

Los números reales en coma flotante se convierten a binario en tres pasos:

1. Convertir al sistema binario
2. Escribir en notación científica
3. Seguir el standard IEEE754 para 32 bits

Por una parte la **parte entera** del número real se convierte a binario y por otra la **parte decimal**, según el algoritmo que se explica en el vídeo

<https://www.youtube.com/watch?v=VMcypTxcbvY>.

En esta práctica se debe leer por un lado la **parte entera** del número real, prescindiendo del signo, y por otro la **parte decimal**, convirtiendo cada una de estas partes a binario, y escribiendo en la pantalla los bits de los números binarios correspondientes, siguiendo el algoritmo descrito **en el vídeo**, que es distinto para la parte entera y para la parte decimal.

En la práctica anterior se utilizaba la instrucción if, mientras que en ésta se utilizará la instrucción do while.

```
parte entera numero float ? 134
bit0=0
bit1=1
bit2=1
bit3=0
bit4=0
bit5=0
bit6=0
bit7=1
parte decimal numero float ? 0.3125
bit-1=0
bit-2=1
bit-3=0
bit-4=1

Process returned 0 (0x0)   execution time : 36.541 s
Press any key to continue.
```

Figura 1. Ejemplo de ejecución del programa

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    float num, num2;
    float dec;
    int bit;
    int numbit=0;

    // parte entera numero float

    do{
        printf("Parte entera numero float: ");
        scanf("%f", &num);
    }while(num<0);

    num = (int)num;

    do{
        num/=2;
        dec=num-(int)num;
        if(dec >= 0.5)
            bit=1;
        else
            bit=0;
        printf("Bit %i = %i\n", numbit, bit);
        numbit++;
    }while(num>=1);

    // parte decimal numero float

    do{
        printf("Parte decimal numero float: ");
        scanf("%f", &num2);
    }while(num2<0 || num2>=1);

    numbit=-1;

    do{
        num2=num2*=2;
        if(num2<1)
            bit=0;
        else{
            bit=1;
            num2-=1;
        }
        printf("Bit %i = %i\n", numbit, bit);
        numbit--;
    }while(num2!=0);

    return 0;
}
```