



Tesis
Automatización de lectura de Curriculum Vitae
para selección de personal en el Sector IT

Calonge, Federico Matias
calongefederico@gmail.com

21 de enero de 2022

Resumen

En la Tesis de Ingeniería en Informática que se presenta, se diseña un *sistema de lectura automática de Curriculum Vitae* accesible vía Web. La finalidad del mismo es ayudar al reclutador laboral a elegir a los mejores candidatos para los puestos laborales de IT que tenga disponible. Esta elección se realiza mediante el uso de algoritmos de *machine learning* y basándose, principalmente, en una medición de similitud entre textos: Curriculum Vitae de los candidatos por un lado, y descripciones de los puestos laborales de IT por el otro. El sistema esta desarrollado utilizando el lenguaje de programación Python, permitiendo verificar la teoría desarrollada.

Abstract

This Computer Engineering Thesis introduces an *automatic Curriculum Vitae reading system* accesible via the Web. The purpose of it is to help the job recruiter to choose the best candidates for the available IT job positions. This choice is made through the use of *machine learning* algorithms and based mainly on a measurement of similarity between texts: Curriculum Vitae of the candidates on the one hand, and IT job descriptions on the other hand. The system is developed using the Python programming language allowing to verify the developed theory.

Índice

1. Introducción.	5
1.1. Objetivos del Proyecto.	6
1.1.1. Objetivo general.	6
1.1.2. Objetivos específicos.	6
1.2. Alcance del Proyecto.	7
1.3. Organización.	7
2. Reclutamiento laboral en IT.	9
2.1. Introducción.	9
2.2. Sistemas de lectura y análisis de CV : Estado del arte.	9
2.2.1. Problemáticas.	10
2.2.2. Trabajos relacionados.	10
3. Algoritmos de Machine Learning.	11
3.1. Introducción.	11
3.2. K-Nearest Neighbor (KNN).	11
3.3. K-Means.	12
3.3.1. Elbow Method.	13
4. Natural Language Processing.	14
4.1. Introducción.	15
4.2. Preprocesamiento de textos.	16
4.3. Similitud entre textos.	17
4.4. Técnicas para medir Similitud entre textos.	18
4.4.1. Cosine Similarity.	18
4.4.2. Word Mover's Distance (WMD).	19
4.5. Algoritmos de vectorización.	20
4.5.1. TF-IDF.	20
4.5.2. Word Embeddings.	21
4.5.2.1. ¿Cómo entrenar Word Embeddings?	22
5. Implementación.	23

5.1.	Obtención del modelo de clasificación.	24
5.1.1.	Introducción.	24
5.1.2.	Esquema.	24
5.1.3.	Obtención de sets de datos.	25
5.1.3.1.	Curriculum Vitae.	25
5.1.3.2.	Descripciones Puestos Laborales.	25
5.1.4.	Preprocesamiento y limpieza de datos.	26
5.1.5.	Cantidad final del set de datos y su uso en las distintas etapas. . . .	27
5.2.	Comparando textos y obteniendo similitudes.	29
5.3.	Armado del modelo de clasificación KNN.	30
5.4.	Clasificación de nuevas muestras y resultados obtenidos.	31
5.5.	Integración al Sistema Web.	32
5.5.1.	Base de datos.	33
5.5.2.	Secciones del sistema	35
5.5.3.	Manejo de los datos.	38
5.5.3.1.	Modelado.	39
5.5.3.2.	Filtrado.	40
5.5.3.3.	Visualización.	41
5.6.	Pipeline Flow final del Sistema.	42
5.7.	Conclusiones.	43
5.8.	Caso de Uso.	44
5.9.	Limitaciones del sistema.	45
6.	Próximos pasos.	46
7.	Anexos.	47

1 Introducción.

El proceso de **selección de personal** se ha vuelto crucial para el manejo de recursos humanos en el mundo laboral moderno. Con la transformación digital de las empresas y del mercado laboral en general, identificar los perfiles más acordes a las necesidades de la empresa se convirtió en uno de los retos más ambiciosos de Recursos Humanos, en especial cuando hablamos del **Sector IT**, donde año a año se van generando nuevos puestos de trabajo.

En estos últimos años se implementaron una gran cantidad de **herramientas de Software que permiten automatizar y gestionar información de los candidatos de una manera mucho más intuitiva e inteligente**. Gracias a la ayuda de este tipo de sistemas, el reclutador consigue a los candidatos más cualificados para cada puesto.

El tema de este Proyecto de Tesis será desarrollar un *sistema de lectura automática de Curriculum Vitae* accesible vía Web. La finalidad del mismo es ayudar al reclutador laboral a elegir a los mejores candidatos para los puestos laborales de IT que tenga disponible. Esta elección se realiza mediante el uso de algoritmos de Machine Learning y basándose, principalmente, en una medición de similitud entre textos: Curriculum Vitae de los candidatos por un lado, y descripciones de los puestos laborales de IT por el otro.

La medición de similitudes de documentos es uno de los problemas más cruciales del **Procesamiento del Lenguaje Natural (NLP)**. Encontrar similitudes entre documentos se utiliza en varios dominios, tales como recomendación de películas, libros o artículos similares, identificación de documentos plagiados o documentos legales, etc.

Para que las máquinas puedan descubrir esta similitud entre documentos, se necesita definir una forma de medir matemáticamente la similitud, la cual debe ser comparable para que la máquina pueda identificar qué documentos son más similares (o menos). Previamente a esto necesitamos representar el texto de los documentos en una forma cuantificable (que suele ser en forma vectorial), de modo que podamos realizar cálculos de similitud sobre él.

Por lo tanto, los pasos necesarios para que las máquinas puedan medir la similitud entre documentos son:

1. Convertir un documento en un objeto matemático (vector).
2. Definir y emplear una medida de similitud.

Para el primer paso se utilizarán los algoritmos de vectorización **TF-IDF** y **Word Embeddings**; y para el segundo paso se emplearán las técnicas **Cosine Similarity** y **Word Mover's Distance -WMD-**.

Una vez obtenidas estas mediciones de similitud entre los Curriculum Vitae de los candidatos y las descripciones de los puestos laborales de IT, estos valores se utilizarán para alimentar un algoritmo de clustering K-means que a su vez, con sus datos de salida (4 clusters), alimentará a un modelo de clasificación KNN. Finalmente este modelo KNN nos servirá para lograr, en base a los valores de similitud de nuevos candidatos, clasificar qué tan similares son dichos candidatos con respecto a la descripción de un puesto de IT: similitud escasa, similitud media, similitud alta, similitud muy alta.

