

CURSO DE PROGRAMACIÓN CON JAVA

EJERCICIO

EJERCICIO MANEJO DE EXCEPCIONES 2



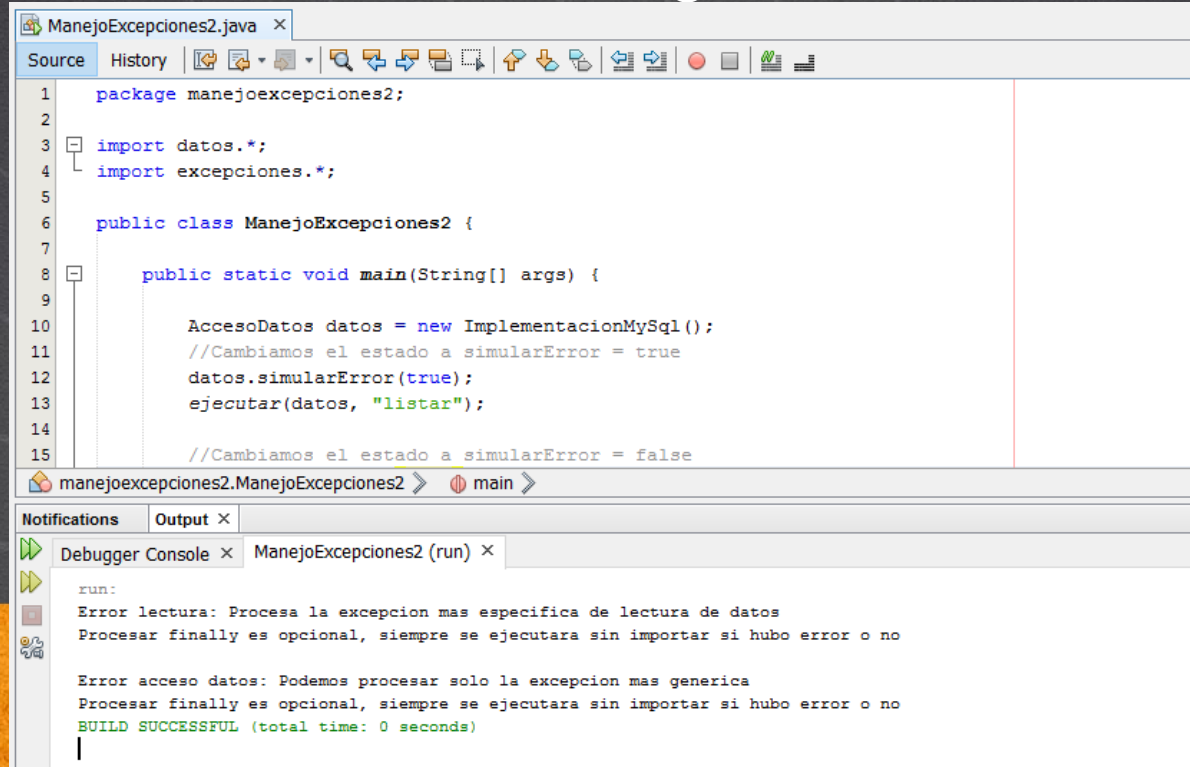
Experiencia y Conocimiento para tu vida

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

Crear un ejercicio para uso más avanzado de excepciones. Al finalizar deberemos observar lo siguiente:



The screenshot displays an IDE window titled 'ManejoExcepciones2.java'. The code defines a package, imports classes from 'datos' and 'excepciones', and creates a 'ManejoExcepciones2' class with a 'main' method. The 'main' method initializes 'AccesoDatos' with 'ImplementacionMySQL()', sets 'simularError' to true, calls 'ejecutar' with 'listar', and then sets 'simularError' back to false. Below the code editor, the 'Output' tab shows the execution results, including error messages for data reading and access, and a final 'BUILD SUCCESSFUL' message.

```
1 package manejoexcepciones2;
2
3 import datos.*;
4 import excepciones.*;
5
6 public class ManejoExcepciones2 {
7
8     public static void main(String[] args) {
9
10         AccesoDatos datos = new ImplementacionMySQL();
11         //Cambiamos el estado a simularError = true
12         datos.simularError(true);
13         ejecutar(datos, "listar");
14
15         //Cambiamos el estado a simularError = false
```

