

## Ejercicio U2\_B10\_E1

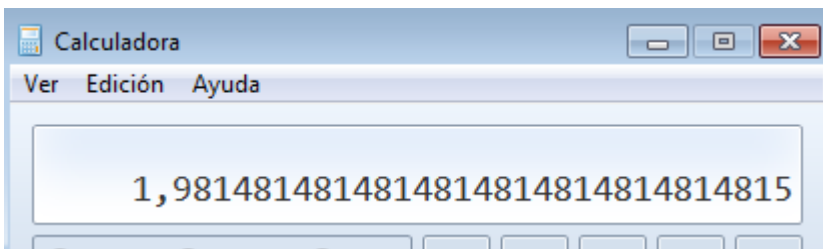
```
import java.math.BigDecimal;
class Unidad2{
    public static void main(String[] args) {
        BigDecimal numero1 = new BigDecimal("10.7");
        BigDecimal numero2 = new BigDecimal("5.4");
        System.out.println("el bigdecimal 1: "+numero1);
        System.out.println("el bigdecimal 2: " + numero2);
        System.out.println("Suma: "+numero1.add(numero2));
        System.out.println("Resta: "+numero1.subtract(numero2));
        System.out.println("Multiplicacion: "+numero1.multiply(numero2));
        System.out.println("Division: "+numero1.divide(numero2));
    }
}
```

el método `divide()` genera una excepción en tiempo de ejecución

la división de números reales  $\mathbb{R}$  puede provocar un resultado con infinitos decimales, como por ejemplo

$$1 / 3 = 0,33333333333333333333333333333333\dots$$

y como por ejemplo los números de ejercicio 10.7/5.4



Aquí se ve claro que es un número periódico y finalmente la calculadora windows redondea con el "5" final.

si hago la operación con una calculadora web , todavía redondea antes



La clase `BigDecimal`, no se atreve a hacer redondeos sin permiso del programador y por eso le avisa de que “no sabe cómo proceder” lanzando una excepción. Los programadores que utilizan `BigDecimal` siempre tienen que especificar la escala(el número de decimales) y el método de redondeo cuando este se requiere.

Hay una versión del método `divide()` en la que se puede especificar escala y redondeo de forma que ahora `BigDecimal` sabe como proceder cuando tenga que redondear un número.

Podríamos por tanto haber escrito:

```
import java.math.BigDecimal;
import java.math.RoundingMode;
```

```
class Unidad2 {
    public static void main(String[] args) {
        BigDecimal numero1 = new BigDecimal("10.7");
        BigDecimal numero2 = new BigDecimal("5.4");
        System.out.println("Suma: " + numero1.add(numero2));
        System.out.println("Resta: " + numero1.subtract(numero2));
        System.out.println("Multiplicacion: " + numero1.multiply(numero2));
    }
}
```

```

// como al dividir pueden salir periodicos mejor indicar escala/redondeo
BigDecimal division = numero1.divide(numero2, 3, RoundingMode.UP);
System.out.println("Division: " + division);
// si sólo indico redondeo la escala la infiere de los operandos
division = numero1.divide(numero2, RoundingMode.UP);
System.out.println("Division: " + division);
}
}

```

## Ejercicio U2\_B10\_E2:

```

import java.math.BigDecimal;
import java.util.Scanner;
class Unidad2 {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.print("precio: ");
        BigDecimal precio=sc.nextBigDecimal();
        System.out.print("impuesto: ");
        BigDecimal impuesto=sc.nextBigDecimal();
        BigDecimal valorImpuesto=precio.multiply(impuesto).divide(new BigDecimal("100"));
        System.out.println("valor impuesto: "+ valorImpuesto);
        BigDecimal pvp= precio.add(valorImpuesto);
        System.out.println("Precio final "+ pvp);
        //todo en un println() ni mejor ni peor
        System.out.println("Precio final "+ precio.add(precio.multiply(impuesto).divide(new BigDecimal("100"))));
    }
}

