

Me lo repites hasta que....

En muchas situaciones, los programas deben repetir ciertas instrucciones, bien hasta que se cumpla determinada condición o bien porque deben ejecutarse un número determinado de veces.

Por ejemplo, podemos necesitar comprobar un dato que le hemos pedido al usuario y no continuar el proceso hasta que el dato sea correcto.

Para expresar estas situaciones utilizaremos las **sentencias iterativas**.

Las instrucciones iterativas, bucles o repeticiones pueden expresarse de varias formas: mediante la expresión `Mientras condición` o utilizando la expresión `Repetir un número de veces` o hasta que se de una condición.

La sintaxis en pseudocódigo que usaremos para ambas expresiones es la siguiente.

```
Mientras (condición) hacer
    <bloque de instrucciones>
Repetir (un número de veces) o hasta que (condición) [usando <n>]
    <bloque de instrucciones>
```

Supongamos, por ejemplo, la situación anterior en la que realizamos una verificación de entrada de datos. El pseudocódigo podría ser el siguiente.

```
escribir("Introduce un número: ")
leer(numero)
mientras numero no sea un número
    escribir("Debes introducir un dato numérico")
    leer(numero)
```

Consideremos, sin embargo, que queremos contar cuantos números divisibles por 3 hay entre 1 y 100. Entonces lo que tendremos es que verificar esa condición para todos los números del 1 al 100. Es decir:



```
ContadorDivPor3=0
repetir 100 veces usando numero
    si numero es divisible por 3
        ContadorDivPor3= ContadorDivPor3+1

escribe("Hay ", ContadorDivPor3, " divisibles por 3)
```

Sea, ahora, un proceso que pide al usuario 10 números y determina cuántos de los números introducidos son pares. El pseudocódigo podría ser el siguiente.

```
escribir(Introduce 10 números enteros)
contadorDePares=0
repetir 10 veces
    escribir("Numero: ")
    leer(numero)
    si el numero es par entonces
        contadorDePares= contadorDePares+1

escribe("De los numeros introducidos son pares: ",
contadorDePares)
```

Para practicar con las sentencias iterativas, analizar y diseñar los algoritmos que se indican en la hoja correspondiente de ejercicios.