manejoexcepciones2.ManejoExcepciones2 > main >

Notifications Output X

Debugger Console X ManejoExcepciones2 (run) X

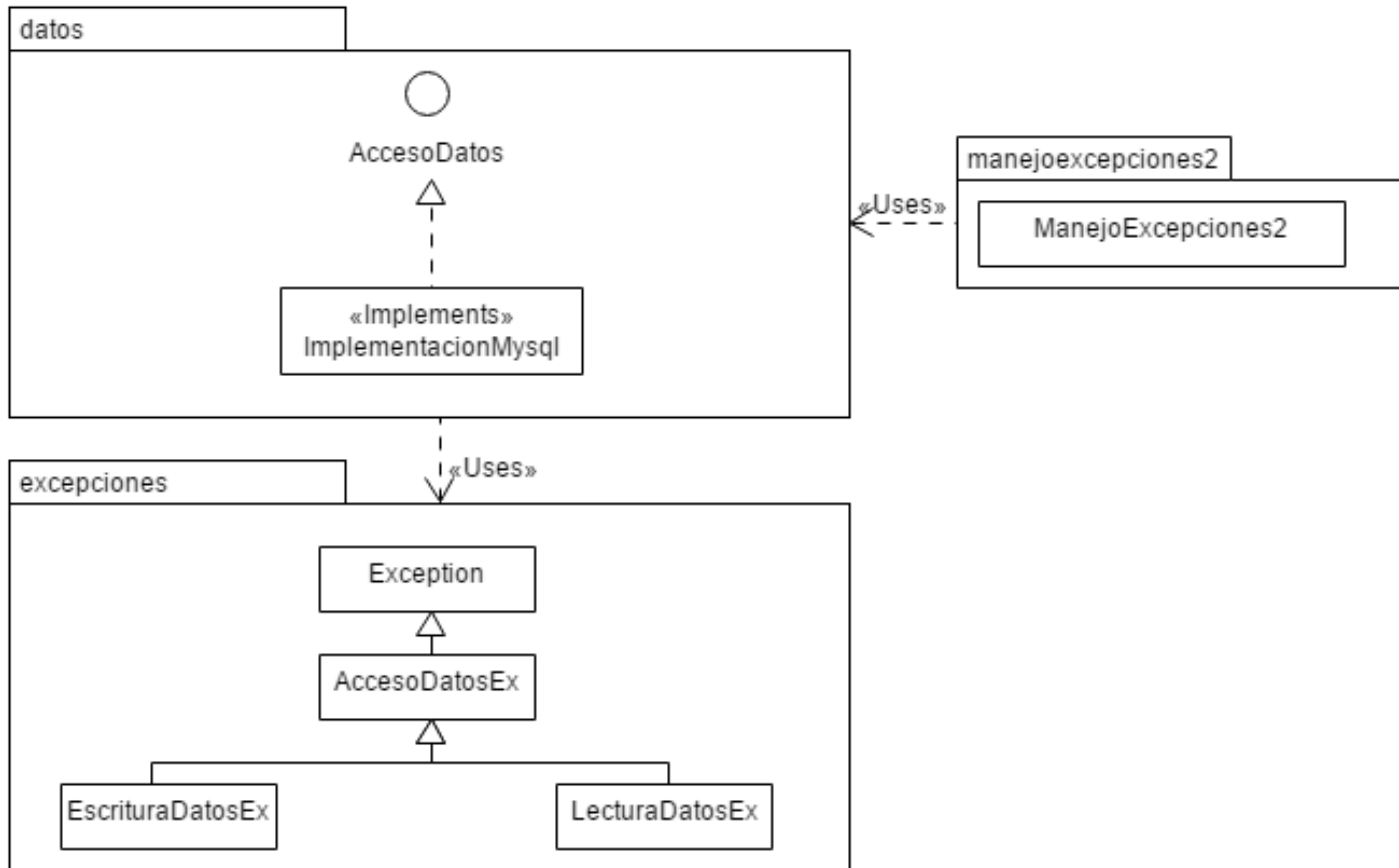
run:

