



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesora:*

Rocío Alejandra Aldeco Pérez

*Asignatura:*

Programación Orientada a Objetos

*Grupo:*

6

*No de Práctica(s):*

11

*Integrante(s):*

Ugalde Velasco Armando

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:*

*Semestre:*

2021-1

*Fecha de entrega:*

8 de enero de 2021

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_



**Práctica de Estudio 10: Excepciones y errores**  
Programación Orientada a Objetos Grupo 6  
Facultad de Ingeniería  
Departamento de Computación

**Objetivo de la práctica:** Implementar el intercambio de datos (lectura y escritura) entre fuentes externas (archivos y/o entrada y salida estándar) y un programa (en un lenguaje orientado a objetos).

Realiza las siguientes actividades después de leer y revisar en clase la **Práctica de Estudio 10: Manejo de archivos**.

Para esta práctica deberás realizar los siguientes pasos:

1. Usando el mismo programa de la práctica 10, ahora deberás leer los números de un archivo e imprimir los resultados en **Alphagrader**.
2. Existen dos casos de prueba uno para convertir hexadecimal a decimal y otro para decimal a hexadecimal, respectivamente. Deberás elegir correctamente los métodos a usar en cada caso para que los resultados sean los correctos.
3. Recuerda agregar las excepciones derivadas del uso de archivos.
5. Cuando estés seguro de que tu programa es correcto, súbelo a **Alphagrader**. Debes incluir los archivos *"input.txt"* y *"input2.txt"* dentro del **ZIP**. Recuerda que si no pasas todos los test no obtendrás los puntos de ejecución.

Para cumplir con la tarea planteada, se modificó el programa de la siguiente forma:



## Práctica de Estudio 10: Excepciones y errores

Programación Orientada a Objetos Grupo 6  
Facultad de Ingeniería  
Departamento de Computación

```
public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    String filename = sc.nextLine();
    try (var br = new BufferedReader(new FileReader(filename)))
    {
        String line;
        while ((line = br.readLine()) != null)
        {
            printRepresentation(line);
        }
    }
    catch (FileNotFoundException e)
    {
        System.out.println("No se encontró el archivo");
    }
    catch (IOException e)
    {
        System.out.println("Error al leer archivo");
    }
}
```

### *Método main*

Se modificó la función **main** de tal forma que, primero, se obtuviera el nombre del archivo a leer, y, posteriormente, se leyera las líneas del archivo mediante un **BufferedReader** y utilizando el bloque **try-with-resources**. Además, para cada línea, se imprimieron sus respectivas representaciones, utilizando el método **printRepresentation**. Cabe mencionar que éste último método contiene exactamente el mismo código del método **main** utilizado en la práctica anterior, lo cual demuestra que no fue necesario cambiar la lógica inicial del programa.

## CONCLUSIONES

Los archivos son una abstracción que nos permite representar información almacenada de forma persistente. Lo anterior, sin duda alguna, tiene presencia en un gran número de aplicaciones y forma parte fundamental de muchos programas. En la biblioteca estándar de Java, se encuentran implementadas algunas clases que facilitan en gran medida la manipulación de archivos. Dentro de estas utilidades, se encuentra el concepto de **Streams**, que nos permiten representar un flujo de datos, los cuales pueden ser de naturaleza binaria, o bien, caracteres.