



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

**ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS**

TESIS

**DESARROLLO DE UN SISTEMA INTELIGENTE
PARA LA EVALUACIÓN DE LOS PERFILES POR
COMPETENCIA LABORAL DE UN PUESTO
GERENCIAL**

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

Autor (es):

Bach. Cabanillas Torres Alvaro Paul Harbert

ORCID: <https://orcid.org/0000-0002-7197-2419>

Bach. Farro Vargas Luis Ademar

ORCID: <https://orcid.org/0000-0003-3245-8393>

Asesor(a):

Mg. Chirinos Mundaca Carlos Alberto

ORCID: <https://orcid.org/0000-0002-6733-8992>

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú 2022

APROBACIÓN DEL JURADO

DESARROLLO DE UN SISTEMA INTELIGENTE PARA LA EVALUACIÓN DE LOS PERFILES POR COMPETENCIA LABORAL DE UN PUESTO GERENCIAL

Bach. Cabanillas Torres Alvaro Paul Harbert

Autor

Bach. Farro Vargas Luis Ademar

Autor

Mg. Chirinos Mundaca Carlos Alberto

Asesor

Mg. Diaz Vidarte Miguel Orlando

Presidente de Jurado

Mg. Bravo Ruiz Jaime Arturo

Secretario de Jurado

Mg. Bances Saavedra David Enrique

Vocal de Jurado



DECLARACIÓN JURADA DE ORIGINALIDAD

Quienes suscriben la **DECLARACIÓN JURADA**, somos **egresado (s)** del Programa de Estudios de INGENIERÍA DE SISTEMAS de la Universidad Señor de Sipán S.A.C, declaramos bajo juramento que somos autores del trabajo titulado:

DESARROLLO DE UN SISTEMA INTELIGENTE PARA LA EVALUACIÓN DE LOS PERFILES POR COMPETENCIA LABORAL DE UN PUESTO GERENCIAL

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética en Investigación de la Universidad Señor de Sipán (CIEI USS) conforme a los principios y lineamientos detallados en dicho documento, en relación a las citas y referencias bibliográficas, respetando al derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y autentico.

En virtud de lo antes mencionado, firman:

Cabanillas Torres Alvaro Paul Harbert	DNI: 71775833	
Farro Vargas Luis Ademar	DNI: 75557753	

Pimentel, 25 de enero de 2023.

DEDICATORIAS

CABANILLAS TORRES ALVARO PAUL HARBERT

Este proyecto de tesis está dedicado a mis padres que día a día me dieron su apoyo, brindándome su fortaleza y entusiasmo de poder seguir adelante afrontando a cumplir los objetivos planteados y con sus valores que me inculcaron desde el hogar para poder desempeñarme como una persona capaz en las diferentes etapas de mi vida que atravesaré.

FARRO VARGAS LUIS ADEMAR

Este presente proyecto de tesis está dedicado principalmente a dios por guiarnos a lo largo de nuestra vida, a mis padres por su amor y sacrificio en todos estos años de vida y mi hermana por brindándome por estar presente dándome consejos, opiniones en esta etapa de mi vida y a todas las personas que nos han apoyado.

AGRADECIMIENTOS

CABANILLAS TORRES ALVARO PAUL HARBERT

Agradezco a Dios por gozar de buena salud, a los ingenieros quien con sus consejos nos brindaron las orientaciones para poder desarrollar nuestra investigación, a mi amiga del alma Christal Ramírez que siempre está conmigo dándome ánimos día a día, a mi compañero Farro Vargas Luis Ademar por ser ese gran amigo incondicional, lo considero como una persona responsable, comprometedor y optimista, que me brindó su invaluable apoyo, a la facultad de Ingeniería de Sistemas de la Universidad Señor de Sipán, por la gran enseñanza que me brindó durante mi formación profesional universitaria.

FARRO VARGAS LUIS ADEMAR

Agradezco a Dios por ser el apoyo y fortaleza en aquellos momentos de dificultad y de debilidad, agradezco a mis padres Angela y Luis y hermana Nereyda por ser un gran apoyo y estar presentes en mis metas por sus valores inculcados que me forman día a día en mi vida, a mi compañero Cabanillas Torres Alvaro por compartir conocimientos y ser un gran amigo en la realización de esta tesis, agradecer a Melissa y Lucero por su apoyo en la presente tesis, a nuestro asesor de tesis Bances Saavedra David Enrique, director de escuela Tuesta Monteza Victor y docentes de la escuela de ingeniería de sistemas de la universidad Señor de Sipán por haber compartido conocimientos a lo largo de nuestra formación profesional.

Resumen

La selección de personal enfocado al área gerencial actualmente necesita de una persona encargada para realizar los procesos que conlleva escoger al candidato más adecuado, lo que implica mucho tiempo y dinero.

En la actualidad el utilizar un sistema inteligente permite ser capaz de analizar y procesar datos mucho más rápido con el fin de resolver el problema de encontrar al mejor participante al puesto de gerencia.

Por ende, este estudio se enfocó en la creación de una Aplicación, cuyo fundamento es un sistema inteligente, donde se evaluaron 5 algoritmos entre ellos Naive Bayes escogiéndose éste como el mejor para utilizar en el software, donde se utilizó un dataset que presenta puntajes de evaluación, así como también datos relevantes de las personas donde permitió entrenar al algoritmo de clasificación para posteriormente ingresar los datos de los postulantes y así determinar si cumplen con las características con las que trabaja el dataset.

En base a ello el diseño del software cuenta con un modelo de clasificación mostrando estadísticas basadas en tiempo de respuesta de 12.40, un consumo de memoria de 653.14 Mb y un consumo de CPU de 11.15% además se evaluó la precisión con un porcentaje de 0.992%, una Rcall o exhaustividad de 0.890% y una exactitud con un porcentaje de 0.942%, con todos estos resultados podemos ver que la eficiencia del software Indicándonos en los resultados que el software es bueno para predecir y escoger al mejor candidato al puesto de gerencia.

Palabras Clave:

Algoritmo de clasificación, Aprendizaje automático, evaluación de perfil laboral, Exactitud, Metodología, Puesto Gerencial, Naive Bayes, sistema Inteligente.

Abstract

The selection of personnel focused on the management area currently requires a person in charge to carry out the processes that involve choosing the most suitable candidate, which implies a lot of time and money. The selection of personnel focused on the management area currently requires a person in charge to carry out the processes that involve choosing the most suitable candidate, which implies a lot of time and money.

Currently, using an intelligent system allows being able to analyze and process data much faster in order to solve the problem of finding the best participant for the management position.

Therefore, this study focused on the creation of an Application, whose foundation is an intelligent system, where 5 algorithms were evaluated, including Naive Bayes, choosing the latter as the best to use in the software, making use of a dataset that presents scores of evaluation as well as relevant data of the people where it allowed to train the classification algorithm to later enter the data of an applicant and thus determining if it meets the characteristics with which the dataset works.

Based on this, the software design has a classification model showing statistics based on a response time of 12.40, a memory consumption of 653.14 Mb and a CPU consumption of 11.15%, the precision was also evaluated with a percentage of 0.992%, an Rcall or recall of 0.890% and an accuracy with a percentage of 0.942%, With all these results we can see that the efficiency of the software Indicating in the results that the software is good at predicting and choosing the best candidate for the management position.

Keywords:

Classification algorithm, Machine learning, profile evaluation, Accuracy, methodology, Managerial position, Naive Bayes, Smart system.

ÍNDICE

I. INTRODUCCION	11
1.1. Realidad Problemática.....	11
1.2. Trabajos Previos.	16
1.3. Teorías Relacionadas al Tema.	31
1.3.1. Naive Bayes:.....	31
1.3.2. Arboles de Decisión:.....	31
1.3.3. Algoritmo Random Forest.....	33
1.3.4. Algoritmo de Regresión Logística.....	34
1.3.5. SVM.....	34
1.3.6. KNN:	35
1.3.7. Lógica difusa:.....	35
1.3.8. Método de toma de decisiones:.....	36
1.3.9. Smote (Synthetic Minority Oversampling Technique):	39
1.3.10. Métricas Derivadas:.....	40
1.3.11. Inteligencia Artificial:.....	42
1.3.12. Aprendizaje Automático (Machine Learning):.....	43
1.3.13. TEST de DISC.....	45
1.3.14. TEST de Valanti	46
1.3.15. TEST de IPV	47
1.4. Formulación del Problema.	48
1.5. Justificación e Importancia del Estudio.	48
1.6. Hipótesis.	49
1.7. Objetivos.	49
1.7.1. Objetivo General.	49
1.7.2. Objetivo Específico.	49
II. MATERIAL Y METODO.....	50

2.1. Tipo y Diseño de Investigación.	50
2.2. Población y Muestra.....	50
2.1.1. Población:.....	50
2.1.2. Muestra:.....	50
2.3. Variables, Operacionalización:.....	52
2.4. Técnicas e Instrumentos de Recolección de Datos, Validez y Confiabilidad.	53
2.4.1. Técnicas:	53
2.4.2. Instrumentos:.....	53
2.5. Procedimiento de Análisis de Datos.....	54
2.6. Criterios Éticos.	58
2.7. Criterios de Rigor Científico.	58
III. RESULTADOS.	59
3.1. Resultados en Tablas y Figuras.....	59
3.2. Discusión de resultados.	66
3.3. Aporte práctico.	68
IV. CONCLUSIONES Y RECOMENDACIONES.....	97
4.1. Conclusiones.....	97
4.2. Recomendaciones.....	98
REFERENCIAS.....	99
ANEXOS.	104

ÍNDICE DE FIGURAS

Figura 1: Conjunto de datos de entrenamiento y prueba..	13
Figura 2. Diagrama de bloques que ilustra el modelo propuesto..	15
Figura 3: Estructura esquemática que muestra las cualidades buscadas de cada candidato ..	22
Figura 4: Estructura de árbol de decisión ..	32
Figura 5: Arquitectura Random Forest..	33
Figura 6: Soporte de división de Hiperplano de máquina vectorial.....	34
Figura 7: Ejemplo de aprendizaje no supervisado.....	44
Figura 8: Evaluación de resultados de la precisión en base al entrenamiento realizado.....	60
Figura 9: Evaluación de resultados del Consumo de Memoria en base al entrenamiento realizado.....	61
Figura 10: Evaluación de resultados del Consumo de CPU en base al entrenamiento realizado.....	62
Figura 11: Evaluación de resultados de la Exactitud en base al entrenamiento realizado.....	63
Figura 12: Evaluación de resultados de la Precisión en base al entrenamiento realizado.....	64
Figura 13: Evaluación de resultados de la Exhaustividad en base al entrenamiento realizado.....	65
Figura 14: Diagrama de objetivos.....	68
Figura 15: Estructura del dataset.	74
Figura 16: Diagrama de Procesos de Negocios..	80
Figura 17: Modelo de caso de uso del negocio.	81
Figura 18: Gestión de Selección de Personal.	81
Figura 19: Modelo del Dominio del Problema.....	81
Figura 20: Paquete de Requerimientos del Sistema	82
Figura 21: Gestión de Selección de Personal..	83
Figura 22: Glosario de Casos de Uso.	84
Figura 23: Diagrama de estado	84
Figura 24: Diagrama de Gantt.....	86
Figura 25: Etapas de desarrollo.	87

<i>Figura 26: Grafica de barras del dataset</i>	88
<i>Figura 27: Diseño de base en Xampp</i>	89
<i>Figura 28: Resultados de Entrenamientos del software.</i>	94

ÍNDICE DE TABLAS

Tabla 1: Criterios y pesos de evaluación.....	18
Tabla 2: Pasos del método TOPSIS según Hwang y Yoon.....	21
Tabla 3: Resultados con el método TOPSIS del candidato más idóneo.	23
Tabla 4: Resultados sin características de selección.	26
Tabla 5: Resultados con características de selección.....	26
Tabla 6: <i>Descripción de la educación de las variables de Normalización.</i>	38
Tabla 7: Descripción de la fórmula para determinar el valor de la utilidad.	38
Tabla 8: <i>Descripción de las variables para hallar el valor final.</i>	39
Tabla 9: Leyenda de las variables utilizadas.	40
Tabla 10: Valores del test de Valanti.....	47
Tabla 11: Valores del test de IPV	48
Tabla 12: Atributos a utilizar para el entrenamiento del sistema.	51
Tabla 13: Descripción de variables del tiempo de respuesta.	54
Tabla 14: Descripción de variables del consumo de memoria.	55
Tabla 15: Descripción de variables de Consumo de CPU.....	55
Tabla 16: <i>Clasificación de variables del indicador Exactitud.</i>	56
Tabla 17: <i>Clasificación de variables del indicador Precisión.</i>	57
Tabla 18: Clasificación de variables del indicador R-Call ó Exhaustividad.....	57
Tabla 19: <i>Resultados del tiempo de Respuesta del algoritmo de clasificación.</i> 59	
Tabla 20: Resultado de consumo de memoria.	60
Tabla 21: Resultado de consumo de CPU	61
Tabla 22: <i>Cuadro de entrenamiento de la Exactitud.</i>	63
Tabla 23: <i>Cuadro de entrenamiento de la Precisión</i>	64
Tabla 24: <i>Cuadro de entrenamiento de la Rcall o Exhaustividad</i>	65
Tabla 25: <i>Tabla de atributo de los especialistas.</i>	69
Tabla 26: Atributos del dataset.....	70
Tabla 27: Tabla comparativa de algoritmos.....	75
Tabla 28: Lista de requerimientos funcionales	76

Tabla 29: Descripción del requerimiento funcional 1	77
Tabla 30: Descripción del requerimiento funcional 2.....	78
Tabla 31: Descripción del requerimiento funcional 3.....	78
Tabla 32: Descripción del requerimiento funcional 4.....	79
Tabla 33: Lista de requerimientos no funcionales.	85
Tabla 34: <i>Matriz de confusión de Entrenamiento 01</i>	95
Tabla 35: Matriz de confusión de Entrenamiento 02.	95
Tabla 36: Matriz de confusión de Entrenamiento 03	95

I. INTRODUCCION

1.1. Realidad Problemática.

Un sistema inteligente es un programa informático que intenta imitar el comportamiento del pensamiento humano para lograr dar respuesta a un problema en específico logrando los objetivos planteados, se caracteriza porque el sistema tiene la posibilidad de analizar y procesar de una manera análoga similar al razonamiento humano sobre un problema y así mejorar sus funciones a través de la experiencia.

Si bien es cierto hoy en día para la selección de personal en una empresa se necesitan de una persona capacitada en esa área, los procesos que se necesita son muy tediosos y el tiempo de respuesta es muy largo por lo que es bueno decir que utilizar un sistema inteligente capaz de procesar tanto los requisitos que solicita la empresa para el puesto de trabajo como también el curriculum vitae del postulante sería de gran ayuda para la organización.

Por su parte, Bachtiar, Pradana, & Yudiari (2019) en la investigación que realizó, Employee Recruitment Recommendation Using Profile Matching and Naïve Bayes, nos comenta que encontrar un trabajo en una empresa es cada vez más complicado no sólo por las cualidades que pide la institución sino también por la gran cantidad de personas que postulan ha dicho puesto, esta situación afecta a la empresa a la hora de escoger un candidato ya que todo su proceso lo realiza de manera manual y eso hace que la selección demore mucho más.

En cierto modo, si la institución cuenta con un sistema que permita un control a la hora de escoger un postulante para un puesto de trabajo hace que sea más eficiente, teniendo en cuenta que el sistema inteligente esté monitoreado por un profesional que está capacitado para corroborar las predicciones donde si la persona es apta o no, sean correctas. Es importante recalcar que para que el sistema realice

buenos resultados, se necesitará un gran número de datos a fin de adecuar los requerimientos solicitados por la empresa.

Asimismo, Waheed, Moshirpour, Moshirpour, Rokne, & Alhadj (2017) en la investigación *Effective Personnel Selection and Team Building Using Intelligent Data Analytics*, nos menciona que para mejorar la eficiencia a la hora de escoger y seleccionar un personal se viene utilizando desde hace unos años técnicas de minería de datos.

Si bien se viene realizando varias pruebas con métodos y técnicas diferentes con el fin de ver cuál es el más ideal para la elaboración de un sistema inteligente abarcado en la Ingeniería de Sistemas considerando la contratación de un empleado. Esto nos lleva a tomar en cuenta la minería de datos, donde teniendo un gran volumen de información se pueda entrenar y obtener soluciones óptimas.

Korkmaz (2019) enfatiza en su investigación *Personnel Selection Method Based On Topsis Multi-Criteria Decision Making Method* que se han visto casos muy significativos donde el jefe es muy exitoso en su puesto de campo de trabajo, pero al no contar con un buen personal de equipo tiende a fracasar.

Debido a la competitividad que existe en la actualidad las empresas están obligadas a contar con profesionales que se desempeñan con éxito en sus puestos y así afrontar estos retos, pero para escoger los mejores no es tarea fácil, por ejemplo al escoger un buen gerente se necesita varios requisitos fundamentales como: haber trabajado anteriormente en un cargo similar, tener experiencia en diferentes áreas de la empresa ya sea marketing, finanzas, entre otros., debe estar preparado para manejar el tema del mercado internacional y sobre todo ser un líder capaz de mejorar continuamente y así hacer sentir a sus trabajadores cómodos en lo que desempeñan con el fin de obtener buenos resultados. Para ello el tener un sistema inteligente que permita considerar estos puntos es de total ayuda no solo porque es más eficiente, sino que toma en cuenta muchos atributos

trabajando de la mano con una metodología adecuada al presentar mejores resultados.

Para mitigar este tipo de problemas, la tecnología nos está dando una mano creando sistemas que pueden ayudar a la empresa a escoger todos los buenos aspectos en los que se tomará en cuenta, evaluando el perfil de la persona tomando en cuenta los requisitos propuestos por la institución y así seleccionar al mejor candidato al puesto.

Al hablar del uso de un sistema inteligente capaz de escoger al mejor candidato al puesto de trabajo se tiene que tomar en cuenta el uso de una metodología existente capaz de gestionar y procesar los datos necesarios con la finalidad de poder escoger al más óptimo, siendo la metodología más eficiente en base a los resultados obtenidos.