1.1 Objetivos del Proyecto.

1.1.1 Objetivo general.

El objetivo de este Proyecto de Tesis es lograr un desarrollo, tanto teórico como práctico, de un *sistema de lectura automática de Curriculum Vitae* accesible vía Web. La finalidad del mismo es ayudar al reclutador laboral a elegir a los mejores candidatos para los puestos laborales de IT que tenga disponible. Esta elección se realiza mediante el uso de algoritmos de Machine Learning y basándose, principalmente, en una medición de similitud entre textos:

- los Curriculum Vitae de los candidatos por un lado,
- y descripciones de los puestos laborales de IT por el otro.

1.1.2 Objetivos específicos.

Los objetivos específicos de este Proyecto de Tesis son:

- Describir el estado del arte actual de los Sistemas de lectura y análisis de Curriculum Vitae en Recruiting.
- Implementar un Sistema de lectura automática de Curriculum Vitae basado en la comparación entre textos, para finalmente obtener una visualización de los mejores candidatos para un puesto laboral de IT determinado.
- Aprender los conceptos y técnicas principales utilizadas dentro del procesamiento de lenguaje natural (NLP) aplicando técnicas de preprocesamiento y limpieza de textos.
- Implementar diferentes técnicas para medir similitudes entre los textos (Cosine Similarity y Word Mover's Distance -WMD-) y diferentes algoritmos de vectorización (TF-IDF y Word Embeddings), analizando su funcionamiento tanto teórica como matemáticamente, ventajas y desventajas.
- Conocer, implementar e integrar el algoritmo de clustering K-means junto al algoritmo de clasificación KNN para obtener un modelo de clasificación de candidatos en base a las medidas de similitud entre los textos.
- Evaluar los Frameworks disponibles para tener una UI accesible vía web e integrar el mismo al Sistema.
- Almacenar datos de candidatos, reclutadores y puestos laborales en una base de datos; contando con un sistema de login y registración para los mismos.

1.2 Alcance del Proyecto.

El alcance de esta Tesis de Grado de Ingeniería incluye el desarrollo de conceptos de análisis de datos y machine learning, procesamiento de lenguaje natural, técnicas de preprocesamiento y limpieza de los datos, algoritmos de vectorización y técnicas para medir la similitud entre textos, algoritmos de clasificación y clustering, integración con frameworks, visualización de datos, y gestión de Base de Datos, de acuerdo a lo enunciado en los objetivos específicos.

1.3 Organización.

Este Proyecto de Tesis fue organizado para trabajarlo en tres secciones:

1. Análisis e investigación inicial.

Esta sección abarca principalmente la parte teórica del trabajo, haciendo hincapié en el análisis e investigación de:

- El estado del arte (actual y pasado) de los sistemas de lectura y análisis de Curriculum Vitae.
- Técnicas usadas para el procesamiento del lenguaje natural (NLP).
- Técnicas para medir similitudes entre textos: Cosine Similarity y WMD.
- Algoritmos de Vectorización: TF-IDF y Word Embeddings.
- Algoritmos de Machine Learning para tareas de clasificación (KNN) y clustering (K-means).

Esta primera sección abarca los capítulos *Reclutamiento laboral en IT*, *Algoritmos de Machine Learning* y *Natural Language Processing* de este Informe de Tesis.

2. Investigación e implementación de distintas técnicas y algoritmos para la obtención del modelo de clasificación KNN.

Esta sección hace referencia a la aplicación práctica dentro del marco teórico desarrollado en la primera sección, mediante la realización de una serie de análisis en documentos de Jupyter Notebook ¹ utilizando Python ², para la obtención final de nuestro modelo de clasificación KNN capaz de clasificar, en base a los valores de similitud de nuevos candidatos, qué tan similares son dichos candidatos con respecto a la descripción de un puesto de IT: similitud escasa, similitud media, similitud alta, similitud muy alta. Esta sección abarca los siguientes items:

- Obtención de sets de datos: curriculums vitae y descripciones laborales.

¹Aplicación cliente-servidor que permite crear documentos web en formato JSON que siguen un esquema versionado y una lista ordenada de celdas de entrada y de salida. Estas celdas albergan, entre otras cosas, código, texto (en formato Markdown), fórmulas y ecuaciones matemáticas. Estos documentos que se generan funcionan en cualquier navegador estándar.

²Lenguaje de programación interpretado y multiplataforma de código abierto, popularizado en los últimos años por su facilidad para trabajar con inteligencia artificial, big data, machine learning y data science, entre muchos otros campos en auge.

- Preprocesamiento de los textos.
- Comparación entre textos y obtención de similitudes entre los mismos mediante el uso de las técnicas para medir distancias y obtener dichas similitudes (WMD y Cosine Similarity) y los algoritmos de vectorización (TF-IDF y Word Embeddings).
- Obtención del modelo de clasificación KNN utilizando como datos de entrada los clusters devueltos por el algoritmo K-means obtenidos en base a las mediciones de similitud previamente realizadas.
- Análisis y primeras visualizaciones de los resultados.

Esta sección abarca el capítulo *Implementación* (desde *Obtención del modelo de clasificación* hasta *Clasificación de nuevas muestras y resultados obtenidos*) de este Informe de Tesis.

3. Investigación e integración al sistema web.

Esta última sección hace referencia a la reutilización de las funciones que contenían la lógica de los distintos algoritmos utilizados junto con el modelo de clasificación KNN obtenidos previamente en la sección 2, para integrar todo esto en el sistema Web que, a su vez, está integrado a una base de datos relacional. De esta manera, el sistema cuenta con una interfaz gráfica permitiendo interactuar entre candidatos y reclutadores y, principalmente, permitiendo que el reclutador sea capaz de obtener un 'TOP' de los 'N' mejores candidatos para los puestos que el reclutador desee consultar (siempre y cuando los candidatos hayan aplicado a dicho puesto). y su integración a un sistema web capaz de manejar las actividades entre los Candidatos y Reclutadores dentro del sistema. Los items principales son:

- Definición de los usuarios que accederán al sistema.
- Definición de los datos que se almacenarán.
- Integración de frameworks y bases de datos.
- Modelado, filtrado y visualización de los datos.
- Evaluación del funcionamiento de todo el Sistema integrado.