Error lectura: Procesa la excepcion mas especifica de lectura de datos
Procesar finally es opcional, siempre se ejecutara sin importar si hubo error o no

Error acceso datos: Podemos procesar solo la excepcion mas generica
Procesar finally es opcional, siempre se ejecutara sin importar si hubo error o no

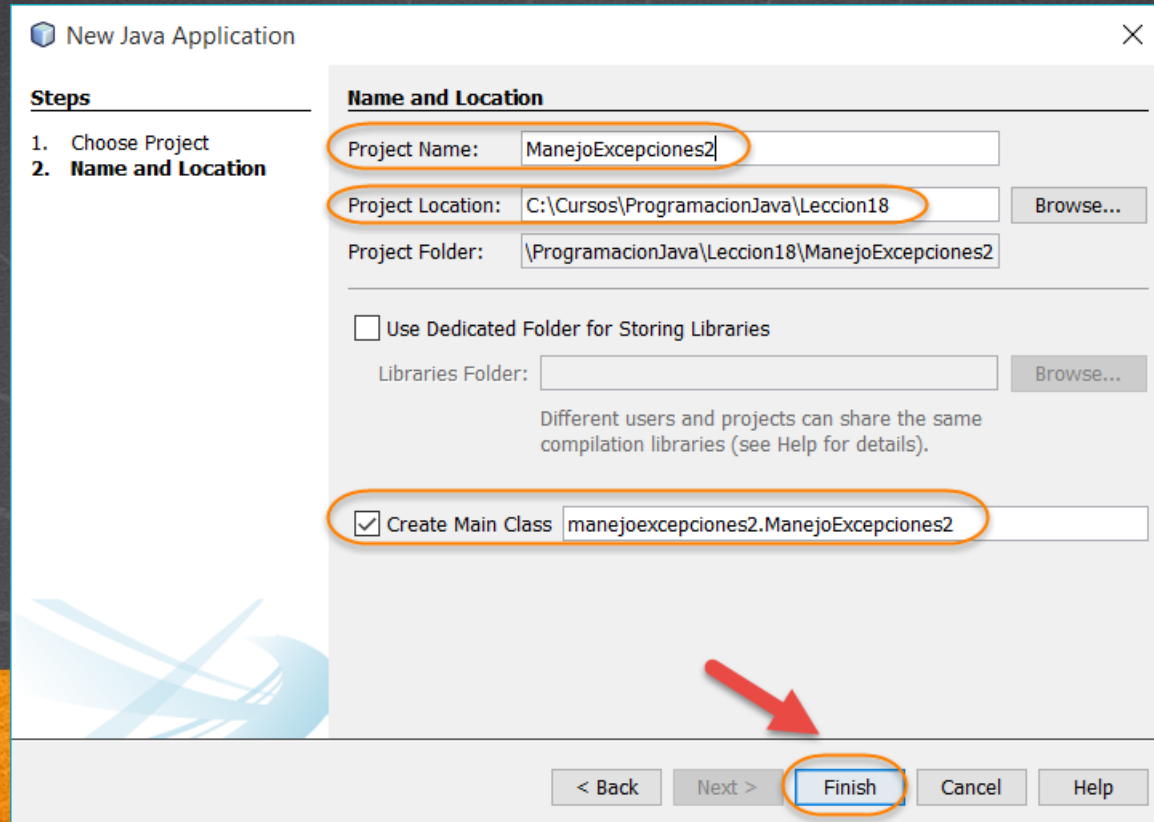
BUILD SUCCESSFUL (total time: 0 seconds)

DIAGRAMA DE CLASES DEL EJERCICIO



PASO 1. CREACIÓN DEL PROYECTO

Vamos a crear el proyecto :



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: ManejoExcepciones2

Project Location: C:\Cursos\ProgramacionJava\Leccion18 Browse...