Por una parte Rabcan, Vaclavkova, & Blasko (2017) realizaron un estudio para seleccionar resultados óptimos utilizando árboles de decisión difuso con la incorporación de algoritmos C4.5 en base a los criterios de empresa de seguridad donde utilizaron un conjunto de entrenamiento y pruebas, procesándolo en la clasificación de árboles de decisión difusos.

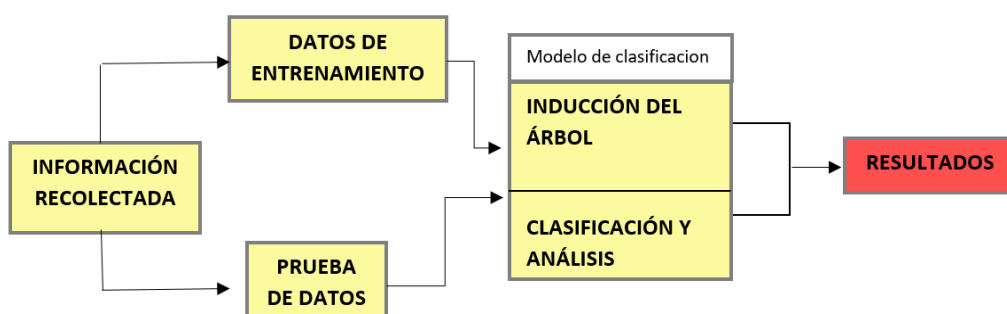


Figura 1: Conjunto de datos de entrenamiento y prueba. Fuente: (Rabcan, Vaclavkova, & Blasko, 2017).

Donde utilizando árboles de decisión con el algoritmo C4.5 dio un resultado de exactitud equivalente de 97.27 % siendo árboles de decisión óptimos para el proceso de selección de personal en el sistema inteligente.

En la investigación de Waheed, Moshirpour, Moshirpour, Rokne, & Alhajj, (2017) propusieron una solución para escoger la selección de personal y en la formación de equipos analizando los rasgos de personalidad de los mejores candidatos de un equipo a postular frente a los demás del equipo para encontrar la mejor combinación en términos de dinámica y así aumentar a gran medida el rendimiento de todo el personal de la empresa.

Con todo lo descrito anteriormente realizaron una serie de pasos:

1. Crearon una base de datos de los equipos ideales.
2. Utilizaron la minería de reglas de asociación para obtener la asociación en los rasgos de personalidad de los equipos.
3. Luego construyeron un clasificador Naive Bayes donde utilizaron rasgos de personalidad escogidos anteriormente para predecir al candidato para trabajar en un equipo ya existente.
4. Con el clasificador creado anteriormente se analiza los resultados de las pruebas de personalidad para los miembros óptimos de un equipo.

En la imagen a continuación se detalla paso a paso los puntos descritos anteriormente.

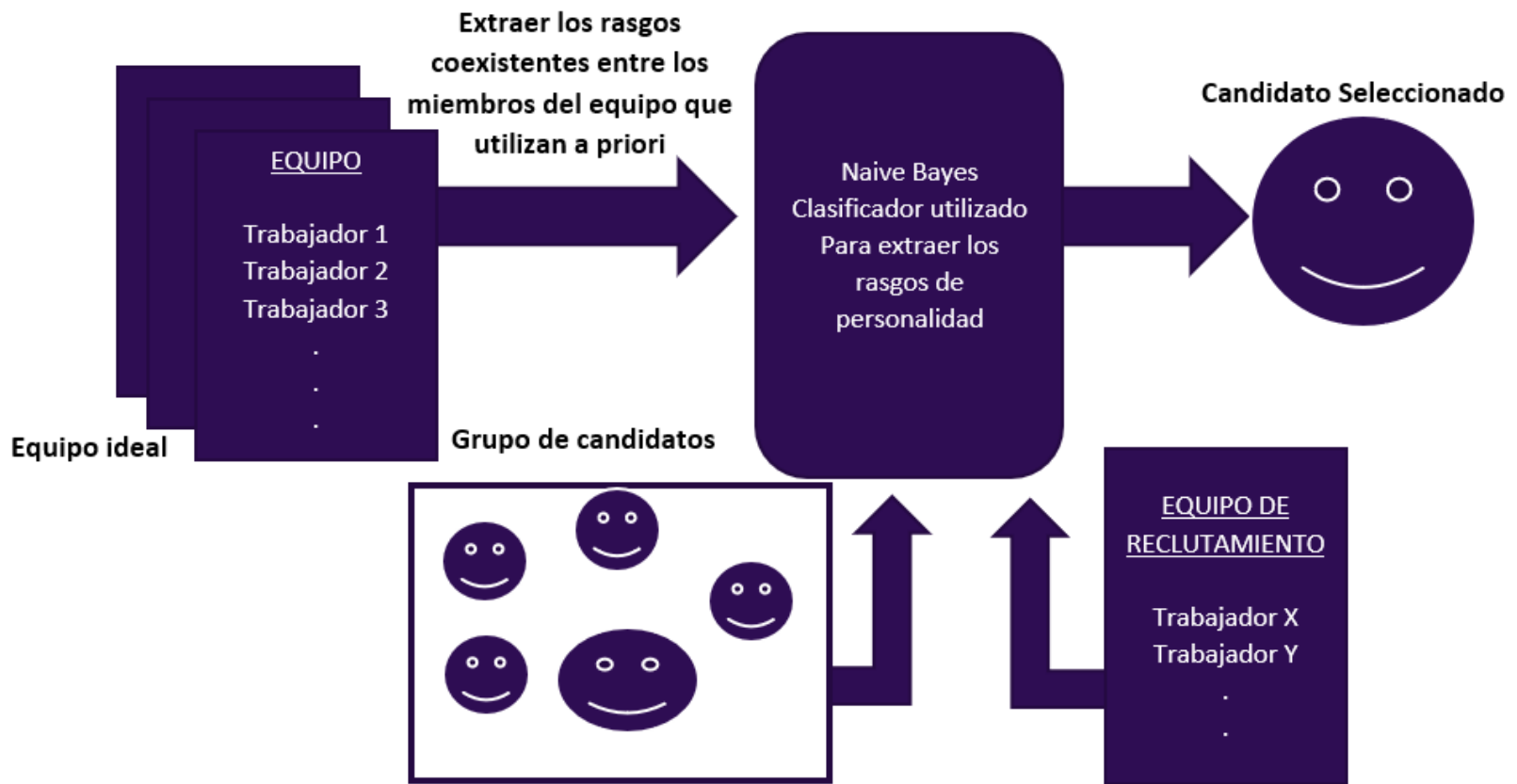


Figura 2. Diagrama de bloques que ilustra el modelo propuesto. Fuente: (Waheed, Moshirpour, Moshirpour, Rokne, & Alhaji, 2017).

Para concluir se puede decir que el uso de un sistema inteligente para la selección de personal en la actualidad es útil para las empresas siendo eficiente en base a los resultados arrojados usando una metodología, teniendo resultados de exactitud, exhaustividad y precisión óptimos para la detección del candidato más aptos y no aptos.

1.2. Trabajos Previos.

Bachtiar, Pradana, & Yudiari (2019) realizaron la investigación Employee Recruitment Recommendation Using Matching and Naive Bayes en Indonesia. Unos de los desafíos que surgen en la participación de selección de personal son la cantidad de personas a evaluar lo que conlleva hacer un proceso manual muy tedioso. En base a eso los investigadores utilizaron Profile Matching para evaluar de manera absoluta a los candidatos y Naive Bayes para establecer al candidato según sus aptitudes. La comprobación de datos se realiza mediante la adecuación de dos variables estadísticas analizando la coincidencia de perfiles (Profile Matching) y por otra parte la precisión se emplea para establecer al solicitante óptimo para ese puesto. Las muestras evaluadas les arrojan un resultado idóneo con un valor efectivo de 0.93 entretanto Naive Bayes les lanza una precisión del 100% para seleccionar a los candidatos más ideales al puesto solicitado. Pese a que hicieron muchas pruebas no pudo implementarlo dinámicamente, porque era necesario encontrar un ítem general que pueda ser utilizado para evaluación y posición laboral.

Así mismo en el artículo de Rabcan, Vaclavkova, & Blasko (2017) la investigación Selection Of Appropriate Candidates For A Type Position Using C4.5 Decision Tree elaborada en Eslovaquia. Efectuaron un estudio para la selección de grupos de empleados óptimos implicando procesos con una alta complejidad organizativa y necesaria precisión. Centrándose en evaluar los requisitos de los candidatos con respecto

a los intereses de servicio de seguridad de los clientes, los autores trabajaron con Árboles de Decisión Difuso (FDT) con incorporación de algoritmo C4.5., basándose en criterios de empresas de servicios de seguridad y con la ayuda del repositorio Kaggle, con el fin de obtener datos de los candidatos más óptimos. Los autores nos mencionan que este estudio se puede tomar en cuenta como una herramienta para la evaluación de un curriculum permitiendo la posibilidad de utilizar el sistema de toma de decisiones logrando un resultado de precisión del 97.27%. Concluyendo lo propuesto en su investigación se pueden utilizar no solo en los servicios de seguridad sino en otras áreas de la empresa.

Por otra Şenel, Şenel, & Aydemir (2018) en su investigación Use And Comparison of Topis And Electre Methods In Personnel Selection realizada en Turquía nos comenta que la finalidad de un proceso de selección precisa principalmente en la divergencia entre los candidatos a evaluar y poder predecir su utilidad a futuro. Teniendo en cuenta los autores realizaron un estudio a una empresa del sector automotriz de Turquía y en base a las indicaciones de dicha institución utilizaron los métodos de TOPSIS y ELECTRE y comprobar que método brinda mejores resultados.

En el estudio de Şenel, B ,Şenel, M. y Aydemir, G. (2018) se realizaron entrevistas a los gerentes comerciales en la empresa de su caso de estudio donde determinaron los siguientes criterios de contratación que se muestran a continuación:

Tabla 1:

Criterios y pesos de evaluación.

CRITERIOS	PESOS	CRITERIOS	PESOS
C1. Estado de educación	0,09953512	C20. Responsabilidad de la tarea	0,019917012
C2. Sección Graduada	0,086828512	C21. Movilidad	0,2127499
C3. Año de nacimiento	0,01300578	C22. Ira	0,019917012
C4. Género	0,011823437	C23. Comportamiento incontrolado	0,022632968
C5. Estado civil	0.011823437	C24. Autodisciplina	0.023990946
C6. Número de niños	0.007094062	C25. Sociabilidad	0.017201056
C7. Patria	0.007882291	C26. Resistencia a la fuerza	0.022180309
C8. Fumar	0.016158696	C27. Armonía	0.023538287
C9. Licencia de conducir	0.013399895	C28. Atención y concentración	0.040685543
C10. Certificado de trabajo pesado y peligroso	0.012217551	C29. Comparación visual	0.037555886
C11. el trabajo está registrado	0.012217551	C30. Sigue las instrucciones visuales	0.0383383
C12. Antecedentes penales	0.016946926	C31. Tendencia mecánico-técnica	0.035991058

C13. Estatus militar para candidatos masculinos	0.014976353	C32. Instrucciones verbales siguientes	0.0383383
C14. Habilidad del ordenador	0.033849129	C33. Imagen general	0.026194145
C15. Información del curso / certificado	0.02981971	C34. Actitud y comportamiento general	0.03077812
C16. Información sobre idiomas extranjeros	0.020148291	C35. Voluntad	0.032087827
C17. Periodo de experiencia laboral (años)	0.029819471	C36. Escuchar y comprender	0.033397535
C18. Experimente la claridad	0.02127499	C37. Fiabilidad	0.032087827
C19. Diseño	0.023538287		

Nota: Tomado de Senel, Senel, & Aydemir (2018, pág. 6).

En la presente tabla del estudio se muestran los criterios en base a entrevistas realizadas a los gerentes comerciales y gente encargada de reclutar personal; los pesos se definieron en base a los criterios a ponderar en el proceso de reclutamiento, los pesos se distribuyen de tal forma que la suma de los valores del criterio sea 1 punto.

Después de realizar dicha comparación con ambos métodos llegaron a la conclusión que es más apropiado utilizar el método TOPSIS para la selección de personal, porque dicho método da resultados más fiables y exactos que el método Electre.

Según Rahmi (2019) en su investigación el autor para la selección de personal utilizó el método FUZZY ELECTRE basado en los autores Hatami y Tavana , definiéndolo como una técnica que requiere menos

ingresos para los problemas que contienen un extenso número de opciones y criterios donde se puede estudiar fácilmente y no obliga a elaborar comparaciones binarias. En el estudio de Baki R se utilizaron 15 etapas donde se dio solución a la selección del mejor candidato al puesto de ingeniero de ventas donde llega a la conclusión de que se puede utilizar en problemas de la vida real para estudios futuros.

Chen, Cheng, y Hung (2016) proponen su investigación un método para la decisión de contratación de personal, el modelo de apoyo a la decisión de contratación de personal de dos fases con las siglas TPPR DSM. Como primer paso excluye solicitantes de participantes que son inadecuados de acuerdo a los requerimientos que pide la empresa mediante el método TOPSIS, después como segundo paso mide bajo criterios específicos usando el método PROMETHEE luego se presenta un índice de apoyo integral para definir el orden de la clasificación de los mejores candidatos, así mismo el autor toma como ejemplo en seleccionar un gerente de marketing en el extranjero para una empresa de fabricación de computadoras, donde se utiliza una variable al azar para poder generar los datos sobre el desempeño de los participantes con respecto a todo lo que pide el puesto. La distribución de rango de cada candidato seleccionado por el método TPPR DSM también se mide por los métodos TOPSIS y PROMETHEE donde llega a la deducción de que el método propuesto es un método eficaz para la toma de decisiones dirigido a los gerentes para que seleccionen a los candidatos adecuados y así reducir tiempo y costo durante la entrevista, también resalta que los gerentes pueden agregar umbrales para eliminar solicitantes que no son aptos durante la primera fase.

Korkmaz (2019) por otra parte en su estudio utilizó el método TOPSIS porque vio que es muy útil para el proceso de selección de personal para una empresa, comenta que este método dio resultados favorables, fue más rápido a la hora de escoger a la persona indicada, además que incrementó la eficiencia y dio mucha ventaja en reducir

costos. El autor se basó en la técnica de toma de decisiones multicriterio TOPSIS por Hwang y Yoon, creadores del método en sí donde consta de 6 pasos.

Tabla 2:

Pasos del método TOPSIS según Hwang y Yoon.

Nº	Pasos
1	Formar la matriz de decisiones.
2	Formación de la matriz de decisión normalizada.
3	Formación de la matriz de decisión estándar ponderada.
4	Encontrar la solución ideal (A +) y la ideal negativa (A-).
5	Calcular las distancias entre alternativas.
6	Calcular la proximidad relativa a la solución ideal.

Nota: Elaboración propia.

En el estudio de Korkmaz (2019) se trabajó con 20 candidatos que postularon para el puesto de operación logística para una firma en Mersin que opera en el sector logístico de los cuales 11 de los 20 no cumplían con los requisitos por lo que se evaluaron 9 candidatos. Para poder obtener dichos datos, se realizó la primera encuesta con la ayuda de un total de 7 miembros del personal de la empresa conformado por un director regional y trabajadores de recursos humanos del cual las cualidades que solicitaba la empresa se detalla a continuación:

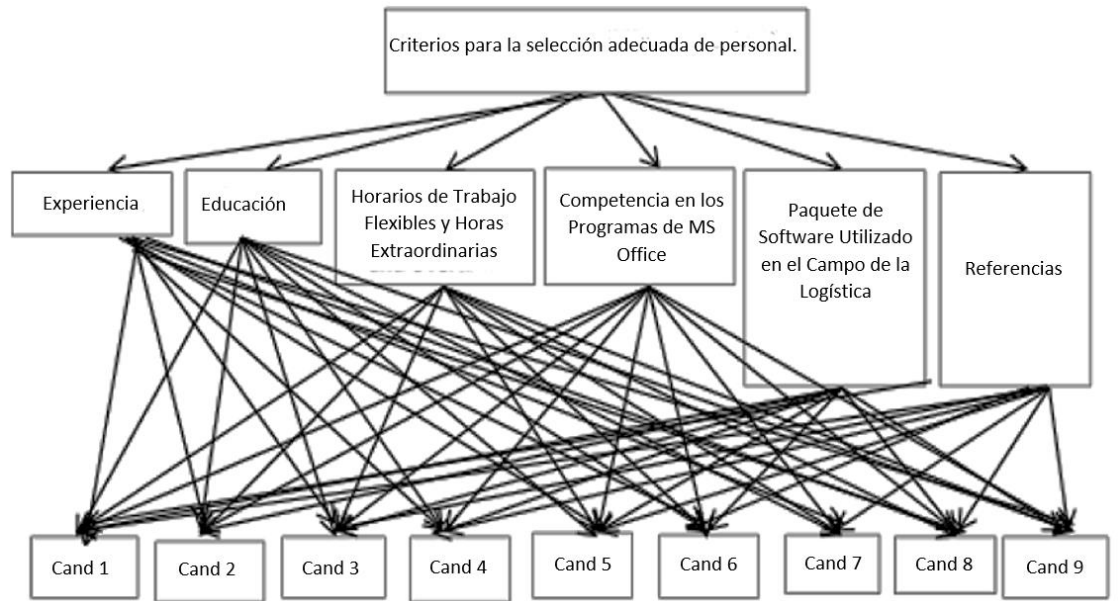


Figura 3: Estructura esquemática que muestra las cualidades buscadas de cada candidato. Fuente: (Korkmaz, 2019).

La imagen nos muestra las cualidades que pidió la empresa para escoger al nuevo candidato, fueron: que tenga experiencia en logística, en educación, que tenga horas de trabajo flexibles y trabajo de horas extraordinarias, buena competencia en programas de MS Office, en paquete de software utilizado en el campo de la logística y por último que tenga cartas de recomendación.

Luego de realizar dichos análisis de acuerdo a los criterios solicitados, la empresa seleccionó los 9 candidatos que fueron objeto de prueba para utilizar el método TOPSIS.

Tabla 3:

Resultados con el método TOPSIS del candidato más idóneo.

