

# Proyecto 2

## Conversiones y Errores

### Métodos Numéricos para Ingeniería

- Hacer una función en Octave llamada **convierteReal(n,m,e,b)**, que reciba cuatro números.
- El primero, llamado **n**, es un número real en decimal (o binario) que se convertirá a binario (o decimal). El segundo, llamado **m**, es el número de bits que tiene la mantisa. El tercero, llamado **e**, es el número de bits que tiene el exponente. El cuarto, llamado **b**, es 0 cuando se quiere convertir de decimal a binario y 1 cuando es de binario a decimal.
- Si **b=1** entonces **n** es binario y tiene que darse como un string (entre comillas). Además, **n** tiene que estar dado en formato mantisa y exponente normalizados (sin punto decimal).
- La representación en binario siempre tiene 1 bit de signo para el número y 1 bit de signo para el exponente, pero no están incluidos en **m** y **e**. Por ejemplo, si **m=8** y **e=4**, entonces el número completo en binario (la palabra) tendrá 14 bits (8+4+1+1).
- La función debe regresar la conversión en una variable llamada **R** y el error cometido en la conversión, sólo en caso de que sea de decimal a binario, en una variable llamada **E**. Si la conversión es de binario a decimal el error no importa y puede regresar lo que sea (pero debe regresar algo porque de no ser así pone un *warning*).
- Si **b=0** entonces **R** debe contener el número en binario y deberá ser un string. Desde luego, **R** debe estar en formato mantisa y exponente normalizados (sin punto decimal).
- El esqueleto de la función sería:

```
function [R, E] = convierteReal(n, m, e, b)
```

```
...
```

```
end
```

#### Ejemplo:

##### Decimal a binario:

Entrada: n=13.45, e = 3, m = 6

Salida: R="00100110101" y E=-0.2

##### Binario a decimal:

Entrada: n = "100110110010010", e=4, m=9

Salida: R=-50.25 y E="No importa"