Esta última sección abarca el capítulo *Implementación* (desde *Integración al Sistema Web* hasta el final del capítulo) de este Informe de Tesis.

2 Reclutamiento laboral en IT.

En este capítulo se va a realizar una introducción a la historia del Reclutamiento laboral, especialmente en el sector de IT, sus etapas y problemáticas. Luego se realizará un análisis del Estado de Arte actual de los Sistemas de lectura y análisis de CV actuales.

2.1 Introducción.

FALTA

El proceso de contratación fue evolucionando a lo largo del tiempo. En el modelo de contratación de primera generación, las empresas anunciaban sus vacantes de puestos laborales en diarios, revistas, radio y televisión. Los candidatos enviaban sus currículums por correo postal y sus currículums se clasificaban manualmente. Una vez preseleccionados los candidatos, el equipo de contratación llamaba a los mismos para realizar rondas de entrevistas. Este fue un procedimiento que llevó mucho tiempo. Luego de esto pasamos a la segunda generación. En esta época las empresas comenzaron a crecer y también lo hicieron las necesidades de contratación. Las empresas empezaron a subcontratar su proceso de contratación naciendo de esta manera las consultoras o agencias de contratación. Estas consultoras requerían que los solicitantes cargaran sus currículums en sus sitios web en formatos particulares. Luego, las consultoras revisaban los datos de los candidatos y preseleccionaban a los mismos para la empresa. El gran inconveniente de este proceso fue que habían numerosas consultoras y cada una tenía su propia y única forma de selección. Para superar todos los problemas anteriores, se llegó a una tercera generación, en la que estamos actualmente. En esta generación se crearon y siguen creándose sistemas con algoritmos inteligentes que ayudan a las empresas y consultoras a analizar la información de cualquier curriculum vitae y clasificarla en función de los puestos laborales disponibles. De esta manera, cuando el empleador publica una oferta de trabajo, estos sistemas clasifican a los currículums basándose en distintas métricas (por ejemplo palabras clave) mostrando así los candidatos más relevantes para el empleador.

2.2 Sistemas de lectura y análisis de CV : Estado del arte.

FALTA

Las empresas y agencias de contratación procesan diariamente una gran cantidad de Curriculum Vitae. Esta no es una tarea para humanos: se requiere de un sistema automatizado e inteligente que sea capaz detectar a los mejores candidatos y con ello realizar una clasificación para los distintos puestos laborales disponibles.

Hay varios enfoques para realizar esta tarea: -Métodos simples: -Sistemas de recomendación. -Extracción de keywords y comparaciones:... - -Comparaciones simples entre los textos de los currículums y las descripciones de puestos. -Comparación de similitudes entre textos. / Sistemas que utilicen algoritmos de similitud entre documentos:

El enfoque número 2 será el utilizado para el sistema desarrollado para este Proyecto de Tesis. Se decidió utilizar este enfoque...

2.2.1 Problemáticas.

FALTA Habiendo realizado la comparación con cualquiera de los enfoques previamente descritos, los currículums son difíciles de analizar. Esto se debe a que varían en los tipos de información, su orden, estilo de escritura, etc.

-Diferentes técnicas para medir similitud de textos.

2.2.2 Trabajos relacionados.

A su vez, hay varios trabajos realizados en este campo. Estos sistemas incluyen: - - - -

La diferencia de este trabajo con el resto de los trabajos anteriormente mencionados es principalmente que se agrega WMD como una de las medidas de similitud entre los textos. WMD es una muy reciente y en el futuro puede llegar a Además, se integró el sistema a una interfaz web, cosa que la mayoría no realizó. Por último, cabe destacar que estos trabajos no tienen un fácil acceso a los datasets ni al código que utilizaron para dichos sistemas, por lo que seguir el trabajo de ellos aplicando las mejoras que mencionan en sus trabajos es una tarea casi imposible. En cambio, este sistema será de código abierto: los datasets y códigos utilizados estarán disponibles en Github y Git LFS.

A su vez, en el enfoque utilizado, pueden haber muchos más sub-enfoques?: (ver <https://towardsdatascience.com/the-best-document-similarity-algorithm-in-2020-a-beginners-guide-a01b9ef8cf05>). -1er enfoque (traditional statistical approach): TF-IDF and Cosine Sim. ->you can easily start your own document similarity on your local laptop. No fancy GPU is necessary. No large memory is necessary. With high-quality data, you will still get competitive results. -Enfoque Deep Learning: USE y BERT... y aca tambien entraría el word movers.->Granted, if you want to do other tasks such as sentiment analysis or classification, deep learning should suit your job.

3 Algoritmos de Machine Learning.

Para comenzar el marco teórico de esta tesis, es necesario explicar qué es Machine Learning y cómo se pueden clasificar a los distintos algoritmos según su tipo de aprendizaje. En este capítulo se va a explicar el objetivo del aprendizaje supervisado y no supervisado en Machine Learning haciendo énfasis en los algoritmos K-Nearest Neighbor (KNN) y K-means.

3.1 Introducción.

FALTA Los métodos que utilizaremos en este Proyecto de Tesis serán...

3.2 K-Nearest Neighbor (KNN).

FALTA K-Nearest Neighbor (o K Vecinos más Próximos en español), blablabal....

3.3 K-Means.

FALTA K-Means (o K-Medias en español), blablaba...

3.3.1 Elbow Method.

FALTA Elbow Method (o Método del codo en español), blablaba...

4 Natural Language Processing.

FALTA

4.1 Introducción.

FALTA

4.2 Preprocesamiento de textos.

FALTA

4.3 Similitud entre textos.

FALTA

4.4 Técnicas para medir Similitud entre textos.

FALTA

4.4.1 Cosine Similarity.

FALTA

Cosine Similarity se basa en la medición del coseno del ángulo entre dos vectores proyectados en un espacio multidimensional para lograr medir la similitud de los documentos (un menor ángulo indica una mayor similitud). En este contexto, estos dos vectores representan matrices que contienen el recuento de palabras de dos documentos. Una de sus limitaciones es que no tiene la habilidad de reconocer si las palabras que compara son semánticamente similares.

4.4.2 Word Mover's Distance (WMD).

FALTA

Anteriormente, mencionamos que una de las limitaciones de Cosine Similarity es que no tiene habilidad de reconocer si las palabras que compara son semánticamente similares. En cambio, Word Mover's Distance (WMD) es un algoritmo más complejo que sí permite reconocer las relaciones semánticas; su limitación son las relaciones sintácticas (VER.....). WMD se basa en word embeddings y permite medir la distancia entre documentos (una menor distancia indica una mayor similitud).

