**Detector de plagio**

El programa se dividirá en diferentes módulos con diferentes funciones para que sea más fácil desarrollarlos por separado. En primer lugar se analizará el texto que se tiene que evaluar para definir sus tópicos, que se hará en su propio módulo, aunque para eso antes se necesita pasar los PDF a txt ([https://www.youtube.com/watch?v=BvuYWC6nyY0)](https://www.youtube.com/watch?v=BvuYWC6nyY0)%20) para poder analizarlos como corpus con las herramientas que ofrece la biblioteca NLTK.

**Definidor Topico**

Para lograr esto se basará en la técnica utilizada por Susan Li (30 de marzo de 2018), <https://towardsdatascience.com/topic-modelling-in-python-with-nltk-and-gensim-4ef03213cd21>. Se implementará una técnica llamada LDA (Latent Dirichlet Allocation).

El proceso consiste primero en limpiar el texto, incluyendo la lematización y la quita de palabras de “pare”. Luego se crea con estos datos un “bag of words” corpus y se lo guarda como un diccionario para un uso futuro. Luego se pedirá al algoritmo LDA, al que se accede con la biblioteca gensim, que encuentre el tópico más probable en el texto, devolverá sus palabras principales.

Después de definir el tópico se necesita recopilar los datos de los anteriores TPs y convertirlo en un corpus, esto se hará en el Modulo Dataset:

**Modulo Dataset**

Clase que convierte en corpus al dataset, referencias para implementación: <https://living-sun.com/es/python/696855-creating-a-new-corpus-with-nltk-python-nlp-nltk-corpus.html>

Luego se utiliza el módulo comparador para buscar el plagio:

**Modulo Comparador**

En éste modulo primero se extrae el texto del archivo que se analizará y cada oración de este se compara con el dataset para buscar plagio. Aquí se remueven las mayúsculas y los signos de puntuación para una evaluación más precisa. Se inicializará un contador que aumentará por cada plagio encontrado para luego calcular el porcentaje del texto plagiado. También se imprimirá el nombre del alumno si es que se encuentra en el texto. Para analizar si hay plagio se utilizan dos tipos de evaluaciones, la de plagio directo y la de parafraseo, explicadas a continuación:

**Evaluar plagio directo**

Primero se examina cada oración para evaluar que no haya plagio directo, se lo considera sólo si la oración tiene más de 6 palabras, como indica el texto de Mathieu Bouville (11 de marzo de 2008), Plagiarism: Words and ideas, <https://arxiv.org/pdf/0803.1526.pdf>

**Evaluar parafraseo**

Si no hay plagio directo se pasa a analizar la posibilidad del parafraseo, se utilizará la lógica propuesta por el usuario sgDysregulation (15 de octubre de 2017), Sin Título, <https://stackoverflow.com/questions/46732843/compare-two-sentences-on-basis-of-grammar-using-nlp>. Se hicieron algunas variantes para adaptarlo al idioma español. En primer lugar se necesita un stemmer, para reducir a la raíz a cada palabra, y se eligió el SnowballStemmer, que tiene soporte para el español. Este se carga en el módulo Comparador y se envía por parámetro para no cargarlo cada vez que se entra a la función que analiza el parafraseo. En segundo lugar se utilizará el wn.synsets (en español) para encontrar de cada palabra grupos de palabras que expresen el mismo concepto.

Con esto se comparan las similitudes entre los posibles synsets de dos palabras, y se calcula un porcentaje de similitud entre las dos oraciones. Si éste es mayor a 80% se considerará que habrá posibilidad de parafraseo y se imprimirán las dos oraciones para que un ojo humano las revise.