Para la Cota Optimista, creamos una nueva función en distancias.py llamada ***levenshtein\_cota\_optimista*** que recibe como argumentos dos cadenas ***x*** , ***y***, y el parámetro ***threshold***.

Esta función sigue los siguientes pasos:

1. La función lambda **f** recibe dos listas ***l1***, ***l2*** y obtiene los caracteres y el número de veces que aparecen en cada cadena (para ***y*** en valores negativos), y se suman.
2. Se obtiene una lista con el resultado de aplicar ***f*** a ***x***, ***y.***
3. Se recorre la lista y se guarda en los contadores ***pos*** y ***neg*** los valores positivos y negativos.
4. Se calcula la cota optimista siguiendo la definición, como el máximo

de ***pos*** y ***neg***.

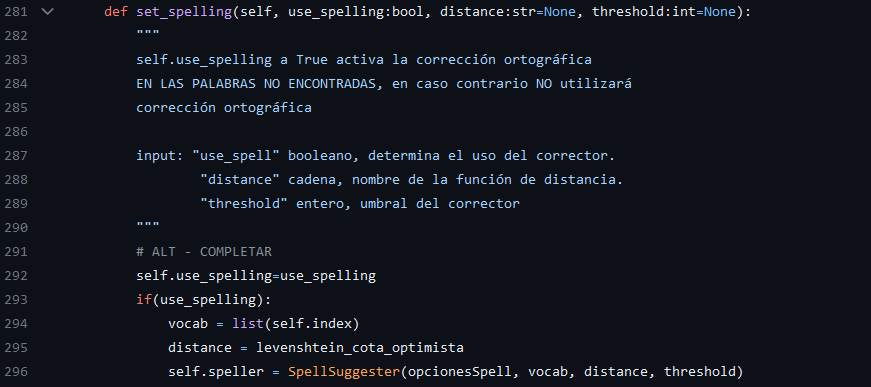
1. Finalmente si la cota es menor que el ***threshold*** se devuelve ***threshold*** +1, si no se calcula la distancia de levenshtein.

La implementación de la cota optimista en distancias.py es la siguiente:Texto

Descripción generada automáticamente

*(Imagen 1: levenshtein\_cota\_optimista en distancias.py)*

En cuanto a la integración con el proyecto de SAR, basta con añadir a la función set\_spelling de SAR\_lib.py, que la distancia se calcule con la cota optimista (línea 295 imagen 2) .



*(Imagen 1: set\_spelling en SAR\_lib.py)*