4.5 Algoritmos de vectorización.

FALTA

Previamente a utilizar Cosine Similarity y WDM para (—completar—) se debe emplear algún algoritmo de vectorización que permita representar las palabras de nuestros textos a un espacio vectorial. De esta forma Cosine Similarity y WMD podrán interpretarlos de la mejor manera. Como algoritmos de vectorización se utilizarán TF-IDF y Word Embeddings.

4.5.1 TF-IDF.

FALTA

El algoritmo TF-IDF asigna valores numéricos a las palabras en función de la frecuencia con que aparecen en los textos para medir la frecuencia de ocurrencia de un término en la colección de documentos, expresando cuán relevante es una palabra para un documento en una colección.

4.5.2 Word Embeddings.

FALTA

Los Word Embeddings son necesarios para utilizar WMD. Los Word Embeddings son una de las variantes más populares para representar textos. Son vectores previamente entrenados y generados mediante un modelo de red neuronal secuencial; de esta manera son capaces de capturar los contextos de una palabra en el documento llegando a poder contener información semántica y sintáctica.

4.5.2.1 ¿Cómo entrenar Word Embeddings?

FALTA

5 Implementación.

La implementación de este Sistema se trabajó en dos grandes partes:

1. Obtención del modelo de clasificación.

En esta primera parte se obtuvieron y preprocesaron datasets de Curriculum Vitae de distintos candidatos y descripciones de puestos de trabajo de IT publicados por distintas empresas, para luego ser comparados y obtener similitudes entre los textos utilizando las técnicas para medir distancias y obtener dichas similitudes (WMD y Cosine Similarity) y los algoritmos de vectorización (TF-IDF y Word Embeddings).

Una vez obtenidas estas mediciones de similitud entre los Curriculum Vitae de los candidatos y las descripciones de los puestos laborales de IT, estos valores se utilizaron para alimentar un algoritmo de clustering K-means que a su vez, con sus datos de salida (4 clusters), alimentan a un modelo de clasificación KNN. Finalmente, con este modelo KNN logramos, en base a los valores de similitud de nuevos candidatos, clasificar qué tan similares son dichos candidatos con respecto a la descripción de un puesto de IT: similitud escasa, similitud media, similitud alta, similitud muy alta.

Estos análisis se realizaron en documentos de Jupyter Notebook utilizando Python; y sirvieron para evaluar el comportamiento del modelo de clasificación y los distintos algoritmos de medición de similitudes para luego ser utilizados en la siguiente etapa.

2. Integración al Sistema Web.

Etapas posteriores a la primera parte. Una vez observado que los resultados fueron los esperables, lo que se hizo fue reutilizar las funciones que contenían la lógica de los distintos algoritmos utilizados junto con el modelo de clasificación KNN obtenidos previamente en la parte 1, para integrar todo esto en el sistema Web. Este sistema web está realizado en Django ³, y cuenta con una base de datos relacional que contiene la información de los candidatos y reclutadores junto con los Curriculum Vitae y puestos que hayan cargado.

De esta manera, nuestro sistema cuenta con una interfaz gráfica permitiendo interactuar entre candidatos y reclutadores y, principalmente, permitiendo que el reclutador sea capaz de obtener un listado con todos los candidatos que aplicaron a un puesto determinado, ordenados de mayor a menor en cuanto a su *similitud* con dicho puesto. Esta *similitud* representa el resultado obtenido de la clasificación por nuestro modelo KNN.

³Framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo-vista-controlador.

5.1 Obtención del modelo de clasificación.

5.1.1 Introducción.

Como inicio definamos qué es un modelo. En nuestro caso, un modelo representa Este modelo se construyó en base a ... La implementación de este Sistema se realizó en Python...

5.1.2 Esquema.

Como podemos ver la figura 5.1 representa el procedimiento utilizado para la obtención de nuestro modelo de clasificación.

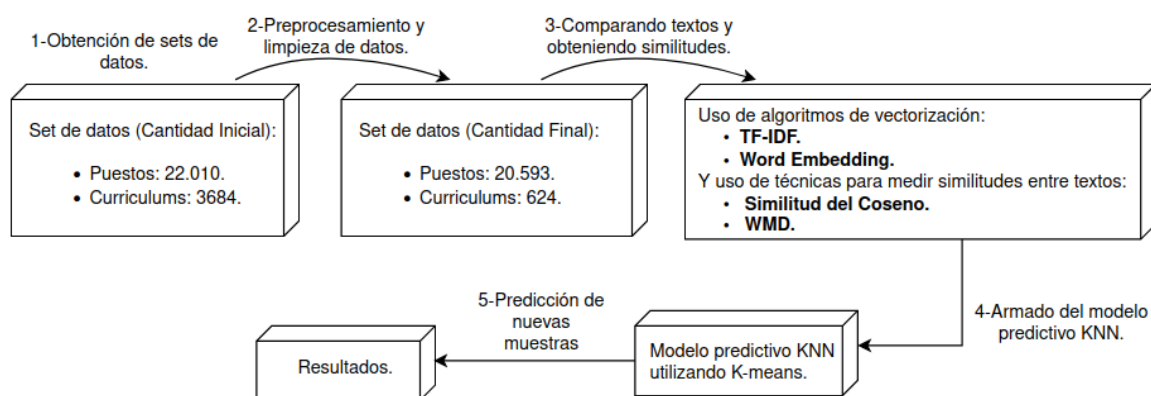


Figura 5.1: Pipeline Flow para la obtención del modelo de clasificación KNN

5.1.3 Obtención de sets de datos.

En primer lugar debemos definir qué es un set o conjunto de datos. Un set o conjunto de datos es una tabla de una base de datos o, matemáticamente, una matriz estadística de datos. Cada columna de la tabla representa una variable del set de datos; y cada fila representa a un miembro determinado del mismo.

Para este Proyecto utilizamos dos grandes sets de datos que se obtuvieron mediante la recolección de distintos archivos alojados en la Web, los cuales estan descriptos a continuación.