Candidatos	Valores C	Top
Candidato 6	0.690349	1
Candidato 9	0.552	2
Candidato 5	0.46533	3
Candidato 3	0.45005	4
Candidato 1	0.446616	5
Candidato 4	0.436578	6
Candidato 8	0.373907	7
Candidato 7	0.334754	8
Candidato 2	0.308872	9

Nota: Tomado de Korkmaz (2019, pág. 12)

En la imagen nos muestra los resultados obtenidos después de aplicar el método TOPSIS a los 9 candidatos y “C” como valor que arrojó TOPSIS y ordenándolos del mejor a menor candidato, dando como candidato más calificado para el puesto al número 6, ubicándose en la primera posición y al candidato número 2 en el último lugar por no cumplir con los requisitos solicitados.

Se dio como conclusión que el proceso de contratación en base al estudio realizado, mostró un incremento del desempeño y satisfacción por parte del personal elegido en el caso de estudio cumpliendo con los requerimientos solicitados por ello el autor sugirió a los investigadores utilizar el método propuesto y más actualizado ya que eso no malgasta los fondos de la empresa y esfuerzos para seleccionar al más adecuado para el puesto.

Waheed, Moshirpour, Moshirpour, Rokne, & Alhaji (2017) en tal sentido en su investigación titulada *Effective Personnel Selection and Team Building Using Intelligent Data Analytics* realizada en Canadá. El agrupamiento de un equipo eficaz es de vital importancia para conllevar una productividad. Sin embargo, si no se tiene esto, no se consigue un mejor uso de tiempo y productividad llevando al fracaso de un proyecto. Por ello realizaron una serie de pasos, primero construyendo una base de datos conteniendo equipos ideales, luego ejecutando el uso de las reglas de minería de datos para escoger conjuntos comunes de los rasgos de personalidad dentro de un equipo y finalizando construyen un algoritmo bayesiano clasificando los rasgos de personalidad para predecir a un candidato idóneo. Por su parte elaboraron un conjunto de datos de prueba de personalidad conteniendo 19,719 personas de 5 a 10 miembros por equipo, más adelante utilizando el clasificador Naive Bayes descubrieron en cuanto a la precisión trabajada era buena, a un tamaño de equipo de 5 a 7 personas disminuyendo a medida que un equipo se extiende. En última instancia concluyeron que, en su investigación aplicando técnicas de minería de datos para analizar los rasgos de personalidad es eficaz y eficiente para la selección de personal y así aumentar el rendimiento de equipo en un proyecto.

(Horat, Kara, & Karakaş, 2019) elaboraron la investigación *Job Pre-Interview System with Artificial Intelligence* realizada en Turquía. Afrontaron la problemática de tener datos de entrevistas realizadas en fase preliminar. Haciendo uso de procesamiento de lenguaje natural Word2vec y uso de antologías. Lograron procesar una data de 30 mil entrevistas realizadas en el sitio web Kariyer.net automatizaron las pre-entrevistas reduciendo pérdidas de tiempo y dinero. Las repuesta obtenidas se transformaron en expresiones discretas haciendo uso de análisis morfológico y procesamiento del lenguaje natural para poder entrenar un modelo de aprendizaje automático.

Chen, cheng, & hung (2016) por una parte en su investigación A Two-Phase Decision-Making Method for Handling Personal Selection Problem, realizada en Taiwan. El proceso de selección de personal lo considera un problema en la toma de decisiones en varios criterios. Por ello propusieron utilizar un marco de aprendizaje automático orientado en árboles de decisiones, Fuzzy y Topsis para ser más competente a la hora de seleccionar al personal. Para el modelo propuesto selecciono a un gerente de marketing en una empresa de fabricación de computadoras donde se seleccionan 8 criterios que incluyen 4 criterios cuantitativos y 4 cualitativos para los métodos Topsis y Promethee en base a ello se recopila información cuantitativa de los solicitantes calculando valor normalizado de los solicitantes para ello la eliminación es del 50% de 10 candidatos seleccionados por otra parte se agregan calificaciones lingüísticas con respecto a cada criterio cualitativo de los candidatos, luego determinar los valores de umbral de cada criterio utilizando el 15% del tiempo de la entrevista a los solicitantes aceptaran la probabilidad del 11.36% y 13.98% por perder a los mejores candidatos que son decididos por ambos métodos. El uso de los métodos puede ahorrar tiempo a la persona encargada a revisar un extenso número de solicitantes permitiendo siendo eficaz eliminando a los solicitantes inadecuados según la información cuantitativa y cualitativa. Y calculándolo automáticamente al empleado adecuado.

Onuralp Yigit & Shourabizadeh (2017) investigación titulada An Approach for Predicting Employee Churn by using data mining elaborada en Turquía. El problema que afronta son los motivos de deserción de empleados. Los autores implementaron técnicas conocidas de clasificación como son: Decision Tree, Naive Bayes, Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbor (KNN) y Random Forest. Utilizados en 1470 registros con 34 teniendo como resultado las siguientes estadísticas mostradas en las tablas.

Tabla 4:

Resultados sin características de selección.

Método	Exactitud	Precisión	Recall	F-measure
Arboles de Decisión	0.765	0.31	0.36	0.33
Naive Bayes	0.791	0.40	0.59	0.48
Regresión logística	0.871	0.74	0.32	0.44
SVM	0.857	0.75	0.17	0.28
KNM	0.844	0.58	0.11	0.18
Random Forest	0.850	0.64	0.14	0.24

Nota: Tomado de Onuralp Yigit & Shourabizadeh (2017, pág. 3).

Tabla 5:

Resultados con características de selección.

método	Exactitud	Precisión	Recall	F-measure
Arboles de Decisión	0.788	0.35	0.37	0.36
Naive Bayes	0.856	0.58	0.58	0.46
Regresión logística	0.871	0.74	0.31	0.44
SVM	0.897	0.98	0.37	0.53
KNN	0.841	0.53	0.12	0.19
Random Forest	0.854	0.63	0.21	0.31

Nota: Tomado de Onuralp Yigit & Shourabizadeh (2017, pág. 4).

Teniendo esos resultados anteriores podemos predecir la probabilidad de que un empleado deje su puesto, darle una mejor valoración y así se quede en su lugar de trabajo.

(Yusuf & Lhaksana, 2020) en su investigación An Automated Interview Grading System in Talent Recruitment using SVM elaborada en Indonesia. La problemática que encontraron fue el alto costo en la necesidad de tener un profesional experto tanto para el proceso y análisis de la entrevista. Para resolver dicha problemática utilizaron los clasificadores Naive Bayes, KNN y SVM. Utilizaron estos métodos para evaluar la aptitud de los participantes en la entrevista mediante un análisis textual, clasificándola palabra por palabra con SVM demostrando que es un mejor clasificador general. Sus muestras de análisis mostraron que con SVM se logró una precisión del 86% mejor que Naive Bayes al 81% y KNN al 79%. Dando como conclusión que el algoritmo SVM tiene mejor rendimiento al utilizar un gran conjunto de datos en comparación con los demás clasificadores.

(Wikan & Kemas, 2020) investigaron en su título Optimization of Decision Tree Algorithm in Text Classification of Job Applicants Using Particle Swarm Optimization realizado en Indonesia. Afrontaron el problema de minimizar los costos a la hora de contratar expertos y así acortar el tiempo para la selección de candidatos. Hicieron uso de un sistema que logró proporcionar recomendaciones de personas aptas utilizando el algoritmo de Árboles de decisión con la ayuda del método Swarm intelligence y Swarm Optimization para mejorar el rendimiento. Comparando varios métodos de árbol de decisión dieron como resultados que aumentó un 7.1% la precisión, siendo la más alta alcanzada fue de 74,3%. Tuvieron como resultado que utilizar Arboles de decisión fue de gran ayuda, además que los hiperparámetros son importantes para determinar mejora el rendimiento de Arboles de decisión.

(Alduayj & Smith, 2019) en su investigación Sentiment Classification and Prediction of Job Interview Performance elaborada de Arabia Saudita. La problemática que tuvieron fue que las empresas tienen

mucha dificultad para la contratación de personas con talentos para un puesto de trabajo, eso complica la tarea del seleccionador incrementando el tiempo para escoger al indicado. Ellos entrenaron muchos modelos de aprendizaje como regresión logística, Naive Bayes, SVM, K nearest neighbour y Árboles de decisión. Para medir el rendimiento de clasificador utilizaron el F1-Measure donde el mayor el resultado lo obtuvo Regresión Logística con 0.814 y el menor fue KNN con 0.619. El principal objetivo de la investigación fue el uso de utilizar varios algoritmos de clasificación para encontrar el mejor resultado de las entrevistas realizadas donde se encontró que Regresión Logística es el mejor para ellos.

(Merchan Blanco & Lamadrid Bedoya, 2019) En su trabajo Diseño de una Metodología de Selección de Personal Obrero en la Empresa Constructora LyM S.A.S. Elaborada En Barranquilla, Colombia. La problemática que afrontaron fue la baja satisfacción laboral en los obreros ya que no se desempeñaban en las áreas establecidas lo que les generaba un nivel de estrés alto e incumplimiento de entregables asignados en el tiempo que les acordaban lo que conllevó a la constructora establecer un proceso de selección de personal. Para buscar una solución decidieron hacer uso de pruebas psicotécnicas tanto el Test de Wartegg como el Test de Valanti. Al hacer uso de las pruebas psicotécnicas se obtuvieron una buena apreciación de las características de los postulantes conllevando a encontrar a un candidato idóneo con virtudes, fácil adaptación en cambios situacionales y teniendo la facultad de poder aprender nuevas cosas. Dentro de la Investigación realizada se puede observar que las herramientas de selección como los Test de Wartegg y Test de Valanti se ajustan a la problemática presentada facilitando una buena evaluación de selección de personal.

(Cardona Mazariegos & Olla de Leon, 2011) En su investigación titulada El eneagrama y su aplicación a procesos de selección de personal a nivel administrativo en Dunbar International realizada en Guatemala. El principal problema que afrontaron es que posicionar a

un candidato idóneo, depende de diversos aspectos propios del mismo participante y lo que demuestre en su comportamiento y adaptación adecuada al puesto. Para poder tener una solución decidieron utilizar test de DISC y test de Eneagrama. En el análisis de los test se evaluaron 15 puestos con el fin de comparar los rasgos de personalidad que se describieron, donde 5 puestos fueron comparados de la misma forma por ambos test pero 4 puestos el test de Disc supero al test de Eneagrama, sin embargo comenta el autor que es ambos test son complementarios para buscar el perfil de personalidad requerida al puesto. Los autores concluyeron que ambos test son útiles, con la diferencia de que los dos tienden a tomar aspectos importantes a la hora de evaluar características de personalidad.

(Padilla Jordán, 2014) Realizó un estudio titulado Relación entre los resultados de la prueba de inventario de personalidad para vendedores (IPV) y el desempeño laboral de un grupo de vendedores que trabajan para una empresa dedicada a comercializar los productos y servicios de telecomunicaciones en Guatemala. Realizada en Guatemala, La Asunción. El autor buscaba conocer si los vendedores de una empresa tendría relación entre los resultados con el test de IPV y el desempeño laboral de los trabajadores , y además conocer si tienen las suficientes aptitudes para comercializar los productos en una empresa de telecomunicaciones. Aplicaron el test de IPV y midieron el desempeño laboral de 40 vendedores en su horario laboral de tres jornadas. Como resultado de este estudio se encontró que tanto el test de IPV como el desempeño laboral no guardan relación una con otra pero si es útil para aplicar a la hora de reclutar y seleccionar una persona para un puesto, especialmente para el área de ventas.

(Torres Flórez, Godoy González, & Gallardo Lichaá, 2019) realizaron en su investigación titulada Procesos de reclutamiento y selección en organizaciones de salud: Caso Villavicencio – Colombia realizada en Colombia. Se realizó un proceso de selección de personal por la necesidad de encontrar a los candidatos, determinando las competencias más importantes. Durante el proceso de selección se aplicó una metodología cuantitativa a través de entrevistas y análisis de documentos, posteriormente llegar a la conclusión de que el proceso de selección que realizan en las organizaciones de salud es pasar por un filtro de reclutamiento utilizando medios externos, hojas de vida, convocatorias anteriores para así obtener los candidatos idóneos sin obviar que el gerente o jefe del área será el último filtro y el encargado de la decisión final.

(Ganga Contreras & Sánchez Álvarez, 2008) en su investigación con nombre Estudio sobre el proceso de reclutamiento y selección de personal en la comuna de Puerto Montt, Región de Los Lagos-Chile realizada en Chile. Los principales problemas es la falta de capacitación en las organizaciones, teniendo personas poco capacitadas en el área provocando grandes pérdidas en la organización, teniendo deslices en los procedimientos y mucha perdida de tiempo en los procesos. En base a la población obtenida de 140 organizaciones que fueron divididas en 9 sectores se comprendió el proceso de reclutamiento de la siguiente manera, desde la forma de reclutamiento tanto interna como externa, pasando por un proceso de entrevista mediante test psicológicos y pruebas psicotécnicas. Por lo tanto es vital que las organizaciones consideren el proceso de selección de personal como algo fundamental considerar la fuerza laboral para lograr los objetivos propuestos por la organización, teniendo en claro las habilidades de la persona que ocupe un puesto en la empresa.

1.3. Teorías Relacionadas al Tema.

1.3.1. Naive Bayes:

Según Yang, An Implementation of Naive Bayes Classifier (2018) nos comenta que a partir del punto de vista de la clasificación, el propósito primordial es hallar el mejor mapeo entre un dato nuevo y un grupo de clasificaciones en un dominio de problema especial. Con el objeto de hacer este mapeo probabilísticamente computable, se hacen varias manipulaciones matemáticas que permite transforma probabilidades conjuntas en las multiplicaciones de probabilidades previas y probabilidades condicionales.

El teorema de Naive Bayes es un algoritmo de clasificación de Aprendizaje Automático más usado por lo simple y rápido que es para la construcción de modelos de predicción. El proceso que utiliza para entrenar los algoritmos es emplear valores verdaderos y así obtener posibles resultados. La fórmula que emplea es la siguiente:

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

1.3.2. Arboles de Decisión:

En el estudio de Yang (2019) nos comenta que un árbol de decisión es un procedimiento que está basado en una secuencia, posee un nodo padre entrelazado a un(os) nodo hijo(s) donde la ruta trazada entre ambos nodos alcanza un resultado definido como booleano.

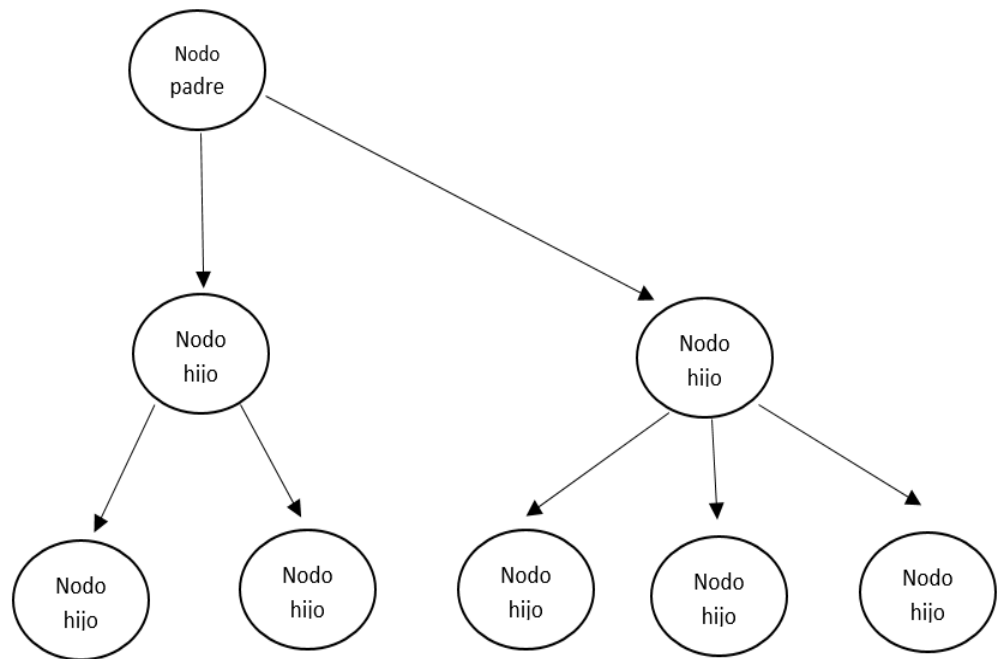


Figura 4: Estructura de árbol de decisión. *Fuente:* Elaboración propia.

Según Feng-Jen (2019) nos dice que para la construcción de un árbol de decisión, se construye a partir de nodos, al primer nodo lo conocen como nodo raíz, este nodo es el que conecta con el resto de los nodos (hoja), donde estos no tienen descendientes, tener en cuenta que cada nodo siempre mantiene una pregunta en secuencia, en tanto cada nodo hoja es asignado a una clase.

Las ventajas que tiene este algoritmo es su fácil comprensión, el uso de exploración de sus datos, menor grado de limpieza es decir los valores que faltan no influye categóricamente en el modelo, no presenta restricción el tipo de cada dato y se basa en un modelo no paramétrico. Las desventajas que pueden presentarse es la pérdida de datos relevantes al categorizar las variables que son del tipo continuas, es inestable en la gestión de cambios porque un dato puede variar categóricamente la estructura total del árbol.

1.3.3. Algoritmo Random Forest.

En su investigación Kristiawan, et al. (2019) afirman que el algoritmo de clasificación Random Forest es uno de los procesos de aprendizaje automático que se utiliza para procesar grandes cantidades de información, este algoritmo funciona uniendo muchos árboles en la data que vamos a entrenar para así producir un elevado nivel de precisión.