Project Folder: \ProgramacionJava\Leccion18\ManejoExcepciones2

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class manejoexcepciones2.ManejoExcepciones2

< Back Next > **Finish** Cancel Help

PASO 2. CREACIÓN CLASE

Vamos a crear una clase:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: AccesoDatosEx

Project: ManejoExcepciones2

Location: Source Packages

Package: excepciones

Created File: amacionJava\Leccion18\ManejoExcepciones2\src\excepciones\AccesoDatosEx.java

< Back Next > **Finish** Cancel Help

PASO 3. CREACIÓN CLASE

Vamos a crear una clase:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

PASO 4. CREACIÓN CLASE

Vamos a crear una clase:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

PASO 5. CREACIÓN CLASE

Vamos a crear una clase:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

PASO 6. CREACIÓN CLASE

Vamos a crear una clase:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

PASO 7. MODIFICAMOS EL CÓDIGO

Archivo AccesoDatosEx.java:

```
package excepciones;  
  
public class AccesoDatosEx extends Exception{  
  
    public AccesoDatosEx(String mensaje){  
        super(mensaje);  
    }  
}
```

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo EscrituraDatosEx.java:

```
package excepciones;

public class EscrituraDatosEx extends AccesoDatosEx{

    public EscrituraDatosEx(String mensaje) {
        super(mensaje);
    }
}
```

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

PASO 9. MODIFICAMOS EL CÓDIGO

Archivo LecturaDatosEx.java:

```
package excepciones;

public class LecturaDatosEx extends AccesoDatosEx{

    public LecturaDatosEx(String mensaje) {
        super(mensaje);
    }

}
```


PASO 10. MODIFICAMOS EL CÓDIGO

Archivo AccesosDatos.java :

```
package datos;

import excepciones.AccesosDatosEx;

public interface AccesosDatos {

    public abstract void insertar() throws AccesosDatosEx;

    public abstract void listar() throws AccesosDatosEx;

    public abstract void simularError(boolean simularError);

}
```

PASO 11. MODIFICAMOS EL CÓDIGO

Archivo ImplementacionMySql.java :

```
package datos;
import excepciones.*;

public class ImplementacionMySql implements AccesoDatos{
    private boolean simularError;

    @Override
    public void insertar() throws AccesoDatosEx {
        if (simularError) {
            throw new EscrituraDatosEx("Error de escritura de datos");
        } else {
            System.out.println("Insertar desde MySql");
        }
    }

    @Override
    public void listar() throws AccesoDatosEx {
        if (simularError) {
            throw new LecturaDatosEx("Error de lectura de datos");
        } else {
            System.out.println("Listar desde MySql");
        }
    }

    public boolean isSimularError() {
        return simularError;
    }

    @Override
    public void simularError(boolean simularError) {
        this.simularError = simularError;
    }
}
```

PASO 12. MODIFICAMOS EL CÓDIGO

Archivo ManejoExcepciones2.java (Parte 1):

```
package manejoexcepciones2;
import datos.*;
import excepciones.*;

public class ManejoExcepciones2 {

    public static void main(String[] args) {
        AccesoDatos datos = new ImplementacionMySQL();
        //Cambiamos el estado a simularError = true
        datos.simularError(true);
        ejecutar(datos, "listar");

        //Cambiamos el estado a simularError = false
        datos.simularError(true);
        System.out.println("");
        ejecutar(datos, "insertar");
    }
}
```

