



---

# Sesión 3

Aprende C desde cero

- Estructuras iterativas
  - for
  - while
  - do while

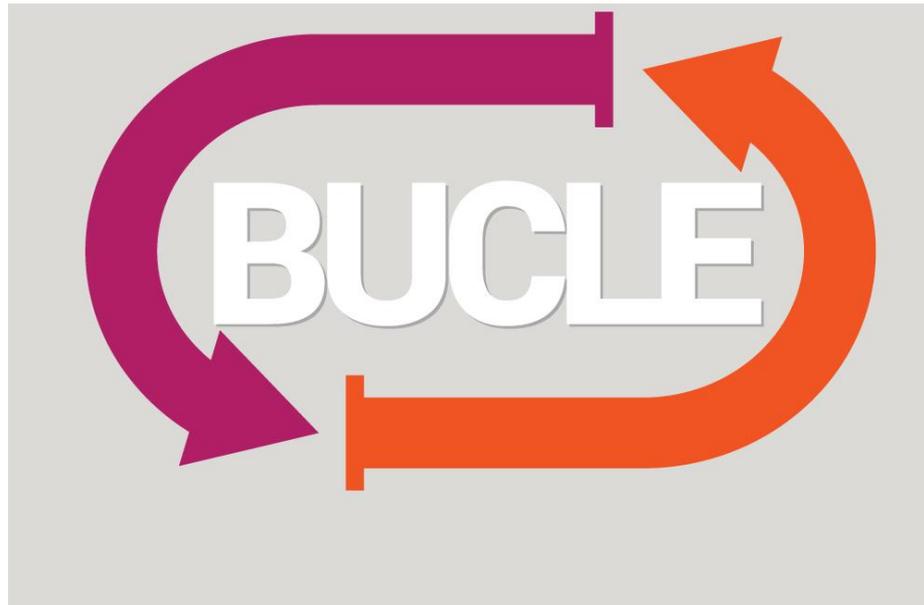
# Estructuras de control iterativas

# Código iterativo

Código iterativo es aquel que se repite un cierto número de veces

Ejemplos:

- Mostrar los n primeros números
- Repetir una operación aritmética varias veces
- Recorrer los elementos de una lista



# Código iterativo

Imaginemos que queremos un código que muestre los primeros 3 números positivos por pantalla.

```
#include <stdio.h>

int main(void)
{
    int i = 0;
    printf("%d,", i);
    i++; //Incrementa el valor de i en 1
    printf("%d,", i);
    i++; //Equivalente a i = i + 1
    printf("%d\n", i);

    return 0;
}
```

# Código iterativo

¿Y si queremos que muestre los 5 primeros?

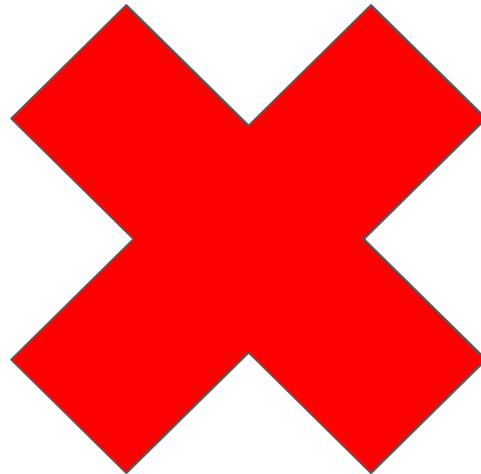
```
#include <stdio.h>

int main(void)
{
    int i = 0;
    printf("%d,", i);
    i++; //Incrementa el valor de i en 1
    printf("%d,", i);
    i++; //Equivalente a i = i + 1
    printf("%d,", i);
    i++;
    printf("%d,", i);
    i++;
    printf("%d\n", i);

    return 0;
}
```

# Código iterativo

¿Y si queremos que muestre los 1000 primeros?



**¡¡Pues para esto sirven las estructuras de control iterativas!!**

Bucle for

# for

Permite la ejecución de un código un determinado número de veces:

```
for (contador = inicio; contador < fin; incremento)
{
    código();
}
```

Dónde:

- **for** es la palabra clave de esta estructura.
- **contador** es una variable para controlar las veces que se ejecuta el código.
- **inicio** es el valor inicial de contador.
- **fin** es el valor final de contador.
- **incremento** es una expresión matemática que se aplica sobre el contador.

