

Unidades

- ¿Qué son las unidades?
 - Las unidades son **grupos de funciones o procedimientos compilados** que pueden ser llamados y utilizados desde cualquier programa en Pascal sin necesidad de escribirlos de nuevo.
 - Las unidades **no son ejecutables por si solas**, dependen del programa que las llama para poder realizar su función.
 - FreePascal incorpora algunas unidades que contienen una gran cantidad de rutinas integradas. Las **principales** son:
 - crt: presentación por pantalla
 - dos, linux: acceso al sistema operativo
 - graph: representación gráfica
 - printer: utilización de la impresora

Escritura de una unidad

- Una unidad consta de:
 - Parte de Interfaz
 - Consta de toda la **información pública** que necesitan los programas que la utilizan. Esta información suele ser la siguiente:
 - El tipo de datos que se crea en la unidad.
 - Constantes u otra información auxiliar que se necesita para crear el tipo de datos
 - Cabeceras de todos los procedimientos y funciones asociadas a la unidad.
 - Parte de Implementación
 - Consta de la **información necesaria para la implementación** de las funciones y procedimientos de la unidad. Esta información suele ser la siguiente:
 - Cualquier declaración que sea necesaria para los procedimientos o funciones pero no para los programas que los utilizan
 - La codificación de los procedimientos y funciones

Estructura y uso de una unidad

• Estructura

```

unit nombre_unidad;
interface
    declaraciones_públicas_de constantes
    declaraciones_públicas_de tipos
    declaraciones_públicas_de variables
    declaraciones_públicas_de proc. y func.
implementation
    declaraciones_privadas
    codificación_de proc. y func.
begin
    sentencias_de_iniciación
end
  
```

```

unit unit_complejo;
interface
procedure comp_leer(var x,y:real);
procedure comp_escribir(x,y:real);
procedure comp_sumar(x1,y1,x2,y2:real;var x,y:real);
implementation
procedure comp_leer(var x,y:real);
begin
    writeln('Introduzca la parte real');
    readln(x);
    writeln('Introduzca la parte imaginaria');
    readln(y)
end;
procedure comp_escribir(x,y:real);
begin
    writeln(x,' + ',y,'i')
end;
procedure comp_sumar(x1,y1,x2,y2:real;var x,y:real);
begin
    x:=x1+x2;
    y:=y1+y2;
end;
begin
    writeln('Libreria de numeros complejos');
end.
  
```

• Uso de una unidad

Se inserta en el programa, tras su cabecera:

```
uses nombre_unidad
```

en el caso de que se utilicen varias :

```
uses nombre_unidad1, ..., nombre_unidadn;
```

```

program suma_complejo;
uses unit_complejo;
var
    x1,y1,x2,y2,x,y:complejo;
begin
    comp_leer(x1,y1);
    comp_leer(x2,y2);
    comp_sumar(x1,y1,x2,y2,x,y);
    comp_escribir(x,y);
end.
  
```

Compilación de una unidad

- Una vez escrita la unidad es necesario **compilarla** para poder utilizarla
- Pasos para la compilación de una unidad
 - Almacenar la unidad en un archivo cuyo nombre debe ser igual al nombre de la unidad y extensión .pas .
 - Compilar dicho archivo generándose (linux):
 - Un archivo con el mismo nombre y extensión .ppu que contiene información acerca de la unidad para el compilador.
 - Un archivo con extensión .o que se unirá al programa principal.
- El programa que utilice la unidad debe tener justo después de la cabecera un enunciado de la forma:
 - **uses** *nombre _unidad*
- El programa que utiliza la unidad **se compila y ejecuta normalmente**. Los archivos .ppu y .o deben de estar en el directorio del programa principal (aunque esto puede cambiarse).