

Ejercicios 2do parcial

1. Explicar el propósito de las clases:

- a. `File`
- b. `FileReader`
- c. `BufferedReader`

2. a. Explicar en detalle el funcionamiento de la función `pedirEntero`:

```
public class Consola {
    public static final java.util.Scanner teclado = new java.util.Scanner(System.in);
    public static final java.io.PrintStream pantalla = new java.io.PrintStream(System.out);

    public static int pedirEntero() {
        while (true) {
            try {
                int n = Integer.parseInt(teclado.nextLine());
                return n;
            } catch (NumberFormatException e) {
                pantalla.println(e.getMessage());
            }
        }
    }
}
```

b. Dadas las declaraciones de los métodos utilizados, indicar qué cambiaría en el código anterior para mejorar la robustez del programa.

```
public class Integer {
    public static int parseInt(String s)
        throws NumberFormatException {...}
}

public class Scanner {
    public String nextLine()
        throws NoSuchElementException, IllegalStateException {...}
}
```

3. Usando conceptos de programación funcional, diseñar una función que reciba una lista de palabras, y devuelva la lista de las 10 palabras más cortas que contienen las

5 vocales.

4. a. Explicar cuál es el propósito de cada uno de los componentes del patrón MVC: Modelo, Vista, Controlador.
- b. Clasificar cada una de las siguientes clases en M, V o C:

```
class UI {
    private Contador c;
    public UI(Contador c) { this.c = c; }
    void mostrar() { System.out.printf("La cuenta actual es: %d", c.get()); }
    void esperar() { new Scanner(System.in).nextLine(); }
}

class Contador {
    private int n = 0;
    public void incrementar() { n++; }
    public int get() { return n; }
}

class Main {
    public static void main(String[] args) {
        var c = new Contador();
        var ui = new UI(c);
        while (true) {
            ui.mostrar();
            ui.esperar();
            c.incrementar();
        }
    }
}
```