5.1.3.1 Curriculum Vitae.

Los set de datos de Curriculum Vitae de los candidatos se obtuvieron de las siguientes fuentes:

1. 228 Curriculums en formato docx y posteriormente convertidos a pdf, obtenidos del sitio Kaggle⁴. Estos pdfs son candidatos de la India con experiencia en el rubro de IT.
2. 2484 Curriculums en formato CSV, obtenidos del sitio Kaggle⁵. Este CSV cuenta con CVs obtenidos del sitio web de postulación de trabajos 'livecareer.com'.
3. 962 Curriculumns en formato CSV, obtenidos del sitio Kaggle⁶. Este CSV cuenta con CVs repartidos en distintas categorías de IT.
4. 10 Curriculums en formato PDF, los cuales los cuales fueron obtenidos como ejemplos mediante una recolección propia de distintos sitios web.

5.1.3.2 Descripciones Puestos Laborales.

Los set de datos de descripciones de puestos laborales se obtuvieron de las siguientes fuentes:

1. 22.000 descripciones en formato CSV; obtenido del sitio Kaggle⁷. El CSV cuenta con descripciones de puestos obtenidos del sitio web de USA de postulación de trabajos del rubro de IT 'Dice.com'.
2. 10 descripciones en formato CSV; obtenidas como ejemplos mediante una recolección propia del sitio Indeed⁸ para puestos de trabajo de IT.

⁴<https://www.kaggle.com/palaksood97/resume-dataset>

⁵<https://www.kaggle.com/snehaanbhawal/resume-dataset>

⁶<https://www.kaggle.com/gauravduttakiit/resume-dataset>

⁷<https://www.kaggle.com/PromptCloudHQ/us-technology-jobs-on-dicecom>

⁸<https://www.indeed.com/q-USA-jobs.html>

5.1.4 Preprocesamiento y limpieza de datos.

Previamente a utilizar las técnicas para medir distancias y obtener similitudes entre textos (WMD y Cosine Similarity) y los algoritmos de aprendizaje (KNN y K-Means) necesitamos que los datos que comparemos e introduzcamos en los algoritmos estén lo más limpios posible; ya que de lo contrario las mismos podrían clasificar o predecir de forma errónea. Este análisis previo sobre los datos debe ser minucioso ya que puede haber valores incoherentes o absurdos.

El procedimiento para la Limpieza de los Curriculum Vitae y las descripciones de los puestos laborales fue el siguiente:

1. Convertimos todo a minúscula.
2. Eliminamos datos no relevantes para nuestros análisis (mails y páginas web).
3. Eliminamos signos de puntuación y caracteres especiales (incluyendo números).
4. Eliminamos stop words.
5. Eliminamos common words no relevantes para nuestros análisis.
6. Aplicamos Lematización y Tokenización.
7. Eliminamos repetidos.
8. Obtenemos y usamos bi-gramas.

Luego de aplicar preprocesamiento y limpieza de datos nos quedarán los siguientes tamaños de nuestros datasets:

- 624 CVs de candidatos (en formato pdf y csv).
- 20593 descripciones de puestos de IT (en formato csv).

5.1.5 Cantidad final del set de datos y su uso en las distintas etapas.

El total de 624 CVs de candidatos y 20593 descripciones de puestos de IT que mencionamos previamente, serán utilizados para el entrenamiento y obtención de vectores mediante TF-IDF (para el posterior cálculo de Cosine Similarity) y para el entrenamiento de Word2Vec y obtención de los Word Embeddings (para el posterior cálculo de WMD).

Por otro lado, para calcular Cosine Similarity y WMD, para utilizarlos en K-means y para entrenar a nuestro algoritmo KNN, utilizaremos únicamente una porción de nuestros datasets:

1-Para el cálculo de Cosine Similarity y WMD:

- 301 CVs de candidatos.
- 201 descripciones de puestos de IT.

Nota: No obstante, al realizar los cálculos de distancias compararemos cada CV con cada Job Description, obteniendo un dataframe total de 3131 filas con sus respectivos valores de WMD y Cosine Sim.

2-Para el uso de K-means y entrenamiento con KNN (eliminamos un CV y una descripción de puesto IT que los utilizamos en '3-'):

- 300 CVs de candidatos.
- 200 descripciones de puestos de IT.

Nota: como se comentó previamente, nos quedarán 3000 filas / puntos para usar en K-means y entrenar KNN; llegando a representar estos 3000 puntos en un plano de 2 dimensiones.

3-Para la clasificación de nuevas muestras mediante KNN:

- 1 CV de candidatos.
- 1 descripción de puesto de IT.

Nota: como se comentó previamente, nos quedarán 131 filas para clasificar.

¿Por qué utilizamos solo una porción de nuestros datasets?: Esto es debido a los drawbacks de WMD y KNN.

- WMD: posee una alta complejidad en el cálculo de la distancia, teniendo un tiempo de ejecución muy elevado. Como ejemplo, al correrlo localmente, el cálculo de WMD para 3131 filas tardó 7 horas; frente a los 3 segundos que tardó el cálculo de Cosine Similarity para la misma cantidad de filas.
- KNN: KNN es una gran opción para datasets pequeños con pocas variables de entrada; pero tiene problemas cuando la cantidad de entradas es muy grande. Cada variable de entrada puede considerarse una dimensión de un espacio de entrada p-dimensional. En grandes dimensiones, los puntos que pueden ser similares pueden

tener distancias muy grandes. Además, cada vez que se va a hacer una predicción con KNN, busca al vecino más cercano en el conjunto de entrenamiento completo. Por esto, se debe utilizar un dataset pequeño para que el clasificador KNN completa su ejecución rápidamente.

En conclusión, al utilizar solo una porción de nuestros datasets para obtener los distintos cálculos de distancias y entrenar KNN, el cálculo de WMD se podrá realizar en un tiempo finito, y nuestro clasificador KNN funcionará rápida y eficientemente al realizar predicciones.

5.2 Comparando textos y obteniendo similitudes.

FALTA

Previamente a utilizar Cosine Similarity y WDM para obtener las medidas de similitud entre los textos, se debe emplear algún algoritmo de vectorización que permita representar las palabras de nuestros textos a un espacio vectorial. De esta forma Cosine Similarity y WMD podrán interpretarlos de la mejor manera. Como mencionamos previamente, como algoritmos de vectorización se utilizarán TF-IDF y Word Embeddings.

