

# Curso de Java Standard



Ing. Octavio Robleto



octavio.robleto@gmail.com



<https://octaviorobleto.com>



# throws

Podemos delegar que se encargue del error quien invoque al método con la palabra reservada **throws** y este agregue su **try catch** o vuelva a delegarlo, esto solo puede usarse en la firma de los métodos.

```
void com.educacionIT.javase.Throw_s.App.cargarArchivo(String url) throws FileNotFoundException
```

Recordemos que al instanciar un objeto del tipo **FileReader** con el constructor que recibe como argumento la dirección de un archivo, el JDK nos indica que tipo de problema puede ocurrir (**FileNotFoundException**).

Dentro del método **cargarArchivo** no se maneja la excepción sino que simplemente se le delega a quien lo invoque.

```
public static void main(String[] args){
    System.out.println("Inicio del Programa");
    cargarArchivo("");
    System.out.println("Fin del Programa");
}

public static void cargarArchivo(String url) throws FileNotFoundException {
    FileReader archivo = new FileReader(url);
}
```




# throws

Cuando se utiliza en el **main** se delega a la máquina virtual de Java que tratara de manejar la excepción y detendrá el software.

```
public static void main(String[] args) throws FileNotFoundException{
    System.out.println("Inicio del Programa");
    cargarArchivo("");
    System.out.println("Fin del Programa");
}

public static void cargarArchivo(String url) throws FileNotFoundException {
    FileReader archivo = new FileReader(url);
}
```



# throws

No solamente podemos delegar excepciones chequeadas sino también excepciones que sabemos que pueden ocurrir, para que quien invoque el método este al tanto de que puede ocurrir un error.

```
String com.educacionIT.javase.interfaces.Archivo.cargar(String  
ruta) throws FileNotFoundException
```

```
autoFamiliar1.cargar("C:/Archivos/auto.pdf");  
autoFamiliar2.cargar("C:/Archivos/auto.pdf");  
transporteCarga.cargar("C:/Archivos/auto.pdf");  
transportePasajeros.cargar("C:/Archivos/auto.pdf");  
transportePasajeros.cargar("C:/Archivos/auto.pdf");
```

# throw

**throw** sin la S, esta palabra reservada nos permite lanzar una excepción, literalmente forzamos el error.

Al instanciar la nueva Excepción se puede aprovechar la sobrecarga del constructor para enviar el mensaje que necesitamos.

```
public void setPuestos(Integer puestos) throws Exception {  
    if (puestos == null) {  
        throw new Exception("El Auto debe poseer puestos");  
    } else if (puestos < 1) {  
        throw new Exception("Solo esta permitido que el Auto tenga 1 o mas puestos");  
    }  
    this.puestos = puestos;  
}
```

```
public void setPatente(Patente patente) {  
    try {  
        if (patente == null) {  
            throw new Exception();  
        } else if (patente.getNumero().length() != 8) {  
            throw new Exception("La patente debe contener 8 caracteres");  
        }  
    } catch (Exception e) {  
        System.out.println(e.getMessage());  
    }  
    this.patente = patente;  
}
```

Podemos delegar la excepción que creamos o simplemente tratarla inmediatamente.