

## Bucles

Fundamentos de la programación y la informática  
Grado en ingeniería aeroespacial en vehículos aeroespaciales 2019-2020  
Escuela Técnica Superior de Ingeniería de Telecomunicación  
Universidad Rey Juan Carlos

---

## Práctica 7.1. Máximo y Mínimo desde teclado

Escribe un programa en un fichero llamado `~/fpi/practica07/maximo_minimo.pas` que vaya pidiendo números reales positivos al usuario. Cuando se introduzca un 0, el programa debe terminar indicando antes el máximo y el mínimo de los números introducidos.

Si el usuario introduce algo diferente a lo solicitado (la entrada no es un número o no es un número real positivo) el programa debe ignorarlo, escribir un mensaje de error y seguir funcionando con normalidad, solicitando un nuevo número.

Si el usuario no introduce ningún número positivo antes del 0, el programa terminará pero no debe mostrar el máximo ni el mínimo, sino un mensaje de error específico.

Ejemplo de funcionamiento:

```
./maximo_minimo
Escribe un número positivo (0 para terminar): jj
Valor no válido.
Escribe un número positivo (0 para terminar): 12.5
Escribe un número positivo (0 para terminar): -1
Valor no válido.
Escribe un número positivo (0 para terminar): 8.0
Escribe un número positivo (0 para terminar): 23.9
Escribe un número positivo (0 para terminar): 0
Máximo: 23.9
Mínimo: 8.0
```

Observaciones

- Tendrás que leer una cadena con el procedimiento `readln()` y convertirlo en número con el procedimiento `val`.
- Naturalmente, el programa debe estar dividido en subprogramas. Escribe (al menos) un subprograma que solicite un valor al usuario, hasta que sea un número correcto. Y otro subprograma que llame al procedimiento anterior y haga los cálculos, hasta que el número sea 0.

## Práctica 7.2. Ángulo de ataque

En este ejercicio practicarás el uso de bucles while.

Tenemos un simulador de vuelo. Estamos trabajando en el control del ángulo de ataque. Nuestro programa recibe

1. El ángulo actual del ala.
2. El ángulo objetivo, que es el ángulo que tenemos que alcanzar.
3. Una constante llamada *tolerancia*.

Estas tres magnitudes están en grados.

Nuestro programa tiene que conseguir que el ángulo actual se acerque *lo suficiente* al ángulo objetivo. Para ello deberá invocar a dos procedimientos:

- `incrementa_angulo`

Es un procedimiento que recibe el ángulo actual y la tolerancia, e incrementa el ángulo. No sabemos cuántos grados lo va a incrementar, es una cantidad desconocida por nosotros, la simulamos con un número aleatorio. Considera esta función una *caja negra* (en sentido informática). Esto es, no te preocupes de cómo funciona por dentro, solo de lo que devuelve.

- `decrementa_angulo`

Análogo al procedimiento anterior.

Escribe un procedimiento llamado *nivela* que, si es necesario, invoque a los procedimientos *incrementa\_angulo* y *decrementa\_angulo* hasta que la distancia entre el ángulo actual y el objetivo esté dentro del margen de tolerancia. Son aceptables los ángulos cuya diferencia con el ángulo objetivo sea menor o igual que el margen de tolerancia.

Ten en cuenta que es posible que el incremento o decremento producido sea mayor del necesario, y por tanto sea necesario rectificar.

Los procedimientos *incrementa\_angulo* y *decrementa\_angulo* ya están implementados. También un procedimiento *inicia\_valores* que da valor inicial a `angulo_actual` y `angulo_objetivo`. Casi todo el problema está resuelto, lo tienes disponible en este enlace.

Lo que debes hacer es:

1. Copiar el programa en tu cuenta, con el nombre `~/fpi/practica07/angulo_ataque.pas`
2. El programa indenta con tabuladores, si lo prefieres puedes convertirlos en espacios de alguna de estas formas:
  - Usando el editor
  - Escribiendo en un terminal

```
expand -t4 angulo_ataque_01.pas > nuevo_fichero.pas
```

Esto crea un nuevo fichero donde los tabuladores ahora son 4 espacios. Cambiando `-t4` por `-t8`, escribiría 8 espacios por cada tabulador. Observa que seguramente querrás renombrar el nuevo fichero, para que pase a tener el nombre original

- Si quisieras hacer el paso inverso, sería todo igual, pero con la orden *unexpand*

Recuerda que en ningún caso puedes mezclar tabuladores y espacios.

### 3. Escribir el procedimiento *nivela*

El resto de detalles sobre el uso de los procedimientos *inicia\_valores*, *incrementa\_angulo* y *decrementa\_angulo*, consúltalos en el mismo fuente del programa.

## Revisión de los nombres de los ficheros

Ejecuta `~/mortuno/revisa practicas fpi` para comprobar que los nombres de los programas son los correctos.