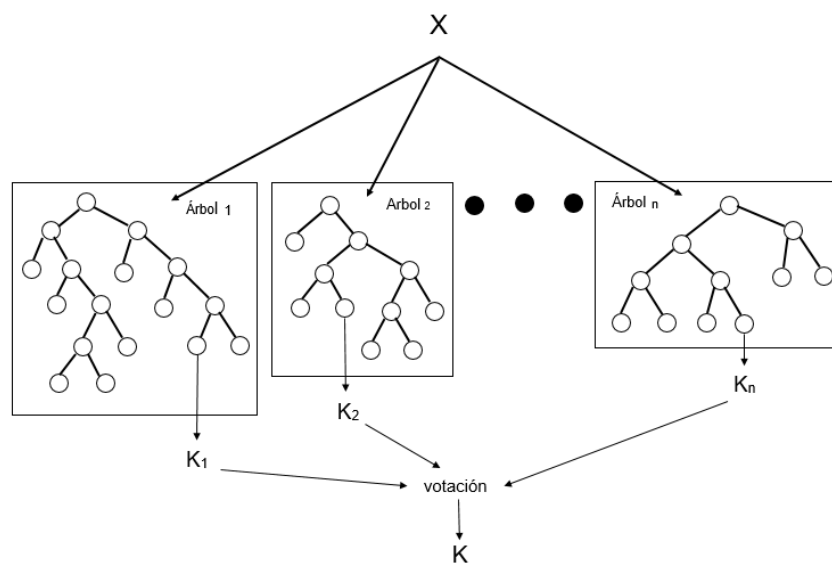


Figura 5: Arquitectura Random Forest. Fuente: (Kristiawan, et al., 2019).

Describiendo la imagen se observa que el número de clasificadores de árboles K es la cantidad de árboles, siendo X el vector de muestra de la entrada y el vector aleatorio KNN es el vector de parámetros de K -total árbol, este último es determinado por KNN y se utiliza para predecir sobre el vector de entrada (X). Este valor está determinado por la votación de todos los árboles.

1.3.4. Algoritmo de Regresión Logística.

El algoritmo de regresión logística según (Kim, et al., 2013) es un tipo de análisis que se emplea para predecir un resultado de una variable dependiente categórica (Mayormente cuando la variable dependiente es binaria) en relación de una o más variables independientes.

Y si hablamos de una definición general, este algoritmo se utiliza para hallar los valores esperados de los parámetros de nuestro modelo, no olvidando que la variable dependiente solo puede tener dos valores posibles.

1.3.5. SVM

Support Vector Machines (SVM) o Máquinas de vectores de soporte es un modelo de clasificación que tiene 2 clases, su modelo básico es definido en el espacio de propiedades del clasificador lineal más grande.

Según los autores CHENG & TONG (2018) el principio de este algoritmo es hallar el hiperplano adecuado en dos tipos de datos positivos y negativos dados, de manera que el hiperplano pueda maximizar ambos datos.

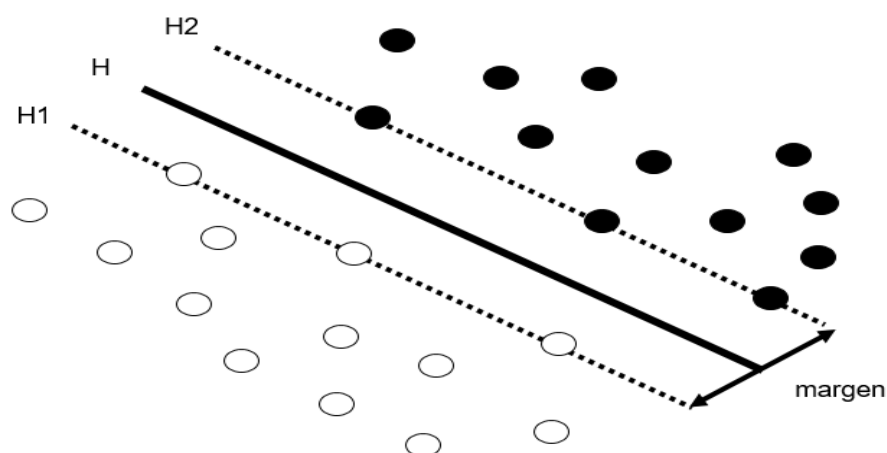


Figura 6: Soporte de división de Hiperplano de máquina vectorial. Fuente: (CHENG & TONG, 2018).

Si se analiza la figura se supone que el Hiperplano H (indicado), H1 y H2 son los dos planos de apoyo que están determinados ambos por los vectores de apoyo que están paralelos al H. El margen es la distancia que hay entre dichos planos de apoyo.

Entonces para dividir mejor ambas clases de datos, es necesario hallar un margen máximo para así garantizar que el Hiperplano divida las dos clases de datos.

1.3.6. KNN:

El algoritmo de KNN o K vecinos más cercanos, según Gironés Roig, Casas Roma, Minguillón Alfonso, & Caihuelas Quiles (2017) es un algoritmo que a partir de una serie de grupo de datos es capaz de clasificar de una manera acertada en una instancia nueva. Está conformado por varios atributos descriptivos y de un solo atributo objetivo que también se le conoce como clase.

Además Gironés Roig, Casas Roma, Minguillón Alfonso, & Caihuelas Quiles (2017) comenta que un dato característico de este algoritmo es que no genera un modelo que sea del mismo aprendizaje con los datos que se ha entrenado, sino que el aprendizaje se da cuando se quiere clasificar una nueva instancia, para clasificar una nueva instancia, primero se calcula la distancia con las instancias de entrenamiento y así poder seleccionar las k instancias que están más cercanas.

1.3.7. Lógica difusa:

Cala Estupiñan (2015) nos dice que la lógica difusa permite responder a diferentes cambios variaciones de la variable a ser utilizada. Si comparamos con la lógica booleana o clásica, ésta permite obtener valores parciales y multi valores. Nos quiere

decir que la lógica booleana es un caso característico dentro de la lógica.

Existen dos tipos de incerteza: la estocástica y la léxica también llamada Imprecisión. La Estocástica se orienta en el acontecimiento del evento, se puede decir como si tuviéramos una pregunta con dos posibles soluciones. En ejemplo sería: ¿dará o no en el blanco? o ¿costará más de lo planeado? La Léxica basada en la ambigüedad de los acontecimientos y en la manera de captación de los eventos por parte de nosotros.

Este tipo de incerteza esta enlazado en la manera como evaluamos conceptos y cómo llegamos a esa conclusión. Un ejemplo de esto sería: ¿qué tan cerca estaremos del blanco? O ¿qué tanto más costara con respecto a lo planeado?

1.3.8. Método de toma de decisiones:

1.3.8.1. Profile Matching:

Por una parte Bachtiar, Pradana, & Yudiari (2019) nos comenta que Profile Matching es un algoritmo enfocado a comparar entre la competencia individual y la competencia esperada para comprender las disimilitudes en la competencia así mismo presenta una alta precisión para precisar la clasificación de candidatos.

Así mismo se puede entender que es un método de toma de decisiones, se asume que existe una variable ideal que permite pronosticar, evaluar y determinar criterios de reflexión. Este método es muy útil, puede apoyar en el proceso de seleccionar un candidato a un puesto idóneo ya que toma decisiones rápidas y confiables.

1.3.8.2. Topsis:

Según Ceballos, Lamata Jiménez, PELTA, & SANCHEZ (2013) el método TOPSIS es muy utilizado gracias a su lógica, porque es racional y entendible, agregando que sus procesos son sencillos y estructurados en un algoritmo, esto permite que una búsqueda de alternativas para cada perspectiva sea con una fórmula matemática sencilla en la que para poder hallar el cálculo se debe de tener en cuenta ítems o valores de los pesos de cada criterio, ya sea que éste sea un costo o utilidad.

1.3.8.3. Smart:

Rahayu, Umar, & Sunardi (2020), en su informe define que, otro de los métodos para la toma de decisiones es SMART, dice que es un método muy simple, además que su comprensión es fácil de entender y no demanda gran conocimiento en el tema de las matemáticas.

Para llevar a cabo este método se definen seis etapas, las cuales son:

Etapas del método SMART:

- 1) Como primer paso se determina los requerimientos a utilizar.
- 2) Se determina el peso de los requerimientos dando un intervalo entre 1 y N.
- 3) Para calcular la normalización se usa la ecuación:

$$N = \frac{w_j}{\sum w_j}$$

Tabla 6:

Descripción de la educación de las variables de Normalización.

Variable	Descripción
N	Normalización
Wj	Valor de peso de los requerimientos
Ewj	Valor de peso total de todos los requerimientos

Nota: Elaboración propia.

- 4) Asigna un parámetro para cada valor de las alternativas de los requerimientos.
- 5) Se determina el valor de utilidad para cada requerimiento, se realiza con la siguiente fórmula:

$$Bi = \frac{Ri - Rmin}{Rmax - Rmin}$$

Tabla 7:

Descripción de la fórmula para determinar el valor de la utilidad.

Variable	Descripción
Bi	Beneficio para cada requerimiento.
Ri	Valor de criterio en i
Rmin	Valor de requerimiento mínimo
Rmax	Valor de requerimiento máximo

Nota: Elaboración propia.

6) Se calcula el valor final con la siguiente ecuación:

$$Vf = Wj.Bi(ai)$$

Tabla 8:

Descripción de las variables para hallar el valor final.

Variable	Descripción
<i>Vf</i>	Valor total de las alternativas del criterio.
<i>B (ai)</i>	Valor total de determinar todas las alternativas del requerimiento.
<i>Wj</i>	Resultado de la normalización del peso del requerimiento.
<i>(ai)</i>	Valor de la utilidad del requerimiento.

Nota: Elaboración propia.

1.3.9. Smote (Synthetic Minority Oversampling Technique):

(Fawcett, 2016) Gracias a smote se creó muchas adaptaciones a distintos algoritmos de aprendizaje, con este método se crea nuevos ejemplos minoritarios insertándolos entre los ya existentes, y por otra parte smote no puede generar ejemplos fuera de una muestra disponible.

Cumple con la función de generar datos artificiales en la muestra minoritaria para así cumplir con la función de obtener un equilibrio en el modelo de aprendizaje automático. Asimismo, smote trabaja calculando la media

ponderara en base a dos datos, dándonos uno nuevo similar a los dos ya existentes.

1.3.10. Métricas Derivadas:

Dentro de las métricas derivadas se trabajó con ciertas variables utilizadas en las fórmulas las cuales se representan a continuación:

Tabla 9:

Leyenda de las variables utilizadas.

VARIABLES		DEFINICIÓN
VP	Verdadero positivo	Son los correctos resultados hechos por el entrenamiento realizado.
VN	Verdadero negativo	Son los resultados positivos pero que el entrenamiento ha clasificado como negativos.
FP	Falsos positivos	Son los correctos resultados clasificados como negativos.
FN	Falsos negativos	Son los incorrectos resultados que ha marcado el entrenamiento como negativos.

Nota: Elaboración propia.

1.3.10.1. Exactitud:

La exactitud es el porcentaje de predicciones correctas que realizó el modelo y se puede entender en la siguiente fórmula:

$$\text{Exactitud} = \frac{\text{Número de predicciones correctas}}{\text{Número total de predicciones}}$$

También se puede interpretar con valores positivos y negativos de la siguiente manera:

$$\text{Exactitud} = \frac{VP + VN}{VP + VN + FP + FN}$$

1.3.10.2. Precisión:

La precisión calcula la cantidad de predicciones correctas que ha realizado el modelo y se interpreta de la siguiente fórmula:

$$\text{Precisión} = \frac{VP}{VP + FP}$$

1.3.10.3. R-call o exhaustividad:

R-call o también llamado exhaustividad busca el tamaño de predicciones correctas que clasificó el algoritmo, su fórmula es:

$$Exhaustividad = \frac{VP}{VP + FN}$$

1.3.10.4. AUC: Área bajo la curva ROC

La curva de ROC se puede interpretar como la exhaustividad frente a la tasa de falsos positivos en diferentes umbrales.

El AUC mide toda la zona del ancho y alto que se encuentra debajo de la curva ROC completa.

El AUC se puede interpretar de tal forma que tiene más probabilidad de que clasifique resultados positivos que negativos aleatoriamente.

1.3.11. Inteligencia Artificial:

Wang, Jin, & Zheng, (2021) nos comenta la inteligencia artificial también la llama como IA como abreviatura, es una rama de la informática que estudia cómo las computadoras intentan aparentar el razonamiento humano, como también la forma de comunicarse de manera no convencional con su entorno.

Definiendo de una forma más simple un sistema inteligente es el intento de imitar las características de la inteligencia humana moldeando a través de una máquina o programa.

1.3.12. Aprendizaje Automático (Machine Learning):

Según Kang & Pecht (2018) nos comenta que, es un conjunto de métodos que nos permiten sustraer información útil para agilizar el desarrollo de técnicas de detección basado en los datos.

1.3.12.1. Tipos de Aprendizaje Automático:

1.3.12.1.1. aprendizaje supervisado:

Así mismo Kang & Pecht (2018) en el aprendizaje supervisado la información que ha sido entrenada a los algoritmos aprendizaje automático incluyen soluciones esperadas donde la clasificación es común en aprendizaje supervisado.

1.3.12.1.2. aprendizaje no supervisado:

Kang & Pecht (2018) el aprendizaje no supervisado a diferencia del aprendizaje supervisado los datos que se manejan en entrenamiento no se encuentra registrados, como se muestra en la figura 7 representación de aprendizaje no supervisado.

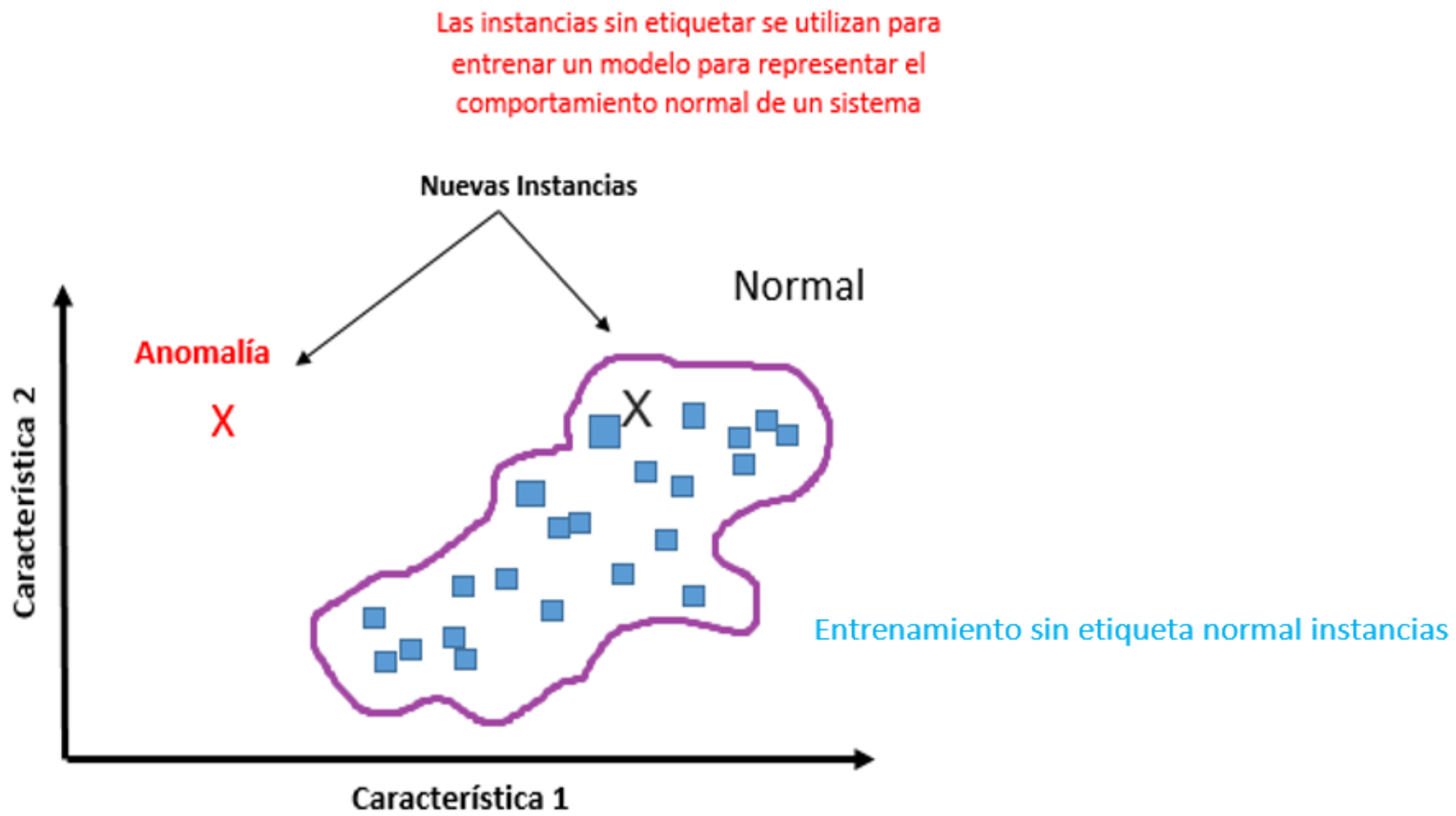


Figura 7: Ejemplo de aprendizaje no supervisado. Fuente: (Kang & Pecht, 2018).

El aprendizaje no supervisado está dirigido en agrupamiento de tareas o también llamados donde su principal función es encontrar grupos parecidos e idénticos.

1.3.12.1.3. aprendizaje semi supervisado

Kang & Pecht (2018) el aprendizaje semi supervisado es un conjunto de métodos y procedimientos que utilizan datos no etiquetados para el aprendizaje, todos estos datos almacenados de entrenamiento comprenden datos sin etiquetar y etiquetados. Así mismo el aprendizaje semi supervisado se puede utilizar para descubrir irregularidades en el sistema.

El aprendizaje semi supervisado es un conjunto de tareas y métodos de aprendizaje que utiliza informaciones no aptas para la capacitación, su conjunto de datos de entrenamiento incluye mucha cantidad de información sin etiquetar y pocos de datos etiquetados. Por ejemplo, detectar irregularidades en un programa es un ejemplo de aprendizaje semi supervisado.

1.3.13. TEST de DISC

El autor (Gil Gaytan & Nuñez Partido , 2017) nos comenta que el test DISC se enfoca en evaluar a la persona de cómo intenta distinguir una situación y hacer frente a ella. A raíz de la respuesta se obtiene estas cuatro emociones principales del individuo de los cuales son dominante, influyente, estable y cumplidor.

El test de Disc es una serie de cuestionarios donde la persona tiene que escoger una alternativa que más lo representa, con el objetivo de dar respuesta al comportamiento de la persona, su modo de juzgar, la manera de influir en la empresa y sociedad, sus emociones, cómo es el actuar bajo presión, entre otros., porque cada persona tiene sus propias características y cualidades.

1.3.14. TEST de Valanti

(Pérez Pérez, 2021) el objetivo del cuestionario de Valores-Antivalores VALANTI cumple la función de identificar sólo los valores importantes a la hora de seleccionar un personal y ver su potencial.

El test de Valanti mide cinco valores esenciales que intenta ver la conducta de la persona frente a una serie de preguntas, intentado medir su juicio crítico ante situaciones que debe elegir dándonos resultados tanto la forma de pensar, sentir, su manera de ser y su conducta frente a su entorno, Por ende este test es utilizado por personas capacitadas para la selección de personal obteniendo sus valores para posteriormente comprobar si se ajusta al puesto a solicitar.

El test nos muestra los cinco valores esenciales que son:

Tabla 10:

Valores del test de Valanti

Valores
Verdad
Rectitud
Paz
Amor
No violencia

Nota: Elaboración propia.

1.3.15. TEST de IPV

ECPA - Les Editions du Centre de Psychologie, (2016) en este libro comenta que el IPV es un instrumento que permite medir al individuo basándose en dimensiones de su personalidad, este test contiene doce dimensiones, donde nueve son de personalidad, uno de Disposición General para la Venta, y dos de factores tanto estilo receptivo y agresivo. Contiene ochenta y siete preguntas sobre su estilo de vida y el mundo laboral, donde la persona escoge la respuesta que más se adecua y de acuerdo a su puntaje que salga en el test donde podemos presenciar la calidad de su carácter y cómo se desarrolla en el mundo de ventas.

El IPV test de personalidad está enfocado a valorar a posibles candidatos a un puesto correspondientes al área de ventas, consta de cuarenta minutos de prueba, centrándose en ver las actitudes esenciales

del individuo en cuanto a su relación con el cliente, así como su disposición para ventas.

Tabla 11:

Valores del test de IPV

Ítems de IPV
Disposición general para la venta.
Comprensión
Adaptabilidad
Control de sí mismo
Tolerancia a la frustración
Compatibilidad
Dominancia
Seguridad
Actividad
Sociabilidad
Receptividad
Agresividad

Nota: Elaboración propia.

1.4. Formulación del Problema.

¿Cuán eficiente es el sistema inteligente para evaluar el perfil laboral de un puesto de gerencia?

1.5. Justificación e Importancia del Estudio.

Nuestra investigación es para ayudar a las empresas a poder elegir de una manera rápida la elección de un candidato a un puesto, Según (Korkmaz, 2019) nos dice que se ha visto casos muy significativos donde el jefe es muy exitoso en su puesto de campo de trabajo, pero al no contar con un buen personal de equipo tiende a fracasar. Para ello se debe de buscar a una persona capaz que tenga iniciativa, trabajo en equipo, capacidad para negociar,

pensamiento crítico y sobre ser un buen líder que pueda manejar a su personal. Para esto se realizará un sistema inteligente validando la información mediante los atributos y aptitudes registradas en el sistema, su función es ingresar los datos de los postulantes, procesando dicha información brindada y arrojando resultados óptimos obteniendo los mejores candidatos.

Esta investigación podrá ayudar para que otros investigadores puedan tener una base relacionada al tema que se está desarrollando.

1.6. Hipótesis.

El sistema inteligente para la evaluación de perfil laboral en el puesto gerencial es eficiente porque presenta una precisión, exhaustividad y exactitud planteadas en las variables de desarrollo de esta tesis.

1.7. Objetivos.

1.7.1. Objetivo General.

Desarrollar un sistema inteligente que permite evaluar eficientemente el perfil laboral para el puesto de gerencia en una empresa.

1.7.2. Objetivo Específico.

- 1)** Entrevistar a personas capacitadas en selección personal.
- 2)** Construir el dataset.
- 3)** Seleccionar el algoritmo de aprendizaje automático a utilizar para desarrollar el sistema inteligente.
- 4)** Elaborar el software utilizando el algoritmo de clasificación seleccionado.
- 5)** Implementar los datos para su procesamiento.

- 6) Efectuar pruebas al software mediante el entrenamiento de datos para medir su eficiencia.

II. MATERIAL Y METODO

2.1. Tipo y Diseño de Investigación.

El tipo de investigación es cuantitativa, debido a que busca escoger al mejor candidato para un puesto de gerencia mediante metodologías que expresarán en valores numéricos.

Así mismo el diseño de la investigación es cuasi experimental porque se tomó los datos y se establecieron parámetros que nos permitieron obtener resultados. Estos datos ya definidos se utilizaron a fin de seleccionar a los mejores candidatos para el puesto que solicita la empresa.

2.2. Población y Muestra.

2.1.1. Población:

La población manipulada en la investigación consta de dieciséis mil doscientos treinta y seis datos de entrevistas realizadas a personas interesadas en el puesto de gerencia.

2.1.2. Muestra:

Se utilizarán el siguiente dataset a partir de las entrevistas realizadas al personal capacitado se obtuvo los 21 atributos que se muestran en la siguiente tabla:

Tabla 12:

Atributos a utilizar para el entrenamiento del sistema.

N°	Atributo
1	Edad
2	Experiencia
3	Verdad
4	Rectitud
5	Paz
6	Amor
7	No violencia
8	Test Disc
9	Disposición general para la venta
10	Comprensión
11	Adaptabilidad
12	Control de sí mismo
13	Tolerancia a la frustración
14	Combatibilidad
15	Dominancia
16	Seguridad
17	Actividad
18	Sociabilidad
19	Receptividad
20	Agresividad
21	Capacitaciones

Nota: Elaboración propia.

Todos estos datos recopilados del dataset se necesitaron para poder entrenar el sistema inteligente donde se obtuvo los mejores candidatos.

2.3. Variables, Operacionalización:

Variables	Indicador	Ítem	Técnica e instrumentos de recolección de datos
Variable Independiente: sistema inteligente	Tiempo de Respuesta	$Tr = Ti - Tf$	Técnica: Observación Instrumento: Electrónico.
	Consumo de Memoria	$Cm = \frac{Cem}{n}$	
	Consumo de CPU	$Ccpu = \frac{Cej}{n}$	
Variable Dependiente: La evaluación de los perfiles por competencia	Precisión	$P = \frac{VP}{FP + VP}$	
	Rcall o exhaustividad	$R = \frac{VP}{FN + VP}$	
	Exactitud	$E = \frac{VN + VP}{VN + FP + VP}$	

2.4. Técnicas e Instrumentos de Recolección de Datos, Validez y Confiabilidad.

2.4.1. Técnicas:

Observación:

Esta técnica se utilizó en el desarrollo de la propuesta de investigación donde se registraron los datos obtenidos, en consiguiente se procedió a analizar y se observó los resultados con el fin de anotar todas las observaciones para poder ver la eficiencia del software.

2.4.2. Instrumentos:

Registro Electrónico:

Este instrumento consiste en registrar toda la información que se obtiene después de desarrollar cada uno de los indicadores, esta información debe ser entendible, clara y concisa. así nos permite precisar y evaluar en cada detalle de dicho indicador para poder saber qué tan bueno y qué tan eficiente es el software.

El registro electrónico se toma como referencia de las investigaciones realizadas por (Onuralp Yigit & Shourabizadeh, 2017) donde los autores plasmaron sus resultados evaluando la matriz de confusión en variables como Exactitud, Precisión y R-call, para medir la eficiencia del software para la evaluación de los perfiles por competencia laboral de un puesto gerencial.

2.5. Procedimiento de Análisis de Datos.

2.5.1. Tiempo de respuesta.

Tiempo que se usa al ejecutar el algoritmo.

$$Tr = \sum_f^i Ti - Tf$$

Donde:

Tabla 13:

Descripción de variables del tiempo de respuesta.

Variable	Descripción
Tr	Tiempo de respuesta
Ti	Tiempo inicial
Tf	Tiempo final

Nota: Elaboración propia.

2.5.2. Consumo de memoria.

Es el consumo de memoria RAM al momento de ejecutar un algoritmo.

$$Cm = \sum_{em}^n \frac{Cem}{n}$$

Tabla 14:

Descripción de variables del consumo de memoria.

Variable	Descripción
Cm	Consumo de memoria
Cem	Consumo en ejecución
n	Total del uso del CPU

Nota: Elaboración propia.

2.5.3. Consumo de CPU

Tiempo que se usa al ejecutar un proceso.

$$C_{cpu} = \frac{C_{ej}}{n}$$

Donde:

Tabla 15:

Descripción de variables de Consumo de CPU.

Variable	Descripción
Ccpu	Consumo del CPU
Cej	Consumo en ejecución
n	Total del uso del CPU

Nota: Elaboración propia.

2.5.4. Exactitud

Es el porcentaje de predicciones correctas ha hecho el software.

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN}$$

Donde:

Tabla 16:

Clasificación de variables del indicador Exactitud.

Variable	Descripción
VN	Verdadero Negativo
VP	Verdadero Positivo
FP	Falso Positivo

Nota: Elaboración propia.

2.5.5. Precisión

Es la cantidad de predicciones correctas que ha hecho el software.

$$Precisión = \frac{VP}{VP + FP}$$

Donde:

Tabla 17:

Clasificación de variables del indicador Precisión.

Variable	Descripción
VP	Verdadero Positivo
FP	Falso Positivo

Nota: Elaboración propia.

2.5.6. R-Call ó Exhaustividad

La exhaustividad o R-Call es la cantidad de verdaderas predicciones que el software ha hecho bien.

$$Exhaustividad = \frac{VP}{VP + FN}$$

Donde:

Tabla 18:

Clasificación de variables del indicador R-Call ó Exhaustividad.

Variable	Descripción
VP	Verdadero Positivo
FN	Falso Negativo

Nota: Elaboración propia.

2.6. Criterios Éticos.

Dentro del presente desarrollo de la investigación se tomó muy en cuenta como criterios éticos el trabajar con las fuentes de información con sus respectivas referencias bibliográficas con la finalidad de eludir el plagio, la información que hemos recopilado fue recolectado dentro del marco legal, sin alterar su contenido, agregar o quitar detalles que infrinjan su confiabilidad. Así como también el criterio de veracidad está muy presente para garantizar que los resultados en la investigación sean correctos.

2.7. Criterios de Rigor Científico.

Se garantizó que con el desarrollo de la investigación se obtuvo resultados que cumplen con los objetivos específicos trazados.

Validez:

El objetivo de este criterio se demostró que la investigación es válida en base a estudios realizados en otras investigaciones, proponiendo un desarrollo de un sistema inteligente para la evaluación de los perfiles por competencia laboral para un puesto de gerencia con algoritmos válidos, se garantizó que el trabajo de investigación fue exacto en determinar resultados.

Fiabilidad:

El proyecto presentó resultados sólidos ya que se utilizaron fórmulas previamente determinadas con el fin de disminuir la similitud en los datos obtenidos y así garantizar la credibilidad en la investigación.

III. RESULTADOS.

3.1. Resultados en Tablas y Figuras.

Lo que se muestra a continuación son los resultados de tiempo de promedio de respuesta, el grado de consumo de memoria, consumo de CPU, además de Exactitud, Precisión y Exhaustividad; estas mediciones son obtenidas en el momento que ejecutamos el algoritmo en el software.

3.1.1. Sistema Inteligente

Para el entrenamiento del algoritmo de clasificación de Naive Bayes se utilizó el dataset creado y así hallar sus variables métricas:

3.1.1.1. Tiempo de Respuesta.

Tabla 19:

Resultados del tiempo de Respuesta del algoritmo de clasificación.

	Tiempo de Respuesta
Entrenamiento 1	9.10
Entrenamiento 2	12.40
Entrenamiento 3	8.26

Nota: Elaboración propia.

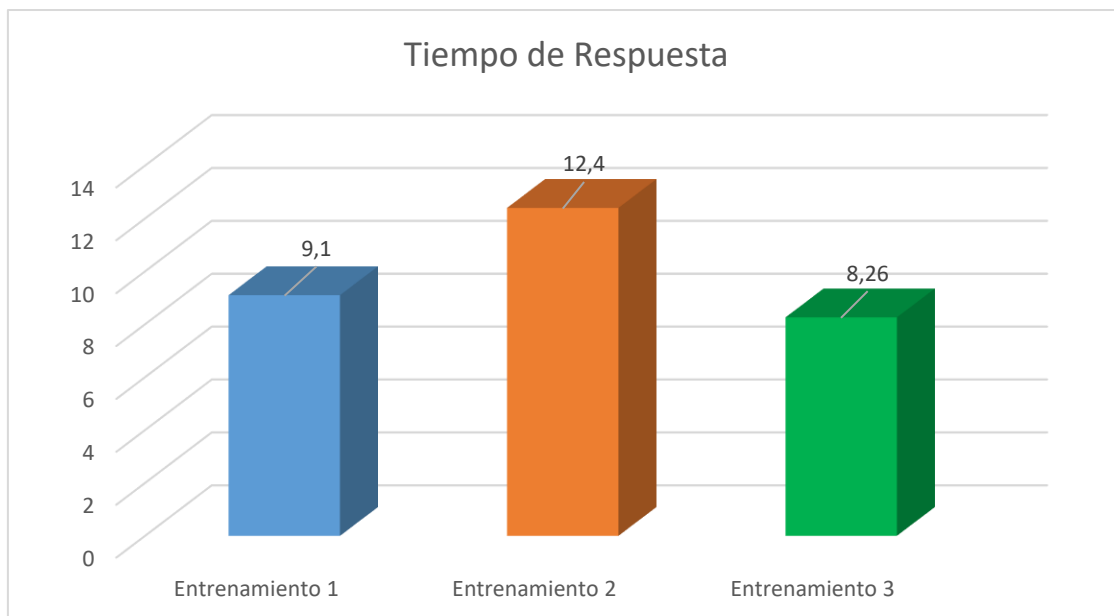


Figura 8: Evaluación de resultados de la precisión en base al entrenamiento realizado. Fuente: Elaboración propia.

En la figura 8, se muestra resultados en referencia tiempo de respuesta, en el momento que ejecutamos el software, obteniendo diferentes resultados en cada entrenamiento con el algoritmo de Naive Bayes, se obtuvo en el primero 9.10 seguido del segundo entrenamiento con 12.40 y en el tercero con 8.26 siendo el mejor el mejor entrenamiento el primero ya que su tiempo de respuesta es mucho mejor que los demás.

3.1.1.2. Consumo de Memoria

Tabla 20:

Resultado de consumo de memoria.

	Consumo de Memoria (mb)
Entrenamiento 1	629.3 mb
Entrenamiento 2	653.14 mb
Entrenamiento 3	550.4 mb

Nota: Elaboración propia.

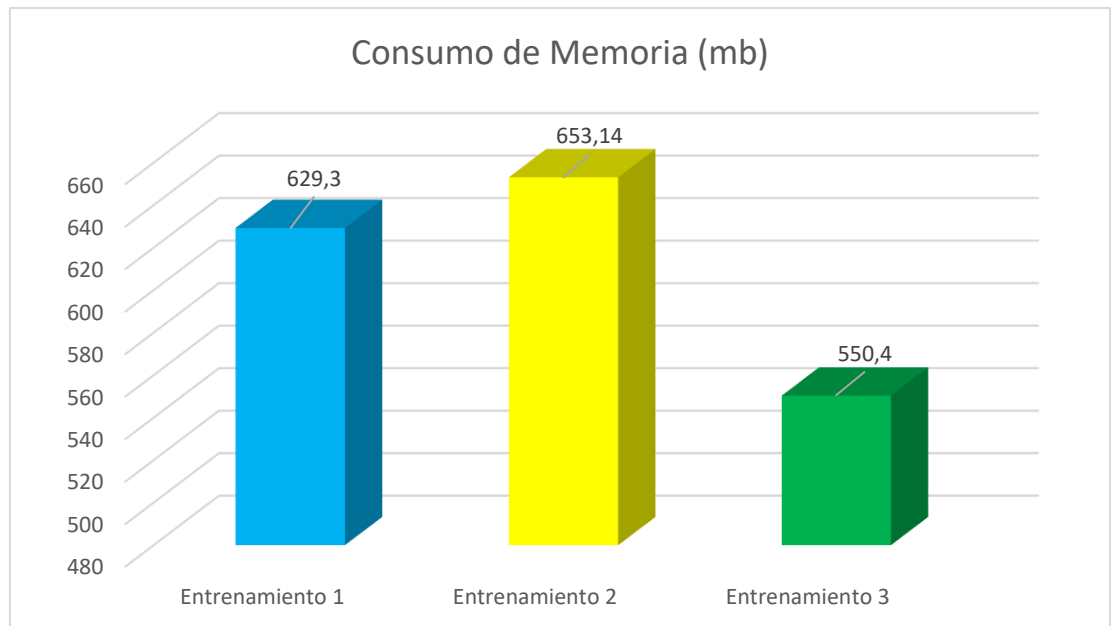


Figura 9: Evaluación de resultados del Consumo de Memoria en base al entrenamiento realizado. Fuente: Elaboración propia.

En la figura 9, se puede ver el consumo de memoria de la RAM que se obtiene al ejecutar el algoritmo varias veces a la hora de realizar el entrenamiento, este consume grandes cantidades de memoria.

Los resultados que obtuvimos para el algoritmo Naive Bayes fueron en el tercer entrenamiento 550,4 MB siendo el que menos consumo obtuvo a diferencia del segundo con 653.14 MB este fue el que más consumió y finalmente en el primero con 629.3 MB de consumo de memoria.

3.1.1.3. Consumo de CPU

Tabla 21:

Resultado de consumo de CPU

	Consumo de CPU
Entrenamiento 1	11.20
Entrenamiento 2	11.15
Entrenamiento 3	11.02

Nota: Elaboración propia.

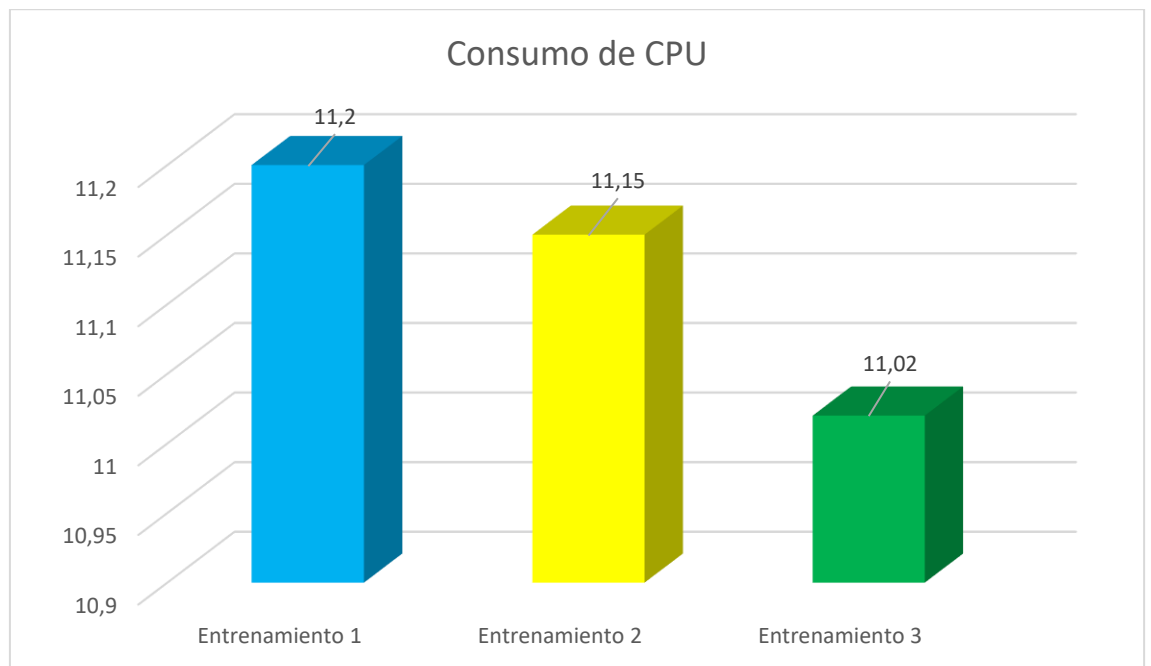


Figura 10: Evaluación de resultados del Consumo de CPU en base al entrenamiento realizado. Fuente: Elaboración propia.

En la figura 10, se muestra resultados en al consumo de CPU, en el momento que ejecutamos el software, obteniendo resultados en cada entrenamiento con el algoritmo de Naive Bayes, se obtuvo en el primero 11.20%, en el segundo 11.15% y en el tercero con 11.02% siendo este el mejor resultado con el más bajo consumo de CPU.

3.1.2. La Evaluación de los Perfiles por Competencia

3.1.2.1. Exactitud

Tabla 22:

Cuadro de entrenamiento de la Exactitud.

	Predicción		Exactitud
	Positivos	Negativos	
1	VP: 1898	FN: 197	0.943
	FP: 19	VN: 1720	
2	VP: 1904	FN: 209	0.942
	FP: 13	VN: 1708	
3	VP: 1899	FN: 190	0.945
	FP: 18	VN: 1727	

Nota: Elaboración propia.

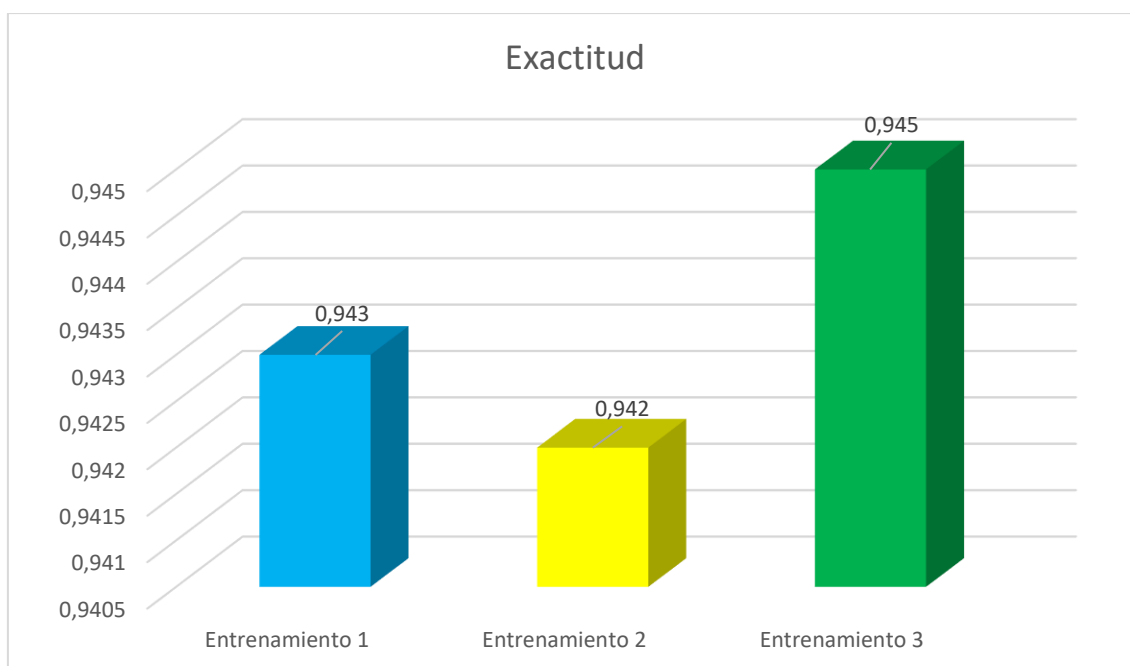


Figura 11: Evaluación de resultados de la Exactitud en base al entrenamiento realizado. Fuente: Elaboración propia.

Como se muestra en la figura 11 donde se ejecutó el algoritmo de clasificación Naive Bayes donde se procedió a hacer el

entrenamiento donde en base a los entrenamientos realizados se escogieron los 3 mejores mostrando el segundo entrenamiento teniendo una exactitud de 0.942 siendo este el más bajo mientras que el entrenamiento que mostró mejor resultados es el entrenamiento 3 obteniendo 0.945.

3.1.2.2. Precisión.

Tabla 23:

Cuadro de entrenamiento de la Precisión

	Matriz	Precisión
	Positivos	
1	VP: 1898 FP: 19	0.989
2	VP: 1904 FP: 13	0.992
3	VP: 1899 FP: 18	0.989

Nota: Elaboración propia.

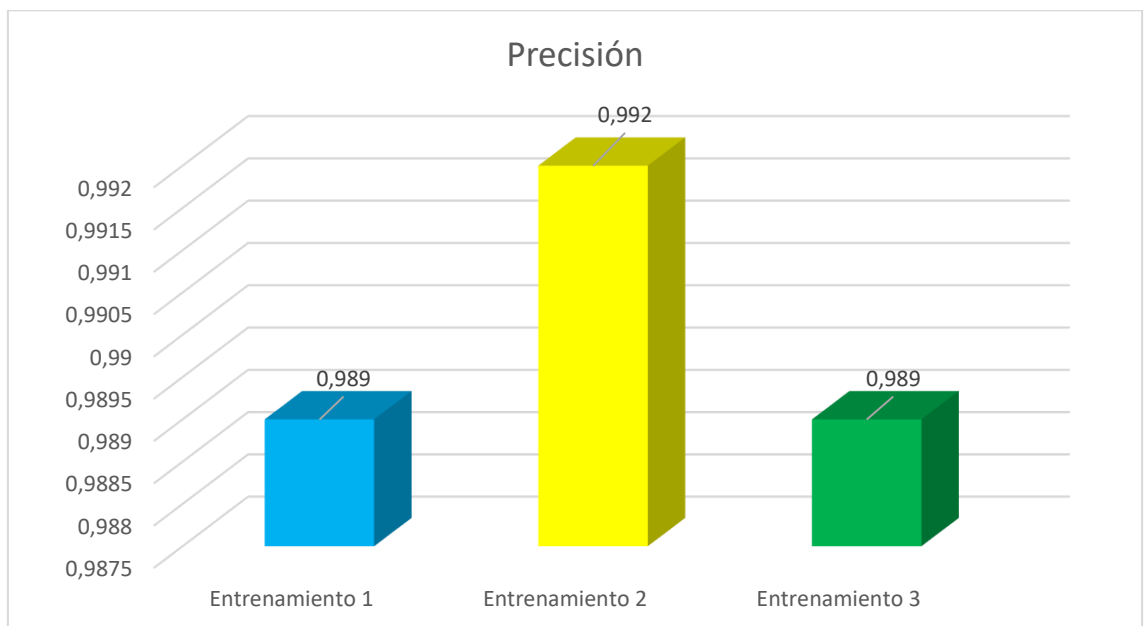


Figura 12: Evaluación de resultados de la Precisión en base al entrenamiento realizado. Fuente: Elaboración propia.

Como se visualiza en la figura 12 que muestra los resultados de ejecución del algoritmo de clasificación Naive Bayes donde se procedió a hacer el entrenamiento donde en base a los entrenamientos realizados se escogieron los mejores mostrando una igualdad en entrenamiento 1 y 3 teniendo una Precisión de 0.989 mientras que el segundo entrenamiento mostró un resultado de 0.992 siendo este el mayor resultado.

3.1.2.3. Rcall o Exhaustividad

Tabla 24:

Cuadro de entrenamiento de la Rcall o Exhaustividad

	Predicción		Rcall o Exhaustividad
	Positivos	Negativos	
1	VP: 1898	FN: 197	0.897
2	VP: 1904	FN: 209	0.890
3	VP: 1899	FN: 190	0.900

Nota: Elaboración propia.

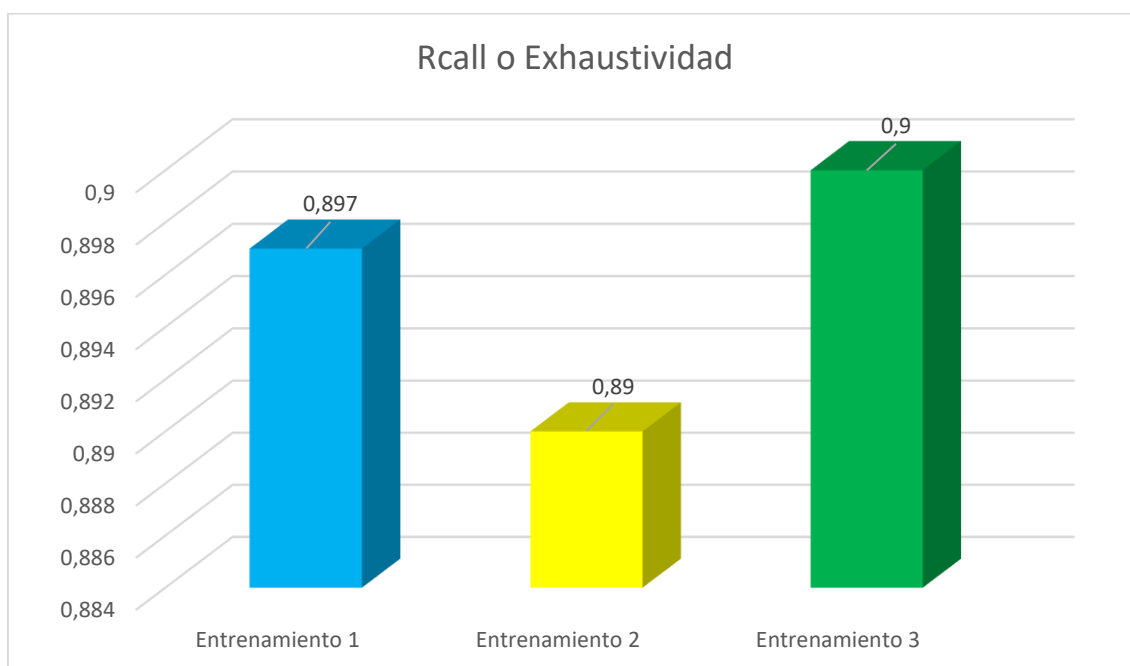


Figura 13: Evaluación de resultados de la Exhaustividad en base al entrenamiento realizado. Fuente: Elaboración propia

Como se visualiza en la figura 13 que muestra los resultados de exhaustividad de los entrenamientos realizados se escogieron los mejores mostrando un alto porcentaje en el tercer entrenamiento teniendo una exhaustividad de 0.900 mientras que el primer entrenamiento mostró un resultado de 0.897 y el segundo de 0.890 siendo este el menor resultado de la exhaustividad.

3.2. Discusión de resultados.

En la presente tesis se realizó un sistema inteligente basándose en un algoritmo de clasificación, donde conjuntamente de investigaciones previas nos comentan que se realizó una serie de análisis, en las que afirman que en el mundo empresarial las dificultades a la hora de escoger un candidato a un puesto de gerencia es un proceso tedioso, donde la persona encargada en realizar todo ese proceso tiene un trabajo muy abrumado y de igual forma que demanda mucho tiempo y dinero. Con base a ello se planteó la hipótesis en la que se desarrolla un software en esta investigación. Conjuntamente se estableció el diseño que se utilizó para el análisis de la información, fue cuasi experimental ya que se determinaron parámetros para la obtención de los resultados. Así mismo después de un análisis exhaustivo de investigaciones previas, se obtuvieron cinco algoritmos de clasificación que resaltaron para el desarrollo del sistema inteligente donde en base a la evaluación hecha en la exactitud de los algoritmos, se optó por el mejor siendo así el de Naive Bayes quien obtuvo el resultado adecuado, tanto en tiempo de ejecución y exactitud así mismo haciendo fácil integración en la etapas del desarrollo del software y teniendo la capacidad de integrar las variables de operalización a la hora de clasificar los datos, entre otros.

Utilizando la ley de Pareto que nos menciona que se escoge un 80% de los datos para entrenamiento y 20% para prueba, teniendo

16236 al haber un desbalance en el las personas aptas al puesto se aplicó una técnica de sobre muestreo donde se obtuvo 19170 de los cuales 3834 datos son de prueba y 15336 para entrenamiento.

Rabcan, Vaclavkova, & Blasko (2017) en la investigación denominada "Selection of appropriate candidates for a type position using C4.5 decision tree" se centraron en el algoritmo de Árboles de Decisión teniendo un resultado de exactitud de 97.27% de esta manera aprobaron que el algoritmo de Árboles de decisión es preciso para la selección de personal.

En base a los resultados que pudieron obtener dichos autores podemos considerar que dicha investigación es buena pero en comparación con la nuestra teniendo como una exactitud superior a la de su investigación, dando por afirmada que la nuestra tiene resultados mas precisos.

Así mismo se puede mostrar en las variables independientes que tanto el tiempo de ejecución siendo de 12.40, consumo de memoria de 653.14 Mb y el consumo de CPU siendo de 11.15% del segundo entrenamiento dando como resultado una exactitud superior al resto de entrenamientos, con una precisión estable y una exhaustividad de 0.890 en consecuencia se puede afirmar que es el mejor resultado obtenido comparando con la investigación de Rabcan, Vaclavkova, & Blasko (2017) es mejor siendo así que considerando los ítems del dataset del desempeño laboral en una empresa se puede obtener si las personas son aptas o no a un puesto gerencial.

3.3. Aporte práctico.

DESARROLLO DE UN SISTEMA INTELIGENTE PARA LA EVALUACIÓN DE LOS PERFILES POR COMPETENCIA LABORAL DE UN PUESTO GERENCIAL

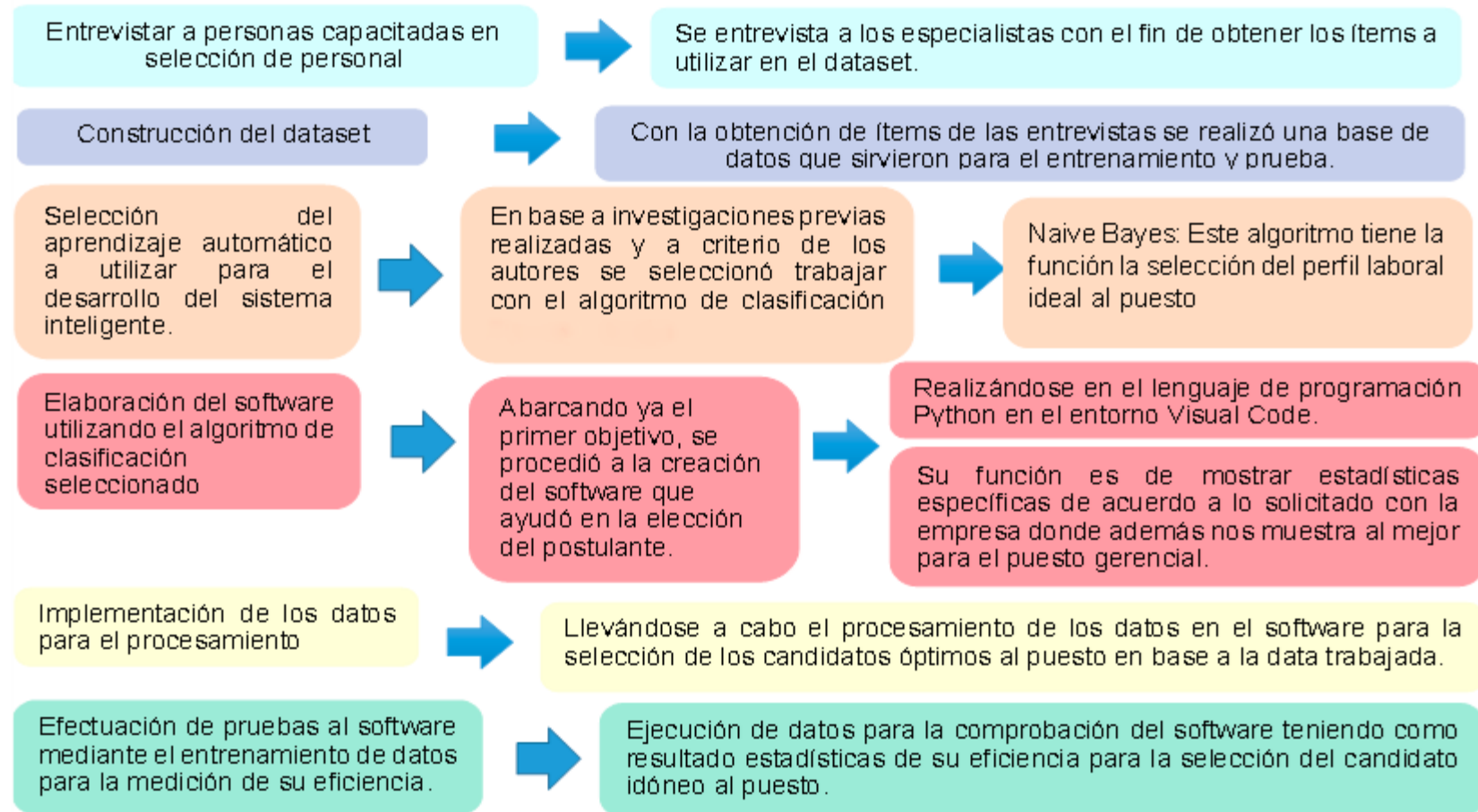


Figura 14: Diagrama de objetivos. Fuente: Elaboración propia

Para poder seleccionar los atributos del dataset se realizaron entrevistas a tres personas capacitadas donde se puede encontrar información en el anexo 03 en selección de personal, se les realizó una serie de preguntas basadas en cómo es el proceso de selección, donde por opinión unánime se seleccionó el puesto de Gerente General en el área de ventas siendo éste el más solicitado por las empresas en base a la experiencia laboral de las capacitadoras, de la misma manera cada persona entrevistada brindó su punto de vista de qué atributos utiliza y cuáles son los más relevantes a la hora de seleccionar al mejor candidato.

Tabla 25:

Tabla de atributo de los especialistas.

ESPECIALISTA	ATRIBUTOS	OBSERVACIÓN
Nereyda Farro Vargas	Test Psicológicas	Esta persona capacitada se centra en utilizar test que van de la mano con la personalidad del candidato.
	Test de fatiga (nivel de estrés)	
	Test de ansiedad	
	Edad	
	Experiencia Laboral	
Melisa Rojas	Test de Valanti	La presente especialista tiende a evaluar de forma más metodológica haciendo referencia a más test psicológicos
	Test de Personalidad Disc	
	Test de IPV	
	Test Nego	
	Test de Barthel	
	Liderazgo	
	Experiencia Laboral	
	Edad	
	Tolerancia	
	Conflictos	
Productividad		
Lucero Castillo	Filtro CV	Esta entrevistadora nos resaltaba todo lo referente a la experiencia laboral, así
	Pruebas Psicológicas	
	Pruebas Psicométricas	
	Edad	

Experiencia Laboral	como también a las
Test de IPV	capacitaciones
Capacitaciones	obtenidas de la persona
Test de Personalidad Disc	a evaluar

Nota: Elaboración propia.

Por consiguiente, estos ítems son esenciales a la hora de entrevistar porque brindan una información tanto objetiva como clara de cómo es la persona postulante al cargo, en aptitud, valores, en el ámbito psicológico y su desempeño profesional.

Por ende, se escogieron los atributos más importantes ajustándolos de manera cualitativa y cuantitativa necesarios para el entrenamiento y prueba del algoritmo de clasificación, escogiéndose los siguientes:

Tabla 26:

Atributos del dataset

Nombre del atributo	¿Qué representa?	Posibles valores	Descripción del posible valor	Tipo de variable
edad	El atributo representa la edad de la persona	33, 34, 35, 36, 37, ..., 80	La edad de la persona	Continua
experiencia	El atributo representa la experiencia laboral de la persona	3, 4, 5, ...	Experiencia laboral de la persona en meses	Continua
verdad	Item que representa los valores	0.- Alto 1.- Bajo 2.- Muy alto	Valores intelectuales	Discreta

	del test de valanti	3.- Muy bajo 4.- Promedio		
rectitud	Item que representa los valores del test de valanti	5.- Promedio Alto 6.- Promedio Bajo	Valores de acción	Discreta
Paz	Item que representa los valores del test de valanti		Valores emocionales	Discreta
amor	Item que representa los valores del test de valanti		Valores afectivos	Discreta
No_violencia	Item que representa los valores del test de valanti		Valores espirituales	Discreta
test_Disc	Representa el resultado del test Disc	- Correcto => 0 - Dominante => 1 - Influyente => 2 - Sensato => 3	Analiza cuál es la personalidad de una persona y cómo se comporta ante determinadas situaciones cotidianas y profesionales.	Discreta

disposicion_general_para_la_venta			Atributo que mide la disposición para la venta	Discreta
Comprension.			Mide el nivel de comprensión de la persona	Discreta
Adaptabilidad			Mide el nivel de adaptabilidad de la persona	Discreta
Control_de_si_mismo.			Mide el control de sí mismo de la persona	Discreta
Tolerancia_a_la_frustracion	Ítems del Test IPV que permite medir las habilidades comerciales de la persona	- Alto => 0 - Bajo => 1 - Medio => 2	Mide la Tolerancia a la frustración	Discreta
Compatibilidad			Mide la Compatibilidad de la persona en el puesto	Discreta
Dominancia			Mide la Dominancia de la persona	Discreta
Seguridad			Mide la Seguridad de la persona	Discreta
Actividad			Mide la actividad de la persona	Discreta
Sociabilidad			Mide la sociabilidad de la persona	Discreta

Receptividad.			Mide la Receptividad de la persona	Discreta
Agresividad.			Mide la agresividad de la persona	Discreta
capacitaciones	Representa las capacitaciones de la persona	1, 2, 3, 4 ...	Muestra las capacitaciones que tuvo la persona	Continua
Clasificador	Variable que representa si la persona es apta o no apta al puesto	- Apto => 0 - No apto => 1	Variable que permite conocer si la persona es apta o no apta al puesto de trabajo	Discreta

Nota: Elaboración propia.

Para el proceso de llenado del dataset teniendo los atributos escogidos anteriormente se realizó un proceso de entrevista a personas interesadas al puesto de Gerente General en el área de Ventas de la mano con los especialistas para corroborar los resultados, así mismo dichas entrevistas se realizaron de manera presencial y así como también virtual obteniendo 6651 aptas y 9585 no aptas sumando 16236 datos al dataset como se muestra en la siguiente imagen.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	edad	experiencia	verdad	rectitud	paz	amor	no_violencia	Test_Disc	Disposicion	comprensor	adaptabilida	Control_de	Tolerancia_a	Combativida	Dominancia	Seguridad	Activi
2	41	12	Promedio Ba	Bajo	Muy bajo	Muy bajo	Bajo	Dominante	medio	bajo	bajo	medio	bajo	bajo	medio	bajo	alto
3	35	3	Alto	Muy alto	Promedio	Bajo	Alto	Sensato	alto	medio	alto	alto	medio	medio	bajo	medio	bajo
4	83	9	Muy bajo	Muy bajo	Muy bajo	Muy bajo	Promedio Ba	Influyente	medio	bajo	bajo	bajo	bajo	alto	alto	bajo	alto
5	44	11	Bajo	Bajo	Muy bajo	Muy bajo	Bajo	Influyente	bajo	bajo	bajo	bajo	bajo	alto	bajo	bajo	alto
6	77	15	Muy bajo	Muy bajo	Muy bajo	Muy bajo	Promedio Ba	Dominante	bajo	bajo	bajo	medio	bajo	bajo	alto	bajo	alto
7	65	8	Promedio	Promedio Al	Promedio Al	Muy alto	Promedio Al	Correcto	alto	medio	medio	alto	medio	medio	bajo	alto	bajo
8	59	14	Promedio Ba	Muy bajo	Muy bajo	Muy bajo	Muy alto	Dominante	bajo	bajo	bajo	medio	bajo	alto	bajo	bajo	alto
9	46	6	Muy bajo	Bajo	Muy bajo	Muy bajo	Bajo	Dominante	medio	bajo	bajo	bajo	bajo	bajo	alto	bajo	alto
10	35	2	Promedio	Promedio Ba	Promedio Al	Promedio Al	Promedio Al	Sensato	alto	medio	alto	alto	alto	medio	medio	alto	bajo
11	41	7	Alto	Promedio Al	Promedio Ba	Promedio Ba	Muy alto	Sensato	alto	medio	medio	alto	alto	medio	medio	medio	bajo
12	43	13	Promedio	Promedio	Promedio	Bajo	Muy alto	Sensato	alto	alto	alto	alto	medio	medio	medio	alto	bajo
13	77	14	Muy alto	Promedio	Muy alto	Promedio Al	Muy alto	Correcto	alto	alto	medio	alto	alto	medio	bajo	alto	bajo
14	70	4	Muy bajo	Muy bajo	Muy bajo	Muy bajo	Muy bajo	Influyente	bajo	bajo	bajo	bajo	bajo	bajo	alto	bajo	alto
15	46	6	Muy bajo	Bajo	Muy bajo	Muy bajo	Bajo	Influyente	medio	bajo	bajo	bajo	bajo	bajo	alto	bajo	alto
16	81	10	Promedio Ba	Bajo	Muy bajo	Muy bajo	Bajo	Dominante	medio	bajo	bajo	bajo	bajo	bajo	bajo	bajo	alto
17	82	9	Promedio Ba	Bajo	Muy bajo	Muy bajo	Promedio Ba	Influyente	bajo	bajo	bajo	bajo	bajo	bajo	bajo	bajo	alto
18	83	4	Promedio Al	Promedio	Promedio	Muy alto	Alto	Sensato	alto	medio	alto	alto	alto	medio	bajo	medio	medio
19	33	1	Muy alto	Promedio Al	Muy alto	Promedio	Alto	Correcto	alto	medio	alto	alto	alto	medio	medio	alto	bajo
20	46	5	Promedio	Promedio Al	Bajo	Alto	Alto	Sensato	alto	alto	medio	alto	medio	medio	medio	alto	medio
21	56	5	Promedio Ba	Muy bajo	Muy bajo	Muy bajo	Muy bajo	Dominante	medio	bajo	bajo	medio	bajo	bajo	medio	bajo	alto
22	69	10	Muy bajo	Bajo	Muy bajo	Muy bajo	Promedio Ba	Dominante	medio	baio	baio	baio	baio	baio	alto	baio	alto

Figura 15: Estructura del dataset. Fuente: Elaboración propia

Para la selección del algoritmo de aprendizaje automático a usar en la presente investigación, se basó en la lectura de las investigaciones de autores que tienen trabajos similares previamente mencionadas en el apartado de investigaciones relacionadas al tema. Así mismo se evaluaron cinco algoritmos de clasificación:

Entre los algoritmos escogidos se encuentra “Naive Bayes” como un algoritmo de clasificación, basándose en el teorema de Bayes, donde se utiliza para calcular un caso que nos proporciona una manera más sencilla de construir modelos con un comportamiento eficiente. Así como también se eligió “Regresión Logística” como un Algoritmo Supervisado, siendo útil para problemas donde la variable adopta valores en un conjunto finito, además cuando se quiere conocer si una variables es dependiente cualitativa o más variables explicativas independientes, además “Random Forest”, definiéndose como una variación del proceso de Bagging, donde hace inspección y así consiguiendo mejores resultados, gracias a que adecúa los árboles generados en el proceso, también “KNN” que es un algoritmo que su aprendizaje se da cuando se quiere

clasificar una nueva instancia y por último "SVM" que su función del algoritmo es hallar el hiperplano adecuado en dos tipos de datos positivos y negativos dados.

Para escoger el algoritmo que realice la predicción del mejor candidato al puesto gerencial, se basó en el uso de los algoritmos antes mencionados, fueron seleccionados por su uso en trabajos previos donde se evaluó el total de uso de CPU, el total de tiempo de ejecución y así mismo la exactitud.

Se utilizó la métrica derivada de "Exactitud", porque se enfoca en medir el porcentaje de la cantidad de predicciones que realizó correctamente el algoritmo.

Tabla 27: *Tabla comparativa de algoritmos.*

Algoritmo	Total, uso de CPU	Total, tiempo de ejecución (en segundos)	Exactitud (Al 80% de datos de entrenamiento y 20% de validación)
RANDOM FOREST	14.5%	4.17	1.0
NAIVE BAYES	11.02%	2.21	0.97
REGRESIÓN LOGÍSTICA	13.3%	8.26	0.88
SVM	17.4%	13.0	0.90
KNN	20.7%	15.53	0.94

Nota: Elaboración propia.

Después de haber realizado un cuadro comparativo donde se colocaron la mejor exactitud, tiempo de ejecución y uso de CPU de cada algoritmo, se visualizó y se escogió el algoritmo de Naive Bayes porque tiene mejor exactitud en comparación con los demás, no obviando que tiene un total de uso óptimo para el uso del software.

Del mismo modo el algoritmo de clasificación tiene las siguientes características acertadas para su desarrollo:

- a) Facilita la integración entre las etapas de desarrollo.
- b) Integra muchas variables a la hora de clasificar datos.
- c) Trabaja de forma óptima cuando se trabaja con variables de entrada categóricas comparada con variables numéricas.
- d) Ordenamiento de variables y mejor selección.
- e) Permite realizar las decisiones óptimas con base a la información existente y a las mejores suposiciones.
- f) Su estructura permite analizar las alternativas, los eventos, las probabilidades y los resultados.
- g) Permite reaccionar a cambios continuos de las variables.

Una vez escogido el algoritmo se plasmó en las etapas de desarrollo del sistema definidas a continuación.

DESARROLLO DE FASES DEL PROYECTO

I. Determinación de Requerimientos del Sistema.

El software se desarrollará y se podrá utilizar por un especialista en selección de personal así mismo el usuario tendrá la opción de:

1.1. Requerimientos Funcionales:

Tabla 28:

Lista de requerimientos funcionales

ID	Requerimiento	Descripción
RF1	El sistema tiene que tener los atributos necesarios para evaluar.	El usuario podrá visualizar los atributos para poder evaluar
RF2	El sistema permitirá analizar y procesar datos para conocer si la persona es apta o no al puesto de gerencia.	El usuario obtendrá un resultado final después de que el sistema evalúe dicha información en un corto periodo de tiempo visualizando

	quienes son las personas idóneas al puesto.
RF3 El sistema debe de tener un apartado a las personas que son aptas y no aptas.	El usuario podrá visualizar en un cuadro de texto donde se verán los resultados y se conocerá quién es apto o no.
RF4 El sistema tiene que tener la opción de ingresar la lista de los postulantes.	El usuario tendrá la opción de ingresar la lista de personas que postulan al cargo.

Nota: Elaboración propia.

Especificación de Requerimientos

Tabla 29:

Descripción del requerimiento funcional 1

Requisito funcional 01: El sistema tiene que tener los atributos necesarios para evaluar.	
1	<p>Entradas:</p> <ul style="list-style-type: none"> - El sistema tendrá la opción de que el usuario seleccione los atributos necesarios para la evaluación de los datos registrados. <hr/> <p>Procesamiento:</p> <ul style="list-style-type: none"> - Cargará los datos que contiene los atributos seleccionados. <hr/> <p>Salidas:</p> <ul style="list-style-type: none"> - Mostrará en una lista las personas que son aptas y no aptas evaluadas de acuerdo al atributo que se asignó.

Nota: Elaboración propia.

Tabla 30:

Descripción del requerimiento funcional 2

Requisito funcional 02: El sistema permitirá analizar y procesar datos para conocer si la persona es apta o no al puesto de gerencia.

2	Entradas: <ul style="list-style-type: none">- Con la finalidad de encontrar el mejor algoritmo de clasificación, se entrenarán y clasificarán cinco algoritmos que será utilizado junto con el dataset realizado. <hr/> Procesamiento: <ul style="list-style-type: none">- El sistema utilizara el mejor entrenamiento para clasificar los datos de los candidatos. <hr/> Salidas: <ul style="list-style-type: none">- los resultados serán mostrados y guardados en la base de datos.
----------	---

Nota: Elaboración propia.

Tabla 31:

Descripción del requerimiento funcional 3

Requisito funcional 03: El sistema debe de tener un apartado a las personas que son aptas y no aptas

3	Entradas: <ul style="list-style-type: none">- Son todos los datos procesados de las evaluaciones a cada postulante realizadas. <hr/> Procesamiento: <ul style="list-style-type: none">- Los datos se mostrarán en un apartado de la interfaz. <hr/> Salidas: <ul style="list-style-type: none">- Los resultados se visualizarán listando los datos guardados que se encuentran en la base de datos.
----------	--

Nota: Elaboración propia.

Tabla 32:

Descripción del requerimiento funcional 4

Requisito funcional 04: El sistema tiene que tener la opción de ingresar la lista de los postulantes.

4	Entradas: <ul style="list-style-type: none">- Registrar el Excel con los datos de los postulantes <hr/> Procesamiento: <ul style="list-style-type: none">- Cargar el Excel registrado de los postulantes <hr/> Salidas: <ul style="list-style-type: none">- Refleja en un listado los resultados de los postulantes
----------	--

Nota: Elaboración propia.

