Refactoring 2

- Estudiante: Javier López
- Problema: Problema 2

Código Inicial

```
import os

def organizar():
    error_file_path = './refactoring/problema2/data/error.lo

    with open(error_file_path, 'r') as f:
        errores = []

        lineas = f.readlines()
        lineas = [line.rstrip().split(' ') for line in linea

        for linea in lineas:
            if linea[0] == 'E' and int(linea[1]) > 50:
                errores.append(linea)

        errores.sort(key=lambda x: int(x[2]))

        for error in errores:
            print(' '.join(error))

def main():
    organizar()

if __name__ == '__main__':
    main()
```

Solución

```
def read_file_lines(file_path):
    with open(file_path, "r") as f:
        return [line.rstrip() for line in f.readlines()]


def extract_and_sort_errors(lines):
    errores = []
    for line in lines:
        if not line.startswith("E"):
            continue

        severidad = int(line.split(" ")[1])
        if severidad > 50:
            errores.append(line)

    errores.sort(key=lambda x: int(x.split(" ")[2]))
    return errores


def print_errors(errores):
    for error in errores:
        print(error)


def organize(file_path):
    lines = read_file_lines(file_path)
    errores = extract_and_sort_errors(lines)
    print_errors(errores)


def main():
    error_file_path = "./refactoring/problema2/data/error.l
    organize(error_file_path)


if __name__ == "__main__":
    main()
```

Refactorings Utilizados

Extract Function

Se realizó la extracción de las diferentes responsabilidades del código en diferentes funciones. Una para leer o cargar los datos a un array con `read_file_lines` otra para preparar transformar lo cargado extraer, filtrar y ordernar `extract_and_sort_errors` en este caso considero que aunque son 3 cosas juntas son tan pocas y sencillas que no vale la pena dividir en más métodos. Y por último el método `print_errors` para mostrar el resultado en la consola.

Todo este proceso de extracción hace más fácil la lectura del código y su mantenibilidad. Pues se puede cambiar las diferentes partes y se entiende cada responsabilidad.

También hay la posibilidad de extraer aún más como encapsular una función si algo es un error o no, pero considero que ya genera mucha indirección y el nivel de granularidad actual lo considero adecuado.