

**FAMILIA PROFESIONAL:**

**CICLOS FORMATIVOS:**

**MÓDULO:**

**Informática y Comunicaciones**

**Desarrollo de Aplicaciones Multiplataforma,**

**Desarrollo de Aplicaciones Web**

**Programación**

## **UNIDAD 7: TRATAMIENTO DE ERRORES Y EXCEPCIONES**

### **ACTIVIDADES**



**AUTORES:** **Fernando Rodríguez Alonso**  
**Sonia Pasamar Franco**

Este documento está bajo licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional License.

Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

**Usted es libre de:**

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato.

El licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia.

**Bajo los siguientes términos:**

- **Atribución** — Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.
- **NoComercial** — Usted no puede hacer uso del material con propósitos comerciales.
- **SinDerivadas** — Si remezcla, transforma o crea a partir del material, no podrá distribuir el material modificado.

No hay restricciones adicionales — No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

# CAPTURA Y PROPAGACIÓN DE EXCEPCIONES

## ACTIVIDAD 1x01

Codifica una clase **Actividad\_1x01** que contenga un método **main** y métodos auxiliares para realizar un programa en Java que lea dos números enteros desde el teclado y que calcule el cociente y el resto de la división entera entre ellos, con control de excepciones:

- 1) Completa el método de clase siguiente:

**String solicitarCadena(String mensaje)**

Este método tendrá como parámetro el mensaje a visualizar en la consola. Devolverá una cadena de caracteres leída desde el teclado (mediante el uso de las clases **InputStreamReader** y **BufferedReader**). No se podrá utilizar la clase **Scanner**.

Si ocurre un error en la lectura del teclado, el método propagará una excepción de entrada/salida al método desde el que se invoca para que la excepción se controle en dicho método.

- 2) Completa el método de clase siguiente:

**short solicitarEnteroCorto(String mensaje, int numMaxErrores)**

Este método tendrá como parámetros el mensaje a visualizar en la consola y el número máximo de errores permitidos al leer desde el teclado. Llamará al método **solicitarCadena** y devolverá el número entero corto correspondiente a la cadena leída desde el teclado.

Si ocurre un error en la lectura del teclado, el método propagará una excepción de entrada/salida al método desde el que se invoca para que la excepción se controle en dicho método.

Si ocurre un error al convertir de cadena a entero corto, el método capturará una excepción de formato de número. El tratamiento de esta excepción consistirá en visualizar en consola un mensaje *“El dato introducido no tiene formato de número entero corto.”*, visualizar la excepción y el mensaje de error, y contar los errores sucedidos. A continuación, se deberá pedir por teclado otro entero corto. Este proceso se deberá repetir mientras el dato introducido no tenga formato de número y no se haya superado el número máximo de errores permitidos.

Si se supera el número máximo de errores permitidos, visualizará en consola un mensaje *“Se ha superado el número de errores permitidos, que es N.”* y devolverá 0.

- 3) Completa el método **main** correspondiente al programa principal.

Este método utilizará el método **solicitarEnteroCorto** para leer desde teclado un dividendo *A* y un divisor *B*, con un máximo de tres errores permitidos. Visualizará en consola un mensaje *“A es divisible entre B.”* si el resto de la división entera entre el dividendo *A* y el divisor *B* es cero. Visualizará en consola un mensaje *“A no es divisible entre B.”* en caso contrario. En cualquier caso, mostrará además en consola el cociente y el resto resultantes de dicha división entera.

Si ocurre un error en la lectura del teclado, el método capturará una excepción de entrada/salida. El tratamiento de esta excepción consistirá en visualizar en consola un mensaje *“Se ha producido un error de entrada/salida.”*, visualizar en consola la excepción y el mensaje de error, y abortar el programa.

Si ocurre un error al realizar la división entera, el método capturará una excepción aritmética. El tratamiento de esta excepción consistirá en visualizar en consola un mensaje *“Se ha producido un error aritmético.”*, visualizar en consola la excepción y el mensaje de error, y abortar el programa.

Dos ejemplos de ejecución del programa podrían ser:

¿Dividendo? 100	¿Dividendo? 100
¿Divisor? 30	¿Divisor? 0
100 no es divisible entre 30.	Se ha producido un error aritmético.
Cociente: 3	java.lang.ArithmeticException:
Resto: 10	/ by zero

Un tercer ejemplo de ejecución del programa podría ser:

```

¿Dividendo? cien
El dato introducido no tiene formato de número entero corto.
java.lang.NumberFormatException: For input string: "cien"
Error número 1.
¿Dividendo? 100
¿Divisor? veinte y cinco
El dato introducido no tiene formato de número entero corto.
java.lang.NumberFormatException: For input string: "veinte y cinco"
Error número 1.
¿Divisor? veinticinco
El dato introducido no tiene formato de número entero corto.
java.lang.NumberFormatException: For input string: "veinticinco"
Error número 2.
¿Divisor? 25
100 es divisible entre 25.
Cociente: 4
Resto: 0
    
```

Un cuarto ejemplo de ejecución del programa podría ser:

```

¿Dividendo? uno
El dato introducido no tiene formato de número entero corto.
java.lang.NumberFormatException: For input string: "uno"
Error número 1.
¿Dividendo? dos
El dato introducido no tiene formato de número entero corto.
java.lang.NumberFormatException: For input string: "dos"
Error número 2.
¿Dividendo? tres
El dato introducido no tiene formato de número entero corto.
java.lang.NumberFormatException: For input string: "tres"
Error número 3.
¿Dividendo? cuatro
El dato introducido no tiene formato de número entero corto.
java.lang.NumberFormatException: For input string: "cuatro"
Error número 4.
Se ha superado el número de errores permitidos, que es 3.
    
```

```
¿Divisor? 25  
0 es divisible entre 25.  
cociente = 0  
resto = 0
```

## CREACIÓN Y LANZAMIENTO DE EXCEPCIONES PERSONALIZADAS

### ACTIVIDAD 2x01

Codifica una clase **Usuario** para tratar la información de los diferentes usuarios de un sitio web. De cada uno de ellos se desea guardar: el nombre, la contraseña y el número de puntos.

