Ejercicio 3 Refactoring

```
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;
        }
}
```

Bad smell: Rompe encapsulamiento la lista words. Refactoring: **Encapsulate Field**.

Bad smell: **Duplicated Code**, ambas clases iteran utilizando prácticamente los mismos pipes del stream. El Refactoring sería reemplazar el stream de calculateAvg() por una llamada al otro método (**Extract Method**).

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

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

```
    }

  public long calculateAvg() {
    return this.characterCount() / this.words.size();
    }
}
```

Bad smell: Variable temporal count. Se reemplaza con un return directamente.

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

  public long characterCount() {
      return this.words
          .stream()
          .mapToLong(w -> w.length())
          .sum();
    }

  public long calculateAvg() {
    return long(this.characterCount() / this.words.size());
    }
}
```