Importante:

- Si el inicio es menor que el fin, acordaros que el incremento debe ser positivo y viceversa, si el inicio es mayor que fin el incremento debe ser negativo.
- La condición de **contador < fin** puede utilizar cualquier condición explicada en el tema pasado para la estructura if

# for

¿Cómo quedaría el ejemplo anterior con un bucle for?

```
#include <stdio.h>

int main(void)
{
    int inicio = 0;
    int fin = 1000;
    int i;

    for (i = inicio; i < fin; i=i+1)
    {
        printf("%d,", i);
    }

    return 0;
}
```

# for

¿Cómo podríamos mostrar solamente los números **impares** dentro de los 1000 primeros números sin utilizar la estructura if? ¿Y con ella?

```
#include <stdio.h>

int main(void)
{
    int inicio = ???;
    int fin = 1000;
    int i;

    for (i = inicio; i ??? fin; i=i+???)
    {
        printf("%d,", i);
    }

    return 0;
}
```

Bucle while

# While

Permite la ejecución de un código un determinado número de veces (*a priori* desconocidas):

```
while(condición) {  
    código;  
}
```

Dónde:

- **while** es la palabra clave de esta estructura.
- **condición** es una comparación entre valores que se resolverá como verdadero o falso.
- **código** es el bloque de instrucciones que se ejecutarán si la condición es verdadera.

Ejemplo:

```
int main() {  
    float num = 12;  
    float umbral = 0.005;  
    int veces = 0;  
    while (num > umbral ) {  
        num = num / 2;  
        veces = veces + 1;  
    }  
    return 0;  
}
```

# Comparadores de condición (recordatorio)

Representar	Marca de formato
Mayor	>
Menor	<
Igual	==
Distinto	!=
Mayor o igual	>=
Menor o igual	<=
Negación	!

# while

¿Cómo quedaría el ejemplo inicial del tema con un bucle while?

```
#include <stdio.h>

int main(void)
{
    int inicio = 0;
    int fin = 1000;

    int i = inicio;
    while (i < fin)
    {
        printf("%d,", i);
        i = i + 1; //i++
    }

    return 0;
}
```

Ejercicio

# Ejercicio: Calculadora 3.0

Partiendo del programa en C de la sesión 2, donde dados dos números calculábamos su suma, resta, producto o división dependiendo la operación elegida, modifícalo para que haga multiplicaciones y divisiones sin los operadores `*` ni `/`.

Para ellos tendremos que utilizar bucles.

Además, en cada iteración deberá mostrar el subtotal de la operación

Ejemplo:

Dados los número 3 y 4 y elegida la multiplicación, la salida del programa debería ser:

```
Bienvenido/a a "Calcooladora v3.0".  
  
Primera cifra: 3.00  
Segunda cifra: 4.00  
Operación: Multiplicación  
  
Subtotal en la iteración 1: 6  
Subtotal en la iteración 2: 9  
  
Resultado: 12  
  
¡Adiós!
```

## Recuerda que puedes:

Interactuar con otros alumnos en el foro del curso:

<http://mooc.uji.es/mod/forum/view.php?id=1654>

Acceder a todos los materiales en el repositorio:

<https://siserte@bitbucket.org/siserte/repositoriocursoc.git>

Contactar con los profesores:

- [adcastel@uji.es](mailto:adcastel@uji.es)
- [siserte@uji.es](mailto:siserte@uji.es)

# do - while

Una variante del bucle while

```
do{  
    código;  
}while (condición);
```

Dónde:

- **do** es la palabra clave que inicia la estructura
- **while** es la palabra clave de esta estructura.
- **condición** es una comparación entre valores que se resolverá como verdadero o falso.
- **código** es el bloque de instrucciones que se ejecutarán si la condición es verdadera.

Ejemplo: menú de opciones

```
do{  
    printf("Elige una opción: (0 - Suma, 1 - Resta)"\n);  
    scanf("%d", &opcion);  
}while((opcion < 0) || (opcion > 1));
```