgbcolor
270 690 moveto
(Feria Internacional) show

/Helvetica findfont 24 scalefont setfont
270 660 moveto
(de ) show

/Helvetica-Bold findfont 24 scalefont setfont
(Empleo) show

/Helvetica findfont 24 scalefont setfont
270 630 moveto
(de Granada) show

% Lugar y Fecha
/Helvetica findfont 16 scalefont setfont
0 0 0 setrgbcolor
270 600 moveto
(Palacio de Congresos) show

/Helvetica findfont 16 scalefont setfont
270 580 moveto
(Mayo 2025) show

showpage
```

Finalmente, utilizamos *showpage* para mostrar el dibujo como se puede ver a continuación:



Feria Internacional de **Empleo** de Granada

Palacio de Congresos
Mayo 2025

7. Bibliografía

<https://swad.ugr.es/swad/tmp/Vi/pzyHGnbASeHAdtUVI6aF9SRxcH0yQOUFtThFTGmrs/P4-postscript.pdf>

https://es.wikipedia.org/wiki/Curva_de_B%C3%A9zier