5.3 Armado del modelo de clasificación KNN.

FALTA

Una vez obtenidas estas mediciones de similitud entre los Curriculum Vitae de los candidatos y las descripciones de los puestos laborales de IT, estos valores se utilizarán para alimentar un algoritmo de clustering K-means que a su vez, con sus datos de salida (4 clusters), alimentarán a un modelo de clasificación KNN. Finalmente, con este modelo KNN lograremos, en base a los valores de similitud de nuevos candidatos, clasificar qué tan similares son dichos candidatos con respecto a la descripción de un puesto de IT: similitud escasa, similitud media, similitud alta, similitud muy alta.

5.4 Clasificación de nuevas muestras y resultados obtenidos.

FALTA

5.5 Integración al Sistema Web.

FALTA

Anteriormente lo que se hizo fue un análisis mediante documentos en Jupyter Notebooks para evaluar el comportamiento del modelo de clasificación y los distintos algoritmos de medición de similitudes.

Al observar que los resultados fueron los esperables, lo que se hizo en esta última etapa fue reutilizar las funciones que contenían la lógica de los distintos algoritmos utilizados junto con el modelo de clasificación KNN obtenidos en la fase previa, para integrar todo esto en el sistema Web.

Como mencionamos previamente, este sistema web está realizado en Django, y cuenta con una base de datos relacional que contiene la información de los candidatos y reclutadores junto con los Curriculum Vitae y puestos que hayan cargado.

De esta manera, nuestro sistema cuenta con una interfaz gráfica permitiendo interactuar entre candidatos y reclutadores y, principalmente, permitiendo que el reclutador sea capaz de obtener un listado con todos los candidatos que aplicaron a un puesto determinado, ordenados de mayor a menor en cuanto a su *similitud* con dicho puesto. Esta *similitud* representa el resultado obtenido de la clasificación por nuestro modelo KNN.

El sistema web contará con 2 tipos de usuario:

- Candidato: quienes cargarán en el sistema sus Curriculum Vitae y aplicarán a los distintos puestos disponibles.
- Reclutador: quienes cargarán en el sistema los puestos de trabajo que tengan disponibles y podrán consultar, entre otras cosas, un listado con todos los candidatos que aplicaron a un puesto determinado, ordenados de mayor a menor en cuanto a su similitud con dicho puesto.

5.5.1 Base de datos.

Nuestros datos los almacenaremos en una base de datos **FALTA definir cual**.

Para modelar y gestionar nuestros datos utilizamos el modelo relacional ⁹.

Sacar la mayoría del documento modelo-entidad-relacion-case-method-richar-barker.pdf

Para comprender los datos que se almacenan en dicha base de datos, los representaremos utilizando un diagrama entidad relación ¹⁰. Previamente a esto explicaremos los elementos del diagrama de entidad relación:

FALTA PONER IMAGEN CON LOS ELEMENTOS: Entidad rectángulo, Unión entre entidades s
las líneas que puede ser obligatoria u opcional, cardinalidad son los 1:M / 1:1 / M:M

- Entidad: objeto concreto o abstracto que figura en nuestra base de datos. Por ejemplo una entidad puede ser un alumno, un cliente, una empresa, etc. Dentro de las entidades estan los atributos, atributos principales o clave primaria (PK) y atributos foraneos o clave secundaria (FK). Las entidades que necesitamos para crear nuestra BD son: Candidato, Puesto, Reclutador y Candidato_Puesto -entidad intermedia entre Candidato y Puesto-.
- Unión entre entidades: pueden ser obligatorias u opcionales. En nuestro diagrama nuestras uniones son todas opcionales, ya que el reclutador puede o no CARGAR un puesto, el candidato puede o no APLICAR a un puesto y a su vez el puesto puede o no ser aplicado por un candidato.
- Cardinalidad: Relación entre entidades o mapeo. La cardinalidad es el tipo de relación entre entidades. Observando la figura 5.2 y considerando que los rectángulos azules son una entidad y los naranjas son otra entidad observamos que pueden haber 4 tipos de cardinalidades posibles:
 1. Uno a uno: a cada entidad azul le corresponde solo una entidad naranja.
 2. Uno a muchos: a cada entidad azul le corresponde una o varias entidades naranjas.
 3. Muchos a uno: a cada entidad naranja le corresponde una o varias entidades azules.
 4. Muchos a muchos: las entidades azules pueden tener varias entidades naranjas y las entidades naranjas también pueden tener varias entidades azules.

⁹Una base de datos relacional es un conjunto de una o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común.

¹⁰Un modelo entidad-relación es una herramienta para el modelo de datos, la cual facilita la representación de entidades de una base de datos.



Figura 5.2: Tipos de cardinalidad.

La cardinalidad entre nuestras entidades son:

- Entre Candidato y Puesto existe una relación muchos a muchos (M:M), ya que un candidato puede aplicar a M puestos y un puesto puede ser aplicado por M candidatos. Es por esto que se creó la tabla intermedia Candidato_Puesto conllevando dos relaciones uno a muchos (1:M) con Puesto y Candidato.
- Entre Reclutador y Puesto existe una relación de uno a muchos (1:M), ya que un reclutador puede cargar M puestos, y un puesto pertenece a un solo reclutador.

En cuanto a las claves pueden haber dos tipos. Por un lado está la clave primaria o atributo principal (PK) es única y toda entidad debe tener la suya. Pueden haber múltiples PKs; estas se llaman PKs compuestas. Y por el otro está la clave secundaria o atributo foráneo (FK). Estas claves identifican a una entidad externa en otra, utilizándose para generar relaciones entre nuestras entidades. Si tenemos una clave FK en una entidad, significa que dicha clave FK es clave PK en otra entidad.

El diagrama de relación que utilizamos para nuestro trabajo lo observamos en la figura 5.3.