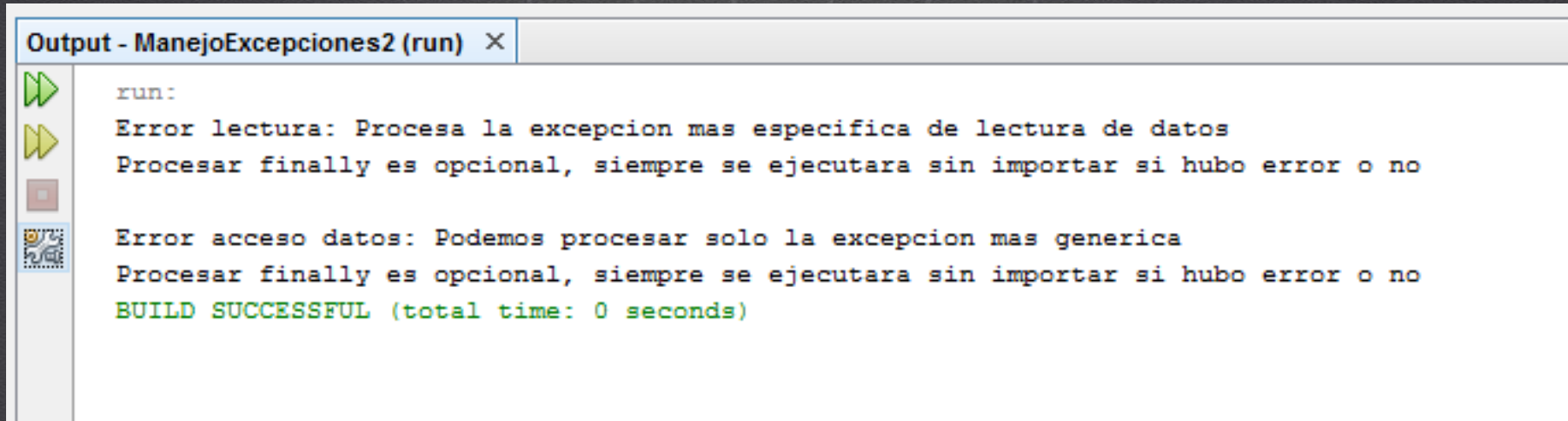
PASO 12. MODIFICAMOS EL CÓDIGO

Archivo ManejoExcepciones2.java (Parte 2):

```
private static void ejecutar(AccesoDatos datos, String accion) {
    if ("listar".equals(accion)) {
        try {
            datos.listar();
        }
        //Si se van a procesar varias excepciones de la misma jerarquia
        //siempre se debe procesar primero la excepcion de menor jerarquia
        //y posteriormente la de mayor jerarquia
        catch (LecturaDatosEx ex) {
            System.out.println("Error lectura: Procesa la excepcion mas especifica de lectura de datos");
        } catch (AccesoDatosEx ex) {
            System.out.println("Error Acceso datos: Procesa la excepcion mas generica de acceso a datos");
        } catch (Exception ex) {
            System.out.println("Error general");
        }
        finally {
            System.out.println("Procesar finally es opcional, siempre se ejecutara sin importar si hubo error o no");
        }
    } else if ("insertar".equals(accion)) {
        try {
            datos.insertar();
        } catch (AccesoDatosEx ex) {
            System.out.println("Error acceso datos: Podemos procesar solo la excepcion mas generica");
        }
        finally{
            System.out.println("Procesar finally es opcional, siempre se ejecutara sin importar si hubo error o no");
        }
    }
    else
        System.out.println("No se proporciono ninguna accion conocida");
}
```


PASO 13. EJECUTAMOS EL PROYECTO

El resultado es como sigue:



```
Output - ManejoExcepciones2 (run) X
run:
Error lectura: Procesa la excepcion mas especifica de lectura de datos
Procesar finally es opcional, siempre se ejecutara sin importar si hubo error o no

Error acceso datos: Podemos procesar solo la excepcion mas generica
Procesar finally es opcional, siempre se ejecutara sin importar si hubo error o no
BUILD SUCCESSFUL (total time: 0 seconds)
```

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

TAREAS EXTRA DEL EJERCICIO

- Probar con el modo debug del IDE y verificar paso a paso.
- Probar con distintos valores y lanzar excepciones padres e hija para así comprobar como funciona el manejo de excepciones según la jerarquía de excepciones que hemos declarado.



Experiencia y Conocimiento para tu vida

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos puesto en práctica conceptos más avanzados de excepciones en Java.
- Pudimos observar cómo trabajar con excepciones de tipo `Exception`, también es posible convertir estas excepciones a tipo `RuntimeException`, simplemente extendiendo de esta clase en lugar de la clase `Excepcion`. Esto puede quedar como un buen ejercicio para que observen cómo el compilador NO nos obliga a procesar las excepciones de tipo `RuntimeException` y así tener un código más limpio.

CURSO ONLINE

PROGRAMACIÓN CON JAVA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx