ESCUELA UNIVERSITARIA DE POSGRADO

MACHINE LEARNING NO SUPERVISADO EN LA DETECCIÓN DE SIMILITUD DE PUESTOS DE EMPLEO DE PROFESIONALES DE TI

Línea de investigación:

Ingeniería de software, simulación y desarrollo de TICs

Tesis para optar el Grado Académico de Doctora en Ingeniería

Autora:

Mamani Rodriguez, Zoraida Emperatriz

Asesora:

Ángeles Lazo, Ana María

(ORCID: 0000-0003-1815-6700)

Jurado:

Mujica Ruiz, Oscar Hugo

Rodriguez Rodriguez, Ciro

Franco Del Carpio, Carlos Miguel

Lima - Perú

2022

Universidad Nacional Federico Villarreal

VRIN | VICERRECTORADO DE INVESTIGACIÓN

ESCUELA UNIVERSITARIA DE POSGRADO

MACHINE LEARNING NO SUPERVISADO EN LA DETECCIÓN DE SIMILITUD DE PUESTOS DE EMPLEO DE PROFESIONALES DE TI

Línea de Investigación:
Ingeniería de software, simulación y desarrollo de TICs

Tesis para optar el Grado Académico de Doctora en Ingeniería

Autora
Mamani Rodriguez, Zoraida Emperatriz

Asesora
Ángeles Lazo, Ana María
(ORCID: 0000-0003-1815-6700)

Jurado
Mujica Ruiz, Oscar Hugo
Rodriguez Rodriguez, Ciro
Franco Del Carpio, Carlos Miguel

Lima – Perú
2022

DEDICATORIA:

Dedico esta tesis a la memoria de mis padres: Francisco Mamani Quenta quien perdió la vida en esta pandemia, así como a mi madre Alicia Rodriguez Yarasca víctima de un fatal accidente de tránsito hace 35 años.

A mis hermanas Silvia y Gina por su apoyo constante, a mi sobrinita Katia por su alegría y a Dios por darme salud y fortaleza para continuar por la senda profesional que me he trazado.

Índice general

Lista de Figuras

Lista de Tablas

RESUMEN

Machine Learning no supervisado es una rama de la inteligencia artificial que utiliza técnicas automatizadas para resolver problemas basados en el descubrimiento de patrones o conglomerados de objetos según su posición geométrica en el espacio vectorial n dimensional, la calidad del agrupamiento depende de la complejidad, dimensionalidad y granularidad del dataset, de las estadísticas y de la distribución de los datos; Clustering es una técnica que recae en este rubro. Por otro lado, Las cualificaciones y perfiles ocupacionales estandarizados y actualizados es uno de los objetivos de las naciones, enfocados en mejorar la calidad y pertinencia de la educación y la formación para el trabajo; globalmente se cuenta con las cualificaciones ocupacionales ISCO-08 de la OIT y a nivel nacional con el CNPO y MNCP. En ese contexto, el presente trabajo realiza una investigación a partir de los puestos de empleo de profesionales de TI suministrados en los portales web por empleadores o grupos de interés, extrae las cualificaciones y su detalle, diseña un modelo dimensional, determina un modelo basado en clusters de puestos de empleo, aplica métricas y una técnica supervisada para evaluar la precisión del modelo, desarrolla un prototipo de aplicación y concluye fundamentando los beneficios que obtendría la academia disponiendo de una demanda social real y las entidades responsables de mantener actualizado el CNPO y MNCP con su implementación, extendiéndolo a otras disciplinas.

Palabras clave: machine learning no supervisado, clustering, k-means, dbscan

ABSTRACT

Unsupervised Machine Learning is a branch of artificial intelligence that uses automated techniques to solve problems based on the discovery of patterns or clusters of objects according to their geometric position in the n-dimensional vector space, the quality of the clustering depends on the complexity, dimensionality and granularity of the dataset, statistics and data distribution; Clustering is a technique that falls into this area. On the other hand, standardized and updated qualifications and occupational profiles is one of the objectives of the nations, focused on improving the quality and relevance of education and training for work; Globally, we have the occupational qualifications ISCO-08 of the ILO and at the national level with the CNPO and MNCP. In this context, the present work carries out an investigation based on the job positions of IT professionals provided in the web portals by employers or interest groups, extracts the qualifications and their detail, designs a dimensional model, determines a model based on clusters of jobs, applies metrics and a supervised technique to evaluate the accuracy of the model, develops an application prototype and concludes by substantiating the benefits that the academy would obtain by having a real social demand and the entities responsible for keeping the CNPO and MNCP updated. with its implementation, extending it to other disciplines.

Keywords: unsupervised machine learning, clustering, k-means, dbscan

I. Introducción

La Agenda de los objetivos para el desarrollo sostenible (ODS) al 2030 hace énfasis en mecanismos de facilitación tecnológica para apoyar diecisiete objetivos mediante iniciativas en ciencia, tecnología e innovación, asimismo la Organización para la Cooperación y el Desarrollo Económico (OECDE) refiere los empleos del futuro, señala la era digital y el impacto en el mercado laboral globalizado de las tecnologías digitales: como: IoT, Big Data, Inteligencia Artificial, BlockChain entre otras. La Organización Internacional del Trabajo (OIT) organismo de la ONU que tiene como funciones formular políticas y establecer programas que promuevan el trabajo decente para todos, adopta la Clasificación Internacional Uniforme de Ocupaciones (CIUO-08) con la finalidad de continuar siendo un modelo útil, base para la formulación de las clasificaciones nacionales y como consecuencia permita la comparación internacional, así como su intercambio. En el Perú se cuenta con el Catálogo Nacional de Perfiles Ocupacionales o Cualificaciones (CNPO), este organiza los perfiles ocupacionales en familias productivas vinculadas a las actividades económicas del país; recientemente se aprobó el Marco Nacional de Cualificaciones del Perú (MNCP) como instrumento para el reconocimiento de las cualificaciones para el desempeño en el mercado laboral, cuyo poblamiento será progresivo y deberá alinearse al CNPO y la certificación de competencia laborales.

Machine Learning no supervisado es una rama de la inteligencia artificial que utiliza técnicas automatizadas para resolver problemas basados en el descubrimiento de patrones o conglomerados de objetos según su posición geométrica en el espacio vectorial n dimensional, la calidad del agrupamiento depende de la complejidad, dimensionalidad y granularidad del dataset, de las estadísticas y de la distribución de los datos; Clustering es una técnica que recae en este rubro, este tipo de técnicas suele utilizarse cuando no se cuenta con datasets entrenados o se cuenta con grandes volúmenes de información, complementariamente se puede aplicar

técnicas supervisadas con fines de predictibilidad en la información según el contexto del negocio de interés.

Las cualificaciones o perfiles ocupacionales estandarizados y actualizados es uno de los objetivos de las naciones, enfocados en mejorar la calidad y pertinencia de la educación y la formación para el trabajo. La COVID-19 ha impactado severamente en el empleo y en los ingresos laborales, la sociedad se debe reinventar, debe replantear sus negocios con enfoques digitales, debe desarrollar la capacidad de resiliencia, acelerar el desarrollo de habilidades y destrezas digitales para recuperar la economía en la llamada "nueva normalidad".

En ese contexto, el presente trabajo realiza una investigación a partir de los puestos de empleo de profesionales de TI suministrados en los portales web por empleadores o grupos de interés, extrae las cualificaciones y su detalle, diseña un modelo dimensional, determina un modelo basado en clusters de puestos de empleo, aplica métricas y una técnica supervisada para evaluar la precisión del modelo, desarrolla un prototipo de aplicación y concluye contrastando con la clasificación CIUO-08 y CNPO, resaltando los beneficios que obtendría la academia disponiendo de una demanda social real y las entidades responsables de mantener operativo y actualizado el CNPO y MNCP con su implementación, extendiéndolo a otras disciplinas.

A continuación, se detalla el contenido del presente trabajo; En el capítulo I se describe el planteamiento del problema el cual comprende la fundamentación y formulación del problema, objetivos generales y específicos, justificación de la investigación. El capítulo II desarrolla el marco teórico; temas como Machine Learning no supervisado, Clustering, Puestos de empleo y Profesionales de Tecnologías de Información son tratados en este capítulo. En el capítulo III se aborda el método aplicado en el desarrollo de la propuesta. En el capítulo IV se describe los resultados. En el capítulo V se formula la discusión de resultados. En el capítulo VI se señalan las conclusiones. En el capítulo VII se precisan las recomendaciones. En el

capítulo VIII se consigna las referencias y finalmente el capítulo IX contiene los principales anexos.

1.1 Planteamiento del Problema

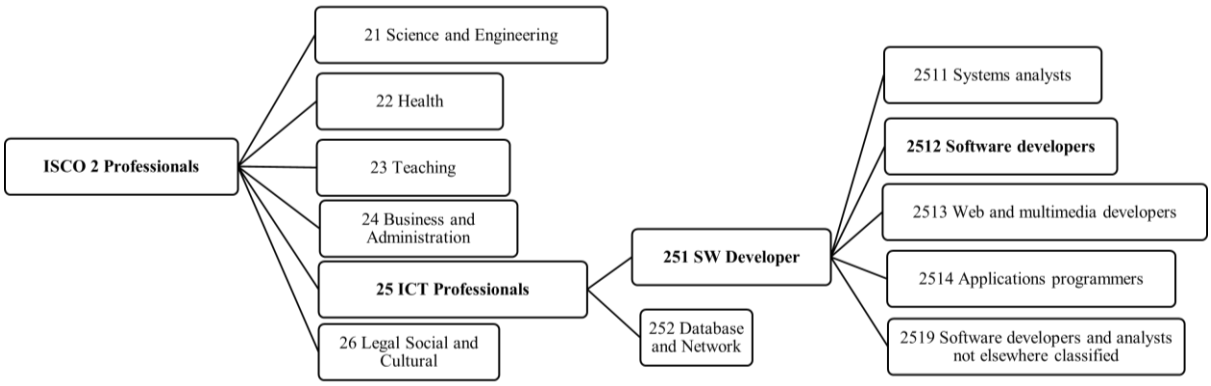
En las Perspectivas de la Organización para la Cooperación y el Desarrollo Económico en Ciencia, Tecnología e Innovación en América Latina 2016 (OECDE, 2016) y Sustainable Development Solutions Network (2017) analizan el impacto de los cambios tecnológicos en las economías y sociedades en un ámbito de quince años, generando nuevas oportunidades económicas; examinan las mega tendencias vinculadas con la ciencia, tecnología e innovación en diversos ámbitos como la Demografía, Recursos naturales y energía, cambio climático y medio ambiente, Globalización, entre otros; así como las tendencias tecnológicas del futuro como: IoT, Big Data, Inteligencia Artificial, BlockChain, complementariamente, la Agenda de los Objetivos para el desarrollo sostenible al 2030 hace énfasis en diecisiete objetivos y el mecanismo de facilitación tecnológica para apoyar los ODS mediante iniciativas en ciencia, tecnología e innovación (Naciones Unidas, 2019). Asimismo en la OECDE (2016) se documenta sobre los empleos del futuro, se precisa que en la era digital, las tecnologías digitales vienen impactando el mercado laboral, debido a que muchas tareas rutinarias, recurrentes han sido automatizadas y que algunas otras aún se mantienen inmunes a la automatización pero no por mucho tiempo por lo que se hace necesario que se innove en el ámbito académico, dotando a los futuros profesionales en determinadas competencias que les permita adaptarse a los vertiginosos cambios del futuro.

La Clasificación Internacional Uniforme de Ocupaciones (ISCO) es un sistema de clasificación de ocupaciones de trabajo, está basado en una clasificación jerárquica de 4 niveles la cual permite clasificar hasta en 436 unidades de agrupación (Organizacion Internacional del Trabajo, 2019). En la Figura 1 se puede apreciar una representación de la Clasificación de ocupaciones de trabajo según ISCO mediante la estructura de datos de tipo árbol con nodos.

Este sistema ha sido utilizado como base del Sistema de Clasificación de Ocupaciones Europeo (ESCO, 2020), sistema de clasificación multilingüe, en contraste al modelo ISCO, añade un nivel adicional de descripciones de ocupaciones y una taxonomía de habilidades, competencias y calificaciones.

Figura 1

Una rama del árbol de clasificación ISCO para profesionales



Nota. La Figura expone mediante una estructura basada en árbol, los perfiles profesionales TI ISCO. Adaptado de "Classifying online job advertisements through machine learning", por Boselli et al., 2018, *Future Generation Computer Systems, 1*(86).

Adicionalmente a los sistemas ISCO y ESCO, se cuenta con varias iniciativas que tienen por objetivo contribuir en mejorar los sistemas actuales con la finalidad de uniformizar a nivel global las ocupaciones de trabajo, sus habilidades y competencias debido a los cambios vertiginosos y enfocados a predecir los empleos del futuro.

1.2 Descripción del Problema

Los nuevos avances tecnológicos orientan a la academia a formar profesionales bajo un enfoque basado en competencias altamente especializadas y sostenibles, centrado en las necesidades de los grupos de interés quienes son representantes de los organismos públicos y/o privados que de alguna manera se ven afectados por la formación profesional que realiza la

academia al suministrar a la sociedad de profesionales de TI con capacidades no sostenibles debido a que no se estila medir el logro de las competencias al egresado.

Los portales laborales son páginas web especializadas y utilizadas por los empleadores como una pizarra electrónica para publicitar los anuncios de trabajo; este tipo de herramienta tecnológica se ha constituido en una valiosa fuente de información que viene siendo aprovechada para diversos fines en el mundo.

Se cuenta con iniciativas en el mundo basados en uniformizar los perfiles de ocupaciones de trabajo, habilidades y competencias, teniéndose grandes avances a nivel global, de los resultados de esas iniciativas se ha determinado que en el mundo se vienen generando dinámicamente nuevos perfiles laborales, nuevas habilidades, nuevas competencias debido a los cambios tecnológicos y las nuevas tendencias que permitan brindar soporte a los diecisiete objetivos de desarrollo sostenible establecido en la Agenda al 2030. (SDSN Australia/Pacific, 2017)

En ese contexto la presente investigación propone un modelo de Machine Learning no supervisado para la detección de similitud en los puestos de empleo de profesionales de Tecnologías de Información.

1.3 Formulación del Problema

1.3.1 Problema General

¿En qué medida la técnica de machine learning no supervisado influye en la detección de similitud de los puestos de empleo de profesionales de Tecnologías de Información?

1.3.2 Problemas Específicos

a) ¿En qué medida extraer los perfiles de empleo desde los portales laborales influye en disponer de información real de los Grupos de Interés?

b) ¿En qué medida el diseño y uso de una técnica de machine learning no supervisado influye en la clasificación de empleos por similitud de habilidades y/o capacidades de profesionales de Tecnologías de Información?

c) ¿En qué medida determinar la semejanza de los puestos de empleo de profesionales de TI permite contribuir en la formación de profesionales acorde a las necesidades de los Grupos de Interés?

d) ¿En qué medida contrastar los perfiles de empleo de profesionales de TI del Perú contra el clasificador estándar internacional permite conocer el grado de disparidad de la empleabilidad de profesionales de TI del Perú?

1.4 Antecedentes

En el contexto peruano no se ubicó trabajos de investigación sobre el tema propuesto, sin embargo, en el contexto internacional si se tiene una variedad de investigaciones como se resume a continuación:

Mansourvar y Yasin (2010) realizaron una investigación con el propósito de permitir al estudiante de la universidad de Malaya, a través de la creación de un portal, poder elegir qué cursos seguir para su formación profesional, así como también poder elegir su especialidad tomando en cuenta la información del mercado laboral de modo que al finalizar su carrera pudiera conseguir un empleo con facilidad, lo cual era muy distante de la realidad, ya que en Malasia el 70% de egresados con estudios superiores no conseguía empleo al terminar sus estudios. El portal fue desarrollado y luego utilizado por usuarios finales satisfactoriamente cumpliendo con los objetivos propuestos en la investigación.

El enfoque no supervisado para generar fragmentos estructurados informativos para motores de búsqueda de empleo propuesto en la investigación de Spirin y Karahalios (2013) consiste en generar datasets entrenados automáticamente a partir de una colección de ofertas de trabajo, describen un algoritmo de aprendizaje automático que consuma el dataset

entrenado, genere un modelo para generar fragmentos de información basados en requisitos y responsabilidades de las ofertas de empleo, el dataset entrenado debe contener instancias marcadas como positivas, las secciones que contengan información de valor de responsabilidades y requisitos de las ofertas de trabajo así como instancias irrelevantes las cuales serán marcadas como negativas. En la parte experimental utiliza la técnica de Maquina de soporte vectorial con un kernel lineal, así como la técnica basada en n-grams de la lingüística computacional.

Lynch (2017) realizó una investigación donde se propuso un sistema que clasifique los títulos de los anuncios de trabajo basados en el dominio, función y nivel de responsabilidad de un puesto de trabajo determinado usando Web Scraping para la obtención de la información y Minería de Datos para el análisis y la clasificación de los datos usando técnicas de machine learning supervisadas como bosques aleatorios y máquinas de soporte vectorial. Además, fue necesario realizar una etapa de preprocesamiento de texto para uniformizar los datos obtenidos. Como conclusión, establece la superioridad del modelo de máquina de soporte vectorial respecto a los bosques aleatorios.

La investigación de Marrara et al. (2017), presenta un enfoque de identificación de potenciales nuevas ocupaciones de trabajo aun no codificadas por la taxonomía internacional estándar ISCO. El enfoque se basó en análisis de texto y provee dos principales contribuciones en el contexto del Mercado laboral: 1°) apoyar a los expertos del Mercado laboral en identificar nuevas ocupaciones potenciales y el proceso de actualizar la taxonomía ISCO, 2°) Modelos de lenguaje son una forma para identificar nuevas ocupaciones o aquella ya codificadas en la taxonomía en términos de habilidades y competencias. En el enfoque propuesto se probó en un dataset de vacantes de empleos en inglés, obteniendo resultados prometedores.

En el trabajo de Chuan et al. (2018) se propuso un modelo semántico para mejorar la adecuación persona-trabajo para el reclutamiento de talentos en línea, para lo cual el autor

establece una representación semántica de los anuncios de empleo y las hojas de vida de los candidatos, en la parte experimental utiliza dataset de una compañía tecnológica de China y varias técnicas de machine learning supervisado como Regresión logística, Árbol de decisión, Adaboost, Bosques aleatorios y Gradient Boosting Decision Tree, para evaluar la precisión y eficiencia de los resultados.

La investigación de Boselli et al. (2018) se centra en la clasificación de ofertas de empleo en línea a través del aprendizaje automático supervisado, su contribución se delimita en la extracción de los anuncios de empleo de los portales web, aplica web scraping, el dataset es entrenado por expertos del dominio consignando los clasificadores ISCO para los perfiles y genera modelos de machine learning con las técnicas de Maquina de Soporte Vectorial (SVM) Linear, SVM RBF Kernel, Bosques aleatorios y Redes Neuronales, concluyendo que se obtuvo la mejor precisión con SVM Linear. Para la extracción de habilidades desde las ofertas de empleo utiliza el clasificador de texto n-gram, se depura los n-grams con baja significancia, participan expertos del dominio para establecer la clasificación de habilidades de ESCO.

En el trabajo de Vinel et al. (2019) sobre la comparación experimental de enfoques no supervisados para descubrir especializaciones de las profesiones que se ubican en el cuerpo de las vacantes laborales, evalúa experimentalmente varios métodos estadísticos de representaciones de vectores de texto: TF-IDF, modelado probabilístico de temas (ARTM), modelos de lenguaje neuronal basados en semántica distribucional (word2vec, fasttext) y representación profunda de palabras contextualizadas (ELMo y BERT multilingüe), utiliza dataset de puestos de empleo en ruso y métodos de clustering como K-means, propagación por afinidad, birch, agrupación aglomerativa y hdbscan; concluye que la mejor solución fue K-means con ARTM siempre que se señale el número de clusters a obtener con antelación, caso contrario word2vec resulta mejor; las métricas utilizadas para evaluar la calidad del agrupamiento son: Mutua Ajustada (AMI), Índice Rand Ajustado (ARI) y Vmeasure.

1.5 Justificación de la Investigación

La necesidad de identificar puestos de empleo semejantes de profesionales de Tecnologías de Información mediante el uso de técnicas de machine learning no supervisado permite contribuir en la uniformidad de los puestos de empleo, conocer a partir de los anuncios de trabajo formulado por los grupos de interés, los perfiles laborales, capacidades, competencias que permitan a la academia a reformular sus currículos afianzando el enfoque basado en competencias, comprendida en el modelo de acreditación nacional y vinculada a la Política de Aseguramiento de la Calidad de la Educación Superior Universitaria del Estado Peruano, aportando asimismo con el cuarto objetivo de desarrollo sostenible de la Agenda al 2030: Educación de Calidad y de manera indirecta con los objetivos:1,2,3, 8 y 10. (Perú. Ministerio de Educación-MINEDU, 2015, Perú; Perú.Sistema Nacional de Evaluación, Acreditación y Certificación de la Calidad Educativa-SINEACE, 2017; SDSN Australia/Pacific, 2017; Naciones Unidas, 2015).

1.6 Limitaciones de la Investigación

La presente investigación está delimitada a contribuir en el contexto peruano, los portales laborales a utilizar en la parte experimental se circunscriben al mercado laboral peruano, la investigación está delimitada a determinar los empleos de profesionales de tecnologías de información similares, utilizando como base para su determinación las características del empleo establecidas en el perfil del anuncio de empleo consignado por los empleadores en los portales de empleo.

1.7 Objetivos

1.7.1 Objetivo General

Proponer un modelo de machine learning no supervisado para la detección de similitud de puestos de empleo de profesionales de Tecnologías de Información

1.7.2 Objetivos Específicos

a) Diseñar una técnica para extraer los anuncios de empleo dirigidos a profesionales de TI desde los portales laborales.

b) Diseñar una técnica de machine learning no supervisado para la detección por similitud, puestos de empleo de profesionales de Tecnologías de Información.

c) Analizar en qué medida la identificación de puestos de empleo semejantes permiten contribuir en uniformizar los puestos de empleo, sus capacidades y competencias de profesionales de Tecnologías de Información.

d) Determinar el grado de disparidad de los puestos de empleo de profesionales de Tecnologías de Información del Perú con la clasificación estándar internacional de empleos de la OIT

II. Marco Teórico

2.1 Marco Conceptual

Como parte del marco conceptual en las siguientes secciones se desarrollarán los principales tópicos en los que se circunscribe la investigación:

2.1.1 Machine Learning No Supervisado

Machine Learning o aprendizaje automático es una rama de la inteligencia artificial, según lo explica Sandhu citado por Alloghani et al. (2020); que utiliza técnicas automatizadas para resolver problemas basado en datos e información histórica sin requerir modificaciones innecesarias en el proceso principal del negocio. Es preciso señalar que la inteligencia artificial involucra la creación de algoritmos y otras técnicas computacionales orientadas a desarrollar inteligencia en las maquinas; estos algoritmos deben ser capaces de pensar, actuar, así como realizar tareas haciendo uso de protocolos establecidos por los humanos.

Los algoritmos de machine learning pueden ser clasificados según el tipo y la participación humana durante el entrenamiento de los datasets en: supervisados, no supervisados, semi-supervisados y de reforzamiento. El aprendizaje automático supervisado se refiere al conjunto de técnicas que tienen por objetivo obtener un modelo de clasificación valido con la finalidad de determinar escenarios futuros, el sistema debe ser capaz de aprender de lo que tiene para generalizar y determinar lo que no tiene. Se cuenta con muchas aplicaciones diferentes, sin embargo, se requiere contar con datos entrenados en un dominio muy específico, establecer las variables predictoras, predeterminar el atributo de salida, los algoritmos intentan predecir y clasificar el atributo categórico, así lo señala Sierra (2006), en este grupo recae los algoritmos de regresión y clasificación.

En la Tabla 1 se puede apreciar un ejemplo de un dataset entrenado con m instancias, la variables predictoras serian: X_1, X_2, X_3,…X_n; mientras que el atributo clase o también llamado categórico "Clase" contiene p categorías.

Tabla 1

Dataset entrenado

X_1	X_2	…	X_n	**Clase**
v_{11}	v_{12}		v_{1n}	cat_1
v_{21}	v_{22}		v_{2n}	cat_2
…	…	…	…	…
v_{m1}	v_{m2}		v_{mn}	cat_p

Nota. Adaptado de "Aprendizaje Automático conceptos básicos y avanzados" (p. 10), por Sierra, 2006, Pearson Prentice Hall.

De acuerdo con Deshpande (2018), El aprendizaje automático no supervisado se enfoca al descubrimiento de patrones, estos son grupos de objetos con características afines, depende en gran medida de las estadísticas y la distribución de los datos de entrada; su aplicación es adecuada cuando se cuenta con una gran cantidad de datos sin entrenar, los algoritmos no supervisados segmentan los datos en clusters basados en similaridad de puntos, no se establece un atributo de salida o atributo categórico o clase también llamado, todas las variables definidas en el análisis son usadas como entradas; entre las técnicas de aprendizaje automático no supervisado se puede mencionar el Clustering.

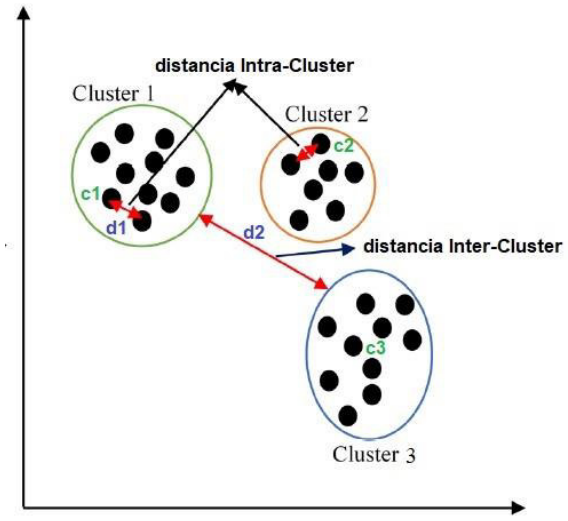
En comparación con el aprendizaje automático supervisado, no necesitamos gastar tiempo y dinero en la preparación del dataset entrenado. Sin embargo, el objetivo del aprendizaje no supervisado parece ser un poco más difícil que el aprendizaje automático supervisado al obtenerse información de los datasets basado en la similitud de puntos de datos, mas no resultados exactos.

2.1.1.1 Clustering. Es una técnica de aprendizaje automático no supervisado, basado en la determinación de clusters generados por similitud de puntos respecto a un centroide del cluster y disimiles con los centroides de los otros clusters. Clustering se trata de una técnica descriptiva y de clasificación, no está sujeta a ningún modelo formal, no se asume la existencia de variables dependientes, ni independientes, no requiere un modelo previo para su análisis; los modelos se crean automáticamente partiendo del reconocimiento de los datos (Perez, 2014; Swamynathan, 2017; Deshpande, 2018).

En clustering los objetos de análisis pueden ser personas, salarios, opiniones, puestos de empleo, petitorios, resoluciones, entre muchos otros; estos deben ser identificados cuidadosamente en función de sus características que representan las principales variables del problema a resolver, así como su influencia en los resultados del algoritmo clustering.

Figura 2

Distancia Intra Cluster e Inter Cluster



Nota. Adaptado de "Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature", por Ezugwu et al., 2021, *Neural Comput & Applic, 1*(33).

Como se puede apreciar en la Figura 2 se expone tres clusters (cluster1, cluster 2, cluster 3), cada cluster tiene un centroide (c1, c2, c3), la distancia intra cluster d1 se refiere a la

distancia mínima que debe presentar cada punto respecto a su centroide, mientras que la distancia inter cluster d2, se refiere a la máxima distancia que deben presentar entre centroides de los clusters respectivamente; cada punto representa un objeto de análisis y debe pertenecer a un único cluster.

Ramadas y Abraham (2018) explican el proceso de clustering, ellos señalan que el proceso se puede establecer en siete pasos fundamentales: i) recolección de la data, ii) vista inicial de la data, iii) representación de la data, iv) tendencia de clustering, v) estrategia del clustering, vi) validación de la data y vii) interpretación del clustering. La recolección de los datos implica la recopilación de datos de diversas fuentes, la vista inicial de la data consiste en valorar la disponibilidad de la data con la que se cuenta, la representación de la data significa preparar la data en función de los requisitos del algoritmo a utilizar para su procesamiento; tendencia de clustering verifica si la data puede ser considerada en un cluster o no, verificando que la naturaleza de los datos sea plausibles de agrupar; la estrategia de clustering se basa en elegir el algoritmo propicio así como sus parámetros correctos a aplicar; validación de la data concierne en examinar y probar los datos manualmente, finalmente se interpretan los clusters o grupos resultantes y se sugiere o realizan análisis adicionales.

Entre otras técnicas clustering se puede señalar: Generación de Nuevos Ejemplos, básicamente esta técnica de aprendizaje automático no supervisado se basa en el uso de datasets entrenados para generar nuevos puntos por similitud a partir de datos ya entrenados, pero estos no serían igual a los datos originales. Asimismo, la técnica detección de anomalías se encuentra en este grupo de técnicas de aprendizaje automático no supervisado, y se enfoca en la identificación de puntos de datos anómalos en un dataset.

En la literatura se cuenta con una variedad de algoritmos que implementan clustering, la mayoría obtienen resultados de buena calidad, sin embargo, muchos de ellos aun cuentan con ciertas limitaciones como tener que establecer a priori el número de clusters a obtener, de

no hacerlo el algoritmo puede presentar dificultades de procesamiento y consecuencia de ello se incurre en el uso de altas cargas de recursos computacionales adicionales. Por otro lado, la técnica clustering tiene como objetivo identificar patrones, comportamientos en información real con características de alta densidad y dimensionalidad por lo que determinar el número de clusters a obtener es una tarea difícil; la calidad de un método clustering está en función de su habilidad para descubrir algunos o todos los patrones ocultos en la información así lo señalan Ezugwu et al. (2021).

2.1.1.2 Distancias y Similaridades. En Sierra (2006) se define los términos distancia y similaridad como se indica a continuación:

Sean m los casos de un conjunto Ω, consideremos $\Omega = \{1,2,\cdots,m\}$, considerando que el clustering tiene como objetivo principal identificar clusters que contengan los casos similares, ello implica necesario medir las similitudes o las distancias que se tiene entre los casos.

Una distancia sobre un conjunto Ω es una función d:

$$d\colon\ \Omega\ x\ \Omega \to \mathbb{R}$$
$$(i,j) \to d(i,j) = d_{ij},$$ tal que verifica las siguientes propiedades:

1 $\mathrm{d}(i,j) \geq 0,\ \forall i,j\ \varepsilon\ \Omega$

2 $\mathrm{d}(i,\mathrm{i}) = 0,\ \forall i\ \varepsilon\ \Omega$

3 $\mathrm{d}(i,j) = \mathrm{d}(j,i),\ \forall i,j\ \varepsilon\ \Omega$

Las propiedades señalan que las distancias no deben ser negativas (1), no puede haber distancia entre los mismos casos (2) y las distancias deben ser simétricas (3). Cuanto mayor sea la distancia $\mathrm{d}(i,j)$, mas diferentes entre sí serán los casos i y j. Si además se cumple la desigualdad triangular: $\mathrm{d}(i,j) \leq \mathrm{d}(i,k) + \mathrm{d}(j,k),\ \forall i,j,k\ \varepsilon\ \Omega$, diremos que la distancia es métrica y que (Ω, d) forma un espacio métrico.

Una similaridad sobre un conjunto Ω es una función s:

s: $\Omega\ x\ \Omega \rightarrow \mathbb{R}$
$(i,j) \rightarrow s(i,j) = s_{ij}$, tal que verifica las siguientes propiedades:

1 $0 \leq s(i,j) \leq 1,\ \forall i,j\ \varepsilon\ \Omega$

2 $1 = s(i,i) \geq s(i,j),\ \forall i,j\ \varepsilon\ \Omega$

3 $s(i,j) = s(j,i),\ \forall i,j\ \varepsilon\ \Omega$

La similaridad no debe ser negativa, establece una escala (1); cada caso se parece asimismo más que a cualquier otro caso (2) y debe ser simétrica (3). La interpretación precisa que cuanto mayor sea la similaridad $s(i,j)$, mas parecidos entre si serán los casos i y j. (pp. 262-263)

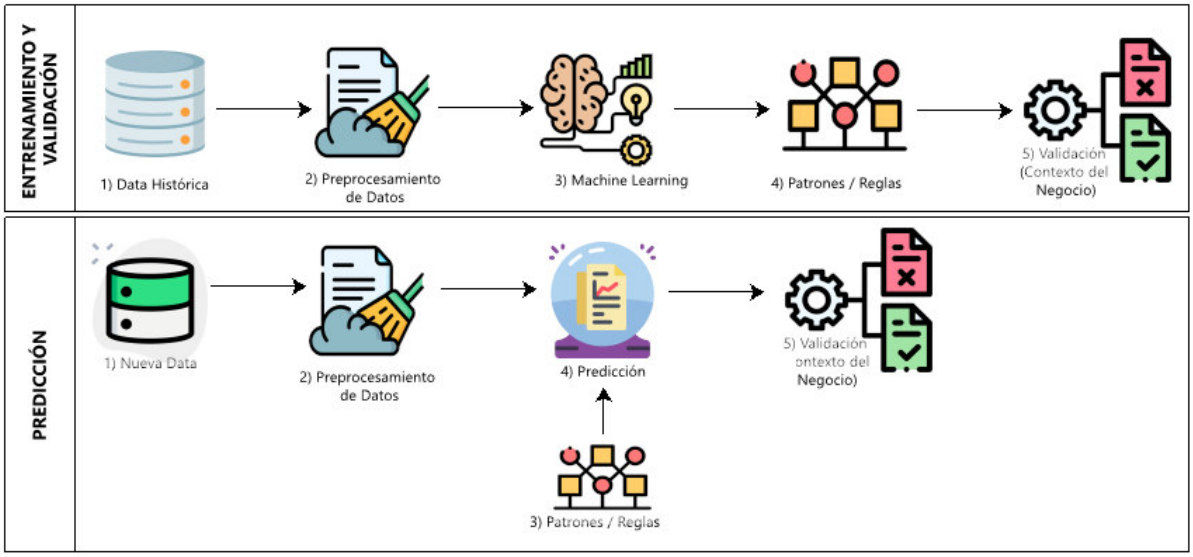
2.1.1.3 Flujo del Proceso del Aprendizaje No supervisado. A diferencia del aprendizaje supervisado, el aprendizaje no supervisado requiere identificar los patrones o comportamiento a partir de la data histórica, esta data no cuenta con el atributo clase, categoría, label o atributo predictor, por lo que debe determinarlo.

En la Figura 3 se expone el Flujo del Proceso del Aprendizaje No supervisado a través de dos sub procesos fundamentales, el primero "Entrenamiento y Validación" se organiza en cinco etapas, la primera es la disposición de la data histórica, la segunda consiste en realizar el preprocesamiento de los datos, esto significa que la data debe ser modelada bajo ciertas características en función del contexto del negocio, de manera que los algoritmos de machine learning puedan procesar sobre ellos emitiendo resultados significativos y con mejor precisión, por lo que esta segunda etapa es fundamental; la tercera etapa consiste en la aplicación del algoritmo de machine learning seleccionado obteniendo como resultado el conjunto de patrones o reglas los cuales deben ser validados por el experto en términos del contexto del negocio. El segundo subproceso "Predicción" comprende el uso de nueva data por determinar el patrón o regla al que corresponde utilizando para su predictibilidad el modelo de aprendizaje obtenido

como resultado del primer subproceso, su resultado igualmente debe ser validado por el experto en el contexto del negocio.

Figura 3

Flujo del Proceso del Aprendizaje No supervisado



Nota. Adaptado de "Mastering Machine Learning with Python in Six Steps, A Practical Implementation Guide to Predictive Data Analytics Using Python", por Swamynathan, 2017, Apress Media, p. 195.

2.1.1.4 K-means. En Swamynathan (2017) se describe a K-means como una técnica clustering cuyo objetivo es organizar la data dentro de clusters, con una similaridad intra-cluster alta y una similaridad inter-cluster baja. Un item de datos solo se asigna a un cluster no a varios; esto genera un numero especifico de clusters disjuntos no jerárquicos. K-means usa la estrategia de dividir y concurrir.

Sierra (2006, p. 290) resume el algoritmo k-means como se indica:

1 De entre los n casos elegir k que llamaremos semillas y denotaremos c_j , $j = 1, \cdots k$, Cada semilla c_j representará al cluster $C_j (j = 1, \cdots k)$.

2 Asignar el caso i al cluster C_j cuando $d(x_i, c_j) = \text{mín}_{l=1,\ldots,k,}\, d(x_i, c_l)$. Es decir, cada caso se asigna al cluster que representa la semilla que tiene más cerca.

3 Los pasos 1 y 2 nos dan una partición inicial de los casos.

4 Calcular la mejora que se produciría en el criterio elegido (minimizar $tr(W)$, minimizar $det(W)$, etc…) al asignar un caso a otro cluster en el que no está actualmente.

5 Hacer el cambio que mayor mejora produce en el criterio.

6 Repetir los pasos 3 y 4 hasta que ningún cambio haga mejorar el criterio elegido.

Complementariamente a las propuestas de Sierra (2006) y Swamynathan (2017), en la literatura se cuenta con diversas variantes del algoritmo k-means, el principal objetivo de estas versiones están enfocadas en buscar la buena calidad de los resultados del clustering, en exceptuar la dependencia de establecer el número de clusters a obtener como resultado del proceso; entre los incidentes presentados en la aplicación del algoritmo se puede mencionar la no convergencia en los resultados, esto puede deberse al número de clusters k elegido o a la ausencia de estructura de clusters en los datos (Sierra, 2006, p. 290).

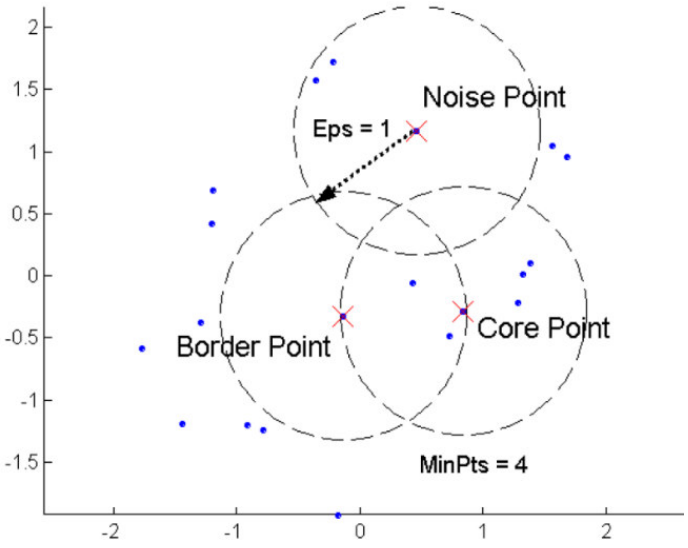
2.1.1.5 DBScan. Es una técnica de clustering basado en la densidad de puntos, los clusters o conglomerados se conforman por regiones densas de puntos en un radio especifico, es de utilidad optar por esta técnica cuando los conglomerados presentan formas irregulares y ruido a través de outliers en los datos, los clusters son identificados de forma arbitraria, son robustos ante la presencia de ruido.

El algoritmo requiere para su procesamiento de parámetros como: el épsilon, el mínimo de puntos que conforman una región densa, los puntos core o cluster, border o frontera, noise o ruido, la eficiencia de ejecución del algoritmo está determinada por O(n log n). Los puntos core o cluster se refiere a los puntos interiores de un conglomerado, siempre que tenga al menos

un número mínimo de puntos en su vecindario de radio epsilon, los puntos border o frontera tienen menos del mínimo de puntos en su vecindario de radio epsilon, el noise o ruido se refiere a los puntos que no forman parte de un cluster o core así como de su frontera como se aprecia en la Figura 4.

Figura 4

Densidad de puntos Core, Border y Noise



Nota. Tomado de "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", por Ester et al., 1996, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*.

En la Figura 5 se expone el Algoritmo DBScan de Ester et al. (1996), *SetOfPoints* es el dataset para descubrimiento de clusters o conglomerados Epsilon *(Eps)* y número de puntos mínimos *(MinPts)* son los parámetros de densidad global determinados manualmente o mediante heurísticas. El algoritmo recorre el dataset, obtiene cada punto con la función *SetOfPoints.get(i)*, verifica si el *ClusterId (ClId)* del punto se encuentra como no clasificado *(UNCLASSIFIED)* e invoca la función *ExpandCluster* pasándole como parámetros: *SetOfPoints, Point, ClusterId, Eps, MinPts*. La función se ejecuta invocando el método

SetOfPoints.regionQuery(Point,Eps) que retorna los puntos vecinos de *Point* en un radio *Eps*. Si la lista de semillas es menor a *MinPts* entonces el punto se marca como ruido *SetOfPoint.changeClId(Point,NOISE);* en otro caso se considera que todos los puntos de la lista de semillas *seeds* con densidad-alcanzable desde *Point* se establece un *ClId=UNCLASSIFIED SetOfpoints.changeCiIds(seeds,ClId)* y se elimina el punto de la lista de semillas *seeds.delete(Point);* Se inicia un bucle repetitivo *WHILE* mientras la lista de semillas *seeds* no se encuentre vacía, se toma el primer punto *currentP* de la lista seeds en *currentP := seeds.first();* a continuación se obtiene los puntos vecinos a *currentP* en una lista *result := setofPoints.regionQuery(currentP,Eps);* se evalúa si el número de semillas de la lista *result* supera o es igual a *MinPts*, de ser igual o superar *MinPts* entonces recorre la lista *FOR*, para cada punto *resultP* en *result* verifica el *ClId* con el conjunto *{ UNCLASSIFIED, NOISE}*, de ser *UNCLASSIFIED* entonces añade el punto a la lista de semillas *seeds.append(resultP)* y modifica el *ClId* del punto *resultP* en el dataset con el *Cluster Id (ClId)* determinado *SetOfPoints.changeCiId(resultP, CiId);* repite el proceso repetitivo *FOR* con los otros puntos de la lista *result*, al finalizar el bucle repetitivo *FOR* elimina el punto *currentP* de la lista de semillas *seeds sedes.delete(currentP);* y repite el bucle *WHILE* con el siguiente punto de la lista *seeds* hasta que esta esté vacía y retorna *True* a la función principal *DBSCAN*, inicializándose el *ClusterId* con el siguiente *ClusterId := nextId(ClusterId)*, continuando la repetitiva *FOR* con el siguiente punto del dataset.

Figura 5

Algoritmo DBScan

```
DBSCAN (SetOfPoints, Eps, MinPts)
  // SetOfPoints is UNCLASSIFIED
  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfPoints.size DO
   Point := SetOfPoints.get(i);
   IF Point.CiId = UNCLASSIFIED THEN
    IF ExpandCluster(SetOfPoints, Point, ClusterId, Eps,
    MinPts) THEN
     ClusterId := nextId(ClusterId)
```

```
      END IF
     END IF
    END FOR
END; // DBSCAN
ExpandCluster (SetOfPoints, Point, CiId, Eps,MinPts) : Boolean;
    seeds : =SetOfPoints. regionQuery (Point, Eps )
    IF seeds.size<MinPts THEN // no core point
     SetOfPoint. changeCl Id (Point, NOISE)
     RETURN False;
    ELSE // all points in seeds are density-
        // reachable from Point
     SetOfpoints. changeCiIds ( seeds, C1 Id)
     seeds .delete (Point)
     WHILE seeds <> Empty DO
      currentP := seeds.first()
      result := setofPoints.regionQuery(currentP,Eps)
       IF result.size >= MinPts THEN
        FOR i FROM 1 TO result.size DO
         resultP := result.get(i)
         IF resultP. CiId IN (UNCLASSIFIED, NOISE} THEN
          IF resultP.CiId = UNCLASSIFIED THEN
           seeds, append (resultP)
          END IF;
          SetOfPoints. changeCiId ( resultP, CiId)
         END IF; // UNCLASSIFIED or NOISE
        END FOR ;
       END IF; // result.size >= MinPts
      seeds, delete (currentP)
     END WHILE; // seeds <> Empty
     RETURN True ;
    END IF
END; // ExpandCluster
```

Nota. Tomado de "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", por Ester et al., 1996, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1(86).

2.1.2 Puestos de empleo

La Organización Internacional del Trabajo (OIT,2019) o ILO en inglés se describe como una entidad que "(…) reúne a gobiernos, empleadores y trabajadores de 187 Estados miembros a fin de establecer las normas del trabajo, formular políticas y elaborar programas promoviendo el trabajo decente de todos, mujeres y hombres.". Asimismo, define el empleo como un conjunto de tareas y deberes realizados, o destinados a ser realizados por una persona. Además, precisa que una ocupación o puesto de empleo es un tipo de trabajo realizado en un empleo. Una persona puede estar asociada con una o varios puestos de empleo desempeñados en el tiempo lo que afianza su hoja de vida.

El cuerpo de Gobierno de la OIT convoco a una reunión de expertos en estadísticas de mercado laboral con la finalidad de actualizar la Clasificación Internacional Estándar de Ocupaciones ISCO del año 1988 (ISCO-88), CIUO-88 en español, a raíz de la propuesta en la diecisieteava conferencia internacional de estadísticas laborales (ICLS) del año 2003 referente a la futura actualización de ISCO-88 debido a la globalización del mercado laboral, la necesidad de comparar datos ocupacionales con propósitos estadísticos-administrativos, la necesidad de contar con un modelo para desarrollar clasificadores de ocupaciones a nivel nacional y regional, la necesidad de disponer de un sistema base que pueda ser utilizado en países que no cuentan con un sistema de clasificación de empleos propio.

El producto resultante de la actualización ISCO-88 es ISCO-08, este sistema utiliza una estructura de clasificación jerárquica que permite clasificar todos los empleos del mundo en 436 unidades de grupo con un nivel de detalle de 130 sub-grupos basados en la similaridad de los niveles de habilidad y especialización de habilidades.

ISCO busca facilitar la comunicación internacional acerca de las ocupaciones suministrando estadísticas con la finalidad de realizar análisis comparativo de la data ocupacional disponible, para lo cual es necesario generar data en una forma que pueda ser utilizada por investigaciones en actividades orientadas a la acción y toma de decisiones relacionadas a la colocación de empleo y la migración internacional. En la Tabla 2 se presenta el CIUO-08 de las profesiones relacionadas a Tecnologías de Información (TI).

Tabla 2

CIUO-08 de profesiones de TI

N°	CIUO	TITULO
1	**1**	**DIRECTORES Y GERENTES**
2	**13**	**DIRECTORES Y GERENTES DE PRODUCCIÓN Y OPERACIONES**
3	133	DIRECTORES DE SERVICIOS DE TECNOLOGÍA DE LA INFORMACIÓN Y LAS COMUNICACIONES

		DIRECTORES DE SERVICIOS DE TECNOLOGÍA DE LA INFORMACIÓN Y LAS
4	1330	COMUNICACIONES
5	**2**	**PROFESIONALES CIENTÍFICOS E INTELECTUALES**
6	**23**	**PROFESIONALES DE LA ENSEÑANZA**
7	235	OTROS PROFESIONALES DE LA ENSEÑANZA
8	2356	INSTRUCTORES EN TECNOLOGÍA DE LA INFORMACIÓN
		ESPECIALISTAS EN ORGANIZACIÓN DE LA ADMINISTRACIÓN PUBLICA Y DE
9	**24**	**EMPRESAS**
		PROFESIONALES DE LAS VENTAS, LA COMERCIALIZACIÓN Y LAS RELACIONES
10	243	PÚBLICAS
		PROFESIONALES DE VENTAS DE TECNOLOGÍA DE LA INFORMACIÓN Y LAS
11	2434	COMUNICACIONES
		PROFESIONALES DE TECNOLOGÍA DE LA INFORMACIÓN Y LAS
12	**25**	**COMUNICACIONES**
13	251	DESARROLLADORES Y ANALISTAS DE SOFTWARE Y MULTIMEDIA
14	2511	ANALISTAS DE SISTEMAS
15	2512	DESARROLLADORES DE SOFTWARE
16	2513	DESARROLLADORES WEB Y MULTIMEDIA
17	2514	PROGRAMADORES DE APLICACIONES
		DESARROLLADORES Y ANALISTAS DE SOFTWARE Y MULTIMEDIA Y ANALISTAS NO
18	2519	CLASIFICADOS BAJO OTROS EPÍGRAFES
19	252	ESPECIALISTAS EN BASES DE DATOS Y EN REDES DE COMPUTADORES
20	2521	DISEÑADORES Y ADMINISTRADORES DE BASES DE DATOS
21	2522	ADMINISTRADORES DE SISTEMAS
22	2523	PROFESIONALES EN REDES DE COMPUTADORES
23	2529	ESPECIALISTAS EN BASES DE DATOS Y EN REDES DE COMPUTADORES NO
24	**3**	**TÉCNICOS Y PROFESIONALES DE NIVEL MEDIO**
25	35	TÉCNICOS DE LA TECNOLOGÍA DE LA INFORMACIÓN Y LAS COMUNICACIONES
26	351	TÉCNICOS EN OPERACIONES DE TECNOLOGÍA DE LA INFORMACIÓN Y LAS
		TÉCNICOS EN OPERACIONES DE TECNOLOGÍA DE LA INFORMACIÓN Y LAS
27	3511	COMUNICACIONES
		TÉCNICOS EN ASISTENCIA AL USUARIO DE TECNOLOGÍA DE LA INFORMACIÓN Y LAS
28	3512	COMUNICACIONES
29	3513	TÉCNICOS EN REDES Y SISTEMAS DE COMPUTADORES
30	3514	TÉCNICOS DE LA WEB
31	7	OFICIALES, OPERARIOS Y ARTESANOS DE ARTES MECÁNICAS Y DE OTROS OFICIOS

32	74	TRABAJADORES ESPECIALIZADOS EN ELECTRICIDAD Y LA ELECROTECNOLOGÍA
33	742	INSTALADORES Y REPARADORES DE EQUIPOS ELECTRÓNICOS Y DE TELECOMUNICACIONES
34	7422	INSTALADORES Y REPARADORES EN TECNOLOGÍA DE LA INFORMACIÓN Y LAS COMUNICACIONES

Nota. Adaptado de Organización Internacional del Trabajo. (30 de enero de 2005). *Estructura de la CIUO-08 y concordancias previas con la CIUO-88*. https://www.ilo.org/public/spanish/bureau/stat/isco/isco08/index.htm

El departamento de estadisticas de la OIT (ILOSTAT, 2020) señala la correspondencia de los niveles de habilidad con los ISCO-08, como se aprecia en la Tabla 3, las profesiones de TI se consideran de nivel de habilidad alto.

Tabla 3

Niveles de habilidad ISCO-08

Amplio nivel de habilidad	**ISCO-08**
Niveles de habilidad 3 y 4 (alto)	1. Gerentes
	2. Profesionales
	3. Técnicos y profesionales asociados
	4. Trabajadores de apoyo administrativo
	5. Trabajadores de servicios y ventas
Nivel de destreza 2 (medio)	6. Trabajadores calificados de la agricultura, la silvicultura y la pesca
	7. Trabajadores de artesanía y oficios conexos
	8. Operadores de plantas y máquinas, y ensambladores
Nivel de destreza 1 (bajo)	9. Ocupaciones elementales
Las fuerzas armadas	0. Ocupaciones de las fuerzas armadas

No está clasificado en otra parte	X. No clasificado en otra parte

Nota. Adaptado de Organización Internacional del Trabajo | ILOSTA. *Clasificación Internacional Uniforme de Ocupaciones*. https://ilostat.ilo.org/es/resources/concepts-and-definitions/classification-occupation/

En el Perú se cuenta con el Catálogo Nacional de Perfiles Ocupacionales o Cualificaciones (CNPO, 2014) elaborado por el Ministerio de Trabajo y Promoción del Empleo (MTPE), este catálogo adopta el Clasificador Industrial Internacional Uniforme (CIIU) rev.4 debido a que este incorpora los últimos cambios de las actividades económicas, la globalización y las tecnologías de información respecto a CIIU rev.3 y otros clasificadores; CIIU rev.4 organiza los perfiles ocupacionales en familias productivas vinculadas a las actividades económicas del país; la primera versión se aprobó en el 2010, siendo reestructurado en el 2012 y 2014; su finalidad es reducir la brecha entre la oferta y la demanda de cualificaciones o puestos de empleo, para lo cual se hace necesario articular el sector laboral con la academia para atender los constantes desafíos enmarcados en el desarrollo nacional.

La Figura 6 muestra la Clasificación del Sector de TI del CNPO, como se aprecia, se considera el sector económico: INFORMACION Y COMUNICACIONES (J), como familia productiva: Tecnologías de la Información y Comunicaciones (TICs) y las divisiones: Programación informática, consultoría de informática y actividades conexas (62) y Actividades de servicios de información (63) para enmarcar las cualificaciones de TI.

Figura 6

Niveles de Clasificación del Sector TI

N	CODSEC	SECTOR	CODFAMPRO	FAMILIA PRODUCTIVA	DIVISIÓN	DESCRIPCIÓN
26	J	INFORMACIÓN Y COMUNICACIONES	J26	Tecnologías de la Información y Comunicaciones - TICs	58	Actividades de edición
					59	Actividades de producción de películas cinematográficas, vídeos y programas de televisión, grabación de sonido y edición de música
					60	Actividades de programación y transmisión
					61	Telecomunicaciones
					62	Programación informática, consultoría de informática y actividades conexas
					63	Actividades de servicios de información

Nota. Tomado de "Catalogo Nacional de Perfiles Ocupacionales (Cualificaciones)", por Perú. Ministerio de Trabajo y Promoción del Empleo - MTPE, 2014.

En el año 2021, el estado peruano ha emitido el Decreto Supremo N° 012-2021-MINEDU que crea el Marco Nacional de Cualificaciones del Perú (MNCP) y la comisión multisectorial de naturaleza permanente denominada "Comisión Nacional para el seguimiento a la implementación del Marco Nacional de Cualificaciones del Perú - MNCP" como instrumento para el reconocimiento de las cualificaciones para el desempeño en el mercado laboral; considera 08 niveles de cualificación, vías de cualificación, dimensiones, subdimensiones, descriptores, criterios de agrupamiento y priorización para su poblamiento.

Las cualificaciones estarán expresadas en términos de conocimientos, destrezas y competencias vinculando al sector productivo con la academia en aras de brindar una mejor formación al recurso humano para un eficiente desempeño laboral. Inicialmente será poblado con las cualificaciones de los sectores primarios como agricultura y minería, progresivamente se incluirán otros sectores. El MNCP está a cargo de una comisión multisectorial y representantes del sector privado quienes realizaran el seguimiento a su implementación, así como su alineamiento con el CNPO y la certificación de competencias laborales (Decreto Supremo N° 012-2021-MINEDU, 2021; MINEDU PMESUT, 2021).

2.1.3 Profesionales de Tecnologías de Información

En la Revista de Negocios de Harvard de 1958 se registró el término Tecnología de Información para describir "nueva tecnología" en los negocios; y se definió como: "(…) implica el procesamiento informático de la información, la programación matemática para la

toma de decisiones y la simulación (…) a través de programas informáticos (…)." (Leavitt y Whisler, 1958, p.41)

El Currículo de Tecnología de la Información 2017, Lineamientos Curriculares de TI 2017 para Programas de Licenciatura en Tecnología de la Información, define el término Tecnología de Información como "La tecnología de la información es el estudio de enfoques sistémicos para seleccionar, desarrollar, aplicar, integrar y administrar tecnologías informáticas seguras para permitir a los usuarios lograr sus objetivos personales, organizativos y sociales." (Task Group on Information Technology Curricula, 2017, p. 18)

Asimismo, describe aspectos relevantes del profesional de TI como se indica:

> El profesional en TI es un solucionador de problemas colaborativos, calificado, investigador que disfruta haciendo que la tecnología funcione de manera efectiva y satisfaga las necesidades de los usuarios en una variedad de entornos; trabajan en colaboración para integrar nuevas tecnologías en el entorno de trabajo, la comunidad y garantizar una experiencia superior y productiva para el usuario y todos los procesos de la organización. En el entorno corporativo, aplican sus conocimientos sobre integración, desarrollo y operación de sistemas e implementan y administran servicios y plataformas de TI que cumplen con las metas y objetivos comerciales de la organización. En la comunidad, los profesionales de TI utilizan su experiencia en la implementación de una amplia gama de soluciones de TI para apoyar los proyectos y actividades de los miembros de la comunidad.
>
> Los profesionales de TI están preparados para realizar tareas de manera ética; están familiarizados con estándares nacionales e internacionales que rigen el desarrollo y las operaciones de las plataformas de TI que mantienen; asimismo pueden explicar y justificar las decisiones profesionales en un lenguaje que la gerencia, usuarios o clientes entiendan. Conocen las implicaciones presupuestarias de las alternativas tecnológicas

> y pueden defender los presupuestos adecuadamente. Tienen una amplia práctica en asegurar adecuadamente las redes de TI, aplicaciones, centros de datos y servicios en línea. Buscan soluciones tecnológicas seguras sin afectar indebidamente la capacidad de los usuarios para lograr sus objetivos. (Task Group on Information Technology Curricula, 2017, p.19)

La actualización de ISCO-08 se hace necesaria debido al impacto de las Tecnologías de Información y Comunicaciones y disciplinas afines en la estructura ocupacional del mercado laboral; tanto las cualificaciones o perfiles de TI señalados a nivel de la OIT en la Tabla 2, así como en el CNPO a nivel nacional de la Figura 6 no se ajustan a la realidad actual por lo que se hace necesario su actualización de acuerdo con los perfiles laborales requeridos en el mercado laboral que permita a la academia la formación de profesionales competentes que pueda cubrir la demanda social y como consecuencia se contribuya al desarrollo del país.

III. Método

3.1 Tipo de Investigación

Considerando lo señalado por Hernández et al. (2014), la presente investigación tiene un enfoque cualitativo, no busca correlacionar variables, utiliza el método inductivo, se basa en estudio de casos, por el tiempo de aplicación de las variables es transversal, pues los datos serán recolectados en un único momento y tiempo; por la naturaleza de los objetivos es una investigación descriptiva, no experimental y es aplicada debido a que se está haciendo uso de conocimientos existentes para encontrar soluciones a los problemas planteados.

Asimismo, de acuerdo con los aportes de Barrientos (2013), considera que por la naturaleza de la información que recoge para responder los problemas de investigación es cualitativa, tipo de investigación basada en Estudio de Casos porque tiene como unidad de estudio a la demanda de puestos de empleo hecha manifiesta por los Grupos de Interés a través de los portales web de trabajo, por lo que las etapas de investigación consideradas son: 1.Enunciar los objetivos de la Investigación, 2.La selección del caso, 3.Recoger los datos y 4.Organizar los datos.

3.2 Población y Muestra

La Población está determinada por los puestos de empleo de los Grupos de Interés, estos son los empleadores que representan al sector público y privado, quienes utilizan los portales de empleo para realizar convocatorias públicas con la finalidad de llevar a cabo un proceso de selección transparente reclutando a los mejores profesionales que cumplan con el perfil requerido.

La técnica de muestreo a utilizar es no probabilística e intencional, por considerarlo clave en el suministro de información de valor para la investigación y está representado por los

anuncios de empleo registrados en los Portales de Trabajo y dirigidos a profesionales de TI del Perú en los dos últimos años.

De lo anteriormente indicado se tendría una muestra de n=8,640 anuncios de empleo publicados entre febrero 2020 y febrero 2021 que serán analizados utilizando machine learning no supervisado permitiendo responder los objetivos de la presente investigación. En la Tabla 4 se puede apreciar el número de puestos de empleo por cada portal de trabajo que publicaron en los últimos dos años.

Tabla 4

Distribución de la Muestra

N°	Portal de Trabajo	N° Puestos de Empleo
1	Google Jobs	3,200
2	Freelancer	2,096
3	Buscojobs	1,289
4	Mipleo	724
5	Linkedin	457
6	Indeed	420
7	Computrabajo	379
8	Convocatoriatrabajo	75
	Total -->	8,640

3.3 Operacionalización de variables

VARIABLE INDEPENDIENTE

Machine Learning no supervisado.

VARIABLE DEPENDIENTE

Puestos de empleo semejantes de profesionales de TI

3.4 Instrumentos

El presente trabajo utilizó un proceso automatizado para la recolección de información conocido como raspado web o Web scraping en inglés, esta técnica se logra escribiendo un programa de computadora que consulta las páginas web en formato HTML y extrae los datos de los anuncios de empleo desde los portales de trabajo. Un segundo instrumento utilizado para la recopilación de información son las Interfaz de Programación de Aplicaciones (APIs), servicios web o API Rest públicas, que organismos nacionales y/o internacionales exponen para brindar la información en el contexto de los datos abiertos (Perú. Presidencia del Consejo de Ministros - PCM, 2017), instrumento utilizado para extraer la información estándar de empleo de ISCO-08 y MT-SP2015. Asimismo, para validar los resultados se utilizó técnicas de validación clustering desarrollados en las obras de King (2015, p. 179-219) y el articulo científico Halkidi et al. (2001), con los resultados obtenidos se procedió a realizar el análisis de los datos y su discusión.

3.5 Procedimiento

El procedimiento seguido en la presente investigación toma como base el proceso de machine learning no supervisado propuesto por Swamynathan (2017, p. 195): se trata de una adaptación de los tradicionales procesos de minería de datos: “Knowledge Discovery Databases” (KDD) y su variante “Cross-Industry Standard Process for Data Mining” (CRISP-DM), cuyas fases han sido personalizadas a la presente investigación como se puede apreciar en la Figura 7 y se especifica a continuación:

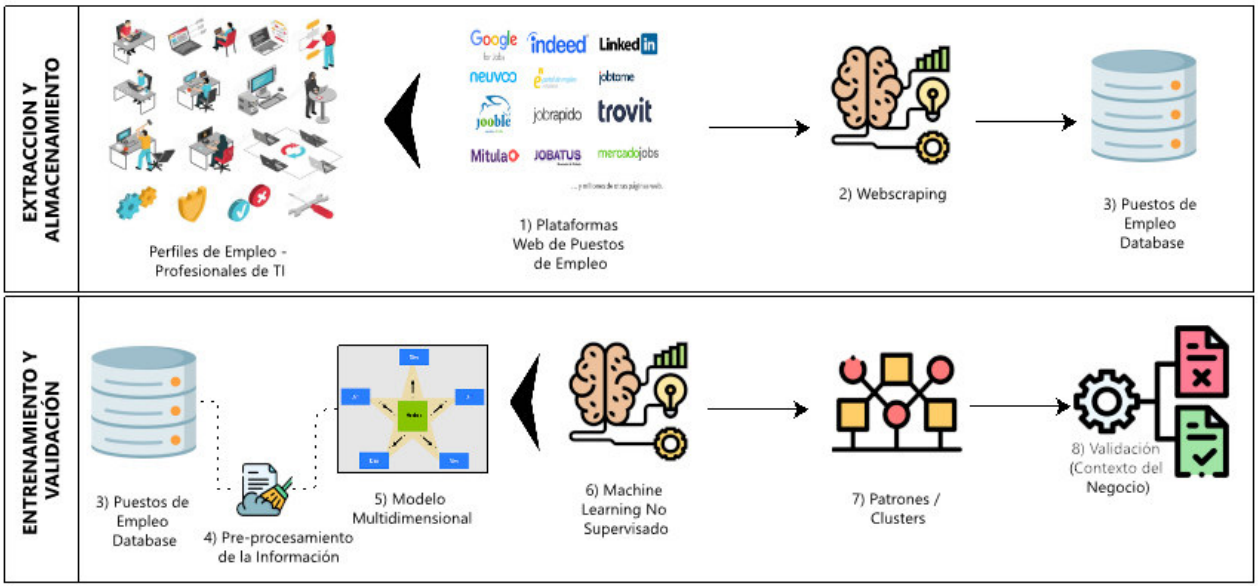
i. Extracción y almacenamiento de los puestos de empleo mediante web scraping
ii. Realizar el Pre-procesamiento de la Data concerniente a los puestos de empleo
iii. Diseñar el modelo dimensional de la Base de Datos
iv. Diseñar el algoritmo de machine learning no supervisado
v. Implementar un prototipo

vi. Identificar los patrones y/o clusters

vii. Evaluación del modelo

Figura 7

Proceso de Machine Learning No supervisado



Nota. Adaptado de "Mastering Machine Learning with Python in Six Steps, A Practical Implementation Guide to Predictive Data Analytics Using Python", por Swamynathan, 2017, Apress Media, p. 195.

En las siguientes secciones se desarrollará cada una de las etapas comprendidas en el proceso seguido por la presente investigación:

3.5.1 Extracción y almacenamiento de puestos de empleo

Esta fase del proceso consistió en extraer la información de los puestos de empleo y su almacenamiento en una base de datos con alojamiento en la nube. La data ha sido recolectada mediante la técnica webscraping desde los portales de trabajo referidos en la Tabla 4, entiéndase un portal de trabajo como la página o plataforma web en la que los empleadores o grupos de interés publican las convocatorias de puestos de empleo, este medio digital se ha

constituido en la principal fuente a la que recurren los ciudadanos, entre profesionales y no profesionales para conseguir un empleo.

Para la presente investigación se seleccionaron ocho portales de trabajo considerando la visibilidad que estos presentan en el ranking de plataformas web de su rubro, fueron consideradas las ofertas de trabajo registradas entre los periodos 2020 - 2021 concernientes a organismos públicos y privados de los distintos rubros de la economía peruana como se puede apreciar en la Tabla 5 y cuyas convocatorias estaban dirigidas a profesionales de TI.

Tabla 5

Puestos de Empleo por Portal de Trabajo

N°	Portal de Trabajo	N° Puestos de Empleo
1	Google Jobs	3,200
2	Freelancer	2,096
3	Buscojobs	1,289
4	Mipleo	724
5	Linkedin	457
6	Indeed	420
7	Computrabajo	379
8	Convocatoriatrabajo	75
	Total -->	**8,640**

Con la finalidad de realizar el proceso de extracción y almacenamiento de la data fue necesario desarrollar un programa en Python que incluía funciones personalizadas para cada portal de trabajo descrito en la tabla referida, esto debido a que cada portal contiene sus propias características de implementación para modelar la información en html; básicamente el procedimiento consistió en acceder a la página de inspección del navegador Google Chrome,

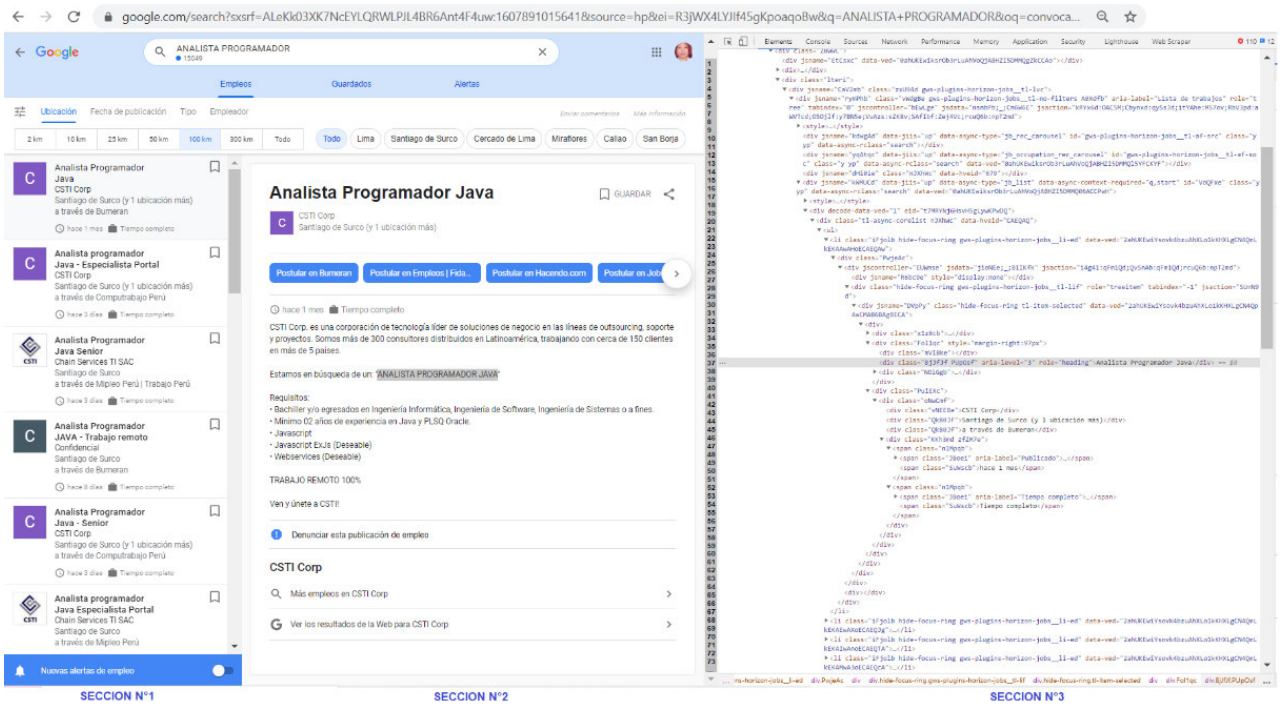
específicamente al panel de elementos a través del cual se puede acceder y leer el Modelo de objetos del documento (DOM) del anuncio de empleo, un DOM "es la representación de datos de los objetos que componen la estructura y el contenido de un documento en la web" según lo indica Mozilla (2021).

Es preciso la comprensión del procedimiento seguido y es en la Figura 8 en la que se puede apreciar la página web con el resultado de búsqueda de la palabra *"Analista Programador"*, la página se ha dividido en tres secciones: la primera sección expone la relación de anuncios de empleos identificados como resultado de la búsqueda; la segunda sección presenta el detalle de la convocatoria de trabajo seleccionada en la primera sección, aquí el empleador precisa los detalles del perfil del profesional requerido así como los beneficios laborales que la empresa ofrece a los interesados y en la tercera sección se observa el panel de elementos de la página de inspección del navegador Google para la lectura del DOM de un anuncio de empleo, como se aprecia en la Figura 8 los anuncios se encuentran organizados por *ítems lists (li)*, uno por cada oferta laboral.

La tercera sección ha sido numerada con la finalidad de explicar la estructura del DOM: se puede apreciar en la línea 22 el *li* del primer anuncio, este se subdivide en divisiones *div,* en la línea 37 se observa el *div* del título del anuncio con el valor *"Analista Programador Java"*, en la línea 43: el *div* correspondiente a la empresa que realiza la convocatoria, en la línea 44: el *div* con la ubicación geográfica de la empresa, en la línea 45: la plataforma web que publica el anuncio de empleo, en la línea 50: el tiempo de publicación del anuncio y en la línea 54: la modalidad del empleo. Cada anuncio de empleo de la sección primera se estructura mediante *ítems lists* como se aprecia en las líneas 68,70,73, etc. Esta manera de estructurar los anuncios permite aplicar técnicas de extracción webscraping a las convocatorias de empleo de los portales web de trabajo.

Figura 8

Documento DOM una Convocatoria de Trabajo



Nota. Tomado de Buscador Google. (2021). Panel de Elementos, *Pagina de Inspeccion del resultado de una busqueda de convocatoria de trabajo en Google*. https://www.google.com/search

La información de los anuncios de empleo extraídos se registró en un esquema de base de datos gestionados por el Sistema de Administración de Base de Datos postgresql y alojado en un servidor en la nube para su disponibilidad permanente.

En la Tabla 6 se expone el esquema utilizado para el alojamiento persistente de la información obtenida, el esquema quedó constituido por cuatro tablas fundamentales que pasamos a describir:

keyword_search: Esta tabla contiene información concerniente a 117 principales palabras claves de búsqueda, clasificadas entre roles y tecnologías, las cuales se invocó desde el programa webscraping con la finalidad de realizar búsquedas automáticas y acotadas a puestos de empleo que refieran a estas palabras clave.

ofertaperfil_tipo: Esta tabla contienen información concerniente a la descripción de las 23 dimensiones que estructuran un anuncio de puesto de empleo, esta estructura permitirá diseñar el modelo dimensional del proyecto.

webscraping: Esta tabla contiene el nombre del portal de trabajo en el que se ubicó los puestos de empleo (pagina_web:cadena), la dirección electrónica del portal de trabajo (url_pagina:cadena), la dirección electrónica de la página principal conteniendo la relación de puestos de empleo publicitados (url_busqueda:cadena), la fecha de la búsqueda (fecha_creacion:fecha) y la palabra clave utilizada en la búsqueda (id_keyword:numérico).

Oferta: contiene información concerniente al título del perfil del puesto de empleo (titulo:cadena), el nombre de la empresa empleadora o grupo de interés convocante de la plaza laboral (empresa:empresa), el lugar en la que se ubica la empresa o grupo de interés (lugar:cadena), la fecha de la publicación del puesto de empleo (fecha_publicacion:fecha), el salario ofertado por la empresa (salario:moneda), la modalidad de trabajo ofrecida por el empleador (modalidad_trabajo:cadena), la dirección electrónica especifica del puesto de empleo (url_oferta:cadena).

oferta_detalle: contiene el detalle de la convocatoria segmentado en tuplas en función de las características del perfil del puesto de empleo como son: requisitos del puesto, experiencia requerida, competencias, habilidades técnicas, conocimientos, idiomas, tipo de contrato, así como los beneficios dirigidos al postulante al puesto de empleo (descripcion:cadena). La información contenida en esta tabla fue sometida al proceso de pre-procesamiento explicado a mayor detalle en la sección 3 de la presente orientada a lograr su uniformidad a fin de lograr el modelo dimensional requerido por el algoritmo de machine learning no supervisado.

Tabla 6

Esquema de Base de Datos del proceso Webscraping

N°	Esquema de Tablas
1	**keyword_search***(id_keyword:numerico,descripcion:cadena,id_tipokeyword:numerico)*
2	**ofertaperfil_tipo**(id_ofertaperfiltipo: numérico,*descripcion:cadena*)
3	**webscraping***(id_webscraping:numerico,pagina_web:cadena, url_pagina:cadena, url_busqueda:cadena, fecha_creacion:fecha, id_keyword:numerico)*
4	**oferta***(id_oferta:numerico,id_webscraping:numerico,titulo:cadena, empresa:empresa,lugar:cadena,fecha_publicacion:fecha,salario:moneda, modalidad_trabajo:cadena,url_oferta:cadena,oferta_detalle:cadena)*
5	**oferta_detalle***(id_ofertadetalle:numerico,id_oferta:numerico,descripcion:cadena)*

En la Figura 9 se expone en lenguaje de programación python parte del código del programa webscraping utilizado para extraer los puestos de empleo desde los portales de trabajo, como se puede apreciar, en las líneas 1 a 6 se importan las librerías requeridas por el programa, en la línea 7 se define la función *maledpeti_portal()*, esta función contiene la lógica de programación para conectarse a la base de datos, obtener los keywords o palabras claves invocando al método *getwords()* (línea 10), en la línea 11 se realiza un proceso iterativo según las palabras recuperadas en la línea 10 y por cada palabra realizará las siguientes acciones: inicializa el arreglo carga con los valores establecidos en las constantes *WS_PORTAL_LABORAL, WS_PAGINAS, WS_PAGINA_INICIAL, WS_OFERTAS, WS_AREA*, las cuales contiene información concerniente al portal de trabajo a procesar, el número de páginas a leer, la página iniciar a leer, la cantidad de ofertas o puestos de empleo a leer, así como el área de búsqueda respectivamente, considerando que un portal de empleo mantiene información de puestos de empleo de distintas áreas disciplinarias. En la línea 20 se setea los parámetros de búsqueda, en la línea 21 se llama al método *registrar_webscraping()*

que se encuentra definido en la clase controller, consecuencia de ello es el poblado de los esquemas *webscraping* y *oferta* descritos en la Tabla 6. Para registrar el detalle del puesto de empleo se realizan las acciones indicadas en la línea 24 discriminando según el tipo de portal de trabajo, considerando que los portales describen una estructura html distinta; este procedimiento se repite por cada palabra clave a buscar y considerada en el arreglo *palabras*. En la línea 25 se puede apreciar el método principal *main*, desde el cual se invoca la función *maledpeti_portal()* pasando como parámetro el portal a procesar con lo cual se pobla el esquema *oferta_detalle* señalado en la Tabla 6 a excepción de *keyword_search.*

Figura 9

Extracto del programa webscraping

```
1. from configuration import *
2. import webscraping_portalname
3. from controller import Controller
4. from dbconnection import Connection
5. from dboperation import DatesDB
6. import datetime
7. def maledpeti_portal(portalname):
8. controller = Controller()
9. con = connect_bd()
10. palabras= controller.getwords(con)
11. for filtro in palabras:
12. carga = {}
13. carga["pagina"] = sitio["WS_PORTAL_LABORAL"]
14. carga["cant_paginas"] = sitio["WS_PAGINAS"]
15. carga["pagina_inicial"] = sitio["WS_PAGINA_INICIAL"]
16. carga["cant_ofertas"] = sitio["WS_OFERTAS"]
17. carga["busqueda_area"] = sitio["WS_AREA"]
18. carga["busqueda"] = ""
19. carga["id_keyword"]=filtro[0]
20. set_url_busqueda(carga, sitio, filtro[1])
21. carga["id_carga"] = controller.registrar_webscraping(con, carga)
22. if sitio["WS_PORTAL_LABORAL"]== PORTAL_NAME
23. listaOferta = webscraping_computrabajo.scraping_ofertas(con,
    carga["url_principal"], carga["url_prefix"],
    carga["url_sufix"],carga["pagina_inicial"], carga["cant_paginas"],
    carga["cant_ofertas"],carga["id_carga"])
24. if __name__ == "__main__":
25. maledpeti_portal(PORTAL_NAME)
```

3.5.2 Pre-procesamiento de la Información

El procedimiento seguido en esta actividad es el señalado por Swamynathan (2017, p. 70), la finalidad consistió en limpiar el texto recopilado en el esquema especificado en la Tabla

5, la eliminación del ruido, así como de valores outliers para garantizar un análisis de similaridad de los puestos de empleo eficiente, entre los aspectos detallados seguidos se puede mencionar:

Conversión de los valores de tipo texto a mayúsculas y eliminación de tildes: Esto se realiza con la finalidad de uniformizar el texto y evitar que se interprete de manera distinta la misma palabra; funciones como lower() y replace() fueron utilizadas para lograr esta finalidad.

Tokenizacion: Este procedimiento consistió en dividir el detalle del puesto de empleo en componentes significativos como: Formación académica requerida, experiencia profesional, conocimientos, capacitaciones, certificaciones, idiomas, funciones del puesto, habilidades, tipo de contrato, horario laboral, modalidad laboral, lugar entre otros requisitos del puesto de empleo.

En la Figura 10 se expone como ejemplo el detalle de un puesto de empleo, en el podemos apreciar que la línea 1 refiere al título del empleo, la línea 2 precisa el lugar del empleo, la línea 3 muestra el tiempo de publicación del anuncio y la modalidad laboral del puesto, asimismo se puede apreciar en la línea 4 varios componentes importantes del puesto de empleo que es necesario tokenizar en componentes simples como formación(línea 5), experiencia laboral(líneas 6 y 7), capacitaciones(línea 8), conocimientos(líneas 9 y 10), la línea 11 se considera elemento del componente idiomas, la línea 12 etiqueta las funciones, estas refieren a los roles y/o responsabilidades que implican el puesto laboral, las mismas que hacen explicitas en las líneas 13 al 19; la línea 26 hacer referencia a las habilidades que debe tener el postulante al puesto laboral, asimismo en la línea 28 precisa el tipo de contrato.

Figura 10

Detalle de un Puesto de Empleo

1. **Analista Programador de Aplicaciones Tics CAS 023 - Manpower Professional**...
2. **Manpower Professional Services**
3. **Lima**
4. **REQUISITOS** :
5. Bachiller Universitario en Ingeniería de Sistemas, Ingeniería de Software, Ingeniería de Computación, Ingeniería Informática o afines por la formación.
6. No menor de tres (03) años de experiencia general en entidades públicas o privadas.
7. Curso no menor a 24 horas de Java y, Curso no menor a 24 horas de Scrum o Kanban o metodologías ágiles.
8. Conocimientos en la plataforma java, javascript, json, HTML5, CSS, consumo de servicios SOAP y REST, Angular, PL / SQL, microservicios, metodologías Agiles de Desarrollo (Scrum) y la Norma Técnica Peruana : NTP 12207.
9. Conocimientos de Inglés a nivel básico.
10. **Funciones**
11. Realizar el mantenimiento adaptativo y perfectivo a los sistemas existentes en la institución, de acuerdo con las necesidades funcionales y operativas con el fin de mantener la operatividad de los sistemas de información.
12. Realizar el proceso de implementación, en el entorno de producción de la funcionalidad, módulo o sistema informático, con el fin de que las áreas usuarias tengan un aplicativo acorde a sus necesidades.
13. Realizar el acompañamiento y la capacitación inicial en el proceso de implementación de la funcionalidad, módulo o sistema informático desarrollado, con el fin de garantizar el correcto funcionamiento de los sistemas.
14. Atender los incidentes y / o requerimientos operativos, reportados por los usuarios en el uso de las aplicaciones informáticas utilizadas en Osinergmin, con el fin de garantizar la operatividad de los sistemas de información.
15. Proponer el uso de estándares y lineamientos de desarrollo, a fin de retroalimentar los procesos asociados al ciclo de vida del software.
16. **Requirements**
17. Grado en Ingeniería Informática
18. **Skills**
19. Teamwork, Customer-oriented, Results-oriented
20. **Contract type**
21. Contrato por obra y servicio

Nota. Adaptado de Buscador Google. (2021). Resultado de una búsqueda de convocatoria de trabajo en Google. https://www.google.com/search

Toda esta información concerniente al detalle del puesto de empleo fue necesario tokenizarla en tuplas simples y significativas, para ello se utilizó como marcadores las etiquetas de html de tipo , <span class>, , , , ; entre las funciones Python utilizadas para este fin se puede mencionar: strip(), replace(), split(), rsplit(), splitlines().

Remoción del ruido: La eliminación del ruido consistió en suprimir los espacios en blanco que se identificó en la parte derecha o izquierda de una cadena de texto, números, signos de puntuación entre otros caracteres irrelevantes del detalle del puesto de empleo. En la Figura 11 se expone el código Python que se utilizó para remover el ruido, por ejemplo en la línea de código 24 se invoca la función remove_tags_html() para remover las etiquetas html que se encuentren en la cadena de texto, en la línea de código 25 se llama a la función incomplete_tags_html() para remover etiquetas html incompletas que se encuentren en el texto del detalle del puesto de empleo, asimismo las funciones remove_non_ascii() y remove_space() para remover los códigos no ascii y los espacios en blanco o caracteres en blanco de inicio y fin de la cadena o párrafo de texto correspondiente al detalle del puesto de empleo.

Figura 11

Código python para remoción de ruido

```
1.  from configuration import *
2.  from controller import Controller
3.  from preprocessing import PreProcessing
4.  from dbconnection import Connection
5.  from dboperation import DatesDB
6.  import datetime
7.  import numpy as np
8.  import string
9.  from bs4 import BeautifulSoup
10. def connect_bd():
11. con = Connection(DATABASE["DB_HOST"],DATABASE["DB_SERVICE"],
    DATABASE["DB_USER"], DATABASE["DB_PASSWORD"])
12. con.connect()
13. return con
14. if __name__ == "__main__":
15. controller = Controller()
16. preprocessing = PreProcessing()
17. con = connect_bd()
18. oferta_detalle = controller.dbofertadetalle
19. datos = oferta_detalle.select_ofertadetalle_dimension(con,3)
20. datos = np.array(datos)
21. i = 1#columna que queremos obtener
22. matrix = [fila[i] for fila in datos]
23. #normalizar data
24. matrix = preprocessing.remove_tags_html(matrix)
25. matrix = preprocessing.remove_incomplete_tags_html(matrix)
26. matrix = preprocessing.remove_non_ascii(matrix)
27. matrix = preprocessing.remove_space(matrix)
28. #modificar la columna de descripcion
```

```
29. for indice in range(0,len(matrix)):
30. datos[indice][1] = matrix[indice]
31. oferta_detalle.update_ofertadetalle_normalized(con,datos)
```

La implementación de las funciones referidas se puede apreciar en la Figura 12 a través de la clase PreProcessing que empaqueta la implementación de las funciones descritas en la Figura 11 entre otras funciones adicionales que son necesarias para realizar el proceso de pre-procesamiento de la información.

Figura 12

Clase PreProcessing python para remoción de ruido

```
1. import inflect
2. from nltk.corpus import stopwords
3. from nltk.stem import LancasterStemmer
4. from nltk.stem import SnowballStemmer
5. from nltk.stem import WordNetLemmatizer
6. from nltk.tokenize import word_tokenize
7. import re
8. import unicodedata
9. import datetime
10.  from googletrans import Translator
11.  from mtranslate import translate
12.  from bs4 import BeautifulSoup
13.  import string

14.  class PreProcessing:
15.  def __init__(self):
16.  pass

17.  def remove_space(self,words):
18.  new_words = []
19.  for word in words:
20.  new_word = word.strip()
21.  new_words.append(new_word)
22.  return new_words

23.  def remove_incomplete_tags_html(self,words):
24.  new_words = []
25.  rep = {"<STRONG": "", "STRONG>": "","ENDIF":""}
26.  for word in words:
27.  rep = dict((re.escape(k), v) for k, v in rep.items())
28.  pattern = re.compile("|".join(rep.keys()))
29.  new_word = pattern.sub(lambda m: rep[re.escape(m.group(0))], word)
30.  new_words.append(new_word)
31.  return new_words

32.  def remove_tags_html(self,words):
33.  new_words = []
34.  for word in words:
35.  new_word = BeautifulSoup(word, "lxml").text
36.  new_words.append(new_word)
37.  return new_words
```

```
38.  def remove_punctuation(self,words):
39.  new_words = []
40.  for word in words:
41.  new_word = re.sub(r'[^\w\s]', '', word)
42.  if new_word != '':
       new_words.append(new_word)
43.  return new_words

44.  def replace_numbers(words):
45.  p = inflect.engine()
46.  new_words = []
47.  for word in words:
48.  if word.isdigit():
       new_word = p.number_to_words(word)
       new_words.append(new_word)
49.  else:
       new_words.append(word)
50.  return new_words

51.  def delete_empty(words):
52.  elements = []
53.  for word in words:
54.  if word is not None and word.strip() != "":
       elements.append(word)
55.  return elements
```

Nota. Adaptado de "Test data preprocessing", por Mayo, 2020, https://gist.github.com/mmmayo13

En la Figura 13 se muestra el resultado, después de realizar el pre-procesamiento de las habilidades requeridas en los puestos de empleo, la información se muestra en la forma de una nube de palabras.

Figura 13

Nube de Palabras de Habilidades

Nota. Elaboración propia con la herramienta Worldcloud de Davies, 2021, https://www.jasondavies.com/wordcloud/

3.5.3 Modelo dimensional

El modelo dimensional toma en consideración catorce dimensiones primarias y una tabla de hechos con una granularidad transaccional, lo que significa que por cada puesto de empleo se genera entradas en la tabla de hechos la cual se constituirá en el dataset a ser evaluado por la técnica de machine learning no supervisada (Kimball y Ross, 2002).

La Tabla 7 describe en la primera columna los nombres de las dimensiones identificadas como relevantes para evaluar la similaridad de los puestos de empleo utilizando técnicas de machine learning no supervisado como clustering, asimismo la segunda columna de la tabla referida describe la información contenida en cada una de las dimensiones y tabla de hechos de la propuesta.

Tabla 7

Dimensiones y Hechos de la propuesta

Dimensión	Descripción
D01 dim_categoria:	- Contiene información concerniente a la categorización de los perfiles de los puestos de empleo
D02 dim_perfil:	- Contiene información concerniente a los Títulos de los puestos de empleo
D03 dim_pagina_web:	- Contiene información concerniente a las páginas web de los portales web de empleo
D04 dim_empresa:	- Contiene información concerniente a los empleadores autores de los puestos de empleo publicados

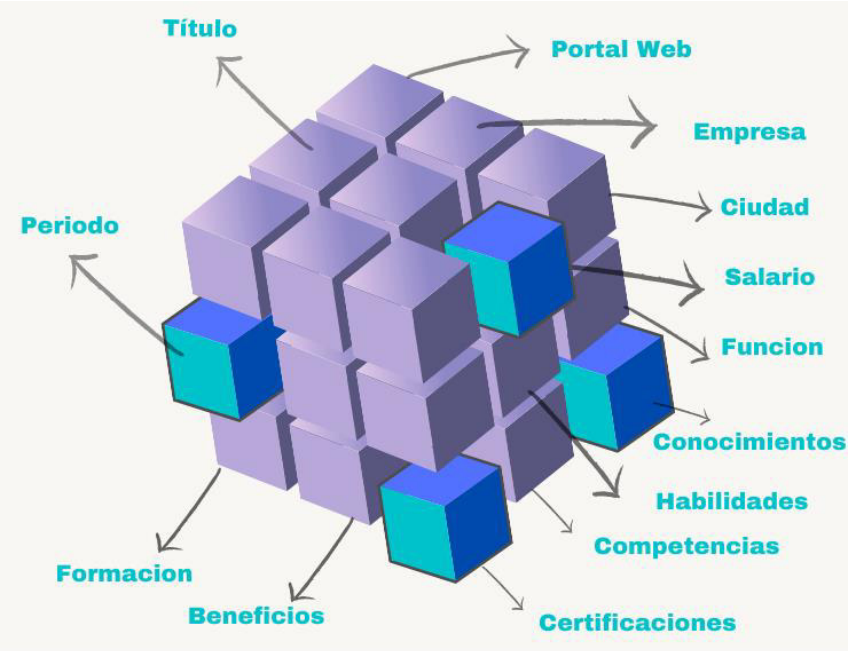
D05 dim_lugar:	-	Contiene información concerniente a las ciudades para desempeñar el empleo
D06 dim_salario:	-	Contiene información concerniente a los salarios ofrecidos por los empleadores
D07 dim_periodo:	-	Contiene información a la fecha de publicación del puesto de empleo
D08 dim_funcion:	-	Contiene información concerniente a las funciones a desempeñar por el candidato al puesto de empleo
D09 dim_conocimiento:	-	Contiene información referente a los conocimientos requeridos para el puesto de empleo
D10 dim_competencia:	-	Contiene información concerniente a las capacidades técnicas requeridas para el puesto de empleo.
D11 dim_habilidad:	-	Contiene información referente a las habilidades blandas, habilidades sociales que debe cumplir el candidato.
D12 dim_certificacion:	-	Certificaciones técnicas requeridas para el puesto de empleo.
D13 dim_beneficio:	-	Contiene información referente a los beneficios que ofrece el empleador a los admitidos al puesto de empleo.
D14 dim_formacion	-	Corresponde a la formación de pregrado mínima requerida para el puesto de empleo.
F01 fact_PuestoDeEmpleo	-	Contiene los hechos transaccionales concernientes a las ofertas de empleo

En la Figura 14 se puede apreciar mediante un cubo multidimensional las características de un anuncio de trabajo, estas características conformaran el dataset a ser utilizado por el

algoritmo de Clustering con la finalidad de identificar patrones según similitud en la información.

Figura 14

Modelo Multidimensional



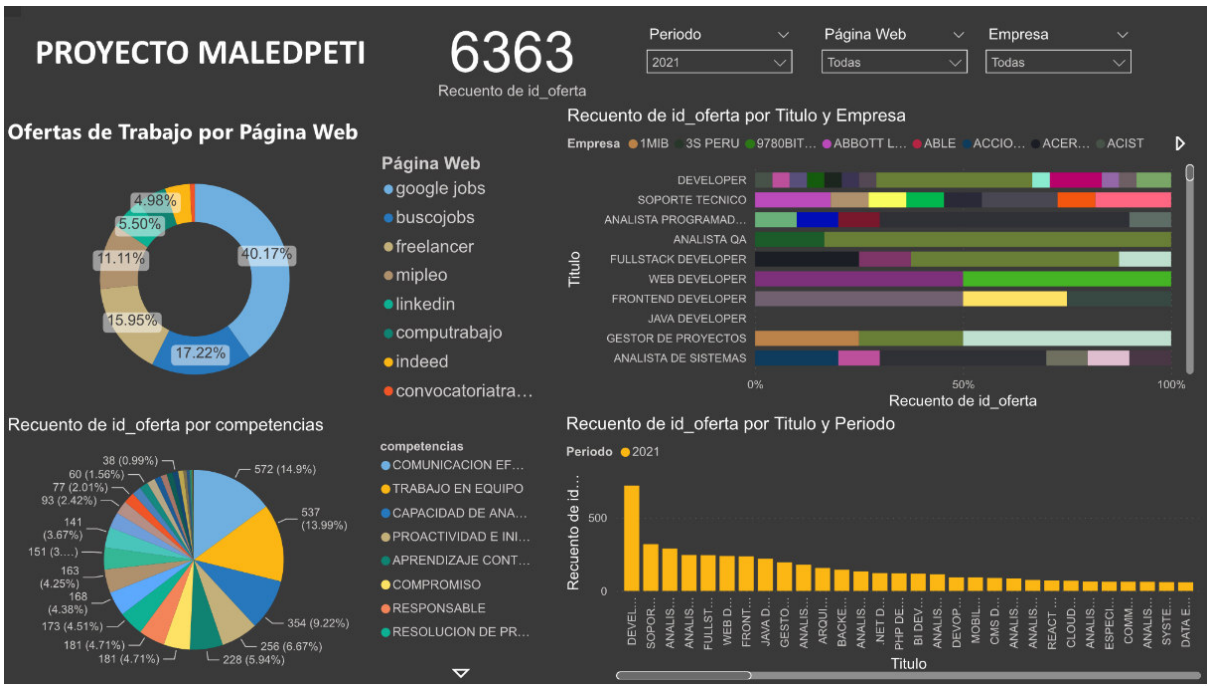
El modelar la información mediante un esquema multidimensional permite apreciar la data desde diferentes vistas como se muestra en la Figura 15 el conjunto de Dashboards del Proyecto elaborado con la herramienta de Microsoft (2021) “Power BI”. La información concierne a los puestos de empleo de profesionales de TI se ha organizado por periodo, página web, empresa, pudiéndose incorporar otras dimensiones. Se observa cuatro vistas fundamentales, la primera expone a razón de porcentaje los puestos de empleo por página web, observando que el mayor porcentaje correspondiente al periodo 2021 corresponde a la página de Google Jobs con un 40.17% mientras que las otras páginas exponen menor porcentaje para el mismo periodo.

Asimismo, se observa mediante un gráfico circular las competencias generales que suelen requerir las empresas, se aprecia que la comunicación efectiva, trabajo en equipo,

capacidad de análisis y proactividad e iniciativa son las más demandadas en los perfiles de empleo de profesionales de TI. Por otro lado, se presenta los principales perfiles de TI requeridos por las principales empresas privadas como bancos, financieras, mineras, empresas de servicios, retails; por el sector público se tiene ministerios, municipalidades, entre otras instituciones gubernamentales. El cuarto dashboard expone la demanda de los perfiles por periodo, resaltando el perfil developer, soporte técnico, analista programador, analista de calidad, fullstack developer entre los más demandados en el periodo 2021.

Figura 15

Dashboards del Proyecto



3.5.4 Algoritmo de machine learning no supervisado

La presente investigación utilizó la propuesta del algoritmo k-means++ de Arthur y Vassilvitskii (2007), básicamente esta variante contempla una forma de inicialización basado en la determinación de centroides aleatorios con probabilidades muy específicas logrando mejoras sustanciales en términos de precisión y velocidad respecto al tradicional método K-means de Lloyd. El algoritmo kmeans++ requiere se precise como parámetro de entrada el

valor de k conglomerados que se desea obtener, por lo que se recurrió a la estrategia denominada Método del Codo de Pranav et al. (2018) para estimar el número ideal de conglomerados k en función del punto de inflexión en la gráfica “k respecto a la distancia promedio al centroide”. La función objetivo del algoritmo es minimizar en los k conglomerados la suma de las distancias al cuadrado cuya formula se indica en 1.

$$\text{SSE} = \sum_{k=1}^{k} \sum_{(x_i \in k)} \|x_i - c_k\|^2, donde\ c_k\ es\ centroide\ del\ cluster. \qquad (1)$$

3.5.5 Prototipo

Se desarrolló un prototipo web utilizando las buenas prácticas de la ingeniería de software continua, con una arquitectura basada en servicios y DevOps como paradigma para integración y entrega progresiva, ágil y colaborativa, tomando como referencia para su implementación lo señalado en el artículo de Mamani et al. (2020). En las siguientes secciones se desarrolla la propuesta de prototipo al cual se le ha denominado MALEDPETI, estas son las siglas del título de la presente investigación.

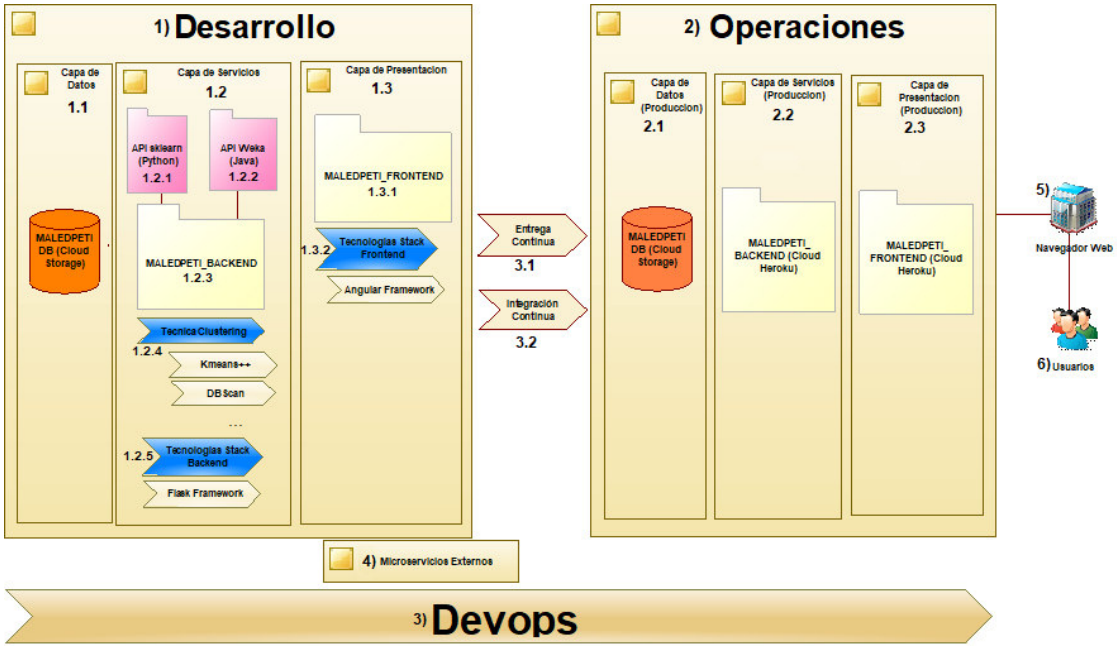
3.5.5.1 Arquitectura del Prototipo. En la Figura 16 se expone la Arquitectura basada en servicios del Proyecto; en ella se establece la relación entre el desarrollo de un producto software y su despliegue continuo mediante entregables de valor hacia el usuario.

En el punto 1) se esquematiza la organización del ambiente de desarrollo en tres capas fundamentales: la Capa de Datos (1.1), la Capa de Servicios (1.2) y la Capa de Presentación (1.3). El espacio 1.1 contiene la Base de Datos con información de los puestos de Empleo modelada mediante dimensiones detalladas en la Tabla 7, la Base de Datos se encuentra gestionada por herramientas de código abierto como: el motor PostgreSQL versión 13.4 sobre un sistema operativo Linux Ubuntu 13.4 64-bit con alojamiento en un servidor cloud DigitalOcean bajo pago; pgAdmin como herramienta para desarrollo y administración del servidor de base de datos.

El espacio 1.2 contiene la Capa de servicios, la misma está compuesta por el backend del proyecto(1.2.3) desarrollado con el micro framework Flask (1.2.5) este se encuentra programado en Python, una de las características de valor es que permite construir aplicaciones web de manera ágil (Flask, 2021), por otro lado Python es un lenguaje programación para desarrollo rápido de aplicaciones e integración de servicios, es de código abierto, amigable y fácil de aprender; asimismo permite reutilizar una diversidad de paquetes de la comunidad de software, dotándole de amplias capacidades de desarrollo de computación matemática y científica como se puede mencionar: SciPy, Pandas, Scikit-learn entre otros ecosistemas (Python, 2019). El backend del presente proyecto utiliza la API sklearn (1.2.1), proporciona algoritmos de machine learning supervisado y no supervisado, asimismo suministra herramientas para preprocesamiento de data, modelos de ajuste, selección, evaluación entre otras utilidades. El espacio 1.3 contiene la Capa de Presentación, aquí se ubica el frontend del proyecto, su desarrollo se realizó con el framework Angular; la finalidad de la arquitectura fue incorporar la cultura devops centrado en integración y entregas continuas al ambiente de producción con despliegue en servidores cloud como digital ocean, heroku, así se aprecia en los puntos 2.1, 2.2 y 2.3 de la propuesta arquitectural, complementariamente se visualiza en los puntos 5) y 6) la interacción del usuario final con el prototipo MALEDPETI mediante un browser o navegador.

Figura 16

Arquitectura basada en servicios del Prototipo



Nota. Adaptado de "Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua", por Mamani et al., 2018, *Industrial Data, 2*(23).

3.5.5.2 Clustering Kmeans. En la presente sección se explicará aspectos relacionados a la codificación del prototipo, enfocándonos en la lógica del backend y frontend del proyecto.

A. Backend del Prototipo. El backend representa la parte dura del prototipo, contiene la lógica de la implementación del algoritmo kmeans++ en lenguaje de programación Python y Java, así como las métricas y gráficos, los cuales serán expuestos como servicios.

En la Figura17 se expone el código fuente del Backend desarrollado en Python con el framework Flask; como se aprecia en las líneas de código: 1 a 10 se importan las librerías específicas de los paquetes flask, scipy, sklearn; en la línea 11 se define el objeto app como una instancia Flask, en la línea 12 intencionalmente se estableció puntos suspensivos para omitir algunas líneas de código con baja relevancia para su explicación, en la línea 13 inicia el código correspondiente al endpoint *kmeans*, este endpoint o ruta también llamado, permite invocar funcionalidad mediante los métodos 'GET' y 'POST', en la línea 14 se define la función *kmeans()* asociada al endpoint con el mismo nombre, en la línea de código 15 se establece la conexión a la base de datos postgresql mediante el adaptador para Python psycopg2, este

requiere consignar los parámetros como el nombre de la base de datos, usuario, clave, el host y el puerto donde se encuentra alojado la base de datos.

En la línea 16 se establece un apuntador a la sesión activa de base de datos para lanzar comandos *structure query language* (sql), en la línea 17 se valida el método requerido, en el caso sea GET devuelve data en formato json utilizando el método *jsonify* de Flask, en caso se trate de una invocación POST (línea 18) se procede a obtener en el objeto *body* el request suministrado desde el frontend en formato json, en las líneas 20 a 28 se aprecia la extracción de los diferentes parámetros del objeto *body* requeridos por el algoritmo de clustering *kmeans* como el enunciado de consulta sql denominado *q1* utilizado para crear el dataset:

```
select o.htitulo_cat, o.htitulo, w.pagina_web, o.empresa, o.lugar,
o.salario, date_part('year',o.fecha_publicacion) as periodo,
f_dimPuestoEmpleo(o.id_oferta,7) as funciones,
f_dimPuestoEmpleo(o.id_oferta,1) as conocimiento,
f_dimPuestoEmpleo(o.id_oferta,3) as competencias,
f_dimPuestoEmpleo(o.id_oferta,2) as habilidades,
f_dimPuestoEmpleo(o.id_oferta,17) as certificaciones,
f_dimPuestoEmpleo(o.id_oferta,5) as beneficio,
f_dimPuestoEmpleo(o.id_oferta,11) as formacion
from webscraping w inner join oferta o on (w.id_webscraping=o.id_webscrapin
g) where o.id_estado is null;
```

Así como el enunciado de consulta sql denominado *q2* con la finalidad de evaluar su comportamiento y precisión en la aplicación de los algoritmos.

```
select o.htitulo, f_dimPuestoEmpleo(o.id_oferta,7) as funciones,
f_dimPuestoEmpleo(o.id_oferta,1) as conocimiento,
f_dimPuestoEmpleo(o.id_oferta,3) as competencias,
f_dimPuestoEmpleo(o.id_oferta,2) as habilidades,
f_dimPuestoEmpleo(o.id_oferta,5) as beneficio from webscraping w inner join
oferta o on (w.id_webscraping=o.id_webscraping) where o.id_estado is null;
```

Otros parámetros como: el número de instancias del dataset (total_data), el número de clusters especificado por el usuario (n_clusters), el mecanismo de determinación inicial de centroides (random_state): {k-means++, random}, el número máximo de interacciones a considerar por el algoritmo (max_iter), en la línea 29 se inicializa el objeto result, el cual será utilizado para el alojamiento de los resultados que retornará al frontend el endpoint *kmeans*.

En la línea 30 se obtiene en el arreglo *field_names* los nombres de los campos que devuelve el query, también se les conoce como labels o etiquetas como se muestra en la Figura 18; la línea 31 expone la invocación al método Dataframe de la clase *pandas* con la finalidad de formatear el dataset en filas y columnas, requiere como parámetros la data y los labels o nombres de las columnas, el resultado es un dataset estructurado, en vista que el dataset está compuesto de datos categóricos, en la línea 32 se instancia al método LabelEncoder() de la clase preprocessing del paquete sklearn, este permite codificar las etiquetas de destino con un valor entre 0 y n-1, donde n es el número total de clases, este codificador se aplica al dataset y el resultado se devuelve en el objeto transformed_data como se muestra en la Tabla 8.

Figura 17

Código fuente del Backend – endpoint kmeans: request

```
1. import os, psycopg2, json, io, base64
2. import pandas as pd
3. from scipy import spatial
4. from sklearn import preprocessing
5. from flask import Flask, request, request, jsonify
6. from flask_cors import CORS
7. from flask_sqlalchemy import SQLAlchemy
8. from sklearn.feature_extraction.text import TfidfVectorizer
9. from sklearn.cluster import KMeans
10.  from matplotlib import pyplot as plt
11.  app = Flask(__name__)
12.  …
13.  @app.route("/kmeans", methods = ['GET', 'POST'])
14.  def kmeans():
15.  con = psycopg2.connect(database="maledpeti", user="modulo4", password=
   "modulo4", host="128.199.1.222", port="5432")
16.  cursor = con.cursor()
17.  if request.method == 'GET': return jsonify(load_data())
18.  if request.method == 'POST':
19.  body        = request.get_json()
20.  query       = cursor.execute(body["query"])
21.  total_data  = cursor.fetchall()
22.  n_clusters  = body["n_clusters"]
23.  init        = body['init']
24.  n_init      = body['n_init']
25.  random_state= body['random_state']
26.  max_iter    = body['max_iter']
27.  axis_x      = int(body['axis_x'])
28.  axis_y      = int(body['axis_y'])
29.  result      = {}

30.  field_names = [i[0] for i in cursor.description]
31.  dataframe = pd.DataFrame(total_data, columns=field_names)
```

```
32.  label_encoder = preprocessing.LabelEncoder()
33.  transformed_data = dataframe.apply(label_encoder.fit_transform)
```

Figura 18

Labels del Dataset

['categoria', 'perfil', 'pagina_web', 'empresa', 'lugar', 'salario', 'periodo', 'funciones', 'conocimiento', 'habilidades', 'competencias', 'certificaciones', 'beneficio', 'formacion']

Tabla 8

Dataset con Labels codificados de 0 a n-1

	categoria	***perfil***	***pagina_web***	***...***	***certificaciones***	***beneficio***	***formacion***
0	*0*	*31*	*0*	*...*	*9*	*0*	*9*
1	*0*	*32*	*6*	*...*	*9*	*7*	*6*
2	*0*	*33*	*0*	*...*	*9*	*0*	*9*
3	*0*	*33*	*0*	*...*	*9*	*7*	*9*
4	*0*	*33*	*0*	*...*	*9*	*7*	*9*
...	*...*	*...*	*...*	*...*	*...*	*...*	*...*
6831	*18*	*119*	*3*	*...*	*9*	*15*	*9*
6832	*18*	*119*	*3*	*...*	*9*	*15*	*9*
6833	*18*	*119*	*4*	*...*	*9*	*5*	*7*
6834	*18*	*119*	*4*	*...*	*9*	*15*	*2*
6835	*18*	*119*	*4*	*...*	*9*	*9*	*9*

En la Figura 19 se expone las líneas de código del *endpoint* del método del codo, en la línea 1 se define el nombre del *endpoint* y los métodos GET y POST soportados, en la línea 2 inicia su implementación, línea 3 se invoca al objeto conexión a base de datos *con* retornando el dataset en el objeto cursor, en la línea 6 previa validación del método POST se obtiene los parámetros en el objeto *body*, las líneas 8 y 9 ejecutan el query asignado al endpoint, a

continuación en las líneas 10 a 14 se desagrega los parámetros en variables individualizadas las cuales serán asignadas al invocar el método *kmeans* (línea 23), la línea 20 se inicializa el arreglo distorsiones *distortions*, en la línea 21 se establece un rango de [2, n_*clustersMax*>, mediante una estructura de control repetitiva se genera modelos *kmeans* con modo de inicialización aleatorio, se invoca al método *fit* pasándole como parámetros el dataset con labels codificados, se añade la métrica *inercia* determinada para cada k número de clusters establecido en el rango indicado en la línea 21, las inercias calculadas se pueden apreciar en la Figura 20 y expresadas mediante un gráfico, el cual es generado mediante la lógica de programación establecidas desde la línea 26 a la 34, la línea 36 y 37 permiten retornar el código binario de la imagen en formato *json*.

Figura 19

Código fuente del Backend – endpoint Método del Codo

```
1.   @app.route("/MetododelCodo", methods = ['GET', 'POST'])
2.   def MetododelCodo():
3.   cursor = con.cursor()
4.   if request.method == 'GET':
5.         return jsonify(load_data())
6.   if request.method == 'POST':
7.      body        = request.get_json()
8.      query       = cursor.execute(body["query"])
9.      total_data  = cursor.fetchall()
10.     n_clustersMax  = body["n_clusters"]
11.     init        = body['init']
12.     n init      = body['n init']
13.     random_state= body['random_state']
14.     max_iter    = body['max_iter']
15.     result      = {}

16.     field_names = [i[0] for i in cursor.description]
17.     dataframe = pd.DataFrame(total_data, columns=field_names
18.     label_encoder = preprocessing.LabelEncoder()
19.     transformed_data = dataframe.apply(label_encoder.fit_transform)
        #metodo del codo
20.     distortions = []
21.     K = range(2,n_clustersMax+1)
22.     for k in K:
23.        kmeanModel = KMeans(n_clusters=k, init=init,
           max_iter=max_iter, n_init=n_init, random_state=random_state)
24.        kmeanModel.fit(transformed_data)
25.        distortions.append(kmeanModel.inertia_)
26.     plt1.plot(K, distortions, 'bx-')
27.     plt1.xlabel('k clusters')
28.     plt1.ylabel('Distorción')
```

```
29.     plt1.title('El método del codo muestra el k clusters óptimo.')
30.     my_stringIObytes = io.BytesIO()
31.     plt1.savefig(my_stringIObytes, format='jpg')
32.     my_stringIObytes.seek(0)
33.     my_base64_jpgData = base64.b64encode(my_stringIObytes.read())
34.     result["elbow_method"] = my_base64_jpgData.decode()
35.     plt1.clf()
36.     response = jsonify(result)
37.     return response
```

Figura 20

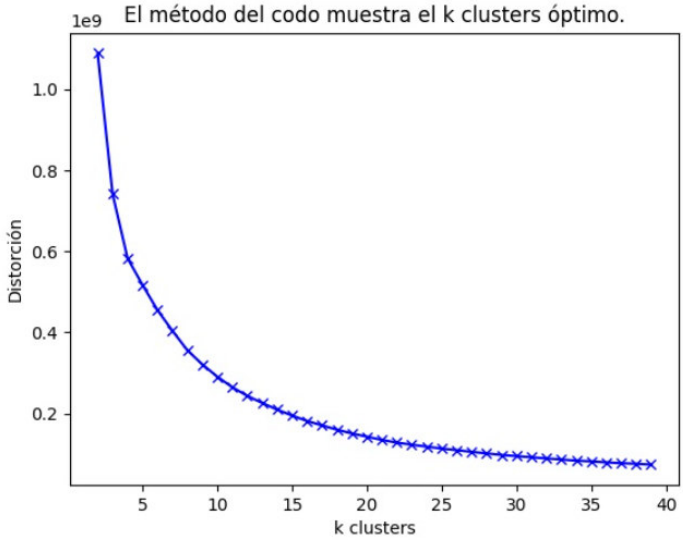
Inercias determinadas para un rango k clusters

[1089196262.0645504, 741436441.0204611, 582923317.8194567, 517516581.47747993, 486746005.4619835, 405454031.1981781, 367329619.6458074, 345876115.1816016, 310901631.6263132, 275812445.0824468, 250462595.74223536, 229322431.49881086, 215375256.83984312, 202071529.76033583, 195924091.36137778, 181497646.5861178]

Una vez determinadas las inercias, estas se grafican en un plano de dos dimensiones, el eje x contiene los k clusters correlativos según el intervalo indicado en la línea 21, el eje y se determina con las inercias calculadas, el resultado se puede apreciar visualmente en la Figura 21, ubicando el punto en el cual se suaviza la curva, dejando de decrecer aceleradamente, basado en la distorsión, ese punto vendría a ser el número de cluster a considerar en el proceso de clustering.

Figura 21

Gráfico del Método del Codo



Una vez determinado el número de clústeres a considerar tomando como referencia el método del codo, en la línea 34 de la Figura 22 se instancia un objeto KMeans sesteándole los parámetros como el número de clústeres a generar, el número máximo de iteraciones y el mecanismo de inicialización de los centroides, a continuación, se invoca el método *fit_predict* pasándole como parámetro el dataset formateado con *labels* codificados, el resultado se asigna a la variable *pred_y* conteniendo el número de clúster al cual pertenece cada índice del dataset como se muestra en la Tabla 9, el índice de la posición cero (0) se ha determinado que pertenece al clúster #4, el índice de la posición uno (1) está más cercano al clúster #3, de igual forma el índice del dataset que ocupa la posición dos (2) y así sucesivamente hasta el último índice del dataset es asignado a un #cluster específico.

Figura 22

Código fuente del Backend – endpoint kmeans:build

```
34.  kmeans = KMeans(n_clusters=n_clusters, init=init, max_iter=max_iter, n
   _init=n_init, random_state=random_state)
35.  pred_y = kmeans.fit_predict(transformed_data)
36.  elements = kmeans.labels_
37.  centroids = kmeans.cluster_centers_
```

Tabla 9

Resultado de Clustering Kmeans

Índice del Dataset	**0**	**1**	**2**	**….**	**6833**	**6834**	**6835**
# cluster	4	3	3	….	1	0	3

La clase KMeans del paquete *sklearn.cluster* permite obtener los centroides de los clusters determinados como se aprecia en las líneas 38 a 47 en la siguiente sección de código:

```
38.  for _centroid in centroids_all_data:
   obj = {}
   obj["point"] = (centroids.tolist())[x]
   obj["distance"] = float(_centroid[0])
   obj["position"] = int(_centroid[1])
```

```
   obj["title_cluster"]= json.loads((dataframe.iloc[centroids_values[x]]).t
   o_json(orient='values'))
   centroids_details.append(obj)
   x+=1
39.  result["centroids"] = centroids_details
40.  result["inertia"] = kmeans.inertia_
41.  result["n_iter"] = kmeans.n_iter_
42.  result["total_instances"] = len(dataframe.index)
43.  result["columns"] = field_names
44.  result["data"] = json.loads(dataframe.sort_values(['cluster'], ascendi
   ng=True).to_json(orient='table'))
45.  clusters = []
46.  for item in range(n_clusters):
   temporal_cluster = 'Cluster {}'.format(item)
   length_actual_cluster = int(dataframe["cluster"].value_counts()[item])
   decimal_frequency_actual_cluster = float(dataframe["cluster"].value_coun
   ts(normalize=True)[item])
   obj = {
          "cluster": temporal_cluster,
          "length": length_actual_cluster,
          "percentage": (round(decimal_frequency_actual_cluster*100, 2)),
          "title_cluster": json.loads((dataframe.iloc[centroids_values[ite
          m]]).to_json(orient='values'))}clusters.append(obj)
47.  result["clusters"] = clusters
48.  response = jsonify(result)
49.  return response
50.  if __name__ == '__main__': app.run()
```

La clase *QueryREST* permite exponer los resultados del *endpoint kmeans* como: centroides, inercia, # de iteraciones, son requeridos para el proceso de clustering, # total de instancias del dataset, los nombres de los campos, el dataset incluyendo el # de cluster de cada una de las instancias que la componen.

```
1. package pe.giinwe.maledpeti.rest;
2. import java.io.IOException;
3. import org.springframework.http.ResponseEntity;
4. import org.springframework.web.bind.annotation.RequestBody;
5. import org.springframework.web.bind.annotation.RequestMapping;
6. import org.springframework.web.bind.annotation.RequestMethod;
7. import org.springframework.web.bind.annotation.RestController;
8. import pe.giinwe.entity.JSONQueryKmeans;
9. import pe.giinwe.entity.ResultKmeans;
10.  import pe.giinwe.entity.ResultKmeansDAO;
11.  @RestController
12.  public class QueryREST {
13.      @RequestMapping(value="/kmeansweka", method=RequestMethod.POST)
14.      public     ResponseEntity<ResultKmeans>     postResult(@RequestBody
   JSONQueryKmeans temporal) throws IOException{
15.          ResultKmeansDAO            process            =            new
   ResultKmeansDAO(temporal.getQuery());
16.          try{
17.              return ResponseEntity.ok(process.getResult(temporal));
18.          }catch(Exception e){
19.              System.out.println("Fallo el metodo 'getResult': "+ e);
20.              return ResponseEntity.notFound().build();
21.          }
```

```
22.      }
23.  }
```

El prototipo considero la integración de las APIs sklearn de Scikit-learn (2020) y weka de la Unidad de Waikato (2021) con la finalidad de evaluar el comportamiento de los algoritmos kmeans que estas implementan con respecto al mismo dataset, para lo cual se hacía necesario desarrollar el endpoint *kmeansweka* en Java con el framework spring boot, la clase *ResultKmeansDAO* del siguiente código expone parte de su implementación.

```
1. import weka.clusterers.SimpleKMeans;
2. import weka.core.Instances;
3. import weka.core.converters.DatabaseLoader;
4. public class ResultKmeansDAO {
5. Instances data=null; SimpleKMeans kmeans;
6. public ResultKmeansDAO(String Query) throws IOException{
7. Connection con = new Connection();
8. DatabaseLoader db = con.getConnection(Query);
9. db = con.getConnection(Query); data = new Instances(db.getDataSet());}
10.  public ResultKmeans getResult(JSONQueryKmeans request){
11.  return getKmeans(request);}
12.  public ResultKmeans getKmeans(JSONQueryKmeans request){
13.  wkmeans = new SimpleKMeans();
14.  try{wkmeans.setPreserveInstancesOrder(true);wkmeans.setSeed(10);
15.  wkmeans.setInitializationMethod(request.getInit());
16.  wkmeans.setNumClusters(request.getN_clusters());
17.  wkmeans.setMaxIterations(request.getMax_iter());
18.  wkmeans.buildClusterer(data);
19.  double[] sizes= wkmeans.getClusterSizes();
20.  Instances instancias = wkmeans.getClusterCentroids();
21.  ResultKmeans result = new ResultKmeans();
22.  result.initNodes(sizes,instancias);result.setTotal_instances(data.size
   ());result.init_centroids(instancias);result.init_columns(data);
23.  result.init_data(data,kmeans.getAssignments());return
   result;}catch(Exception e){System.out.println("Error en metodo
   'getKmeans': "+ e);return null;}}}
```

B. Frontend del Prototipo. En esta sección se expone el código fuente del Frontend del proyecto, para su desarrollo se utilizó el framework angular por presentar el enfoque de aplicaciones de una sola página (SPA) propicio para aplicaciones que interactúan con el servidor mediante una API rest, este tipo de enfoques ejecutan la mayor parte de la lógica de la interfaz del usuario en el navegador web haciendo más liviana su ejecución, descentralizando la carga de procesamiento del servidor.

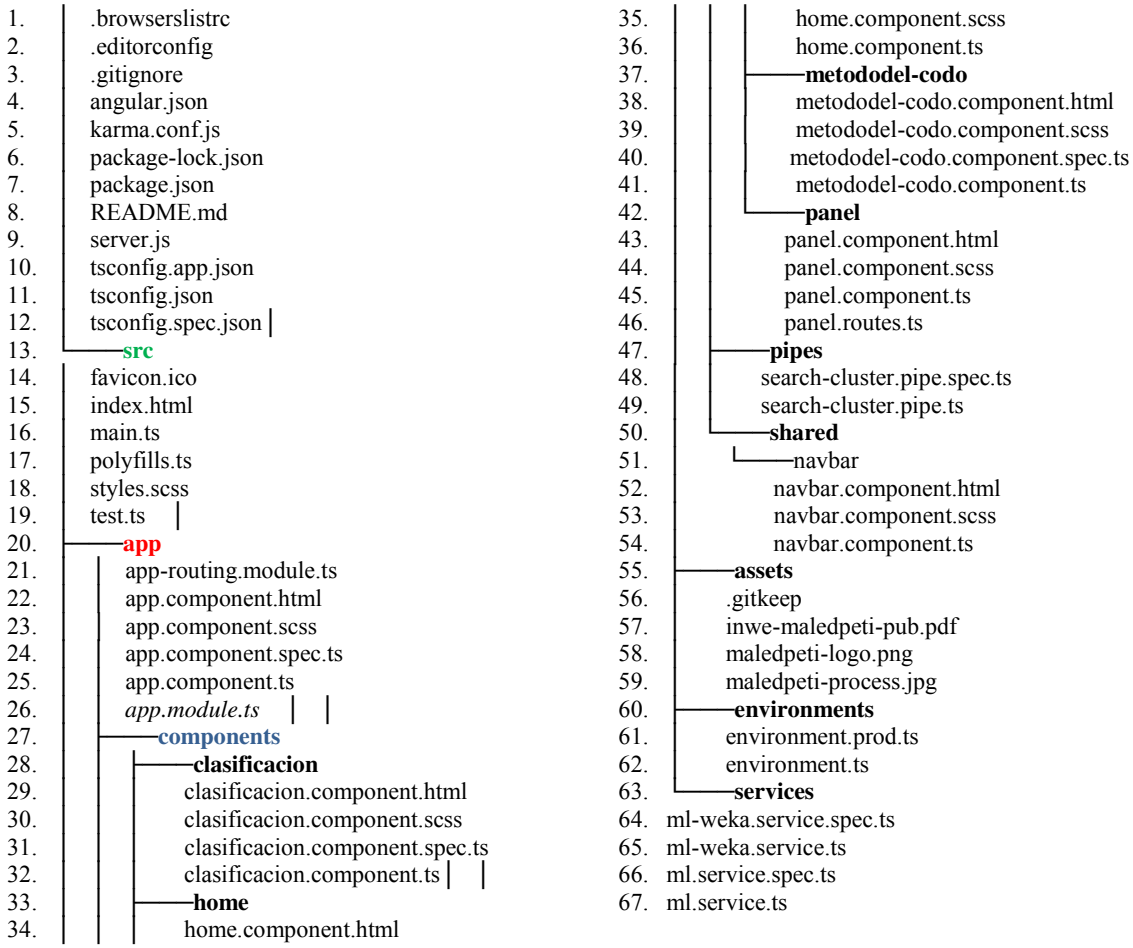
En el esquema de la Figura 23 se expone la estructura del código fuente del *Frontend* del proyecto a través de una lista de archivos, está compuesto en módulos, componentes y

servicios e inyección de dependencias, en la línea 26 del esquema se aprecia el módulo que contiene la clase principal del frontend.

Angular define la logica de un componente en cuatro archivos con extension .html, .css, .ts y .spec.ts, esto se puede divisar en las lineas 29 a 32 del esquema, componente clasificacion; la extension .ts corresponde a archivos typescript, se trata de un lenguaje de programacion construido como una capa superior a javascript, entre las bondades que presenta javascript se tiene la deteccion de errores, limpieza, sencilles, y solidez en el codigo.

Figura 23

Esquema del Frontend del Prototipo



En la clase *AppModule* se especifica como se vincula la lógica de los componentes que implementan las funcionalidades del frontend del prototipo, para ser encapsulados en el decorador @NgModule, como se aprecia en las líneas 16 al 21, según su naturaleza se organizan en declaraciones de tipo: imports, providers, exports y bootstrap, este último indica la lógica del componente a ejecutarse al iniciar la ejecución del frontend.

```
1. import { NgModule } from '@angular/core';
2. import { BrowserModule } from '@angular/platform-browser';
3. import { FormsModule ,ReactiveFormsModule } from '@angular/forms';
4. import { HttpClientModule } from '@angular/common/http';
5. import {NgxPaginationModule} from 'ngx-pagination';
6. import { AppRoutingModule } from './app-routing.module';
7. import { AppComponent } from './app.component';
8. import { PanelComponent } from './components/panel/panel.component';
9. import { NavbarComponent } from './shared/navbar/navbar.component';
10. import { HomeComponent } from './components/home/home.component';
11. import { SearchClusterPipe } from './pipes/search-cluster.pipe';
12. import { MetododelCodoComponent } from './components/metododel-
   codo/metododel-codo.component';
13. import { ClasificacionComponent } from
   './components/clasificacion/clasificacion.component';
14. import { Chart } from 'chart.js';

15. @NgModule({
16. declarations:
   [AppComponent,KmeansComponent,PanelComponent,NavbarComponent,HomeCompon
   ent,SearchClusterPipe,MetododelCodoComponent,ClasificacionComponent],
17. imports:
   [AppRoutingModule,BrowserModule,FormsModule,ReactiveFormsModule,HttpCli
   entModule,NgxPaginationModule],
18. exports: [NgxPaginationModule],
19. providers: [],
20. bootstrap: [AppComponent]})
21. export class AppModule { }
```

En la clase *ClasificacionComponent* se expone en typescript parte del codigo del componente clustering kmeans a nivel frontend. En la linea 3 a traves del constructor de la clase se invoca dos objetos: *mlService* y *mlWekaService* cada uno refiere al servicio backend correspondiente, el primero linkea al servicio backend que implementa el algoritmo clustering kmeans con el uso de la API Scikit-learn en python y el segundo linkea al servicio backend que implementa el algoritmo clustering kmeans en java con la API Weka, ambas implementaciones personalizadas y orientadas al descubriemiento de patrones y/o comportamiento en la detección

de Similitud de Puestos de Empleo de Profesionales de TI, en las lineas 9 y 20 se aprecian los servicios en formato typescript que invocan a los backends referidos.

```
1. …
2. export class ClasificacionComponent implements OnInit {
3. … constructor(private formbuilder: FormBuilder,private
mlService:mlService,private mlWekaService:mlWekaService,private _sanitizer:
DomSanitizer,)
4. ngOnInit(): void {this.sendData();  }
5. sendData() {…});}
6. …
7. runKmeans(){
8. …
9. this.mlService.runKmeans(this.form.value).subscribe((result: any)=>{
10.  this.response = result;
11.  this.data = result?.data.data;
12.  this.columns = result?.columns.filter((item:any) => item !==
"cluster");
13.  let no_sorted_clusters = result?.clusters;
14.  this.clusters = no_sorted_clusters.sort((a:any, b:any) =>
b?.percentage-a?.percentage);
15.  this.centroids_idx=result?.clusters.map((val:any)=>val.cluster);
16.  this.imgScikit = ''+ this.response?.graphic;
17.  this.porcentaje=result?.clusters.map((val:any)=>val.percentage);
18.  var
colores=result?.clusters.map((val:any)=>'#'+(0x1000000+(Math.random())*0xff
ffff).toString(16).substr(1,6));
19.  }, (err:any)=>{Swal.close();Swal.fire({icon: 'error',title:
'Aviso...',text: '!Ocurrió un error!',})});
20.  this.mlWekaService.runKmeans(this.form.value).subscribe((result:
any)=>{
21.  this.response_weka = result;
22.  let data no sorted= result?.data.data;
23.  …
24.  this.centroids_idx_weka=result?.clusters.map((val:any)=>val.cluster);
25.  this.porcentaje_weka=result?.clusters.map((val:any)=>val.percentage);
26.  var
colores=result?.clusters.map((val:any)=>'#'+(0x1000000+(Math.random())*0xff
ffff).toString(16).substr(1,6));
27.  }, (err:any)=>{console.error("ERROR: respuesta
inesperada");Swal.close();Swal.fire({icon: 'error',title: 'Aviso...',text:
'!Ocurrió un error WEKA-KMEANS!',})});}}
```

La implementacion de la clase *MLService* se puede apreciar en el siguiente codigo:

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
@Injectable({providedIn: 'root'})
export class MLService {
  readonly URL = 'http://128.199.1.222:5001';
  constructor(private http: HttpClient) { }
  runMetododelCodo(form: any) {return this.http.post(this.URL+
'/MetododelCodo', form);}
  runKmeans(form: any) {
    return this.http.post(this.URL+ '/Clasificacion', form);}}
```

Asimismo la implementacion de la clase *MLWekaService* se expresa según se indica:

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
@Injectable({providedIn: 'root'})
export class MLWekaService {
  readonly URL = 'http://128.199.1.222:8080';
  constructor(private http: HttpClient) { }
  runKmeans(form: any) {return this.http.post(this.URL+ '/kmeansweka',
form);}}
```

C. Funcionalidades del Prototipo. En la presente sección se explicará las funcionalidades del prototipo implementadas a la fecha, las mismas se pueden apreciar en las Figuras 24 y 25.

La primera muestra la pantalla principal, la segunda figura expone un formulario en el cual se debe consignar información requerida por el backend para realizar el proceso del método del codo, clustering o generar conglomerados de los puestos de empleo según la similitud que guardan respecto a las dimensiones o características del dataset; como la casilla: Query: contiene el enunciado de la consulta en formato sql que permitirá generar el dataset en tiempo de ejecución, la casilla Número de Cluster para consignar el # de conglomerados que se desea obtener, la casilla Init para consignar el mecanismo de inicialización de los centroides: {k-means++, random}, la casilla Número Máximo de Iteraciones, aquí se establece el límite de iteraciones que realizara el algoritmo en la búsqueda de los conglomerados minimizando la distancia *intra cluster* y maximizando la distancia *inter cluster*, hasta que los centroides de la iteración *q* sean los mismos a los centroides de la iteración *q-1* o el número de iteraciones sea igual al máximo establecido, concluyendo su procesamiento, evitando entrar a un bucle infinito, incrementando sustancialmente su costo de cómputo.

Figura 24

Pantalla principal del Prototipo

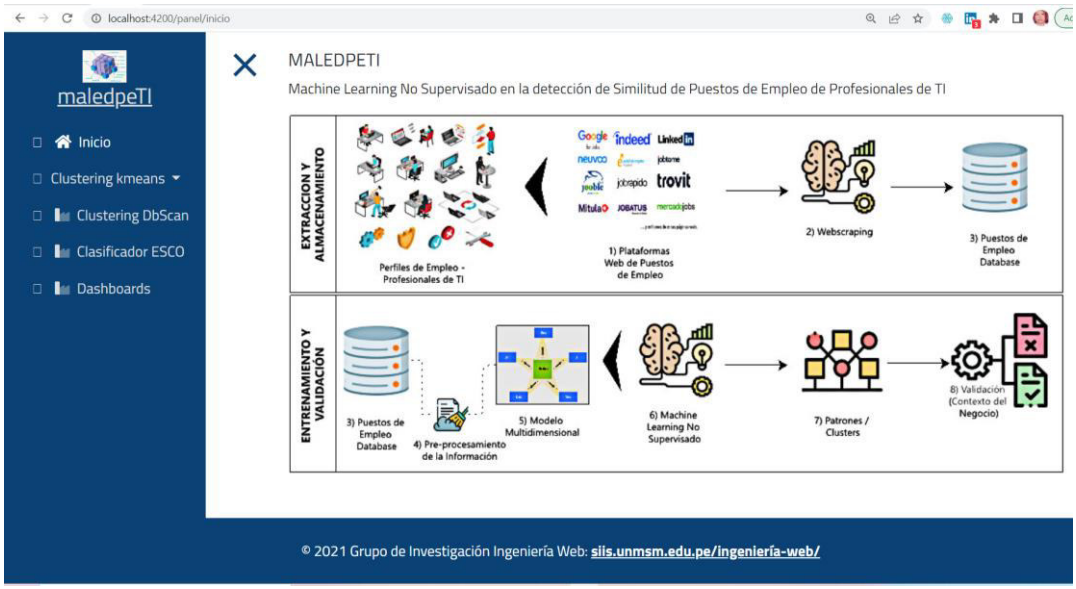
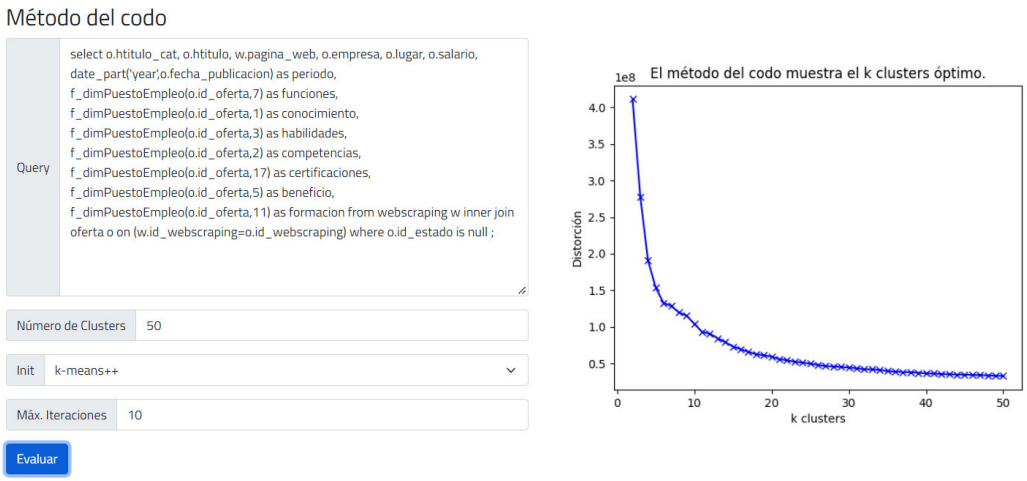


Figura 25

Formulario para procesamiento del método del Codo



3.5.5.3 Clustering DBscan. En la Figura 26 se presenta parte del código fuente correspondiente a la ruta o *endpoint dbscan* que implementa el algoritmo DBScan detallado en la Figura 5, como se puede apreciar en la línea de código 4 se obtiene el *request* suministrado desde una interfaz de usuario en el objeto *body*, en la línea 5 se obtiene el valor del parámetro *épsilon* y se asigna en la variable *eps*, este valor es fundamental para determinar los puntos vecinos basado en la distancia indicada en el parámetro *eps*.

En la línea 6, se obtiene el parámetro *min_samples*, este valor comprende el número de puntos mínimos *(MinPts)* para formar una región densa que representa el cluster o

conglomerado, en la línea 7 se obtiene el dataset desde la base de datos, la función *getPuestosEmpleo()* define la lógica para recuperar la información desde la base de datos, en la línea 8 se formatea la data a una estructura de datos basada en filas y columnas, en la línea 9 se codifica los labels o valores de las características de cada una de las columnas que la conforman, representándolos mediante un índice de la posición que guarda la instancia en el dataset, una muestra de dicha representación se puede apreciar en la Tabla 8 como dataset con etiquetas o labels codificados de 0 a n-1.

Figura 26

Endpoint DBScan – preparacion del dataset

```
1. @app.route("/dbscan", methods = ['GET', 'POST'])
2. def dbscan():
3. if request.method == 'POST':
4. body = request.get_json()
5. eps         = body["dbscan-eps"]
6. min_samples = body['dbscan-min_samples']
7. total data = getPuestosEmpleo()
8. dataset= pd.DataFrame(total_data, columns=['perfil','funciones','conoci
   miento','habilidades','competencias','beneficio']).reset_index(drop=Tru
   e)
9. X = dataset.apply(LabelEncoder().fit_transform).values
10.  sc_x = StandardScaler()
11.  X = sc_x.fit_transform(X)
```

En la línea 12 del código se ejecuta la técnica DBScan pasándole como parámetros *épsilon* y *min_samples*, asimismo se invoca al método *fit_predict(X)* del dataset el resultado se obtiene en el objeto *y_dbscan*, en las líneas 13 al 21 concierne a setear las métricas en el arreglo *result*, en la líneas 23 a 24 se utiliza una estructura de control repetitiva para determinar para cada cluster, el índice del cluster, el número de puntos que la conforma así como el porcentaje que representa la cantidad de puntos del cluster determinado respecto al total de puntos del dataset, finalmente se asigna el arreglo *clusters* al objeto *result* para ser expuesto en formato json por el *endpoint* .

```
12.  y_dbscan = DBSCAN(eps=eps, min_samples=min_samples).fit_predict(X)
13.  dataset['data'] = y_dbscan
14.  n_clusters_     = len(set(y_dbscan)) - (1 if -1 in y_dbscan else 0)
15.  n_noise_        = list(y_dbscan).count(-1)
```

```
16.  silhouette      = metrics.silhouette_score(X, y_dbscan)
17.  result= {}
18.  result["n_clusters"]      = n_clusters_
19.  result["noise_points"]    = n_noise_
20.  result["silhouette"]      = silhouette
21.  result["total_instances"] = len(dataset.index)
22.  clusters = []
23.  for item in range(n_clusters_):
   temporal_cluster = 'Cluster {}'.format(item)
   length_actual_cluster = int(dataset["data"].value_counts()[item])
   decimal_frequency_actual_cluster = float(dataset["data"].value_counts(no
   rmalize=True)[item])
   obj = {
                        "n_cluster": temporal_cluster,
                        "n_puntos": length_actual_cluster,

   "porcentaje(%)": (round(decimal_frequency_actual_cluster*100, 2))
   }
   clusters.append(obj)
24.  result["clusters"] = clusters
```

En la línea 25 se hace uso de una estructura de control repetitiva para elaborar el gráfico basado en dos dimensiones (2D) correspondiente a los perfiles de los puestos de empleo: *eje x* y las funciones: *eje y*, o responsabilidades que corresponden a dicho perfil según la información consignada en los portales de empleo, en la línea 30 se retorna el objeto *result* del endpoint el cual se puede apreciar en la Figura 27, así como el grafico se puede observar en la Figura 28 respectivamente .

```
25.for cluster in range(n_clusters_):
      color = cm.nipy_spectral(float(cluster) / n_clusters_)
      plt.scatter(X[y_dbscan==cluster, 0], X[y_dbscan==cluster, 1], s=25, c=np
      .array([color]),label=f"Cluster {cluster}")
      plt.legend(title='Clusters', loc='upper left', fontsize='xx-small')
26.plt.title('Clusters de Perfiles vs Funciones')
27.plt.xlabel('Perfiles')
28.plt.ylabel('Funciones')
29.plt.show()
30.return  jsonify(result)
```

Figura 27

Clusters DBScan basados en la similitud de Puestos de Empleo y métricas

CLUSTER	CANTIDAD	PORCENTAJE
0	3865	55.31700000000001 %
3	119	1.703 %
1	100	1.431 %
4	65	0.9299999999999999 %
5	62	0.8869999999999999 %
9	62	0.8869999999999999 %
10	35	0.501 %
6	32	0.45799999999999996 %
13	32	0.45799999999999996 %
2	30	0.42900000000000005 %
14	29	0.415 %
7	24	0.34299999999999997 %
12	23	0.329 %
16	23	0.329 %
8	19	0.272 %
11	19	0.272 %
15	11	0.157 %

Métricas

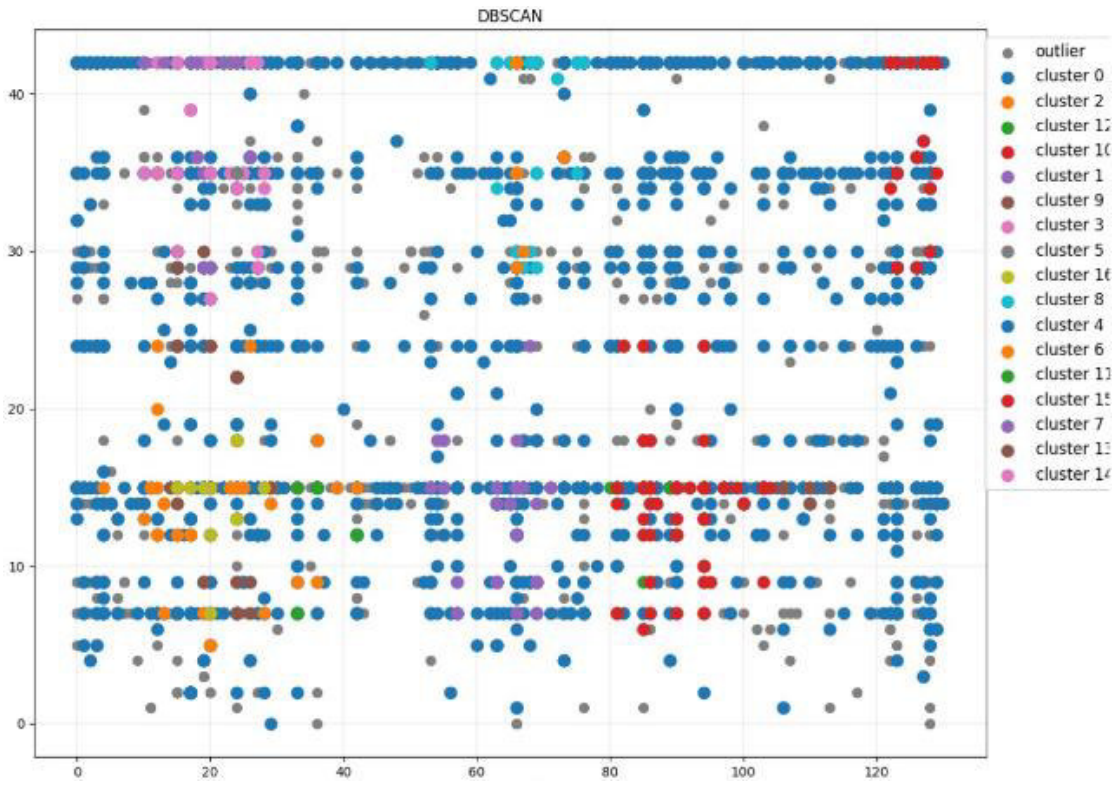
Coeficiente de silueta: -0.40590

Clusters: 17

Outliers: 2437 (34.879%)

Figura 28

Clusters DBScan de Perfiles y Funciones

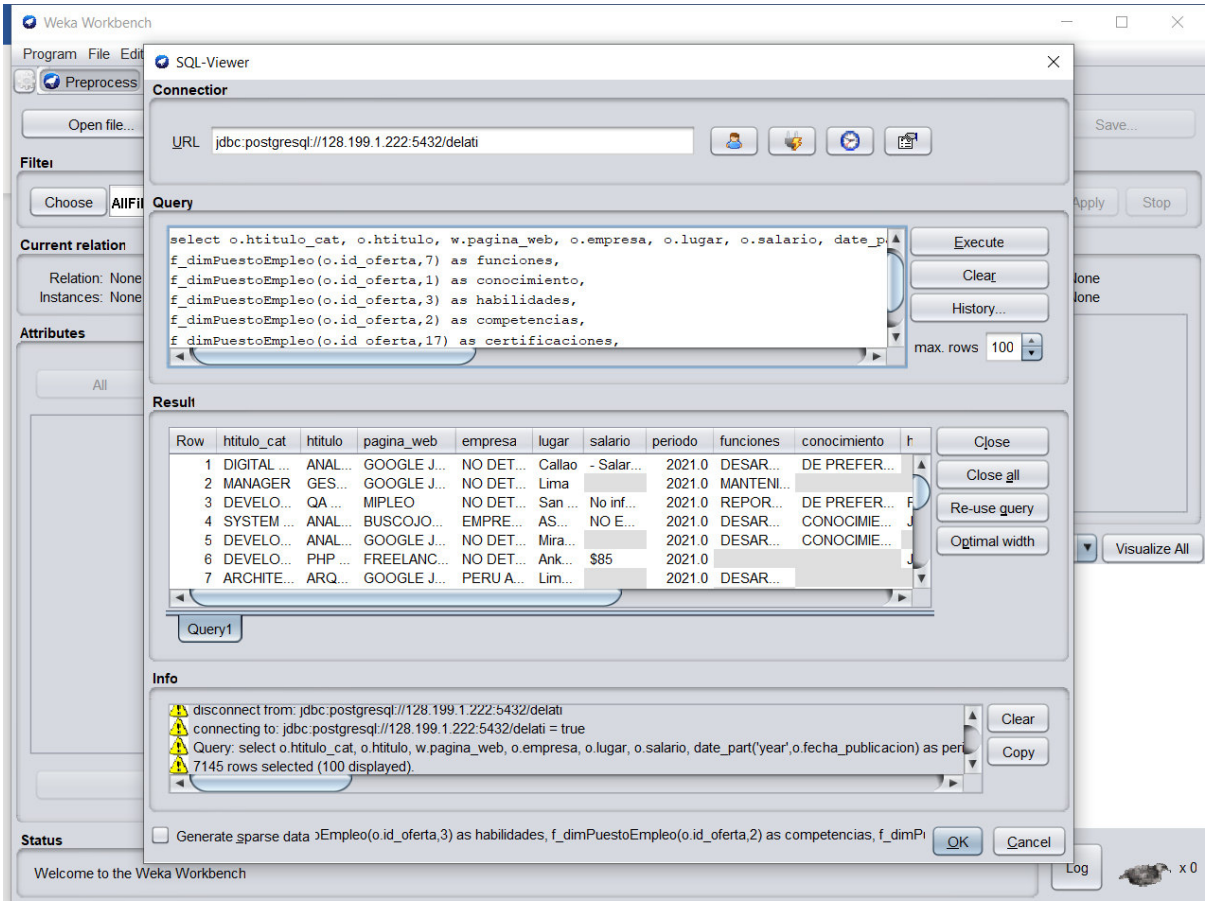


3.5.6 Clustering API Weka

Uno de los problemas formulados en la presente investigación estableció determinar la influencia de la técnica machine learning no supervisada en la clasificación de empleos por similitud de habilidades y/o capacidades de profesionales de Tecnologías de Información. En la presente sección se explicará el uso de la técnica de aprendizaje no supervisado *kmeans* utilizando el software Weka, su aplicación determinará un modelo de aprendizaje basado en conglomerados o clusters según la similitud que guardan las instancias concernientes a los puestos de empleo, para su cálculo utiliza la distancia entre puntos. Para la determinación de los modelos se utilizaron los enunciados de consultas q1 y q2 definidas en la sección funcionalidades del prototipo.

Figura 29

Interfaz Weka – obteniendo el dataset



Nota. Elaboración propia con el uso del software Weka, *Weka 3: Machine Learning Software in Java (versión 3.8.5),* por The University of Waikato, 2021, http://www.cs.waikato.ac.nz/ml/weka.

El algoritmo kmeans que implementa Weka es el propuesto por Arthur y Vassilvitskii (2007), desde la interfaz se dispone de varias funciones para el cálculo de las distancias: Chebyshev, Euclidean, Filtered, Manhattan y Minkowski, la función predeterminada es la Euclidean. Otro aspecto relevante es el método de inicialización: Random, k-means++, Canopy, Farthest first, estos parámetros son requeridos por el algoritmo, así como el número de clusters a determinar y el número máximo de iteraciones. En la Tabla 10 se muestra los parámetros de ejecución del algoritmo en weka, la consulta q1, el n° de clusters a obtener:15, el método de inicialización: k-means++, la función distancia: euclidean, concluida la ejecución se retornó las métricas: n° de instancias del dataset: 7145, n° de iteraciones realizados para obtener los quince conglomerados: 7 y la suma de errores elevado al cuadrado de los intra-cluster, esto se refiere a la suma de todas las distancias de los puntos a sus centroides, de todos los clusters, elevado al cuadrado: 34,696.82.

Tabla 10

Clustering Kmeans con Weka – Parametros

Query	N° Instancias	N° Clusters	Método Inic.	Función Distancia	N° Iter	(suma errores)^2 (Intra-Cluster)
q1	7145	15	k-means++	euclidean	6	34,696.82

En la Tabla 11 se expone los quince clusters determinados, por cada cluster, el número de instancias que la componen y el porcentaje que representa respecto al dataset. Se puede apreciar que el cluster N° 5 con 1,239 instancias equivale a un 17% del dataset es el más

representativo seguido del cluster N° 6 con 1,226 y en la posición #11 el cluster N°11 con 918 instancias representando el 13% del dataset.

Tabla 11

Clustering Kmeans con Weka – Similitud de Puestos de empleo (q1)

N° Cluster	Clustered Instances
0	364 (5%)
1	286 (4%)
2	599 (8%)
3	281 (4%)
4	131 (2%)
5	1239 (17%)
6	1226 (17%)
7	334 (5%)
8	442 (6%)
9	444 (6%)
10	167 (2%)
11	918 (13%)
12	183 (3%)
13	306 (4%)
14	225 (3%)

En la Tabla 12 se aprecia el Cluster N° 5, este expresa que el 17% de los puestos de empleo de profesionales de TI corresponden al año 2020 y perfil FULLSTACK DEVELOPER, publicitado en GOOGLE JOBS, para desempeñarse en una empresa de la Ciudad de Lima-

Perú, no especifica el salario, la función principal que deberá realizar el postulante al empleo es DESARROLLO E IMPLEMENTACION DE PROYECTOS DE SOFTWARE, se requiere que el interesado en la plaza, evidencie CONOCIMIENTO DE BASE DE DATOS, su competencia principal debe ser DESARROLLO EN LENGUAJE DE PROGRAMACION JAVA, la COMUNICACION EFECTIVA es una de las habilidades blandas importantes para el empleo, el postulante debe contar con CERTIFICACION EN SCRUM, la empresa ofrece ESTABILIDAD LABORAL, y como formación para el puesto se requiere EGRESADOS O BACHILLERES DE INGENIERIA DE SISTEMAS, INFORMATICA O AFINES. Análogamente se puede interpretar los otros catorce conglomerados, los cuales se encuentran detallados en el Anexo C.

Tabla 12

Clustering Kmeans con Weka – Cluster N°5 (q1)

Cluster N°5 Instancias: 1239 (17%)
htitulo_cat: DEVELOPER
htitulo: FULLSTACK DEVELOPER
pagina_web: GOOGLE JOBS
empresa: NO DETALLADO
lugar: Lima
salario: NO ESPECIFICADO
periodo: 2020
funciones: DESARROLLO E IMPLEMENTACION DE PROYECTOS DE SOFTWARE
conocimiento: CONOCIMIENTO DE BASE DE DATOS
competencias: JAVA
habilidades: COMUNICACION EFECTIVA

certificaciones: CERTIFICADO EN SCRUM

beneficio: ESTABILIDAD LABORAL

formacion: EGRESADO O BACHILLER DE INGENIERIA DE SISTEMAS, INFORMATICA O AFINES

Aplicando el flujo del Proceso del Aprendizaje No supervisado establecido en la Figura 7, se puede indicar que la Tabla 11 contiene el modelo clustering resultante que incorpora un atributo *clase* al dataset con la etiqueta *"cluster"*; esto permite clasificar nuevos puestos de empleo y predecir a que cluster pertenecen. Para lograr esta finalidad se utilizó un dataset sin clasificar "maledpeti-test" con quince instancias y el algoritmo de clasificación supervisado J48, el resultado se aprecia en la Figura 30, la tasa de error en la predicción es bastante baja y como consecuencia el modelo es bueno, presenta una precisión promedio del 80% mientras que el 20% restante expone un error de precisión entre [0.333, 0.533]

Figura 30

Clasificación J48 (q1)

Number of Leaves : 23,149

Size of the tree : 23,250

Time taken to build model: 0.36 seconds

=== Predictions on test set ===

inst#	**actual**	**predicted**	**error**	**prediction**
1	1:?	1:cluster1		0.978
2	1:?	8:cluster8		0.533
3	1:?	2:cluster2		1
4	1:?	12:cluster12		1

5	1:? 2:cluster2	0.875
6	1:? 9:cluster9	0.572
7	1:? 6:cluster6	0.333
8	1:? 6:cluster6	1
9	1:? 13:cluster13	0.806
10	1:? 8:cluster8	1
11	1:? 8:cluster8	0.533
12	1:? 15:cluster15	0.773
13	1:? 11:cluster11	0.894
14	1:? 8:cluster8	0.939
15	1:? 8:cluster8	0.773

Se aplicó el algoritmo kmeans al dataset resultante del enunciado de consulta q2, este dataset solo considera seis dimensiones y 7,006 instancias, los parámetros establecidos para la ejecución del algoritmo son los indicados en la sección superior de la Tabla 13 como el método de inicialización de Centroides: k-means++, la función distancia es: Euclidean**,** los clusters resultantes se muestran en la sección inferior izquierda de la Tabla referida. El Cluster N° 3 con 2,318 corresponde al 33% es el más representativo y señala que es el perfil DEVELOPER es el más solicitado por los empleadores, el rol principal que desempeñarían los postulantes a plazas con este perfil sería DESARROLLO E IMPLEMENTACION DE PROYECTOS DE SOFTWARE, se requiere contar con CONOCIMIENTO DE BASE DE DATOS, debe evidenciar competencia en el lenguaje de programación JAVA, una de las habilidades blandas solicitadas es la COMUNICACION EFECTIVA, por parte del empleador estos ofrecen: ESTABILIDAD LABORAL.

Las métricas resultantes indican que fueron cuatro iteraciones las generadas por el algoritmo con una suma de error al cuadrado de 16,027.0 lo cual es mucho más bajo respecto al dataset q1 de catorce dimensiones.

Tabla 13

Clustering Kmeans con Weka – Cluster N°3 (q2)

Método de Inicialización de Centroides: k-means++

Función Distancia: Euclidean

Instances: 7006

Number of iterations: 4

sum of squared errors: 16027.0

Clustered Instances		**Cluster N°3 Instancias: 2318 (33%)**
0	420 (6%)	perfil: DEVELOPER
1	1380 (20%)	funciones: DESARROLLO E IMPLEMENTACION DE
2	1682 (24%)	PROYECTOS DE SOFTWARE
3	2318 (33%)	conocimiento: CONOCIMIENTO DE BASE DE DATOS
4	151 (2%)	competencias: JAVA
5	226 (3%)	habilidades: COMUNICACION EFECTIVA
6	194 (3%)	beneficio: ESTABILIDAD LABORAL
7	81 (1%)	
8	51 (1%)	
9	88 (1%)	
10	44 (1%)	
11	53 (1%)	
12	178 (3%)	
13	17 (0%)	

14 123 (2%)	

3.5.7 Evaluación del modelo

Concluida la fase experimental de las técnicas aplicadas en el presente trabajo es preciso señalar que la calidad de los conglomerados mucho depende de la calidad de la información, del número de dimensiones del dataset, a mayor número de dimensiones el algoritmo realiza mayor esfuerzo en determinar su posicionamiento geométrico en el espacio vectorial, realizar el cálculo de las distancias respecto a cada instancia representada por un punto hacia cada centroide para determinar a cual exhibe mayor aproximación, que dependiendo de la función distancia, el método para determinar el centroide en el caso de k-means así como el número de puntos mínimos en el caso de DBScan, pueden requerir altas capacidades de recursos de cómputo para lograr una mejor precisión en los resultados, es por ello que en la Tabla 14 se muestra las métricas obtenidas considerando varios criterios en los parámetros requeridos por cada técnica.

La técnica Kmeans bajo la implementación de la API Scikit-learn en Python determinó una inercia alta considerando la consulta q1 que tiene catorce dimensiones, requiriendo veintitrés iteraciones para determinar los conglomerados; considerando el método de inicialización, este influye en el número de iteraciones y la inercia obteniendo (+34%) con el método de inicialización aleatorio, esto demuestra baja eficiencia respecto al método de inicialización k-means++. Se aprecia también que el número de dimensiones q2 ha influenciado positivamente en el número de iteraciones y el valor de la inercia resultante.

En la experimentación con el algoritmo DBScan no se logró generar conglomerados considerando la consulta q1, básicamente el algoritmo consideró la distribución completa como outliers o ruido; sin embargo al considerar la consulta q2 como se aprecia en la tabla (líneas 5 a 7) los resultados fueron más razonables; la variación en las métricas resultantes quedaron

discriminados por el valor del parámetro *min_samples*, que se refiere al número de puntos mínimos que conforman un área o conglomerado, bajo este criterio, se evaluó con tres valores distintos obteniendo, con veinte número de puntos mínimos, el algoritmo determinó ocho clusters, 1819 puntos outliers o puntos ruido y una silhouette de -0.13, mientras que en el caso de 10 puntos mínimos se obtuvo quince clusters, el ruido se reduce a 678 así como la silhouette a -0.12 y una tercera prueba consideró cinco puntos mínimos, obteniendo 48 clusters, un ruido de 555 puntos y una silhouette de -0.25.

Tabla 14

Resultados de Tecnicas Clustering

N°	Técnica	Api/ Soft.	LP	Query	Instancias	N° dim	N° Clus.	Método Inic.	F.Dist	min_ samples	N° Iter.	Inercia	SE^2 (Intra-Cluster)	noise points	silhouette
1			Python	q1	7,145.00			k-means++			23	202,016,721.31	14,213.26		
2				q1	7,145.00	14		random			25	202,704,820.00	14,237.44		
3				q2	7,006.00			k-means++			14	2,453,152.91	1,566.25		
4	Kmeans	Scikit-learn	Python	q2	7,006.00	6	15	random	euclidean		11	2,454,240.49	1,566.60		
5							8			20				1819	-0.13047
6							15			10				678	-0.12345
7	DBScan	Scikit-learn	Python	q2	7,006.00	6	48			5				555	-0.25108
8		Weka	Java	q1	7,145.00	14		k-means++			6		34,696.82		
9	Kmeans	Weka	Java	q2	7,006.00	6	15	k-means++	euclidean		4		16,027.00		

En la experimentación con el software Weka considerando la consulta q1 se obtuvo la métrica “suma de errores al cuadrado” más alta respecto a la consulta q2, asimismo se aprecia el impacto en el número de iteraciones requerido para conformar los quince conglomerados o clusters; como consecuencia se puede afirmar que el número de dimensiones impacta significativamente en los resultados del algoritmo.

3.6 Análisis de datos

Para el análisis de datos se utilizó la información obtenida mediante la aplicación de la técnica webscraping para la extracción de la información de empleo de profesionales de Tecnologías de Información desde los portales web, el uso de la técnica de machine learning no supervisado como clustering para determinar la similitud de los puestos de empleos a partir de la formación, experiencia, conocimiento, habilidades, capacidades, idiomas, entre otros requisitos contenidos como parte del perfil del empleo solicitado por los empleadores y establecidos en la Tabla 5.

Se definió un modelo multidimensional para aplicar analítica de datos con el apoyo de la herramienta para inteligencia de negocios “Power BI", el detalle se puede apreciar en la sección 3.5.3; Asimismo se hizo uso de técnicas de machine learning no supervisado para clasificar por similitud los puestos de empleo, herramientas como la API científica Scikit-learn y weka fueron utilizadas, el detalle de su aplicación se puede apreciar en las secciones 3.5.5 y 3.5.6 respectivamente.

Complementariamente se utilizó en el análisis de datos las cualificaciones de la OIT, ISCO-08, la CNPO y la MTPE-SP2015 las cuales permitieron brindar respuesta a los problemas de investigación planteados en el presente trabajo. Un resumen de las principales técnicas y herramientas para el análisis de datos utilizado en el presente trabajo se muestra en la Tabla 15.

Tabla 15

Técnicas y Herramientas para Análisis de Datos

N°	Técnica	API/Herramienta	Etapa del Proceso	Detalle
1	Modelo Webscraping	BeautifulSoup	Extracción	sección 3.5.1
2	Preprocesamiento	nltk	Limpieza de Data	sección 3.5.2
3	multidimensional	Power BI	Analítica	sección 3.5.3
4	Kmeans	Scikit-learn	Clustering	sección 3.5.5.2
5	DBScan	Scikit-learn	Clustering	sección 3.5.5.3
6	Kmeans	Weka	Clustering	sección 3.5.6
7	J48	Weka	Clasificador supervisado	sección 3.5.6

IV. Resultados

La presente investigación consideró como objetivo principal proponer un modelo de machine learning no supervisado para la detección de similitud de puestos de empleo de profesionales de Tecnologías de Información.

Con la finalidad de atender el objetivo principal se estableció cuatro objetivos específicos, el primero refiere al diseño de una técnica para extraer los anuncios de empleo dirigidos a profesionales de TI desde los portales laborales; para lo cual en la sección 3.5.1 se detalla el procedimiento seguido, este consistió en identificar los principales portales de trabajo, analizar la estructura DOM de las páginas web, desarrollar un programa en Python que aplique webscraping para la extracción y almacenamiento de las cualificaciones de los perfiles de empleo suministrados por los empleadores o grupos de interés en una base de datos, por lo que consideramos que este primer objetivo fue logrado y permitió aplicar inteligencia de negocios para evaluar diferentes indicadores como se detalla en la sección 3.5.3.

El segundo objetivo se centró en diseñar y aplicar machine learning no supervisado para la detección de similitud de puestos de empleo de profesionales de Tecnologías de Información basado en el posicionamiento geométrico en el espacio vectorial, para su logro se aplicó el proceso planteado en la sección 3.5, las técnicas utilizadas fueron: kmeans y dbscan considerando la API Python Scikit-learn; se desarrolló el prototipo MALEDPETI soportado bajo una Arquitectura basada en microservicios con enfoque de entrega e integración continua en un contexto de ingeniería de software progresiva, definiendo un framework base a nivel backend desarrollado en Python y un frontend bajo angular que permitan incorporar otras técnicas de machine learning tanto supervisadas como no supervisadas, las especificaciones se detallan en la sección 3.5.5. complementariamente se utilizó el software “Weka” para determinar los conglomerados o clusters de los perfiles de empleo en dos escenarios distintos

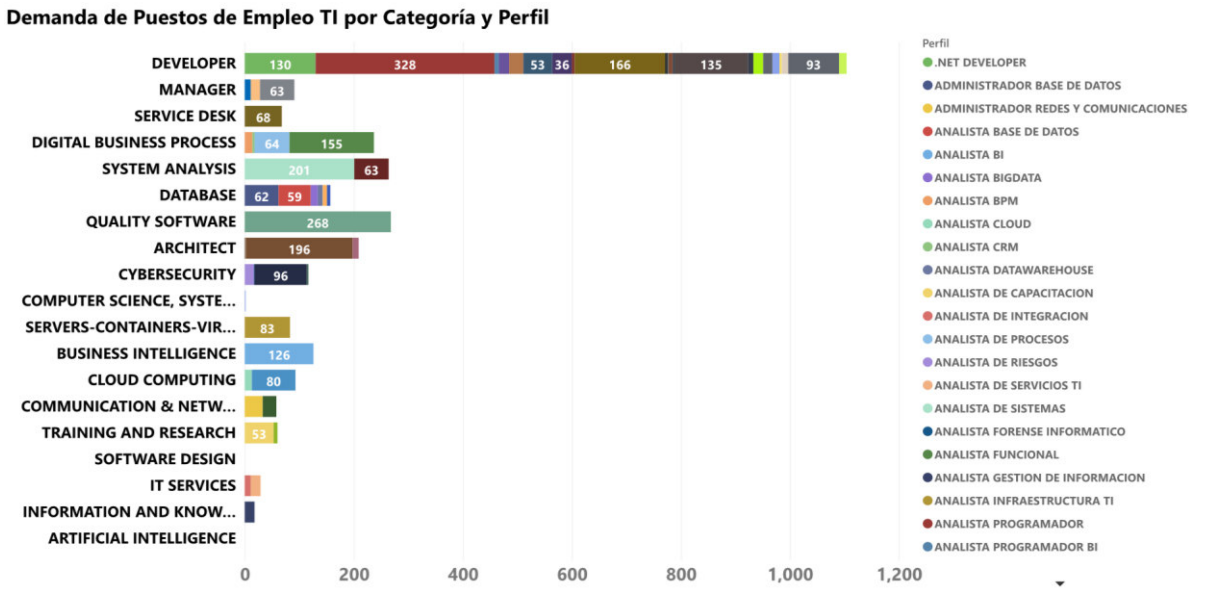
(q1, q2), las métricas resultantes se pueden apreciar en la Tabla 14, la significancia de los resultados se explica a detalle en la sección 3.5.6.

Como tercer objetivo se propuso realizar un análisis respecto a la identificación de puestos de empleo semejantes que permitan contribuir en la uniformidad de puestos de empleo, capacidades y competencias; para este fin se aplicó técnicas de aprendizaje no supervisado para determinar conglomerados o clusters afines como se detalla en la sección 3.5.5 y 3.5.6 complementariamente se aplicó "data analytics" sobre el modelo multidimensional con el apoyo de la herramienta de Inteligencia de negocios "Power BI" obteniéndose los resultados que se a continuación se detalla:

Se identificaron 133 perfiles de TI las cuales fueron agrupadas en 19 categorías como se aprecia en la Figura 31, siendo la Categoría DEVELOPER el grupo que reúne la mayor cantidad de perfiles de TI, en otras posiciones importantes también se encuentran ANALISTAS DE SISTEMAS, CALIDAD DE SOFTWARE, PROCESOS DE NEGOCIOS DIGITALES, Y ARQUITECTOS; concluimos que la mayor demanda laboral recae sobre estas categorías y sus respectivos perfiles.

Figura 31

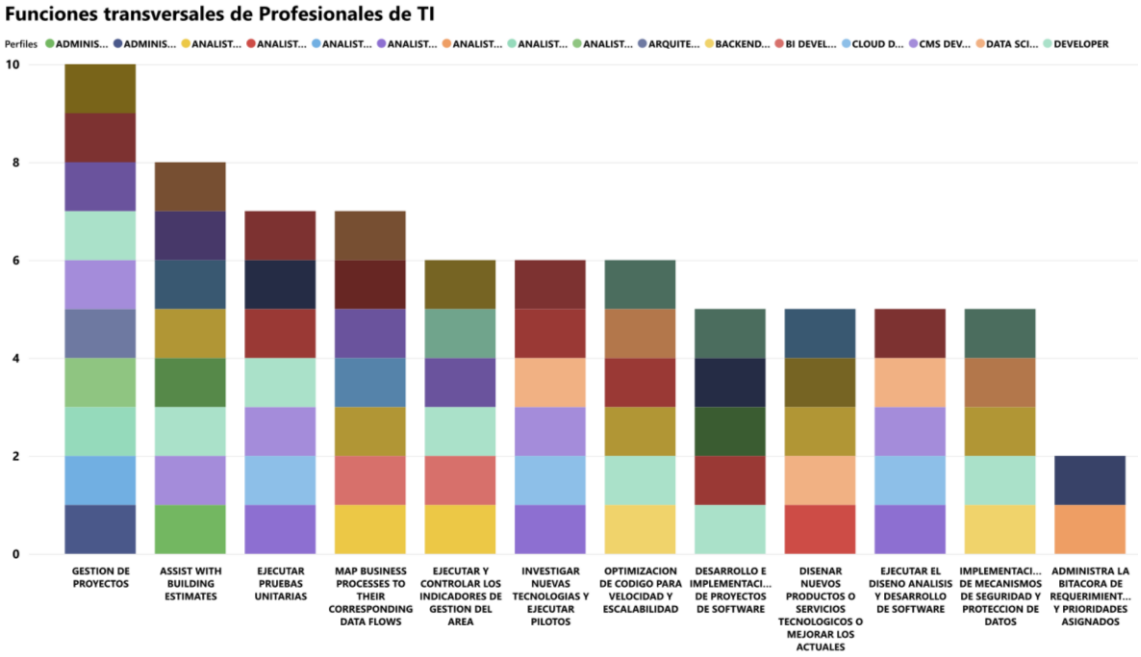
Perfiles de ofertas de empleo Categorizados



Las principales funciones requeridas por los perfiles de empleo se aprecia en la Figura 32, una misma función es requerida por varios perfiles de los puestos de empleo, por ejemplo: la función “GESTION DE PROYECTOS” es una de las labores que desempeñaría los perfiles: “Web Developer”, “System Administrator”, “Gestor de Proyectos”, “Developer”, “Cms Developer”, “Arquitecto de Software”, “Analista de Seguridad Informática”, “Analista Qa”, “Analista Cloud” y “Administrador de Redes y Comunicaciones”, Análogamente la función: “EJECUTAR PRUEBA UNITARIAS” los perfiles afines a DEVELOPER.

Figura 32

Funciones o roles requeridos por varios perfiles

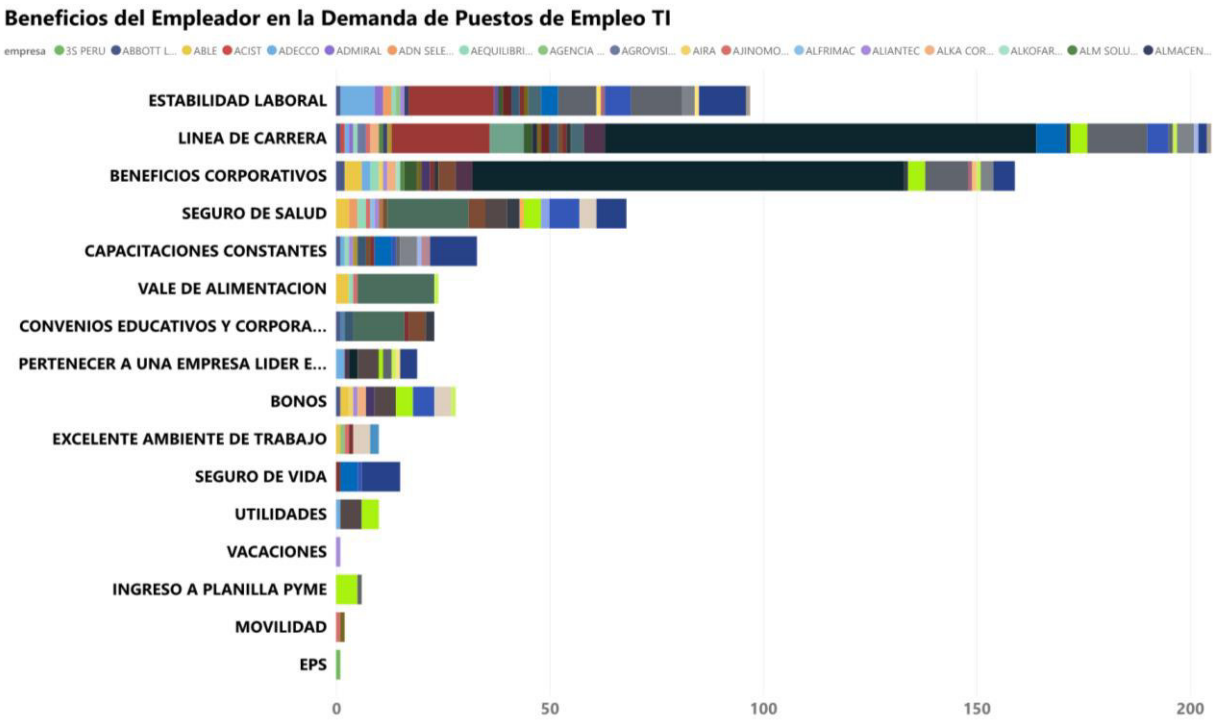


Otra variable importante en una oferta de empleo son los beneficios ofrecidos por las empresas para generar interés en profesionales de TI para que postulen a la plazas, el mayor porcentaje de lo ofrecido corresponde a los derechos de ley según el tipo de la modalidad de plaza ofrecida, de la Figura 33 se puede apreciar que es la “ESTABILIDAD LABORAL” el que mayormente ofrecen, seguido por “LINEA DE CARRERA” en la empresa, “BENEFICIO CORPORATIVOS” así como “SEGURO DE SALUD” y “CAPACITACIONES

CONSTANTES" son las que evidencian mayor oferta, mientras que muy pocas ofrecen "UTILIDADES".

Figura 33

Beneficios ofrecidos por los empleadores



Se ha podido determinar la alta variabilidad de los salarios ofrecidos por las empresas u organizaciones públicas, como se puede observar en la Figura 34, un gran porcentaje de empresas no precisa el salario en la oferta de empleo, estas estilan señalar "no especificado", "salario acorde al mercado laboral", "salario acorde a la responsabilidad", "salario acorde a experiencia y conocimientos", "salario acorde a experiencia", "salario acorde a proyecto", "salario sujeto a evaluación", otras empresas consignaron los salarios ofrecidos, sin embargo para los perfiles Developer y afines los salarios ofertados oscilan en promedio entre <3,000 ; 6,000> soles como se aprecia en la Figura 35.

Figura 34

Salarios ofrecidos por empleadores

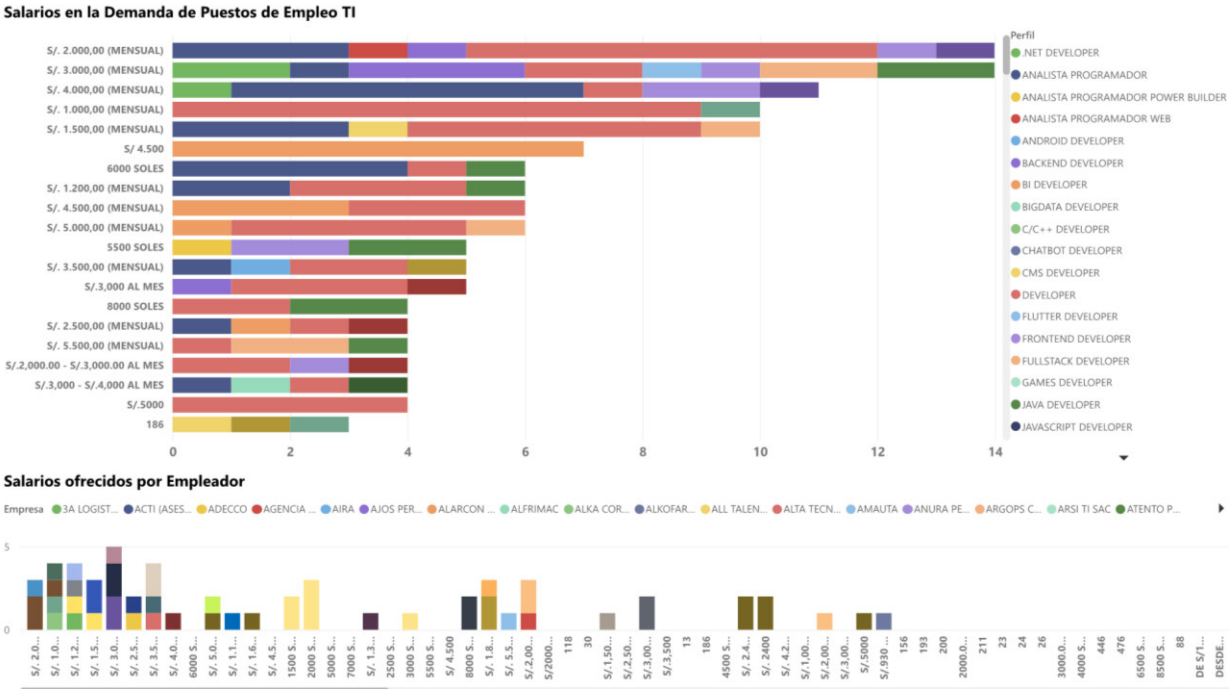
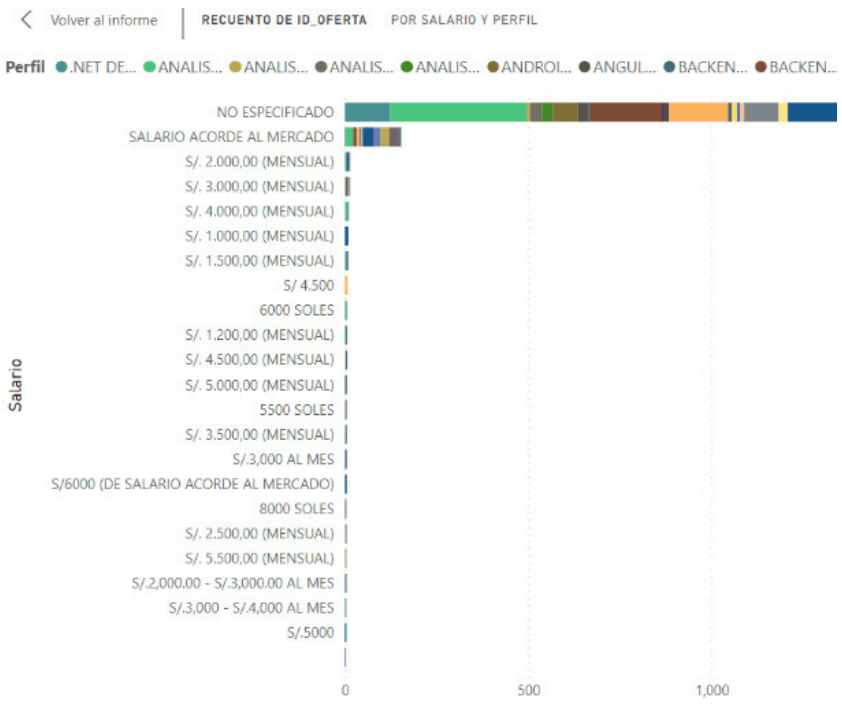


Figura 35

Variabilidad de salario para perfil Developer o afín

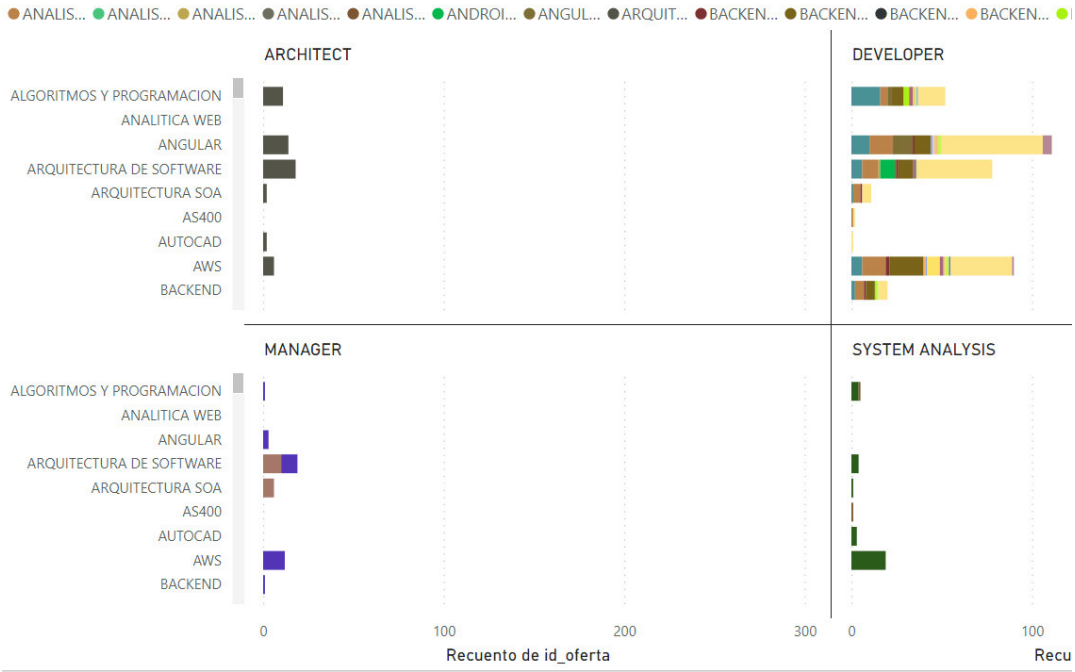


Las competencias que debe evidenciar el postulante a un puesto de empleo de TI se pueden apreciar en la Figura 36, por ejemplo, la competencia técnica “Diseñar la Arquitectura del Software” es más demandante en la categoría DEVELOPER, seguido de ARCHITECT,

MANAGER y con menor énfasis en la categoría SYSTEM ANALYSIS, bajo la misma lógica se puede determinar la competencia “Algoritmos y Programación”, “Aws” entre otros.

Figura 36

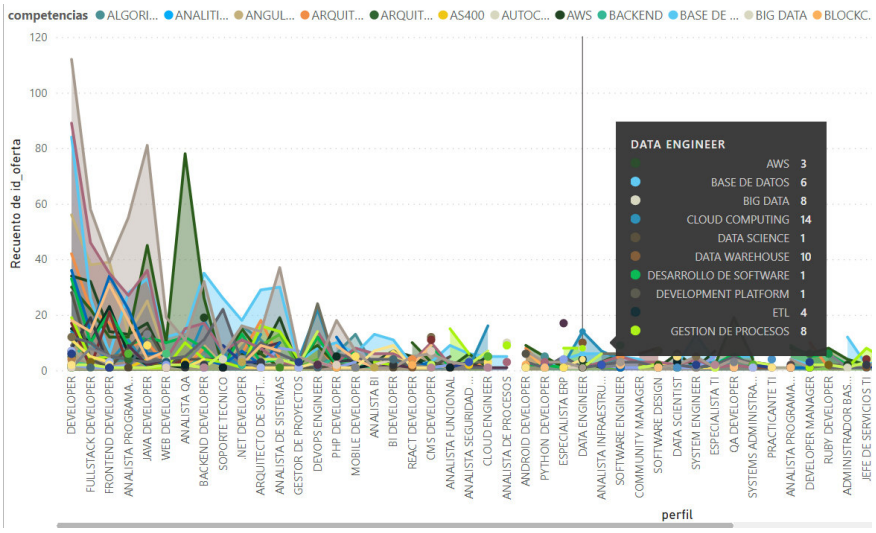
Competencias técnicas por Categorías de perfiles



Asimismo, en la Figura 37 el puesto de empleo del perfil DATA ENGINEER expone las competencias indicadas en el rectángulo de color negro; AWS, BASE DE DATOS, BIG DATA, CLOUD COMPUTING, DATA SCIENCE, DATA WAREHOUSE, DESARROLLO DE SOFTWARE, DEVELOPMENT PLATFORM, ETL y GESTION DE PROCESOS, los números indican el número de puesto de empleo que refiere esa competencia. De manera similar se puede analizar los otros perfiles.

Figura 37

Competencias técnicas por perfiles

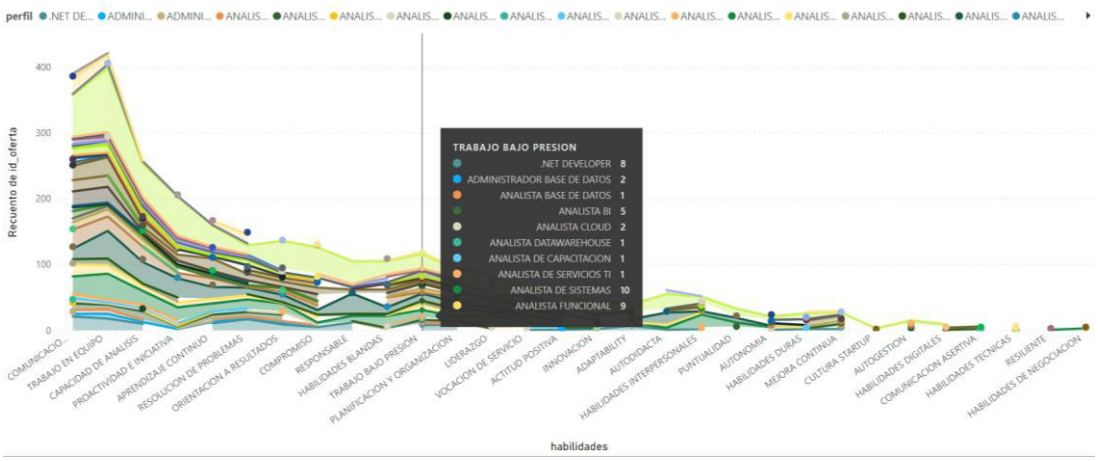


Las habilidades es otro elemento de información de alto valor en un puesto de empleo, las habilidades requeridas se pueden apreciar en la Figura 38, “COMUNICACIÓN EFECTIVA”, “TRABAJO EN EQUIPO” y “CAPACIDAD DE ANALISIS” se encuentran entre las más demandadas.

En el cuadrante negro se puede apreciar “TRABAJO BAJO PRESION”, como una habilidad requerida por varios perfiles de puestos de empleo como: “.Net Developer”, “Administrador de Base De Datos”, “Analista de Base de Datos”, “Analista BI”, “Analista Cloud”, “Analista Datawareohouse”, “Analista de Capacitación”, “Analista de Servicios TI”, “Analista de Sistemas” y “Analista Funcional”.

Figura 38

Habilidades transversales



Finalmente, el cuarto objetivo consistió en determinar el grado de disparidad de los perfiles de puestos de empleo de profesionales de Tecnologías de Información del Perú con la clasificación estándar internacional de empleos de la OIT. Para su determinación se procedió a identificar las cualificaciones ISCO-08 el cual se detalla en la Tabla 2, asimismo se analizó las cualificaciones del CNPO establecidas por el MTPE los cuales se pueden apreciar en la Figura 6 y las ocupaciones de TI del sector privado 2015 informadas al MTPE, se muestra en la Tabla 16. Para cada ocupación se consignó en la columna ISCO-08 el CIUO correspondiente, determinando una coincidencia del 72.42% mientras que el 28.57% de ocupaciones no mantiene correspondencia con ISCO-08.

Asimismo, es importante resaltar que el perfil ANALISTA SISTEMAS INFORMATICOS evidencia como el perfil de TI más demandante en el sector privado del periodo 2015, mientras que el perfil INGENIERO SISTEMAS INFORMATICOS, ocupa la segunda posición y el perfil PROGRAMADOR INFORMATICA/ANALISIS DE SISTEMAS la tercera posición como se aprecia en la Tabla 16.

Tabla 16

Promedio de Ocupaciones TI del Sector Privado 2015

N°	Ocupaciones	Isco-08	Promedio
1	Analista Sistemas Informáticos	2511	9555
2	Ingeniero Sistemas Informáticos		7129
3	Programador Informática/Análisis de Sistemas	2511	2338
4	Operador Equipos Informáticos/Computadoras	3511	2112
5	Técnico Servicios Informáticos para Usuarios	3512	1983
6	Analista Sistemas Informáticos/Computadoras	2511	1770
7	Programador Informática/por Computadora	2512	1380

8	Técnico Sistemas/Excepto Informáticos		1340
9	Técnico Programación por Computadoras	3514	1321
10	Operador Almacenamiento de Datos	3513	1096
11	Técnico control de equipos Informáticos	7422	1057
12	Técnico Análisis Informático	3511	987
13	Analista Sistemas Informáticos/Banco de Datos	2521	806
14	Programador Informática/Análisis de Base de Datos	2521	664
15	Analista Sistemas Informáticos/Telecomunicaciones	2523	629
16	Administrador Banco de Datos	2521	443
17	Operador Computadora	3511	441
18	Ingeniero Aplicaciones de la Informática		332
19	Ingeniero Sistemas/excepto Informáticos		288
20	Programador desarrollo de la Lógica del Proceso		263
21	Analistas transmisiones/Sistemas Informáticos		252
22	Programador codificador de los Programas	2512	227
23	Director de Departamento Informática	1330	171
24	Operador Equipos Informáticos/Unidades Periféricas	3511	160
25	Programador prueba y ejecución del Programa en Computadora		132
26	Profesor Educación Superior/Informática	2356	117
27	Ingeniero de Ordenadores Electrónicos		39
28	Creador Sistemas/Excepto Informáticos		25

Nota. Tomado de Perú. Plataforma Nacional de Datos Abiertos. (2020). *Número de Trabajadores del Sector Privado por Meses 2015, Según CIIU-Clasificación Internacional Industrial Uniforme*. https://www.datosabiertos.gob.pe/dataset

La contrastación de las cualificaciones determinadas en la presente investigación con respecto a ISCO-08 se puede apreciar en la Tabla 17, que el proyecto identifico 133 perfiles, de manera análoga se incorporó una columna ISCO-08 para consignar el CIUO, una segunda columna para establecer el identificador de las cualificaciones del sector privado del periodo 2015 MTPE-SP2015 y una tercera columna para consignar el identificador del CNPO; en el Anexo E se puede apreciar las cualificaciones completas de la presente investigación.

Tabla 17

Perfiles TI del Sector Público y Privado 2020-2021

N°	Perfil	ISCO-08	CNPO	MTPE SP2015	N° PEmpleo
1	Developer	2512	J2662	7	1180
2	Web Developer	2513	J2662		764
3	Analista Programador	2519	J2662	3	450
4	Soporte Técnico	2523		17	443
5	Frontend Developer	2513	J2662		399
6	Analista Qa			25	391
7	Fullstack Developer	2512	J2662	7	366
8	Java Developer	2512	J2662	7	360
9	Gestor de Proyectos				324
10	Arquitecto de Software				306
11	Php Developer	2513	J2662		257
12	Analista de Sistemas	2511		1	248
13	Cms Developer		J2662		242
14	Backend Developer	2513	J2662		233

15	Mobile Developer		J2662		218
16	Analista Funcional				209
17	.Net Developer	2513	J2662		205
18	Analista BI				187
19	BI Developer		J2662		184
20	React Developer	2513	J2662		161
21	Community Manager				149
22	Devops Engineer				147
23	Python Developer	2513	J2662		138
24	Android Developer		J2662		136
25	Especialista ERP				132
26	Analista Seguridad Informática				128
27	Data Engineer				123
28	Software Design				122
29	Cloud Engineer				117
30	Practicante TI				109
31	Gerente de TI			23	104
32	Analista Infraestructura TI	2522			102
33	System Engineer			2	102
34	Administrador Base de Datos	2521		16	99
35	Analista de Capacitación	2356		26	94
36	Systems Administrator	2522			94
37	Analista TI				93
38	Analista de Procesos				87
39	Analista Service Desk				87

40	Analista Base de Datos		13	85
41	Web Services Developer	J2662		80
42	Especialista Ecommerce			77
43	Software Engineer		1	76
44	Data Scientist			75
45	Especialista Ciberseguridad			75
46	Qa Developer	J2662	25	68
133	Research Expert			1

V. Discusión de resultados

a) En el trabajo de Chuan et al. (2018) se propuso un modelo semántico para mejorar la adecuación persona-trabajo para el reclutamiento de talentos en línea, para lo cual el autor establece una representación semántica de los anuncios de empleo y las hojas de vida de los candidatos, en la parte experimental utiliza dataset de una compañía tecnológica de China y varias técnicas de machine learning supervisado como Regresión logística, Árbol de decisión, Adaboost, Bosques aleatorios y Gradient Boosting Decision Tree, para evaluar la precisión y eficiencia de los resultados; sin embargo nuestro trabajo se orienta al descubrimiento de patrones en la información de los anuncios de empleo utilizando machine learning no supervisado, específicamente se utilizó los algoritmos kmeans y dbscan.

b) La investigación de Boselli et al. (2018) se centra en la clasificación de ofertas de empleo en línea a través del aprendizaje automático supervisado, su contribución se delimita en la extracción de los anuncios de empleo de los portales web, aplica web scraping, el dataset es entrenado por expertos del dominio consignando los clasificadores ISCO para los perfiles y genera modelos de machine learning con las técnicas de Maquina de Soporte Vectorial (SVM) Linear, SVM RBF Kernel, Bosques aleatorios y Redes Neuronales, concluyendo que se obtuvo la mejor precisión con SVM Linear. Para la extracción de habilidades desde las ofertas de empleo utiliza el clasificador de texto n-gram, se depura los n-grams con baja significancia, participan expertos del dominio para establecer la clasificación de habilidades de ESCO; mientras que la presente investigación se enfoca en machine learning no supervisado, uso de técnicas clustering como kmeans y dbscan para el descubrimiento de conglomerados por similitud de puntos en un espacio de más de dos dimensiones representados por los puestos de empleo publicados en los portales web.

c) La propuesta de Lynch (2017) se centra en resolver un problema organizacional de recursos humanos quienes determinan de manera subjetiva los perfiles de empleo, salario, nivel y responsabilidad de los empleados, basándose en el detalle del puesto, generándose sesgos e inconsistencias, es así que su investigación se enfoca en el análisis de la predictibilidad de los títulos de los puestos de empleo a partir del detalle del puesto, obtiene de una página web los puestos de empleo, aplica varias transformaciones con el uso de Lenguaje de procesamiento natural (LPN) obteniendo un dataset de palabras claves determinadas a partir de las frecuencias de los términos en la información, al modelo resultante le aplica técnicas de machine learning supervisado como Maquina de soporte vectorial y Bosques aleatorios para predecir los treinta puestos de empleo más frecuentes. Los resultados no fueron muy alentadores, expresaron baja precisión, atribuyéndolos a la alta dimensionalidad de la información y a la complejidad de la técnica LPN utilizada; sin embargo, sus resultados pueden constituirse en una base para diseñar modelos futuros de machine learning. La similitud con el presente trabajo estaría básicamente en el enfoque de utilizar la información de las ofertas de empleo desde las páginas web, más las técnicas de machine learning utilizadas son supervisadas y el contexto del negocio se delimita al reclutamiento de personal en la organización.

d) La investigación de Marrara et al. (2017) propone un enfoque de reconocimiento de ocupaciones sobre la taxonomía ISCO basado en el modelo lingüístico, el enfoque describe una posible mejora del proyecto WoLMIS. La evaluación experimental demostró el potencial del enfoque para identificar posibles nuevas profesiones a partir de las ofertas de trabajo analizadas. Contrastando con nuestra propuesta, las similitudes se centran en que ambas propuestas utilizan la información de los puestos de empleo de los portales de empleo, consideran la clasificación ISCO, aunque el dominio es distinto, así como la técnica para implementar su enfoque propuesto.

e) El enfoque no supervisado para generar fragmentos estructurados informativos para motores de búsqueda de empleo propuesto en la investigación de Spirin y Karahalios (2013) consiste en generar datasets entrenados automáticamente a partir de una colección de ofertas de trabajo, describe un algoritmo de aprendizaje automático que consuma el dataset entrenado, genere un modelo para generar fragmentos de información basados en requisitos y responsabilidades de las ofertas de empleo, el dataset entrenado debe contener instancias marcadas como positivas, las secciones que contengan información de valor de responsabilidades y requisitos de las ofertas de trabajo así como instancias irrelevantes las cuales serán marcadas como negativas. En la parte experimental utiliza la técnica de Maquina de soporte vectorial con un kernel lineal, así como la técnica basada en n-grams de la lingüística computacional. Según la síntesis señalada difiere de nuestra propuesta toda vez que nuestro alcance no solo se circunscribe a la extracción automatizada de las ofertas de empleo y su clasificación en componentes simples que la conforman.

f) En el trabajo de Vinel et al. (2019) sobre la comparación experimental de enfoques no supervisados para descubrir especializaciones de las profesiones que se ubican en el cuerpo de las vacantes laborales, evalúa experimentalmente varios métodos estadísticos de representaciones de vectores de texto: TF-IDF, modelado probabilístico de temas (ARTM), modelos de lenguaje neuronal basados en semántica distribucional (word2vec, fasttext) y representación profunda de palabras contextualizadas (ELMo y BERT multilingüe), utiliza dataset de puestos de empleo en ruso y métodos de clustering como K-means, propagación por afinidad, birch, agrupación aglomerativa y hdbscan; concluye que la mejor solución fue K-means con ARTM siempre que se señale el número de clusters a obtener con antelación, caso contrario word2vec resulta mejor; las métricas utilizadas para evaluar la calidad del agrupamiento son: Mutua Ajustada (AMI), Índice Rand Ajustado (ARI) y Vmeasure. Si bien la propuesta de los autores utiliza dos de las técnicas no supervisadas

utilizadas en nuestra investigación, sin embargo, en la contextualización del dataset no utilizan modelos dimensionales como si lo hace nuestra investigación, por otro lado, su propuesta se delimita a identificar especializaciones en el cuerpo del puesto de empleo mientras que nuestro trabajo aborda catorce dimensiones como funciones o roles, competencias, habilidades entre otros.

VI. Conclusiones

- En esta investigación se propuso un modelo de machine learning no supervisado para la detección de similitud de puestos de empleo de profesionales de TI basado en la posición geométrica de las instancias en el espacio vectorial y determinadas por similitud de puntos; para lo cual nuestro enfoque aplicó una perspectiva metodológica con resultados de valor en cada una de ellas.
- Se aplicó técnicas de extracción de información de los puestos de empleo publicados en los principales portales laborales, para este fin se desarrolló una aplicación en python y la estrategia se centró en la fragmentación del detalle del perfil del puesto de empleo en tuplas individualizadas, tomando como referencias las etiquetas del documento DOM de la página web y su registro en un esquema de base de datos diseñado para esta finalidad.
- Se realizó el preprocesamiento de la información extraída, utilizando técnicas de minería de textos.
- Se diseñó un modelo basado en dimensiones, una dimensión por cada variable o elemento de información que conforma la estructura de un perfil de empleo.
- Se experimentó con los algoritmos de clustering kmeans y dbscan en diferentes áreas de trabajo como Google Colab, Weka para evaluar su potencia y capacidad en el proceso de agrupamiento de datasets no entrenados.
- Se implementó un Prototipo bajo arquitectura basado en servicios, este prototipo se desacopla en dos principales capas: El backend contiene la lógica que invoca APIs de machine learning como scipy en Python y Weka en Java; la implementación del backend permitió exponer los resultados a razón de clusters, métricas y graficas producto de la ejecución de los algoritmos sobre el dataset formado en función de dimensiones con alto nivel de granularidad.

- Se evaluó los modelos tomando como referencia las métricas de cada algoritmo, estableciendo resultados alentadores considerando la complejidad del dataset, el número de dimensiones, los datos categóricos de alta longitud y altas capacidades de recursos computacionales requeridos para el logro de una mejor precisión en los resultados.
- Se sometió el modelo a un nuevo dataset haciendo uso del algoritmo supervisado J48 obteniendo una precisión del 80% en la clasificación de los puestos de empleo del nuevo dataset.
- Se contrastó los 133 perfiles laborales y 19 categorías determinados con los de ISCO-08 y CNPO apreciando un amplio desfase en el rubro TI y la inexistencia de un catálogo de cualificaciones uniforme a nivel del estado peruano, recientemente en Julio 2021 se aprobó por decreto supremo el MNCP con proyección a ser poblado progresivamente con las cualificaciones de los perfiles de empleo.
- Se aplicó "data analitycs" sobre el modelo dimensional con el apoyo de la herramienta para inteligencia de negocios "Power BI", obteniendo una serie de perspectivas de la realidad de los puestos de empleo de TI en los dos últimos años entre los que se resaltan: Dashboards con principales indicadores de puestos de empleo TI, funciones transversales de los perfiles de puestos de empleo, demanda de competencias, demanda de habilidades, demanda de funciones, beneficios del empleador, salarios en la demanda de puestos de empleo, mapa de calor de la demanda de puestos de empleo TI en Perú, estos resultados permitirían a la academia a actualizar los currículos en aras de reducir la brecha entre la demanda social y la pertinencia del perfil del egresado para el logro de un desempeño profesional competente en beneficio del desarrollo nacional.

VII. Recomendaciones

- Si bien la presente investigación contribuye con un enfoque basado en las técnicas de machine learning no supervisadas como clustering kmeans y dbscan para determinar la similitud de los puestos de empleo, basado en su posicionamiento geométrico en el espacio vectorial, logrando resultados alentadores, sin embargo, se hace necesario continuar investigando otras técnicas basadas en la lingüística computacional, la similitud semántica mutilenguaje entre otros métodos del campo de la inteligencia artificial que permitan mejorar los resultados obtenidos, asimismo involucrar otras dimensiones como: Experiencia, Certificación, Capacitación, Idiomas, Posgrado, aperturándose nuevas líneas de investigación.
- Nuestro enfoque puede ser replicado a perfiles de empleo de otras áreas disciplinarias, con la finalidad de identificar su real demanda social y sobre esta base actualizar los currículos de la academia, crear nuevos programas educativos de nivel pregrado y posgrado, así como lograr la pertinencia del perfil de egreso y como consecuencia reducir la brecha entre la oferta y la demanda de cualificaciones para atender los constantes desafíos enmarcados en el desarrollo nacional.
- Los organismos involucrados en las políticas laborales y educativas pueden considerar nuestra propuesta, implementándolo en aras de sincerar el Catálogo Nacional de Perfiles Ocupacionales (CNPO) con la real demanda social y agilizar el poblamiento del Marco Nacional de Cualificaciones del Perú (MNCP), manteniendo una actualización sostenible, producto de los vertiginosos cambios en la economía mundial.

VIII. Referencias

Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A. y Aljaaf, A. (2020). A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science. *Supervised and Unsupervised Learning for Data Science (USA)*, 3-21. https://doi.org/10.1007/978-3-030-22475-2

Arthur, D. y Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 1027-1035. https://dl.acm.org/doi/10.5555/1283383.1283494

Barrientos, E. (2013). *Investigacion Educativa*. Universidad Nacional Mayor de San Marcos.

Boselli, R., Cesarini, M., Mercorio, F. y Mezzanzanica, M. (2018). Classifying online job advertisements through machine learning. *Future Generation Computer Systems,* (86), 319-328. https://doi.org/10.1016/j.future.2018.03.035

Buscador Google. (15 de enero de 2021). *Panel de Elementos, Pagina de Inspeccion del resultado de una busqueda de convocatoria de trabajo en Google*. https://www.google.com/search

Chuan, Q., Hengshu, Z., Tong, X., Chen, Z., Liang, J., Enhong, C. y Hui, X. (2018). Enhancing Person-Job Fit for Talent Recruitment: An Ability-aware Neural Network Approach. *SIGIR'18,* 25-34. https://doi.org/10.1145/3209978.3210025

Comision Europea. European Skills, Competences, Qualifications and Occupations (ESCO). (15 de enero de 2021). C*lasificación europea de capacidades/competencias, cualificaciones y ocupaciones*. https://esco.ec.europa.eu/es/node/4

Davies J. (05 de Enero de 2021). *Word Cloud Generator [Software]*. https://www.jasondavies.com/wordcloud/

Decreto Supremo N° 012-2021-MINEDU. Decreto Supremo que crea el Marco Nacional de Cualificaciones del Perú - MNCP y la comisión multisectorial de naturaleza permanente denominada"Comisión Nacional para el seguimiento a la implementación del Marco Nacional de Cualificaciones del Perú - MNCP". (09 de Julio de 2021), https://www.gob.pe/institucion/minedu/normas-legales/2138266-012-2021-minedu

Deshpande, M. (2018). *Machine Learning for Human Beings, Build Machine Learning Algorithms with Python.* Zenva. https://pythonmachinelearning.pro

Ester, M., Kriegel, H., Sander, J. y Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining,* 226-231. https://dl.acm.org/doi/10.5555/3001460.3001507

Ezugwu, A., Shukla, A., Agbaje, M., Oyelade, O., José-García, A. y Agushaka, J. (2021). Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature. *Neural Computing and Applications,* (33), 6247–6306. https://doi.org/10.1007/s00521-020-05395-4

Flask. (15 de enero de 2021). *Flask web development, one drop at a time*. https://flask.palletsprojects.com/en/2.0.x/

Halkidi, M., Batistakis, Y. y Vazirgiannis, M. (2001). On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, (17), 107-145. https://doi.org/10.1023/A:1012801612483

Hernández, R., Fernández, C. y Baptista, M. (2014). *Metodología de la Investigación* (6ª ed.). McGraw-Hill.

Kimball, R. y Ross, M. (2002). *The data warehouse toolkit : the complete guide to dimensional modeling* (2ª ed.). Wiley.

King, R. (2015). *Cluster Analysis and Data Mining, An Introduction* (1ª ed.). Mercury Learning and Information.

Leavitt, J. y Whisler, T. (1958). Management in the 1980's. *Harvard Business Review, 36*, 41-48. https://hbr.org/1958/11/management-in-the-1980s

Lynch, J. (2017). *An Analysis of Predicting Job Titles Using Job Descriptions.* [Tesis de Maestría, Dublin Institute of Technology]. Repositorio Institucional de Disertaciones. https://arrow.dit.ie/scschcomdis

Mamani, Z., Del Pino, L. y Gonzales, J. (2020). Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua. *Industrial Data, 23*(2), 141–149. https://doi.org/10.15381/idata.v23i2.17278

Mansourvar, M. y Yasin, N. (2010). Web portal As A Knowledge Management System In the Universities. *International Journal of Information and Communication Engineering*, 10(4). https://publications.waset.org/5677/web-portal-as-a-knowledge-management-system-in-the-universities

Marrara, S., Pasi, G., Viviani, M., Cesarini, M. y Mercorio, F. (2017). A language modelling approach for discovering novel labour market occupations from the web. *WI '17: Proceedings of the International Conference on Web Intelligence,* 1026–1034. https://doi.org/10.1145/3106426.3109035

Mayo, M. (15 de enero de 2020). *text_data_preprocessing_5.* https://gist.github.com/mmmayo13

Microsoft (20 de julio de 2021). *Microsoft Power BI Desktop*. (versión 2.109.782.0) https://powerbi.microsoft.com/

Mozilla (27 de enero de 2021). *Introducción al DOM*. https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

Organización de las Naciones Unidas [ONU]. (20 de julio de 2019). *Objetivos de Desarrollo Sostenible*. https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/

Organizacion Internacional del Trabajo [OIT] ILOSTAT. (15 de agosto de 2020). *Clasificación Internacional Uniforme de Ocupaciones (CIUO).* https://ilostat.ilo.org/es/resources/concepts-and-definitions/classification-occupation/

Organización Internacional del Trabajo [OIT]. (05 de marzo de 2019). *Misión e impacto de la OIT*. https://www.ilo.org/

Organizacion Internacional del Trabajo [OIT]. (15 de marzo de 2019). *Estructura de la CIUO-08 y concordancias previas con la CIUO-88.* https://www.ilo.org/public/spanish/bureau/stat/isco/isco08/index.htm

Organización para la Cooperación y el Desarrollo Económicos [OECDE]. (2016). *Perspectivas de la OCDE en Ciencia, Tecnología e Innovación en América Latina 2016.* http://dx.doi.org/10.1787/sti_in_outlook-2016-en

Perez, L. (2014). *Tecnicas de mineria de datos e inteligencia de negocios IBM SPSS Modeler.* Ibergarceta Publicaciones.

Perú. Ministerio de Educación [MINEDU]. (2015). *Política de Aseguramiento de la Calidad de la Educación Superior Universitaria.* http://www.minedu.gob.pe/reforma-universitaria/pdf/politica.pdf

Perú. Ministerio de Trabajo y Promoción del Empleo [MTPE]. (2014). *Catalogo Nacional de Perfiles Ocupacionales (Cualificaciones).* https://www2.trabajo.gob.pe

Perú. Ministerio de Trabajo y Promoción del Empleo [MTPE]. (2020). *Número de Trabajadores del Sector Privado por Meses 2015, Según CIIU-Clasificación Internacional Industrial Uniforme*. https://www.datosabiertos.gob.pe/dataset

Perú. Presidencia del Consejo de Ministros [PCM]. (2017). *Modelo y Estrategia de Datos Abiertos Gubernamentales del Perú.* https://sgp.pcm.gob.pe/wp-content/uploads/2017/05/Modelo-y-Estrategia-DAG-del-Peru.pdf

Peru. Programa para la mejora de la calidad y pertinencia de los servicios de educacion superior universitaria y tecnologica a nivel Nacional [PMESUT]. (2021). *Marco Nacional de Cualificaciones del Perú (MNCP).* https://www.pmesut.gob.pe/mncp

Perú. Sistema Nacional de Evaluación, Acreditación y Certificación de la Calidad Educativa-[SINEACE]. (2017). *Modelo de Acreditacion para Programas de Estudios de Educacion Superior Universitaria.* https://www.gob.pe/institucion/sineace/informes-publicaciones/914029-modelo-de-acreditacion-para-programas-de-estudios-de-educacion-superior-universitaria

Pranav, N., Archana, S., Madhav, C. y Sunil, B. (2018). Empirical Analysis of Data Clustering Algorithms. *Procedia Computer Science*, (125), 770–779. https://doi.org/10.1016/j.procs.2017.12.099

Python. (15 de agosto de 2019). *Python*. https://www.python.org/

Ramadas M. y Abraham A. (2018). Metaheuristics and Data Clustering. In: Metaheuristics for Data Clustering and Image Segmentation. *Intelligent Systems Reference Library,* (152), 7–55. https://doi.org/10.1007/978-3-030-04097-0_2

Scikit-learn. (15 de julio de 2020). *Machine Learning in Python*. https://scikit-learn.org/stable/

SDSN Australia/Pacific. (2017). *Como empezar con los ODS en las Universidades, Una Guia para las Universidades, Los centros de Educacion Superior y el Sector Académico.* https://reds-sdsn.es/wp-content/uploads/2017/02/Guia-ODS-Universidades-1800301-WEB.pdf

Sierra, B. (2006). *Aprendizaje Automático conceptos basicos y avanzados.* Pearson Prentice Hall.

Spirin, N. y Karahalios, K. (2013). Unsupervised Approach to Generate Informative Structured Snippets for Job Search Engines. *WWW '13 Companion: Proceedings of the 22nd International Conference on World Wide,* 203-204. Rio de Janeiro, Brazil: ACM. https://doi.org/10.1145/2487788.2487891

Swamynathan, M. (2017). *Mastering Machine Learning with Python in Six Steps, A Practical Implementation Guide to Predictive Data Analytics Using Python.* Apress https://doi.org/10.1007/978-1-4842-2866-1

Task Group on Information Technology Curricula. (2017). *Information Technology Curricula 2017 IT2017 Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology.* Association for Computing Machinery. https://doi.org/10.1145/3173161

The University of Waikato. (06 de enero 2020). *Weka 3: Machine Learning Software in Java* (versión 3.8.5). http://www.cs.waikato.ac.nz/ml/weka/

Vinel, M., Ryazanov, I., Botov, D. y Nikolaev, I. (20-22 de noviembre de 2019). Experimental Comparison of Unsupervised Approaches in the Task of Separating Specializations Within Professions in Job Vacancies. *8th Conference, AINL 2019,* Tartu, Estonia*, Proceedings,* 99–112. https://doi.org/10.1007/978-3-030-34518-1

IX. Anexos

Anexo A.

Resultado experimental extendido, Clustering Kmeans con Scikit-learn (q1).

Inercia: 74,144,244.898
Número de iteraciones : 10.00
Total de instancias : 7129

#	Cluster	Especificación del Cluster	# Inst.	%
1	Cluster 8	Developer | Fullstack Developer | Google Jobs | No Detallado | Peru - Lima | No Especificado | 2021 | Desarrollo e Implementacion de Proyectos de Software | Conocimientos de Php Javascript Html Java Css Net | Mysql | - | - | Linea de Carrera | - | 8 |	987	13.84%
2	Cluster 13	Developer | Analista Programador | Linkedin | Notaria del Villar | Peru - Lima | No Especificado | 2021 | Desarrollo e Implementacion de Proyectos de Software | Conocimientos de Php Javascript Html Java Css Net | Php | Responsable | - | - | - | 13 |	850	11.92%
3	Cluster 6	Database | Data Scientist | Google Jobs | bbva en Peru | No Definido | No Especificado | 2021 | Gestion de Proyectos | Conocimiento en Gestion de Proyectos | - | - | - | Linea de Carrera | Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines | 6 |	840	11.78%
4	Cluster 12	Database | Data Engineer | Google Jobs | Indra Peru | No Definido | No Especificado | 2021 | - | Conocimiento en Metodologias Agiles | Modelamiento de Base De Datos | - | - | Estabilidad Laboral | Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines | 12 |	796	11.17%
5	Cluster 4	Developer | Developer | Buscojobs | Ssays Sac | Peru - Lima | No Especificado | 2021 | Desarrollo e Implementacion de Proyectos de Software | - | Microservicios | - | - | Estabilidad Laboral | - | 4 |	670	9.40%
6	Cluster 5	Developer | Developer | Mipleo | Enterprise Solutions Development Sac | Peru - Lima | No Especificado | 2021 | Mantenimiento y Soporte de Aplicativos | - | Microservicios | - | - | - | - | 5 |	524	7.35%
7	Cluster 1	Developer | Developer | Mipleo | No Detallado | Lima - Miraflores | No Especificado | 2021 | Desarrollo e Implementacion de Proyectos de Software | - | React | - | - | Estabilidad Laboral | Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines | 1 |	461	6.47%
8	Cluster 14	Developer | Developer | Mipleo | Vg All Services Eirl | Lima - San Martin de Porres | S/. 1.200,00 (Mensual) | 2021 | - | Conocimientos de Php Javascript Html Java Css Net | Visual Studio .Net | - | - | - | - | 14 |	460	6.45%
9	Cluster 10	Manager | Community Manager | Mipleo | Factotum | Lima - San Isidro | S/. 1.100,00 (Mensual) | 2021 | Desarrollo e Implementacion de Proyectos de Software | - | Redes Sociales | - | - | - | - | 10 |	292	4.10%
10	Cluster 9	Developer | Microservices Developer | Freelancer | No Detallado | Indore, India | $1465 | 2021 | Reportar al Project Manager a	282	3.96%

		cargo de los Proyectos en curso en donde se encuentre asignado | - | Microservicios | - | - | - | - | 9 |		
11	Cluster 7	Developer | Developer | Google Jobs | Pacifico Seguros | Cusco - Santiago | No Especificado | 2021 | Desarrollo e Implementacion de Proyectos de Software | Conocimientos de Php Javascript Html Java Css Net | - | - | - | Linea De Carrera | - | 7 |	232	3.25%
12	Cluster 3	Developer | Mobile Developer | Freelancer | No Detallado | Noida, India | $410 | 2021 | - | - | Mobile Apps | - | - | - | - | 3 |	230	3.23%
13	Cluster 11	Developer | Games Developer | Freelancer | No Detallado | Diyarbakir, Turkey | $30 - $250 | 2021 | - | - | Mysql | - | - | - | - | 11 |	197	2.76%
14	Cluster 2	Developer | Php Developer | Freelancer | No Detallado | Saint John, Canada | $165 | 2021 | - | - | Mobile Apps | - | - | - | - | 2 |	172	2.41%
15	Cluster 0	Manager | Community Manager | Mipleo | Chain Services Ti Sac | Cusco - Santiago | No Especificado | 2021 | Creacion/Actualizacion de manuales de usuarios de los desarrollos | - | Redes Sociales | - | - | - | Tecnico o Bachiller en Ing. de Sistemas y/o Afines | - |	136	1.91%

Anexo B.

Resultado experimental extendido, Clustering Kmeans con Scikit-learn (q2)

Inercia: 2,543,012.596
Número de iteraciones: 10.00
Total de instancias: 6987

#	Cluster	Nombre del Cluster	# Inst.	%
1	Cluster 2	Analista Programador | Desarrollo e Implementacion de Proyectos de Software | Conocimientos de Php Javascript Html Java Css Net | - | - | Utilidades | 2 |	808	11.56%
2	Cluster 14	Analista Programador | Reportar al Project Manager a cargo de los proyectos en curso en donde se encuentre asignado | Conocimientos de Php Javascript Html Java Css Net | Visual Studio .Net | - | - | 14 |	782	11.19%
3	Cluster 8	Especialista Transformación Digital | Gestion de Proyectos | Conocimientos de Php Javascript Html Java Css Net | Metodologias Agiles | - | - | 8 |	747	10.69%
4	Cluster 3	Analista Service Desk | Gestion de Proyectos | - | Manejo de Herramientas como Project Trello Miro Figma | - | Seguro de Salud | 3 |	608	8.70%
5	Cluster 0	Data Engineer | Desarrollar los Flujos de Datos y la integracion de los diferentes componentes y elementos de un sistema de Big Data o Machine Learning | Conocimiento Intermedio Python | - | - | - | - |	594	8.50%
6	Cluster 5	Software Design | Participar en el Analisis y Diseño de Soluciones | - | UI Design | Trabajo en Equipo | Seguro de Salud | 5 |	436	6.24%
7	Cluster 4	Gestor de Proyectos | Desarrollo e Implementacion de Proyectos de Software | de Preferencia tener conocimientos de ITIL | - | - | Seguro de Salud | 4 |	435	6.23%
8	Cluster 10	Developer | Reportar al Project Manager a cargo de los proyectos en curso en donde se encuentre asignado | Conocimientos de Php Javascript Html Java Css Net | Visual Studio .Net | - | Linea de Carrera | 10 |	403	5.77%
9	Cluster 7	Soporte Tecnico | Mantenimiento de los Equipos de Computo | - | Linux | Trabajo Bajo Presion | - | 7 |	372	5.32%
10	Cluster 13	Gestor de Proyectos | Reportar al Project Manager a Cargo de los Proyectos en curso en donde se encuentre asignado | - | - | Resolucion de Problemas | - | 13 |	362	5.18%
11	Cluster 6	Analista Programador | Gestion de Proyectos | De preferencia tener conocimientos de ITIL | Base de Datos | - | - | 6 |	338	4.84%
12	Cluster 11	Soporte Tecnico | Desarrollo e Implementacion de Proyectos de Software | Conocimientos en Seguridad Web | - | - | Vale de Alimentacion | 11 |	336	4.81%
13	Cluster 12	Enterprise Architecture | Desarrollo e Implementacion de Proyectos de Software | Conocimientos de Php Javascript Html Java Css Net | Business Development | Resolucion de Problemas | - | 12 |	261	3.74%
14	Cluster 1	Devops Engineer | Reportar al Project Manager a cargo de los proyectos en curso en donde se encuentre asignado | - | Cloud Computing | - | - | 1 |	259	3.71%
15	Cluster 9	Qa Manager | Gestion de Proyectos | - | Calidad De Software | - | - | 9 |	246	3.52%

Anexo C.

Resultado experimental extendido, Clustering Kmeans con Weka (q1)

Intra-Cluster: 34,696.82
Número de iteraciones : 6.00
Total de instancias : 7129

#	Cluster	Especificación del Cluster	# Inst.	%
1	cluster 10	Developer,'Web Developer',Freelancer,'No Detallado','Lima - Ate','No Especificado',2020,'Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Certificado en Scrum','Estabilidad Laboral','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',10	1344	18.85%
2	cluster 12	Developer,Developer,'Google Jobs','No Detallado','Peru - Lima','No Especificado',2020,'Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Certificado en Scrum','Estabilidad Laboral','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',12	1062	14.90%
3	cluster 11	System Analysis','Analista de Sistemas','Google Jobs','No Detallado','No Definido','No Especificado',2020,'Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Certificado en Scrum','Estabilidad Laboral','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',11	816	11.45%
4	cluster 4	Developer,'Soporte Tecnico',Buscojobs,'No Detallado','Peru - Lima','No Especificado',2020,'Gestion de Proyectos','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Certificado en Scrum','Estabilidad Laboral','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',4	787	11.04%
5	cluster 8	Developer,Developer,Mipleo,'No Detallado','Peru - Lima','No Especificado',2020,'Desarrollo E Implementacion De Proyectos De Software','Conocimiento De Base De Datos',Java,'Comunicacion Efectiva','Certificado En Scrum','Estabilidad Laboral','Tecnico O Bachiller En Ing. De Sistemas Y/O Afines',8	712	9.99%
6	cluster 1	Manager,'Gestor De Proyectos',Buscojobs,'No Detallado','Peru - Lima','No Especificado',2020,'Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Certificado en Scrum','Estabilidad Laboral','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',1	354	4.97%
7	cluster 15	Manager,'Gestor de Proyectos','Google Jobs','No Detallado','Peru - Lima','No Especificado',2020,'Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Certificado en Scrum','Estabilidad Laboral','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',15	306	4.29%
8	cluster 14	Developer,'Fullstack Developer','Google Jobs','No Detallado','No Definido','No Especificado',2020,'Desarrollo e Implementacion de	280	3.93%

		Proyectos de Software','Conocimientos de Php Javascript Html Java Css Net',Java,'Comunicacion Efectiva','Certificado en Scrum','Estabilidad Laboral','Tecnico o Bachiller en Ing. de Sistemas y/o Afines',14		
9	cluster 2	Database,'Administrador Base de Datos','Google Jobs','Michael Page','Peru - Lima','No Especificado',2020,'Apoyo y Validacion del correcto despliegue de la Solucion','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Certificado en Scrum','Estabilidad Laboral','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',2	249	3.49%
10	cluster 3	Developer,Developer,Mipleo,'No Detallado','Lima - Miraflores','No Especificado',2020,'Desarrollo e Implementacion de Proyectos de Software','Conocimientos de Php Javascript Html Java Css Net',Java,'Comunicacion Efectiva','Certificado en Scrum','Estabilidad Laboral','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',3	239	3.35%
11	cluster 7	Digital Business Process','Analista Funcional','Google Jobs','No Detallado','Peru - Lima','No Especificado',2020,'Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Devops',Java,'Comunicacion Efectiva','Certificado en Scrum','Estabilidad Laboral','Ingeniero de Sistemas Informatico o Carreras Afines',7	234	3.28%
12	cluster 13	Developer,'Backend Developer',Linkedin,'Bairesdev Sa','Peru - Lima','No Especificado',2020,'Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Certificado en Scrum','Beneficios Corporativos','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',13	229	3.21%
13	cluster 5	Developer,Developer,Buscojobs,'No Detallado','Peru - Lima','No Especificado',2020,'Desarrollo e Implementacion de Proyectos de Software','Conocimientos de Php Javascript Html Java Css Net',Java,'Comunicacion Efectiva','Certificado en Scrum','Linea de Carrera','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',5	200	2.81%
14	cluster 6	Cybersecurity,'Analista Seguridad Informatica','Google Jobs','No Detallado','Peru - Lima','Salario acorde al Mercado',2020,'Analizar Realidad del Negocio (Requerimientos, Procesos, Propuestas de Mejoras)','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Certificado en Scrum','Estabilidad Laboral','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',6	192	2.69%
15	cluster 9	Developer,'.Net Developer','Google Jobs','No Detallado','Peru - Lima','No Especificado',2020,'Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Devops',Java,'Comunicacion Efectiva','Certificado en Scrum','Seguro de Salud','Egresado o Bachiller de Ingenieria de Sistemas, Informatica o Afines',9	125	1.75%

Anexo D.

Resultado experimental extendido, Clustering Kmeans con Weka (q2)

Intra-Cluster: 16,027.00
Número de iteraciones : 4
Total de instancias : 6987

#	Cluster	Nombre del Cluster	# Inst.	%
1	cluster 2	Developer,'Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Estabilidad Laboral',2	2730	39.07%
2	cluster 1	'Analista de Sistemas','Gestion de Proyectos','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Estabilidad Laboral',1	1606	22.99%
3	cluster 6	'Analista Programador','Desarrollo e Implementacion de Proyectos de Software','Conocimientos de Php Javascript Html Java Css Net',Java,'Comunicacion Efectiva','Estabilidad Laboral',6	487	6.97%
4	cluster 4	'Analista Qa','Analizar Realidad del Negocio (Requerimientos, Procesos, Propuestas de Mejoras)','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Estabilidad Laboral',4	457	6.54%
5	cluster 3	'Analista Funcional','Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Devops',Java,'Comunicacion Efectiva','Estabilidad Laboral',3	393	5.62%
6	cluster 9	'Soporte Tecnico','Mantenimiento de los Equipos de Computo','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Estabilidad Laboral',9	339	4.85%
7	cluster 7	'Gestor de Proyectos','Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Base de Datos',Java,'Comunicacion Efectiva','Beneficios Corporativos',7	256	3.66%
8	cluster 8	'Fullstack Developer','Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Base de Datos',React,'Comunicacion Efectiva','Estabilidad Laboral',8	182	2.60%
9	cluster 5	'Soporte Tecnico','Desarrollo e Implementacion de Proyectos de Software','Conocimiento de Excel Avanzado',Java,'Comunicacion Efectiva','Estabilidad Laboral',5	147	2.10%
10	cluster 13	'Fullstack Developer','Reportar al Project Manager a Cargo de los Proyectos en curso en donde se encuentre asignado','Conocimiento de Base de Datos',Java,'Habilidades Blandas','Estabilidad Laboral',13	114	1.63%
11	cluster 12	'Analista BI','Apoyo y Validacion del correcto despliegue de la Solucion','Conocimiento de Base de Datos','Power BI','Comunicacion Efectiva','Estabilidad Laboral',12	98	1.40%
12	cluster 14	'Analista QA','Quality Control','Conocimiento en Metodologias Agiles',Java,'Comunicacion Efectiva','Linea de Carrera',14	85	1.22%
13	cluster 15	'Web Developer','Apoyo y Validacion del correcto despliegue de la Solucion','Conocimiento de Base de Datos',Microservicios,'Comunicacion Efectiva','Estabilidad Laboral',15	44	0.63%

14	cluster 11	'Especialista Transformación Digital','Desarrollar los Flujos de Datos y la Integracion de los diferentes componentes y elementos de un Sistema de Big Data o Machine Learning','Conocimiento de Base de Datos',Microservicios,Responsable,'Capacitaciones Constantes',11	30	0.43%
15	cluster 10	Developer,'Quality Control','Conocimiento de Devops',React,'Comunicacion Efectiva','Estabilidad Laboral',10	19	0.27%

Anexo E.

Perfiles de Empleo TI de los Grupos de Interés

N°	Perfil	ISCO 08	CNPO	MTPE SP2015	N° PEmpleo
1	DEVELOPER	2512	J2662	7	1180
2	WEB DEVELOPER	2513	J2662		764
3	ANALISTA PROGRAMADOR	2519	J2662	3	450
4	SOPORTE TECNICO	2523		17	443
5	FRONTEND DEVELOPER	2513	J2662		399
6	ANALISTA QA			25	391
7	FULLSTACK DEVELOPER	2512	J2662	7	366
8	JAVA DEVELOPER	2512	J2662	7	360
9	GESTOR DE PROYECTOS				324
10	ARQUITECTO DE SOFTWARE				306
11	PHP DEVELOPER	2513	J2662		257
12	ANALISTA DE SISTEMAS	2511		1	248
13	CMS DEVELOPER		J2662		242
14	BACKEND DEVELOPER	2513	J2662		233
15	MOBILE DEVELOPER		J2662		218
16	ANALISTA FUNCIONAL				209
17	.NET DEVELOPER	2513	J2662		205
18	ANALISTA BI				187
19	BI DEVELOPER		J2662		184
20	REACT DEVELOPER	2513	J2662		161
21	COMMUNITY MANAGER				149
22	DEVOPS ENGINEER				147
23	PYTHON DEVELOPER	2513	J2662		138
24	ANDROID DEVELOPER		J2662		136
25	ESPECIALISTA ERP				132
26	ANALISTA SEGURIDAD INFORMATICA				128
27	DATA ENGINEER				123
28	SOFTWARE DESIGN				122
29	CLOUD ENGINEER				117
30	PRACTICANTE TI				109
31	GERENTE DE TI			23	104

N°	Perfil	ISCO 08	CNPO	MTPE SP2015	N° PEmpleo
32	ANALISTA INFRAESTRUCTURA TI	2522			102
33	SYSTEM ENGINEER			2	102
34	ADMINISTRADOR BASE DE DATOS	2521		16	99
35	ANALISTA DE CAPACITACION	2356		26	94
36	SYSTEMS ADMINISTRATOR	2522			94
37	ANALISTA TI				93
38	ANALISTA DE PROCESOS				87
39	ANALISTA SERVICE DESK				87
40	ANALISTA BASE DE DATOS			13	85
41	WEB SERVICES DEVELOPER		J2662		80
42	ESPECIALISTA ECOMMERCE				77
43	SOFTWARE ENGINEER			1	76
44	DATA SCIENTIST				75
45	ESPECIALISTA CIBERSEGURIDAD				75
46	QA DEVELOPER		J2662	25	68
47	ESPECIALISTA TI				66
48	ANGULAR DEVELOPER	2513	J2662		63
49	ADMINISTRADOR REDES Y COMUNICACIONES	2523	J2661		61
50	MACHINE LEARNING DEVELOPER		J2662		53
51	GAMES DEVELOPER		J2662		52
52	DEVELOPER MANAGER				51
53	JAVASCRIPT DEVELOPER	2513	J2662		48
54	RUBY DEVELOPER	2513	J2662		47
55	SEARCH ENGINE EXPERT				47
56	JEFE DE SERVICIOS TI				40
57	C/C++ DEVELOPER	2513	J2662		39

N°	Perfil	ISCO 08	CNPO	MTPE SP2015	N° PEmpleo
58	DEVOPS DEVELOPER		J2662		39
59	BLOCKCHAIN DEVELOPER		J2662		37
60	ANALISTA PROGRAMADOR POWER BUILDER	2512	J2662	3	36
61	ANALISTA PROGRAMADOR WEB	2513	J2662	3	35
62	ANALISTA REDES Y COMUNICACIONES	2523	J2661	15	35
63	ESPECIALISTA TRANSFORMACIÓN DIGITAL				33
64	DATA ARCHITECT				29
65	ANALISTA GESTION DE INFORMACION				28
66	COBOL DEVELOPER	2512	J2662	7	28
67	IOT DEVELOPER		J2662		27
68	BI MANAGER				26
69	ANALISTA DE SERVICIOS TI		J2663	2	25
70	SALES TI ENGINEER	2434			24
71	BIGDATA ENGINEER				23
72	ANALISTA DE RIESGOS				22
73	CMS MANAGER				22
74	BUSINESS DEVELOPER				21
75	SECURITY ENGINEER				20
76	ANALISTA BPM				18
77	ANALISTA DE INTEGRACION				18
78	BIGDATA DEVELOPER		J2662		18
79	CLOUD DEVELOPER		J2662		18
80	JEFE INFRAESTRUCTURA TI				18
81	ANALISTA CLOUD				17
82	CLOUD ARCHITECT				17
83	ESPECIALISTA REDES Y COMUNICACIONES	2523	J2661		17
84	IOS DEVELOPER		J2662		17
85	ESPECIALISTA BI				16
86	ESPECIALISTA IA				16
87	ANALISTA BIGDATA				14
88	CLOUD DATA ENGINEER				14

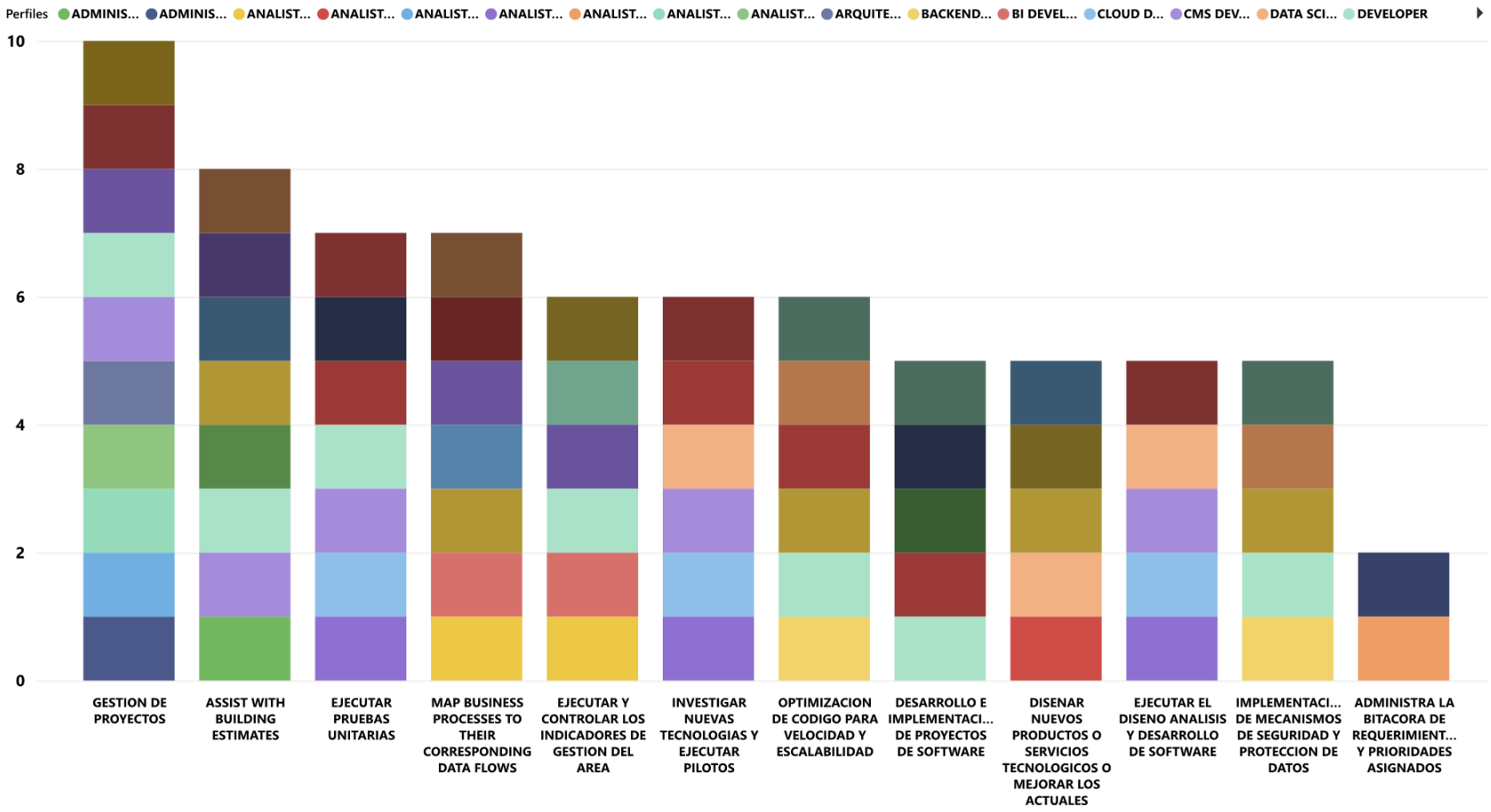
N°	Perfil	ISCO 08	CNPO	MTPE SP2015	N° PEmpleo
89	FLUTTER DEVELOPER	2513	J2662		14
90	PRODUCT MANAGER				14
91	RPA DEVELOPER		J2662		14
92	MICROSERVICES DEVELOPER		J2662		13
93	QA MANAGER				13
94	GESTOR SERVICE DESK				12
95	NODE.JS DEVELOPER	2513	J2662		12
96	WEBSCRAPING DEVELOPER		J2662		12
97	ANALISTA CRM				11
98	DATA ENGINEER & DEVELOPER				11
99	ANALISTA DATAWAREHOUSE				10
100	BACKEND NODE.JS	2513	J2662		10
101	ANALISTA PROGRAMADOR BI		J2662		9
102	BACKEND JAVA DEVELOPER	2513	J2662		9
103	SQL DEVELOPER		J2662		9
104	BACKEND AWS DEVELOPER		J2662		8
105	ENTERPRISE ARCHITECTURE				8
106	AUDITOR DE SISTEMAS				7
107	CHATBOT DEVELOPER		J2662		7
108	COORDINADOR ACADEMICO				7
109	PROCESS MANAGER				7
110	FRONTEND ANDROID		J2662		6
111	COMPUTER SCIENCE				5
112	MICROSERVICES ARCHITECT				5
113	COORDINADOR BASE DE DATOS				4
114	ESPECIALISTA ECM				4
115	ESPECIALISTA GESTION DE INFORMACION				4
116	JEFE SOPORTE TECNICO				4
117	BACKEND PYTHON	2513	J2662		3
118	JEFE AUTOMATIZACION RPA				3

N°	Perfil	ISCO 08	CNPO	MTPE SP2015	N° PEmpleo
119	SCALA DEVELOPER	2513	J2662		3
120	SUPERVISOR BASE DE DATOS SIG				3
121	ARQUITECTO ANDROID				2
122	ARQUITECTO BACKEND				2
123	BACKEND MOBILE DEVELOPER		J2662		2
124	BACKEND PHP	2513	J2662		2
125	COORDINADOR SERVICE DESK				2
126	DIRECTOR CLOUD PLATFORM				2
127	GENEXUS DEVELOPER		J2662		2
128	ANALISTA FORENSE INFORMATICO				1
129	DATAMINING EXPERT				1
130	EMBEDDED DEVELOPER		J2662		1
131	ESPECIALISTA LCMS				1
132	PERL DEVELOPER		J2662		1
133	RESEARCH EXPERT				1

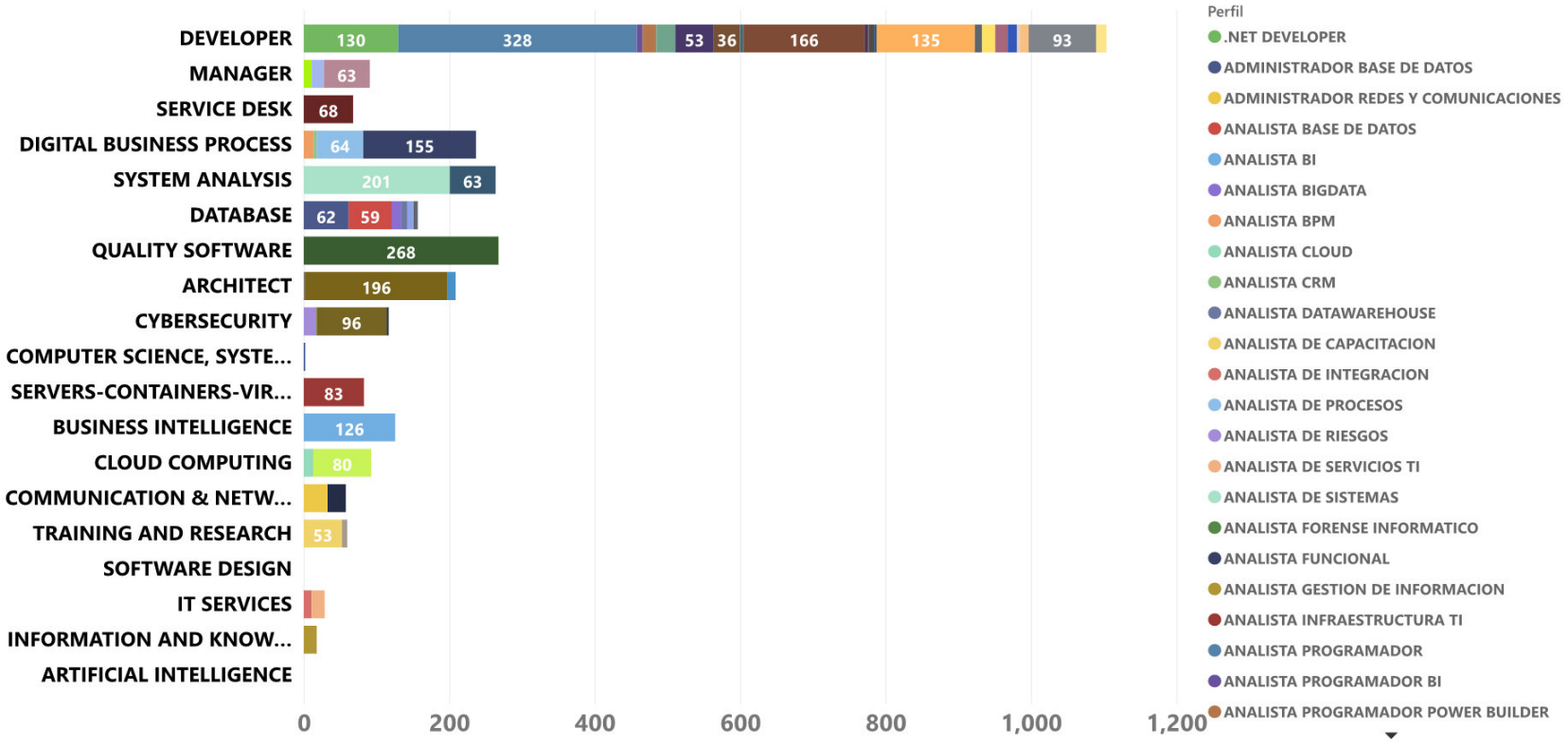
Anexo F.

Dashboards del proyecto

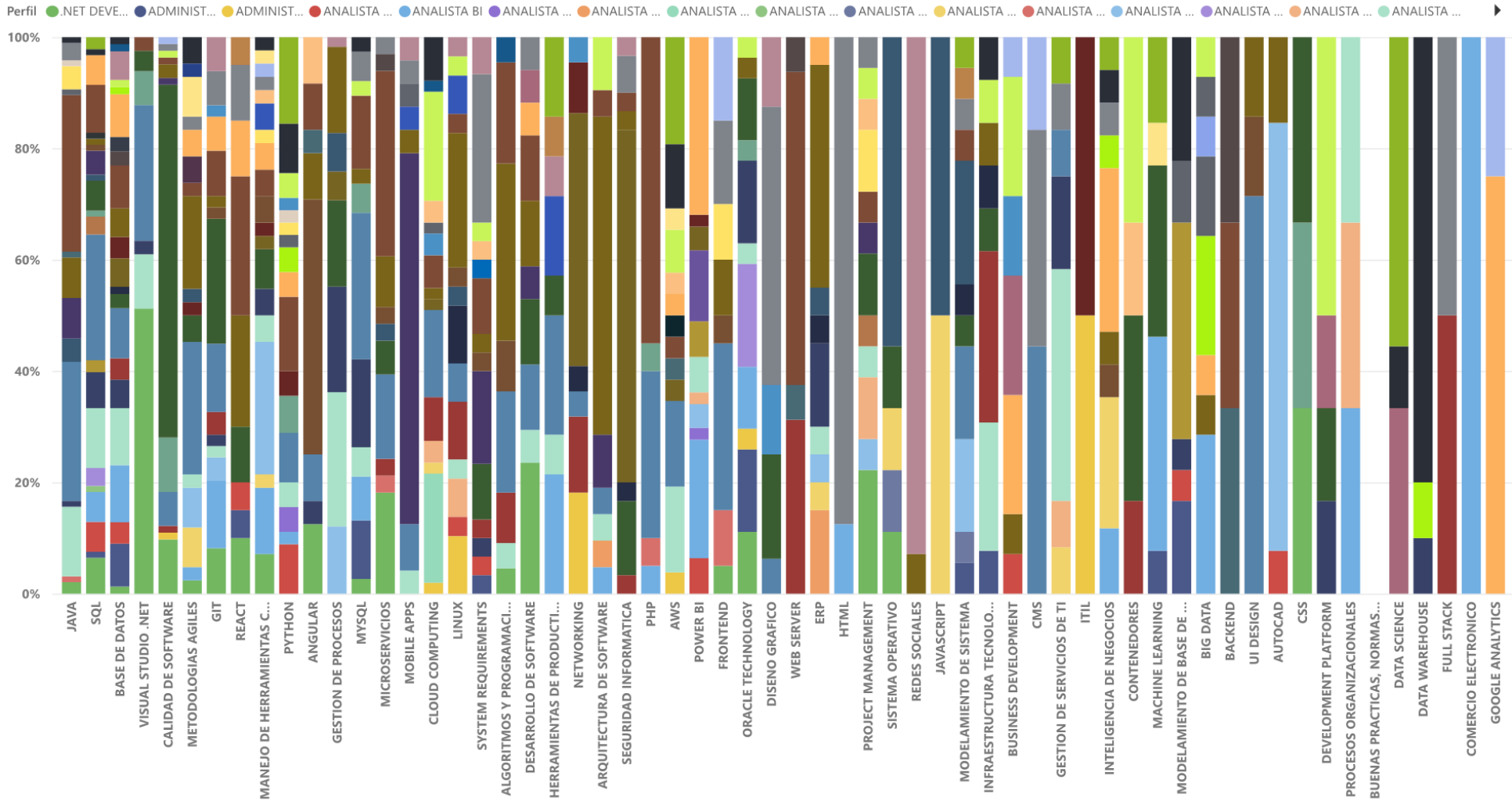
Funciones Transversales de Profesionales de TI



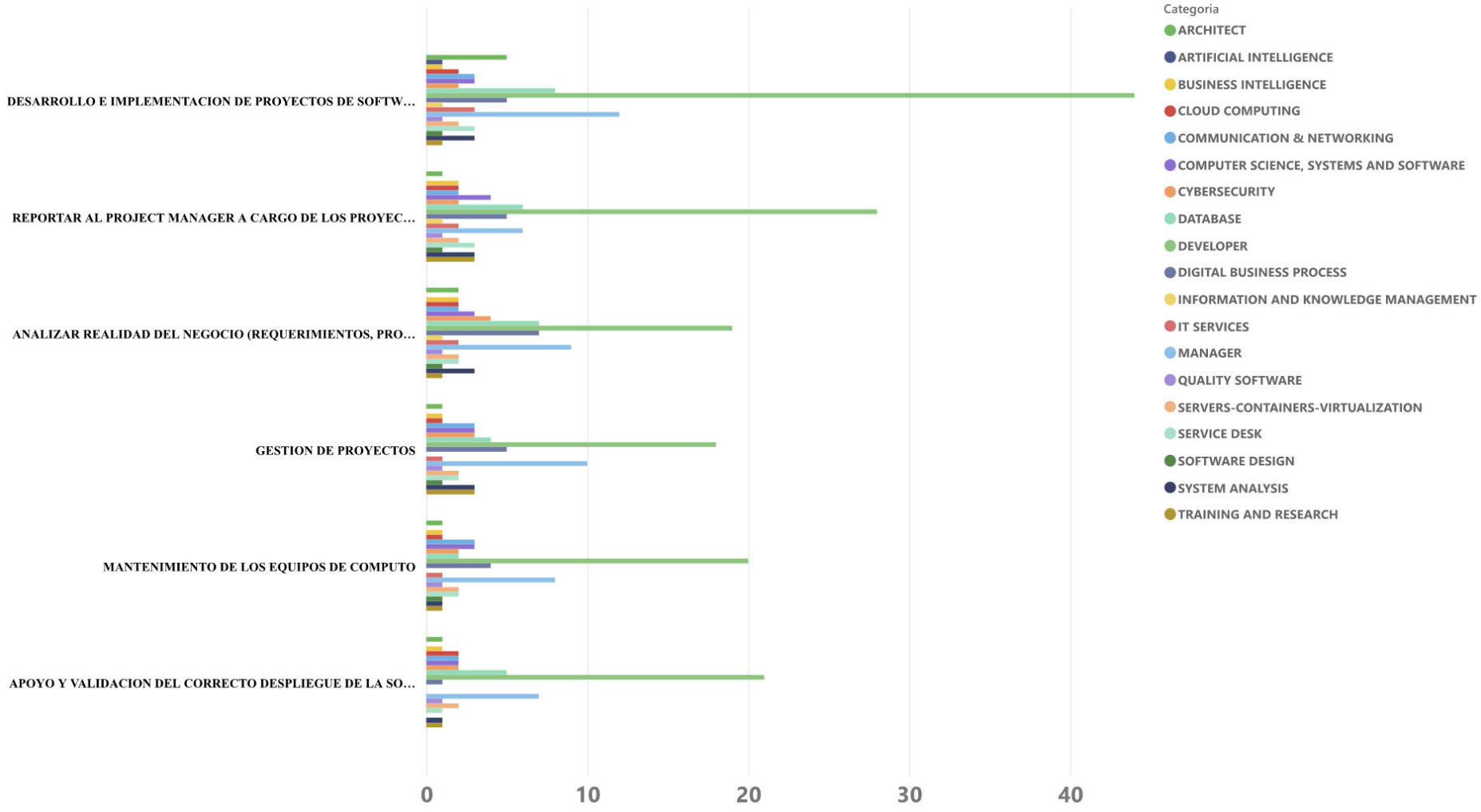
Demanda de Puestos de Empleo TI por Categoría y Perfil



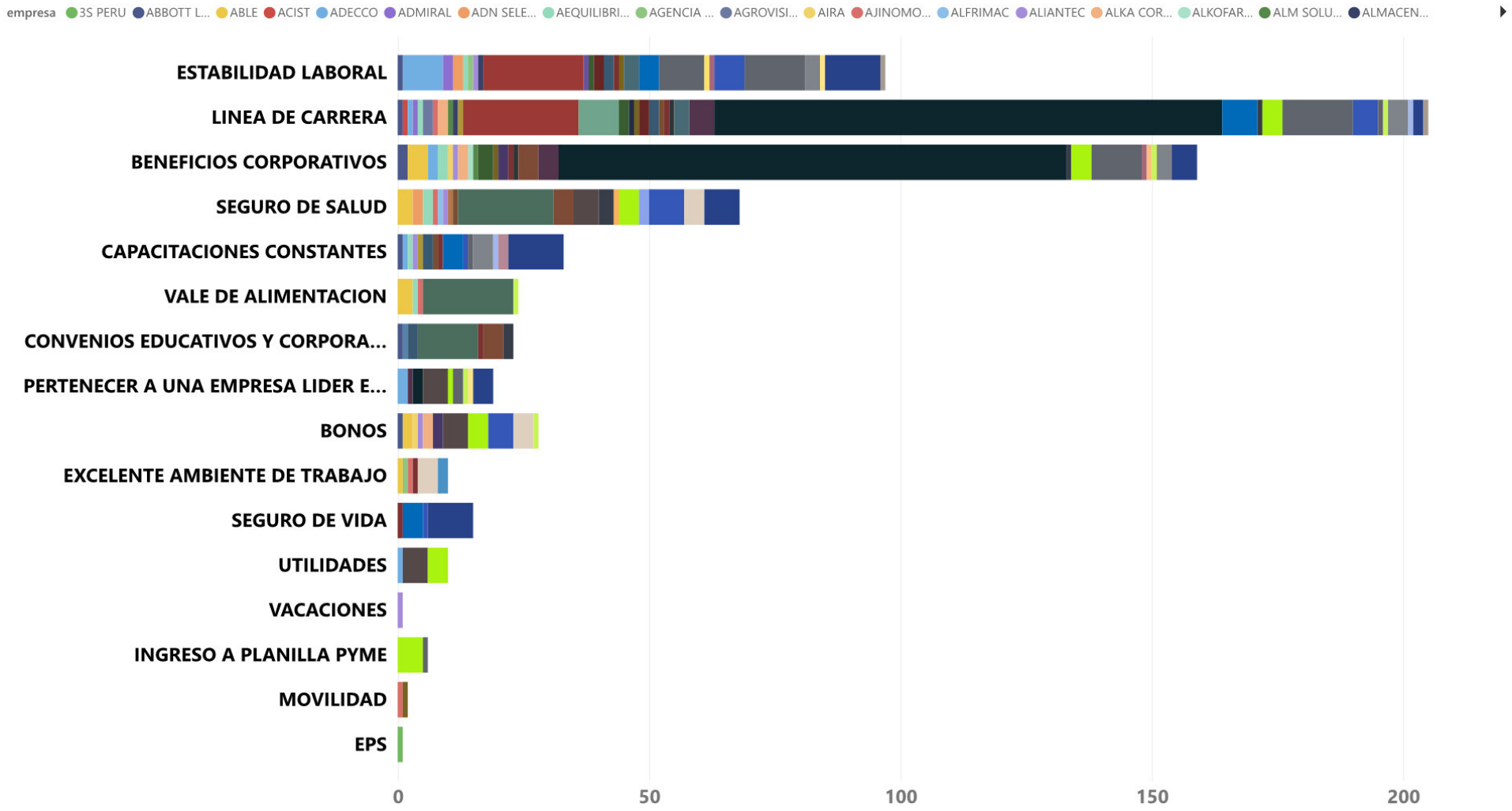
Demanda de Competencias en Puestos de Empleo TI



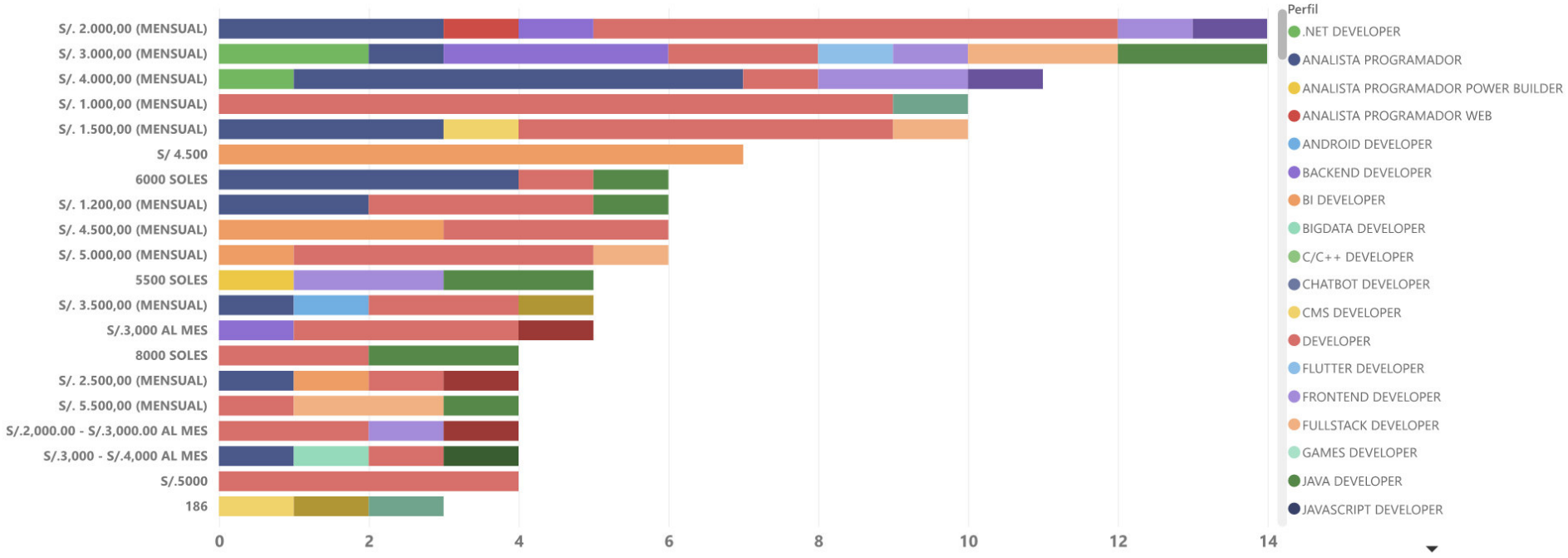
Demanda de Funciones en Puestos de Empleo TI



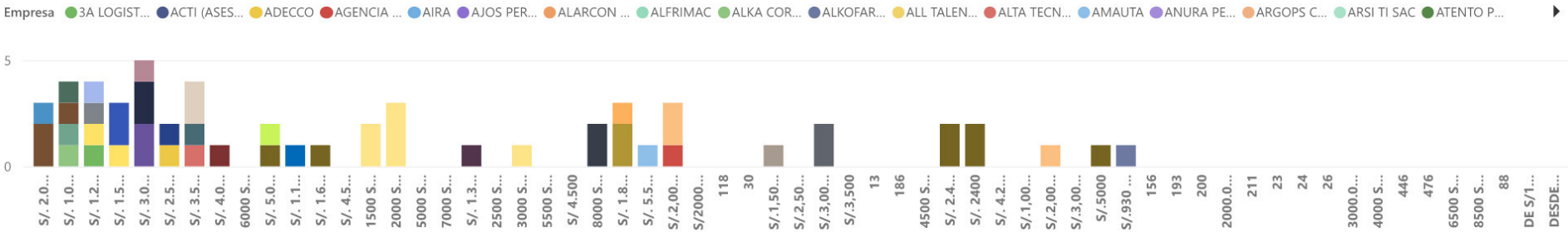
Beneficios del Empleador en la Demanda de Puestos de Empleo TI



Salarios en la Demanda de Puestos de Empleo TI



Salarios ofrecidos por Empleador



Demanda de competencias en Puestos de Empleo TI

Demanda de Habilidades en Puestos de Empleo TI