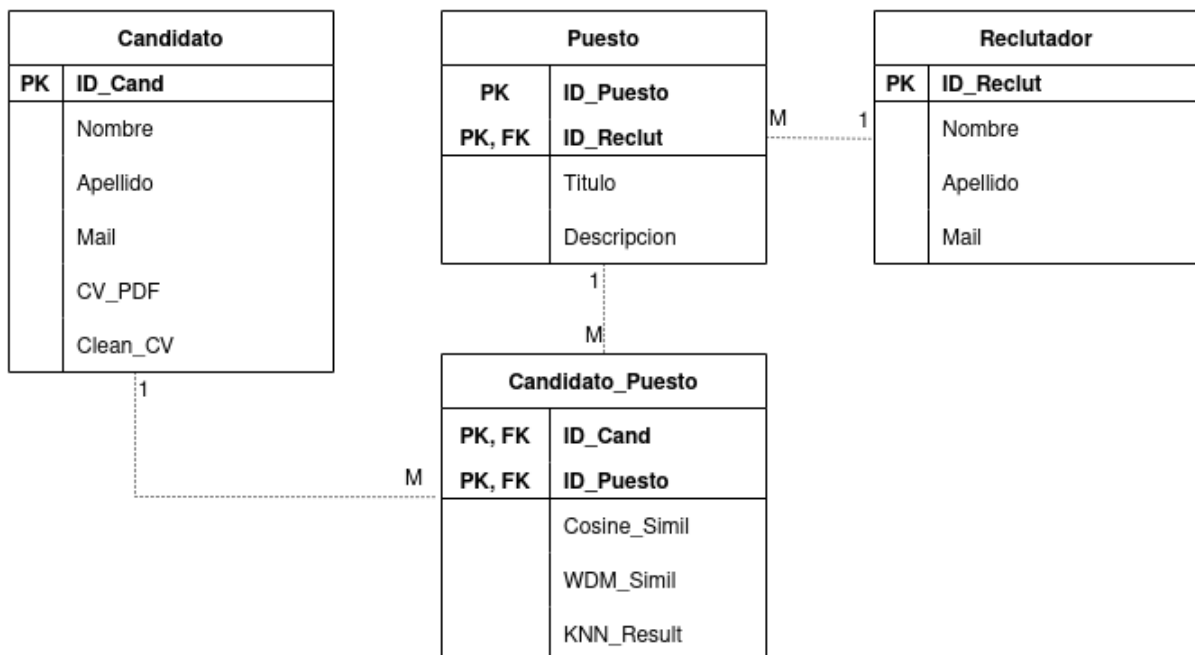
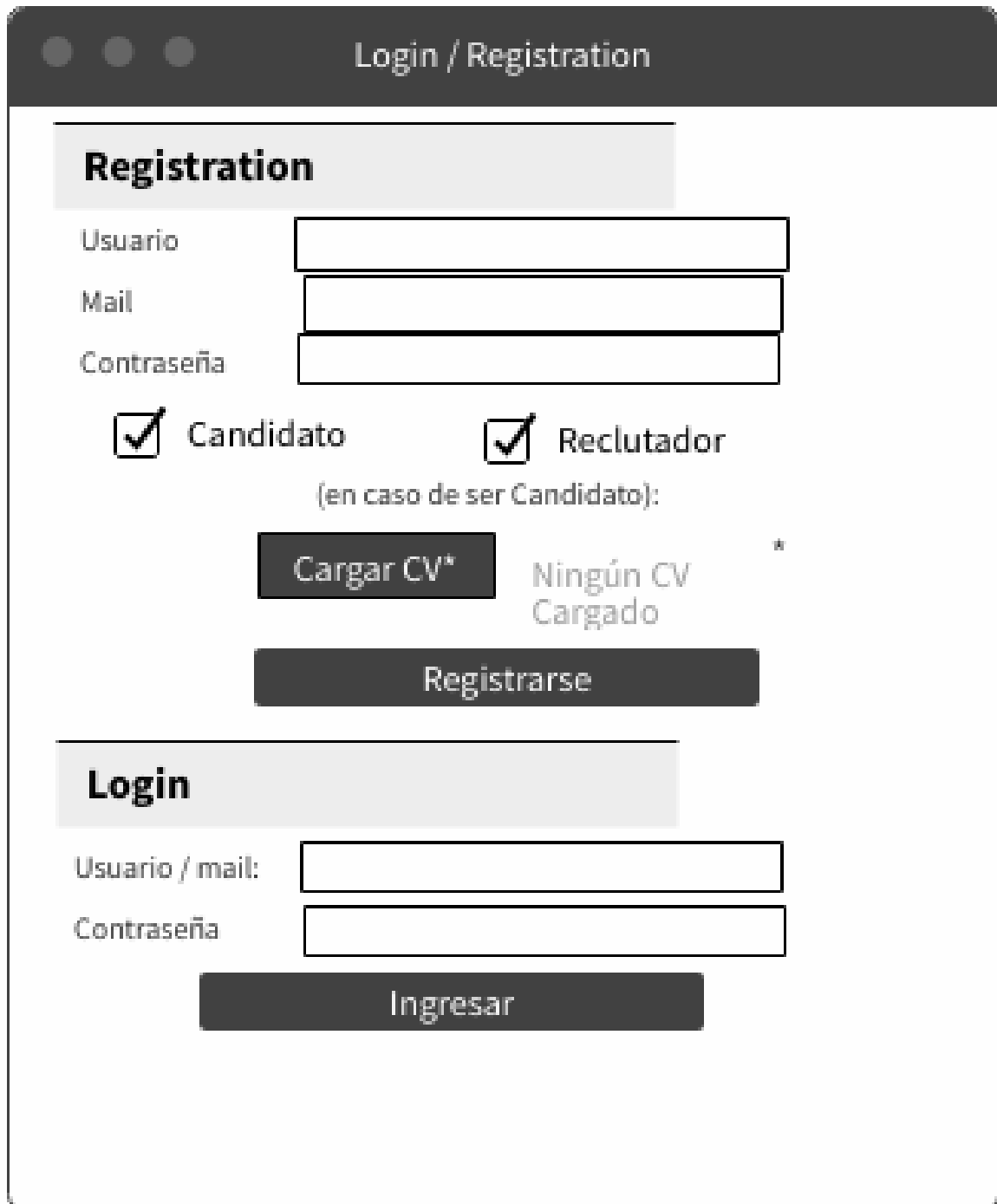


Figura 5.3: Diagrama de relación utilizado.

5.5.2 Secciones del sistema

Para registrarse o loguearse al sistema, se implementará la interfaz provista en la figura 5.4.



The image shows a wireframe of a web interface titled "Login / Registration". It is divided into two main sections: "Registration" and "Login".

Registration Section:

- Fields for "Usuario", "Mail", and "Contraseña" (Password).
- Two checkboxes: ☒ "Candidato" and ☒ "Reclutador".
- Text "(en caso de ser Candidato):" below the checkboxes.
- A button labeled "Cargar CV*" and a status message "Ningún CV Cargado" with an asterisk.
- A large "Registrarse" button.

Login Section:

- Fields for "Usuario / mail:" and "Contraseña".
- A large "Ingresar" button.

Figura 5.4: Logueo y Registración.

