

Ejercicio 3

```
public class Document {
    List<String> words;

    public long characterCount() {
        long count = this.words
            .stream()
            .mapToLong(w → w.length())
            .sum();
        return count;
    }

    public long calculateAvg() {
        long avgLength = this.words
            .stream()
            .mapToLong(w → w.length())
            .sum() / this.words.size();
        return avgLength;
    }
}
```

Paso uno: Análisis

- Ambos métodos utilizan la misma cadena de funciones para calcular la cantidad de letras que hay en el documento. No creo que haga falta extraer un método porque con reemplazar la cadena en el segundo método con una llamada al primer método basta → [Replace Inline Code with Function](#)
- Por otro lado, el método **calculateAvg()** puede causar una excepción si la lista de palabras no está inicializada o tiene 0 palabras dentro., por lo que habría que hacer algo para evitarlo:
 1. Preguntar dentro del método si el tamaño de la lista **words** es mayor a 0

2. Reemplazar el valor del tamaño de la lista cuando sea 0 → Esta opción me parece mejor porque no tenemos que agregar ninguna variable extra para que el método retorne 0, solo hace falta poner un operador ternario en el dividendo

Esto igualmente no sería un refactoring ya que estamos cambiando el comportamiento del código (¿creo?)

Paso dos: Refactoring

1. **Code Smell: Código Repetido** → Se repite la misma cadena de mensajes en los dos métodos
 - a. **Refactoring a aplicar: Replace Inline Code with Function** → Reemplazar la cadena de funciones por una llamada al método **characterCount()**

Resultado final

```
public class Document {
    List<String> words;

    public long characterCount() {
        long count = this.words.stream()
            .mapToLong(w → w.length()).sum();
        return count;
    }

    public long calculateAvg() {
        long avgLength = this.characterCount() /
            (this.words.size() > 0 ? this.words.size() : 1);
        return avgLength;
    }
}
```