

# Programación 1

## Tema 8

---

Reales



Escuela de  
Ingeniería y Arquitectura  
**Universidad Zaragoza**





# Índice

---

- Tipos reales
  - Dominio valores
  - Representación
  - Operaciones
  - La biblioteca `cmath`
  - Limitaciones
- Problemas

# Dominio de valores

---

- Subconjunto de  $\mathbb{R}$
- Acotado superior e inferiormente
- Precisión finita
  - Discretización
- Dominio de valores dependiente de la representación

# Representación

---

## □ Externa

- Arábica decimal  
(con punto decimal en lugar de coma)
- $\langle \text{constanteReal} \rangle ::= [ "+" | "-" ]$   
 $\langle \text{dígito} \rangle \{ \langle \text{dígito} \rangle \} "." \langle \text{dígito} \rangle \{ \langle \text{dígito} \rangle \}$   
 $[ ("E" | "e") [ "+" | "-" ] \langle \text{dígito} \rangle \{ \langle \text{dígito} \rangle \} ]$

## □ Ejemplos

- 2.5    3.1415926535    -2.0    0.75
- 6.022e23    1.6726e-27    9.1093e-31



# Representación

---

- Interna

- IEEE 754

- $\text{Mantisa} \times B^{\text{exponente}}$

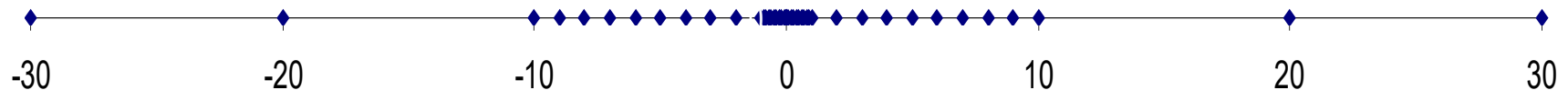
- $B = 2$

- Mantisa y exponente con un número concreto de bits

# Dominio de valores

## Ejemplo (en base 10)

- **Mantisa** =  $\{-9, -8, \dots, -2, -1, 0, 1, 2, \dots, 8, 9\}$
- **Exponente** =  $\{-1, 0, 1\}$
- **Base** = 10
- $-90, -80, -70, -60, -50, -40, -30, -20, -10,$   
 $-9, -8, -7, -6, -5, -4, -3, -2, -1,$   
 $-0.9, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, -0.2, -0.1,$   
 $0,$   
 $0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9,$   
 $1, 2, 3, 4, 5, 6, 7, 8, 9,$   
 $10, 20, 30, 40, 50, 60, 70, 80, 90$



# Reales en C++

## □ Compilador GNU GCC

### ■ float

- $-3,40282 \times 10^{38} \dots +3,40282 \times 10^{38}$
- mínimo valor absoluto mayor que cero:  $1,17549 \times 10^{-38}$
- **6 dígitos de precisión (decimal)**

### ■ double

- $-1,79769313 \times 10^{308} \dots +1,79769313 \times 10^{308}$
- mínimo valor absoluto mayor que cero:  $2,22507386 \times 10^{-308}$
- **15 dígitos de precisión (decimal)**

### ■ long double

- $-1,1897315 \times 10^{4932} \dots +1,1897315 \times 10^{4932}$
- mínimo valor absoluto mayor que cero:  $3,36210314 \times 10^{-4932}$
- **18 dígitos de precisión (decimal)**



# Operaciones

---

- Aritméticos:  $+$ ,  $-$ ,  $*$ ,  $/$
- Relacionales:  $==$ ,  $!=$ ,  $<$ ,  $<=$ ,  $>$ ,  $>=$
- Funciones aritméticas de la biblioteca estándar `cmath`



# Biblioteca estándar cmath

---

- Funciones trigonométricas
  - **double sin(double a)**
    - Devuelve el valor de  $\sin a$ , con  $a$  en radianes
  - **double cos(double a)**
    - Devuelve el valor de  $\cos a$ , con  $a$  en radianes
  - **double tan(double a)**
    - Devuelve el valor de  $\tan a$ , con  $a$  en radianes

# Biblioteca estándar cmath

---

- Funciones exponencial y logarítmicas
  - **double exp(double x)**
    - Devuelve el valor de  $e^x$
  - **double log(double x)**
    - Devuelve el valor de  $\ln x$
  - **double log10(double x)**
    - Devuelve el valor de  $\log_{10} x$
  - **double log2(double x)**
    - Devuelve el valor de  $\log_2 x$

# Biblioteca estándar cmath

---

- Funciones que calculan raíces y potencias
  - **double** sqrt(**double** x)
    - Devuelve el valor de  $\sqrt{x}$
  - **double** pow(**double** x, **double** y)
    - Devuelve el valor de  $x^y$
- Valor absoluto
  - **double** abs(**double** x)
    - Devuelve el valor  $|x|$

# Biblioteca estándar cmath

---

- Funciones de aproximación y redondeo a valores reales sin decimales
  - **double floor(double x)**
    - Devuelve  $\lfloor x \rfloor$ , el mayor real sin decimales que sea menor o igual que  $x$ .
  - **double ceil(double x)**
    - Devuelve  $\lceil x \rceil$ , el menor real sin decimales que sea mayor o igual que  $x$ .
  - **double round(double x)**
    - Devuelve el real sin decimales más próximo a  $x$ .
  - **double trunc(double x)**
    - Devuelve  $\lfloor x \rfloor$ , el real resultante de eliminar los decimales de  $x$ .

# Problema

---

Escribir un programa C++ que, dadas las coordenadas de dos puntos del plano, indique la distancia entre ellos

Introduzca las coordenadas de un punto: **0.0 0.0**

Introduzca las coordenadas de otro punto: **1.0 3.0**

La distancia entre los puntos es de 3.1623 unidades.