El nombre de cada usuario será una cadena de texto que se utilizará para identificar el usuario. No podrá haber dos o más usuarios con el mismo nombre en el sitio web. La contraseña de cada usuario deberá tener una longitud entre 6 y 20 caracteres (con ambos límites incluidos).

Para esta clase **Usuario**:

- Codifica un constructor que reciba como parámetros un nombre, una contraseña y un número de puntos.
- Codifica un método de objeto que devuelva una cadena de texto con el resumen de los atributos del objeto. Para ello, sobrescribe el método **toString** que procede de la clase **Object**.

Codifica una clase **ExcepcionUsuario** para tratar las situaciones anómalas relativas a la creación de usuarios. Esta clase deberá heredar de la excepción **RuntimeException** y tratará las siguientes situaciones anómalas:

- Al crear un usuario, si la contraseña no tiene una longitud entre 6 y 20 caracteres (con ambos límites incluidos), se lanzará una excepción de usuario con el mensaje:  
**Error al crear usuario: contraseña no tiene entre 6 y 20 caracteres.**
- Al crear un usuario, si el número de puntos es negativo, se lanzará una excepción de usuario con el mensaje:  
**Error al crear usuario: número de puntos negativo.**

Codifica una clase **Clientela** que permita gestionar un conjunto de usuarios. Además de un vector de usuarios, esta clase contendrá un índice que tendrá una doble función: marcará la tanto la primera posición libre del vector de usuarios, como el número de elementos que éste contiene instanciados.

Para esta clase **Clientela**:

- Codifica un constructor que reciba como parámetro el número de usuarios que puede contener (es decir, la capacidad del vector). Al instanciar un objeto de esta clase, el índice de la clientela será 0.
- Codifica el método sobrescrito **toString** que devuelva una cadena de texto con la información de todos los usuarios de la clientela. Para cada usuario, se indicará su posición dentro del vector de usuarios y un resumen con los valores de todos los atributos de dicho usuario.
- Codifica los métodos de objeto necesarios, según las opciones de menú indicadas en el programa principal.

Codifica una clase **ExcepcionClientela** para tratar las situaciones anómalas relativas a la gestión de la clientela de usuarios. Esta clase deberá heredar de la excepción `RuntimeException` y tratará las siguientes situaciones anómalas:

- Al insertar un usuario en la clientela, si el vector está lleno de usuarios, se lanzará una excepción de clientela con el mensaje:  
**Error al insertar usuario: vector de usuarios lleno.**
- Al insertar un usuario en la clientela, si existe un usuario con el mismo nombre de usuario en la clientela, se lanzará una excepción de clientela con el mensaje:  
**Error al insertar usuario: nombre de usuario repetido.**
- Al eliminar un usuario de la clientela, si el vector está vacío, se lanzará una excepción de clientela con el mensaje:  
**Error al eliminar usuario: vector de usuarios vacío.**
- Al eliminar un usuario de la clientela, si la posición no indica un usuario instanciado dentro del vector, se lanzará una excepción de clientela con el mensaje:  
**Error al eliminar usuario: posición no indica usuario en la clientela.**

Codifica una clase **Actividad\_2x01** que incluya un programa principal **main**. Este programa utilizará una clientela de usuarios (con capacidad para 100 usuarios) y gestionará el siguiente menú de opciones:

**0) Salir del programa.**

**1) Insertar un usuario en la clientela.**

Leerá por teclado el nombre, la contraseña y el número de puntos del usuario a insertar.

Intentará realizar la inserción del usuario, creado a partir de los datos leídos por teclado, en la clientela.

Si la clientela está llena (con 100 usuarios) o si la clientela ya tiene un usuario con el mismo nombre, se lanzará una excepción de clientela con el mensaje correspondiente.

Si la inserción se ha realizado con éxito, visualizará en consola el mensaje:

**Se ha insertado el usuario en la clientela con éxito.**

**2) Consultar todos los usuarios de la clientela.**

Realizará la consulta de todos los usuarios de la clientela.

Si no hay usuarios en la clientela, visualizará en consola el mensaje:

**La clientela está vacía.**

En caso contrario:

- Visualizará en consola el listado de todos los usuarios contenidos en la clientela, indicando para cada usuario, su posición dentro de la clientela y un resumen con los valores de todos los atributos de dicho usuario.
- Visualizará en consola el número de usuarios consultados de la clientela.

**3) Eliminar un usuario, por posición, de la clientela.**

Leerá por teclado la posición del usuario a eliminar.

Intentará realizar la eliminación del usuario, indicado mediante la posición leída desde teclado, de la clientela.

Si la clientela está vacía (sin usuarios) o si no hay ningún usuario en dicha posición dentro de la clientela, se lanzará una excepción de clientela con el mensaje correspondiente.

Si la eliminación se ha realizado con éxito, visualizará en consola el mensaje:

**Se ha eliminado el usuario de la clientela con éxito.**

Este programa principal deberá validar que la opción de menú elegida sea válida (comprendida entre 0 y 3). Si no lo es, visualizará en consola el mensaje:

**La opción de menú debe estar comprendida entre 0 y 3.**

Este programa principal deberá tratar todas las excepciones posibles. El tratamiento de cada excepción consistirá en capturarla y visualizar en consola el tipo de la excepción y su mensaje correspondiente. Este tratamiento deberá repetirse para cada opción de menú de forma iterativa.