```

## Ejercicio U2\_B10\_E3:

Simplemente tomo la entrada como String con next()

```

import java.util.Scanner;
import java.math.BigDecimal;
class Unidad2 {
    public static void main(String args[]) {
        BigDecimal precio;
        BigDecimal impuesto;
        BigDecimal pvp;
        Scanner entrada = new Scanner(System.in);
        System.out.println("precio: ");
        precio=new BigDecimal(entrada.next());
        System.out.println("impuesto: ");
        impuesto=new BigDecimal(entrada.next());
        pvp=precio.add((precio.multiply(impuesto)).divide(new BigDecimal("100")));
        System.out.println("pvp: "+pvp);
    }
}

```

## Ejercicio U2\_B8\_E4:

```

import java.util.Scanner;
import java.math.BigDecimal;
import java.math.RoundingMode;

class Unidad2 {

    public static void main(String args[]) {
        BigDecimal precio;
        BigDecimal impuesto;
        BigDecimal pvp;
        Scanner entrada = new Scanner(System.in);
        //pido precio y lo configuro con dos decimales
    }
}

```

```

System.out.print("precio en euros: ");
precio = new BigDecimal(entrada.next());
precio = precio.setScale(2, RoundingMode.UP);
System.out.println("precio redondeado: "+precio);

//pido impuesto y lo configuro con dos decimales
System.out.print("impuesto en %: ");
impuesto = new BigDecimal(entrada.next());
impuesto=impuesto.setScale(2,RoundingMode.UP);
System.out.println("impuesto en % redondeado: "+impuesto);

//consultar en el API la versión de divide() que se usa, tiene 3 parámetros.
//de paso que divido entre 100 para pasar a tanto por ya configuro la escala
//también pude haber multiplicado por 0.01 que es equivalente a dividir por 100
impuesto = impuesto.divide(new BigDecimal("100"), 2, RoundingMode.UP);
System.out.println("impuesto en tanto por 1 y redondeado: "+impuesto);

//calculo ahora el valor del impuesto en €
BigDecimal valorImpuesto = precio.multiply(impuesto);
System.out.println("impuesto en euros sobre el precio: "+valorImpuesto);
valorImpuesto = valorImpuesto.setScale(2, RoundingMode.UP);
System.out.println("impuesto en euros sobre el precio redondeado: "+impuesto);

//al fin pvp, no precisa redondeo por que siempre suma dos números ya normalizados
pvp = precio.add(valorImpuesto);

System.out.println("pvp: " + pvp);
}
}

```

## Ejercicio U2\_B8\_E5:

```

import java.math.BigDecimal;

class Cuenta {
    private String numeroCuenta;
    private String titular;
    private BigDecimal saldo;

    Cuenta(String numeroCuenta, String titular, BigDecimal saldo) {
        this.numeroCuenta = numeroCuenta;
        this.titular = titular;
        this.saldo = saldo;
    }

    Cuenta() {
        this("sin numero", "sin titular", BigDecimal.ZERO);
    }

    void setSaldo(BigDecimal saldo) {
        this.saldo = saldo;
    }

    BigDecimal getSaldo() {
        return saldo;
    }

    public String toString() {
        return "("+numeroCuenta + ", " + titular + ", " + saldo+")";
    }
}

public class Unidad2 {
    public static void main(String[] args) {
        Cuenta c1 = new Cuenta("111-222", "Epi", new BigDecimal("50.0"));
        System.out.println("el saldo inicial de Epi es: " + c1.getSaldo());

        Cuenta c2 = new Cuenta("999-888", "Blas", new BigDecimal("100.0"));
        System.out.println("Datos de la cuenta c2: " + c2);

        c1.setSaldo(new BigDecimal("250.0"));
    }
}

```

```
System.out.println("el nuevo saldo de Epi es: " + c1.getSaldo());
```

```
Cuenta c3 = new Cuenta();
```

```
System.out.println("datos de cuenta creada sin parámetros: " + c3);
```

```
    }  
}
```