Tratamiento de errores
- Excepciones -

Introducción

```
int c;
int a = 1;
int b = 0;
c = a / b;
System.out.println("¿llega aquí?");
```

```
int c;
int a = 1;
int b = 0;
try {
        c = a / b;
} catch (ArithmeticException e) {
        System.out.println("¡División por cero!");
}
System.out.println("¿llega aquí?");
```

Uso general

```
try {
    ... // bloque try
} catch (TipoExcepción nombreParámetro) {
    ... // bloque catch
}
```

```
try {
    ... // bloque try
} catch (TipoExcepción1 nombreParámetro1) {
    ... // bloque catch
} catch (TipoExcepción2 nombreParámetro2) {
    ... // bloque catch
}
```


Uso general

```
int c;
int a = 1;
int b = 0;
try {
        c = a / b;
} catch (Exception e) {
        System.out.println("Estoy en el primer catch");
} catch (ArithmeticException e) {
        System.out.println("Estoy en el segundo catch");
}
```


Uso general

```
int c;
int a = 1;
int b = 0;
try {
      c = a / b;
} catch (TipoException2 e1 | TipoException1 e2) {
      System.out.println("Estoy en el primer catch");
}
```

Uso general

```
try {
    ... // bloque de trabajo
    return;
} catch (IOException e) {
    ... // bloque si salta la excepción
} finally {
    ... // código para cerrar archivos
}
```


Uso general

```
void metodo1(int a, int b) {
        int c;
        try {
                c = a / b;
        } catch (ArithmeticException e) {
                System.out.println("División por cero");
        }
        System.out.println("a/b = " + c):
}
```

```
void metodo2() {
        int x, y;
        ...
        metodo1(x, y);
        ...
}
```


Uso general

```
void metodo1(int a, int b) throws ArithmeticException {
        int c;
        c = a / b;
        System.out.println("a/b = " + c);
}
```

```
void metodo2() {
        int x, y;
        ...
        try {
                metodo1(x, y);
        } catch (ArithmeticException e) {
                        System.out.println("División por cero");
        }
        ...
}
```