# Problema

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
...
/*
 * Programa que solicita al usuario las
 * coordenadas reales de dos puntos y le
 * informa de la distancia euclídea ente
 * los mismos.
 */
int main() {
    ...
}
```

# Problema

```
int main() {  
    // Petición de los datos  
    cout << "Introduzca las coordenadas de un punto: ";  
    double x1, y1;  
    cin >> x1 >> y1;  
    cout << "Introduzca las coordenadas de otro punto: ";  
    double x2, y2;  
    cin >> x2 >> y2;  
  
    // Cálculo de la distancia  
    double dist = distancia(x1, y1, x2, y2);  
  
    // Escritura de resultados;  
    cout << "La distancia entre los puntos es de " << fixed  
        << setprecision(4) << dist << " unidades." << endl;  
    return 0;  
}
```

# Problema

```
/*  
 * Pre: ---  
 * Post: Devuelve la distancia euclídea  
 * entre los puntos (x1, y1) y  
 * (x2, y2).  
 */  
double distancia(double x1, double y1,  
                 double x2, double y2) {  
    double dX = x1 - x2;  
    double dY = y1 - y2;  
    return sqrt(dX * dX + dY * dY);  
}
```





# Limitaciones

---

- Desbordamiento

# Desbordamiento

```
#include <iostream>
#include <cmath>
using namespace std;

/*
 * Programa que muestra el resultado de un desbordamiento
 * al trabajar con datos de tipo real.
 */
int main() {
    double x = DBL_MAX;
    cout << "x = " << x << endl;
    double y = 1.5 * x;
    cout << "y = " << y << endl;
    return 1;
}
```



# Desbordamiento

```
x = 1.79769e+308
```

```
y = inf
```



# Limitaciones

---

- ❑ Desbordamiento
- ❑ Precisión

# Precisión

```
#include <iostream>
using namespace std;

/*
 * Muestra un resultado con problemas de
 * precisión.
 */
int main() {
    double x = 1e20;
    double y = 1;
    double z = x + y - x;
    cout << "z = " << z << endl;
    return 1;
}
```



# Precisión

$$Z = \emptyset$$



# Precisión. Otro ejemplo

```
#include <iostream>
using namespace std;

/*
 * Programa que muestra cuál es el primer natural no
 * representable como double.
 */
int main() {
    double x = 9007199254740992; //  $2^{53}$ 
    double y = 9007199254740993; //  $2^{53} + 1$ 

    cout << fixed;
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;

    return 1;
}
```

**Basado en:** Respuesta de kennytm a «Which is the first integer that an IEEE 754 float is incapable of representing exactly?». *Stack Overflow*. 2010.  
<https://stackoverflow.com/questions/3793838/> (consultado el 24-10-2019).



# Precisión. Otro ejemplo

$x = 9007199254740992.000000$

$y = 9007199254740992.000000$





# Limitaciones

---

- ❑ Desbordamiento
- ❑ Precisión
- ❑ *NaN*

## *Not a number*

```
/*  
 * Programa que muestra un resultado con  
 * una codificación no válida de datos de  
 * tipo real.  
 */  
int main() {  
    double x = sqrt(-1);  
    cout << x << endl;  
    return 1;  
}
```



# Desbordamiento

`x = nan`

# Suma de series

---

## □ Exponencial

■  $e^x = 1 + x^1/1! + x^2/2! + x^3/3! + x^4/4! + \dots$

## □ Coseno

■  $\cos x = 1 - x^2/2! + x^4/4! - x^6/6! + \dots$

## □ Seno

■  $\sin x = x^1/1! - x^3/3! + x^5/5! - x^7/7! + \dots$

## □ $\pi$

■  $\pi = 4 \times (1 - 1/3 + 1/5 - 1/7 + 1/9 + \dots)$

# Coseno

```
/*  
 * Pre: El valor de «x» viene expresado en radianes.  
 * Post: Devuelve una aproximación al valor de cos x.  
 */  
double cos(double x) {  
    ...  
}
```

# Coseno

```
double cos(double x) {  
    // Se tiene en cuenta el desarrollo en serie de la función coseno:  
    //  $\cos x = 1 - x^2/2! + x^4/4! - x^6/6! + x^8/8! - x^{10}/10! + \dots$   
    const double COTA = 1.0E-15;  
  
    unsigned i = 0; // índice del término  
    double termino = 1.0; //  $\text{termino} = (-1)^i \cdot x^{2i} / (2i)!$   
    double resultado = termino; //  $\text{resultado} = \text{suma términos calculados}$   
  
    while (abs(termino) > COTA) {  
        // Se incrementa «resultado» con el siguiente término de la serie:  
        i++; // siguiente índice  
        //  $\text{termino} = (-1)^{i-1} \cdot x^{2i-2} / (2i-2)!$   
        termino = -termino * x * x / (2 * i * (2 * i - 1));  
        //  $\text{resultado} = \text{suma de los términos calculados}$   
        resultado += termino;  
    }  
    return resultado;  
}
```



# Coseno

## $\cos(\pi/4)$

i	termino	resultado
0	1,000000000000000000	1,000000000000000000
1	-0,308425137534042	0,691574862465958
2	0,015854344243816	0,707429206709773
3	-0,000325991886927	0,707103214822846
4	0,000003590860449	0,707106805683294
5	-0,000000024611370	0,707106781071925
6	0,000000000115012	0,707106781186936
7	-0,00000000000000390	0,707106781186546
8	0,00000000000000001	0,707106781186547
9	0,00000000000000000	0,707106781186547



# Resumen

---

- Tipos reales
  - Dominio valores
  - Representación
  - Operaciones
  - Desbordamiento y precisión
  - La biblioteca cmath
- Problemas