ES UN BOCETO, FALTA PONER LA IMAGEN REAL

El Candidato tendrá acceso al menú indicado en la figura 5.5.

Candidato

Mi Perfil

Nombre y apellido

DNI

Fecha nacimiento

Teléfono

Email

Domicilio

Actualizar datos

Cargar y analizar nuevo CV

Puestos disponibles

ID Puesto	Puesto	Descripcion Pu...	Ubicación
1	Programador F...	Descripcion larga	Buenos Aires
2	DB Engineer	Descripcion larga	Buenos Aires
3	Data Scientist	Descripcion larga	Buenos Aires

(se podrá filtrar/ordenar en cada columna)

Postularse

Postulaciones

ID Puesto	Puesto	Descripci...	Ubicación	Fecha Pos...
1	Programa...	Descripci...	Buenos Ai...	15/06/2021

(se podrá filtrar/ordenar en cada columna)

Figura 5.5: Vista del Candidato.

ES UN BOCETO, FALTA PONER LA IMAGEN REAL

Por su parte, el Reclutador tendrá acceso al menú indicado en la figura 5.6.

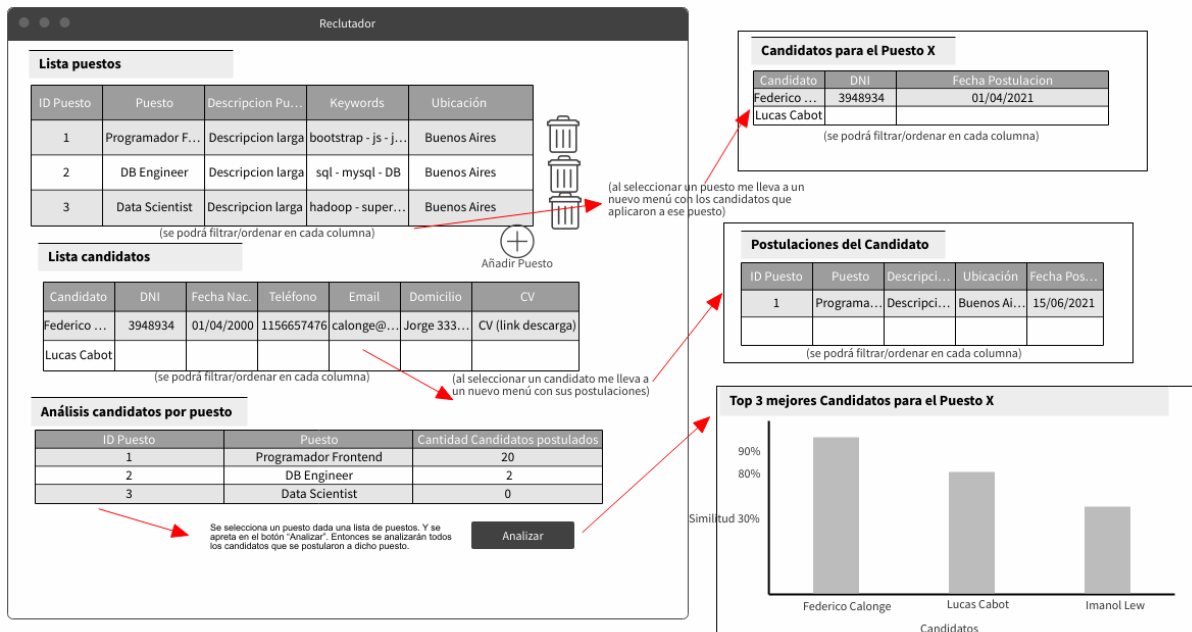


Figura 5.6: Vista del Reclutador.

ES UN BOCETO, FALTA PONER LA IMAGEN REAL

5.5.3 Manejo de los datos.

FALTA

5.5.3.1 Modelado.

FALTA

5.5.3.2 Filtrado.

FALTA

5.5.3.3 Visualización.

FALTA

5.6 Pipeline Flow final del Sistema.

FALTA

Una vez que el reclutador dentro de la sección observada en la figura 5.6 haga click en ".Analizar", el sistema reflejará el pipeline indicado en la figura 5.7 para obtener como resultado un listado con todos los candidatos que aplicaron al puesto, ordenados de mayor a menor en cuanto a su *similitud* con dicho puesto. Esta *similitud* representa el resultado obtenido de la clasificación por nuestro modelo KNN.

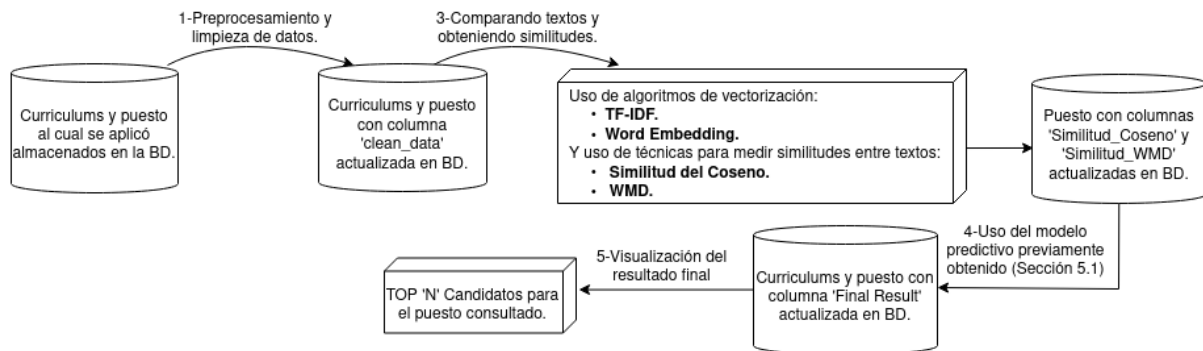


Figura 5.7: Pipeline Flow final del Sistema.

5.7 Conclusiones.

FALTA

5.8 Caso de Uso.

FALTA

5.9 Limitaciones del sistema.

FALTA

6 Próximos pasos.

FALTA - acá poner mejoras

7 Anexos.

FALTA

Referencias

- [1] Luis Argerich, Natalia Golmar, Damián Martinelli, Martín Ramos Mejía, & Juan Andrés Laura. (2019, Enero). *75.06, 95.58 Organización de Datos*. Apunte del Curso Organización de Datos, Universidad de Buenos Aires, Facultad de Ingeniería. (pp. COMPLETAR páginas).