# Java Excepciones Aprenda a crear, lanzar y controlar excepciones

## Conclusión 1: Pila de ejecución

* Qué es, para qué sirve y cómo funciona la pila de ejecución.
* Qué es la depuración (debug) y para qué sirve.
* Cómo utilizar Eclipse y su perspectiva de debug.
* Cómo cambiar entre perspectivas de Eclipse.

|  |
| --- |
|  |

## Conclusión 2: Tratamiento de excepciones

* Qué son las excepciones, para qué sirven y por qué se utilizarlas.
* Cómo analizar el seguimiento de excepciones o stacktrace.
* Manejar excepciones con bloques try-catch.
* Manejar una excepción lanzada dentro del bloque catch.
* Manejar múltiples excepciones con más de un bloque catcho usando Multi-Catch usando el pipe (|).

|  |
| --- |
|  |

## Conclusión 3: Lanzando excepciones

* Cómo lanzar excepciones.
* Cómo asignar un mensaje a la excepción.

|  |
| --- |
|  |

## Conclusión 4: Checked y unchecked

## Conclusión 5: Aplicando excepciones

## Conclusión 6: Finally y try with resources

## Anexos:

## Código:

* Trow: palabra resevada que quiere decir lanza

## Vocabulario:

* Pila de ejecución: Stack o pila-🡪ciclo de ejecución método java

|  |
| --- |
| Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamente |

* Punto de quiebre
* Debug: El debug es la ejecución de nuestro sistema, pero donde él encuentra este breakpoint él va a parar y nos va a mostrar todo lo que hay en la memoria, en ese momento. Vamos a verlo aquí en práctica. Para eso, para el debug, nosotros tenemos que ir al botón de acá, que es este bichito, debug flujo. Este bichito está al lado de play, del botón de play.
* Step int: entra al pedazo de codigo
* Step over: Pasa a la siguiente línea
* Error
* Las excepciones también son objetos
* Stacktrace: traza de la pila
* Try y catch: intenta y atrapa