Diagrama de Procesos de Negocio – DPN

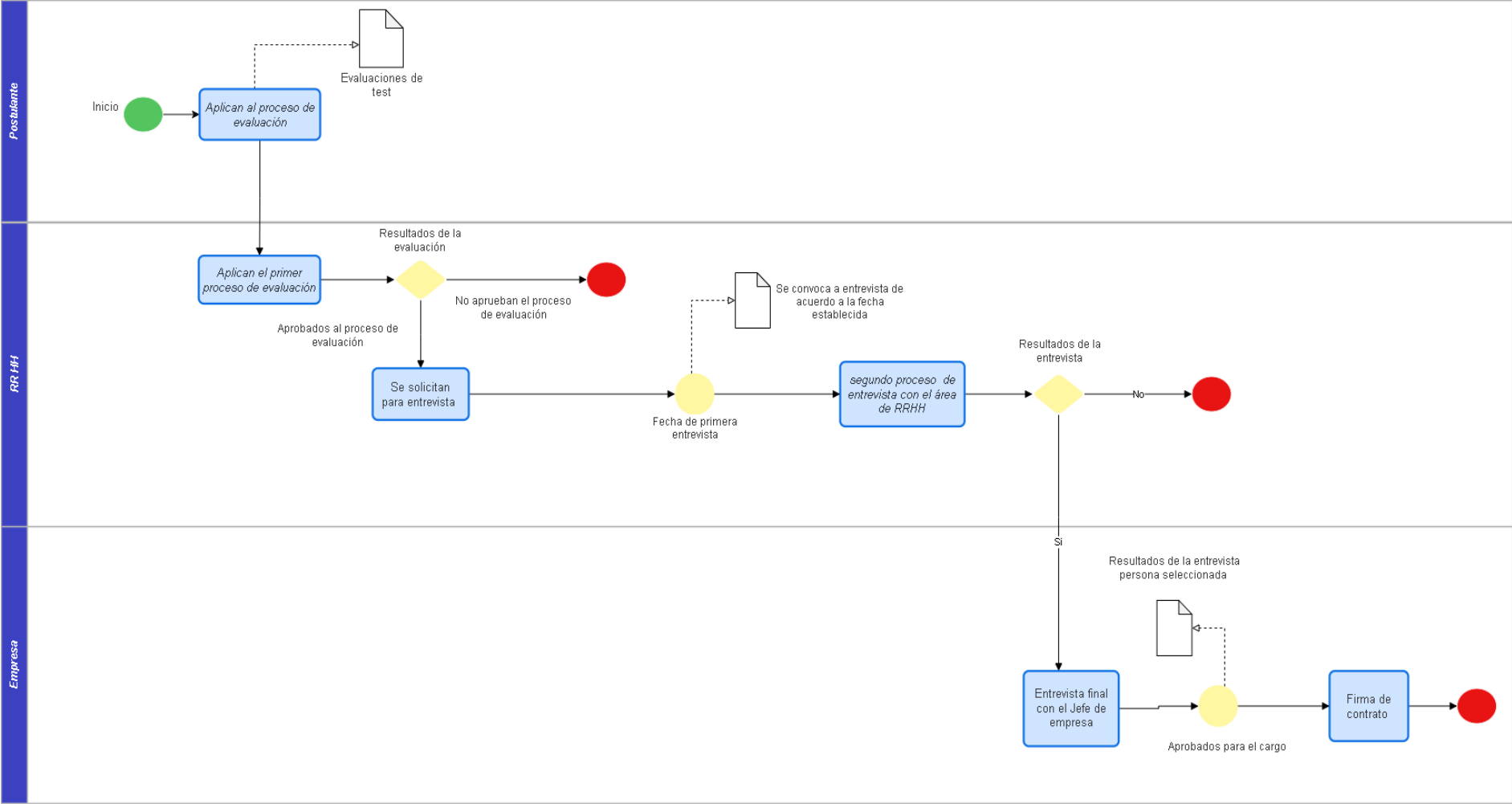


Figura 16: Diagrama de Procesos de Negocios. Fuente: Elaboración propia.

1.1.2. Modelamiento del Negocio

1.1.2.1. Modelo de Casos de Uso del Negocio (MCUN).

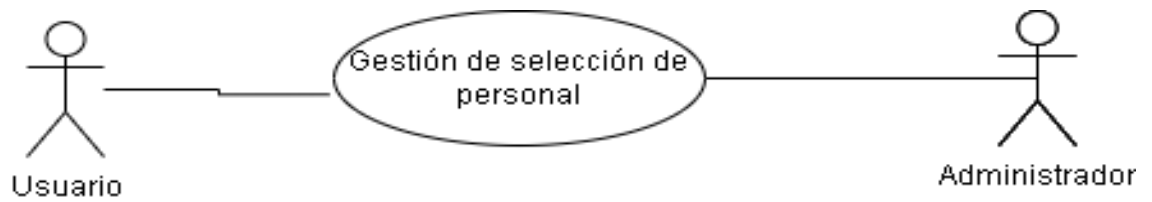


Figura 17: Modelo de caso de uso del negocio. Fuente: Elaboración propia

1.1.2.2. Modelo de Objeto del Negocio (MON).

- MON – GESTION DE SELECCIÓN DE PERSONAL

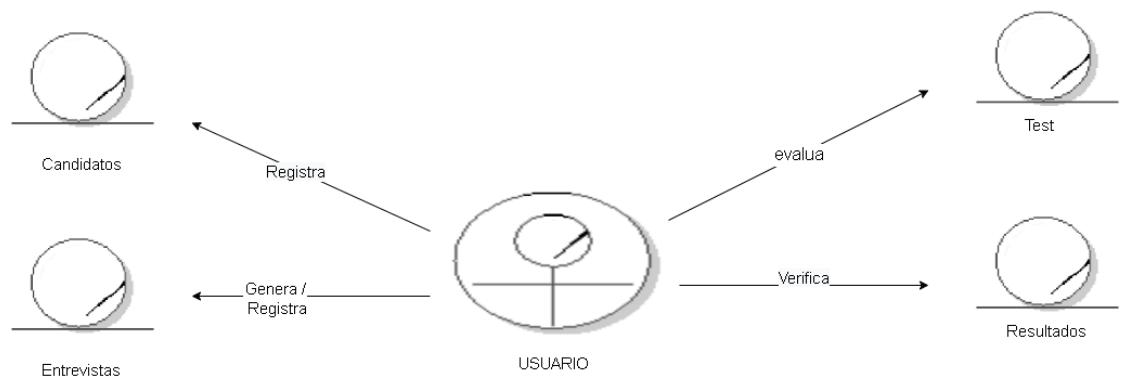


Figura 18: Gestión de Selección de Personal. Fuente: Elaboración propia

1.1.2.3. Modelo del Dominio del Problema (MDP).

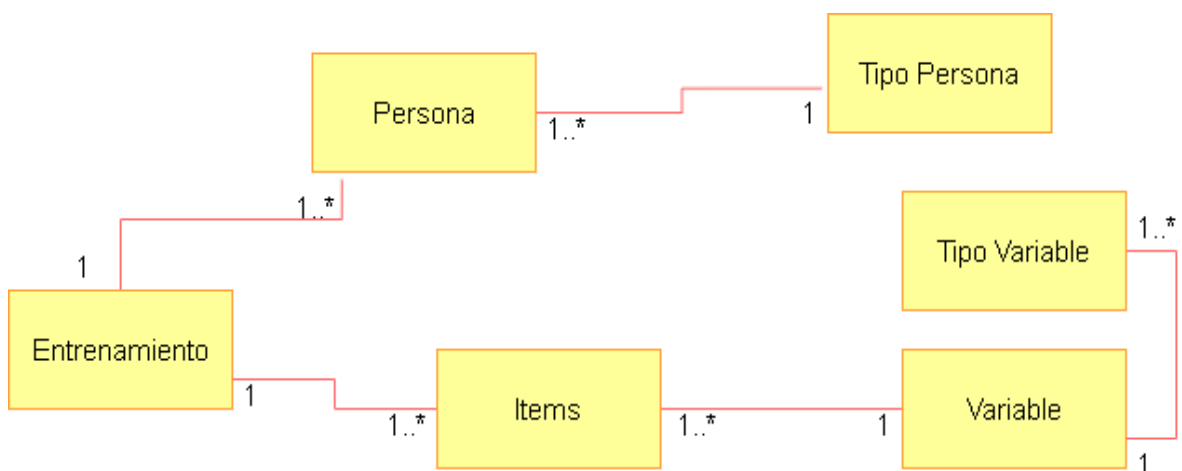


Figura 19: Modelo del Dominio del Problema. Fuente: Elaboración propia.

1.1.3. Requerimientos

1.1.3.1. Package de Requerimientos

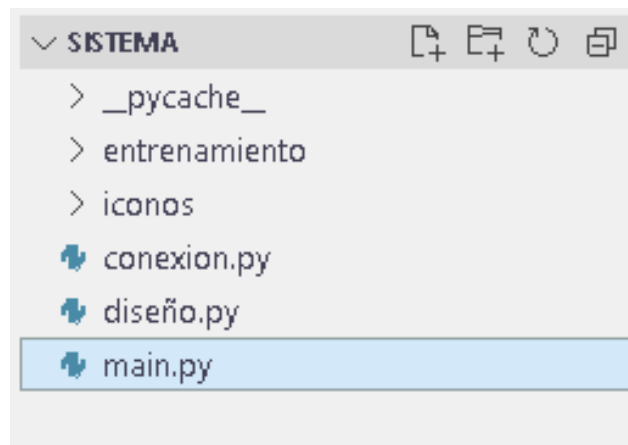


Figura 20: Paquete de Requerimientos del Sistema . Fuente: Elaboración propia.

1.1.3.2. Diagramas de Casos de Uso (DUC).

- **DUC - GESTION DE SELECCIÓN DE PERSONAL**

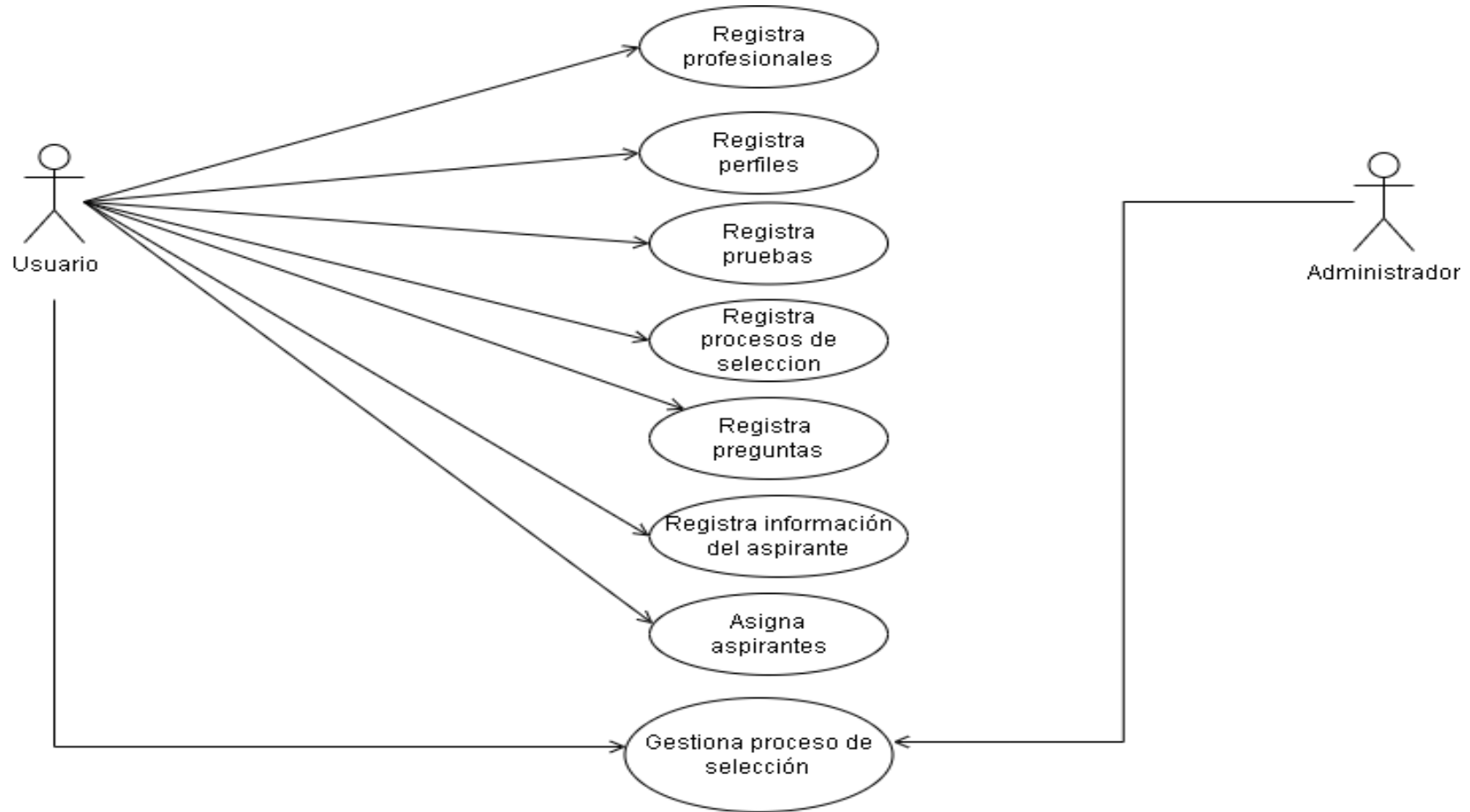


Figura 21: Gestión de Selección de Personal. Fuente: Elaboración propia.

1.1.3.3. Glosario de Casos de Uso

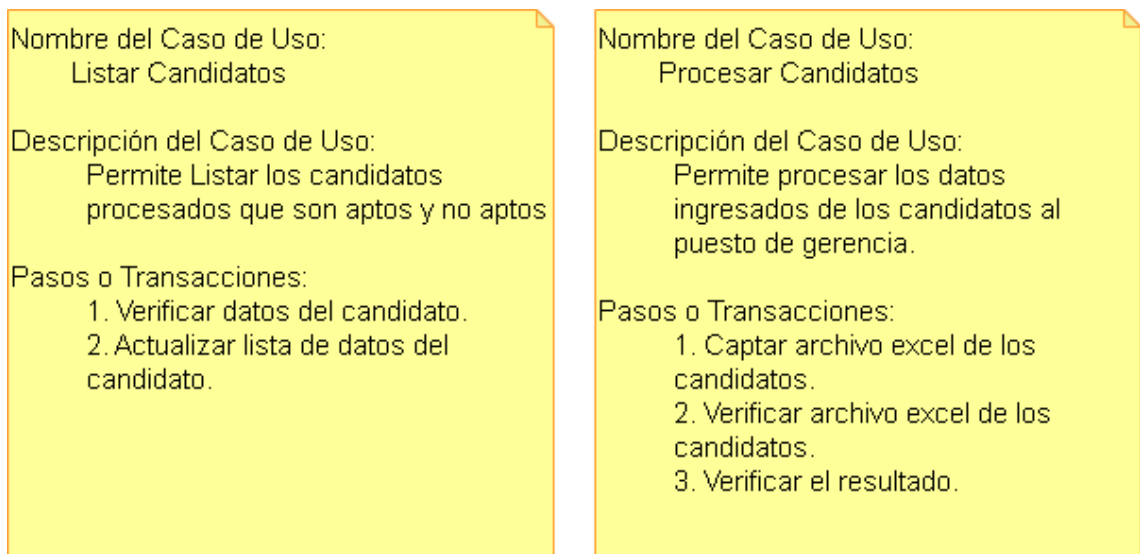


Figura 22: Glosario de Casos de Uso. Fuente: Elaboración propia.

1.1.4. Diagrama de estado

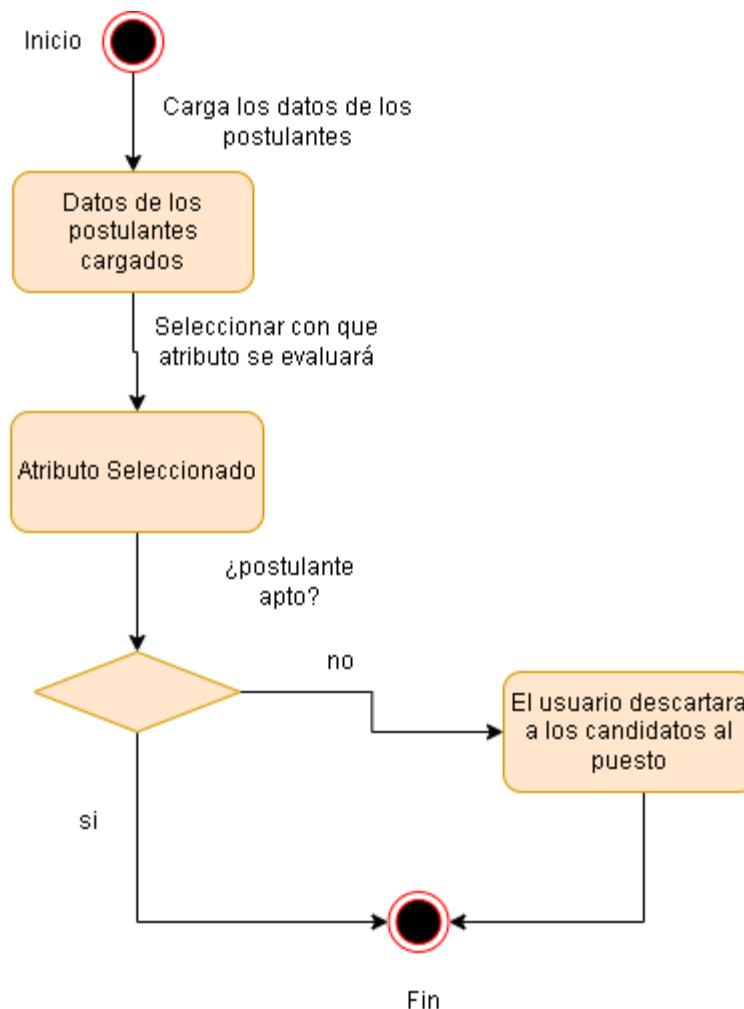


Figura 23: Diagrama de estado. Fuente: Elaboración propia.

1.2. Requerimientos no Funcionales.

Tabla 33:

Lista de requerimientos no funcionales.

ID	Requerimiento	Descripción
RNF1	La aplicación necesita ejecutarse en un sistema operativo windows 10 ó 11	Dado que el sistema operativo de windows 7 ya no tiene soporte y la versión 8 no es muy estable, se debe de elegir los windows más actuales ya que aún mantienen actualización.
RNF2	Los requerimientos de hardware son: - Procesador: 1 GHz o más - RAM: 3 GB o más - Disco duro 256 GB o más	Los requerimientos mínimos de hardware deben de soportar el sistema operativo Windows 10.
RNF3	Interfaz de usuario	La interfaz tiene que ser amigable con el usuario.

Nota: Elaboración propia.

Para mantener un orden a la hora de desarrollar el Proyecto se realizó un Diagrama de Gantt donde se especificó los procesos en sus respectivas fechas como se visualiza en la siguiente figura:

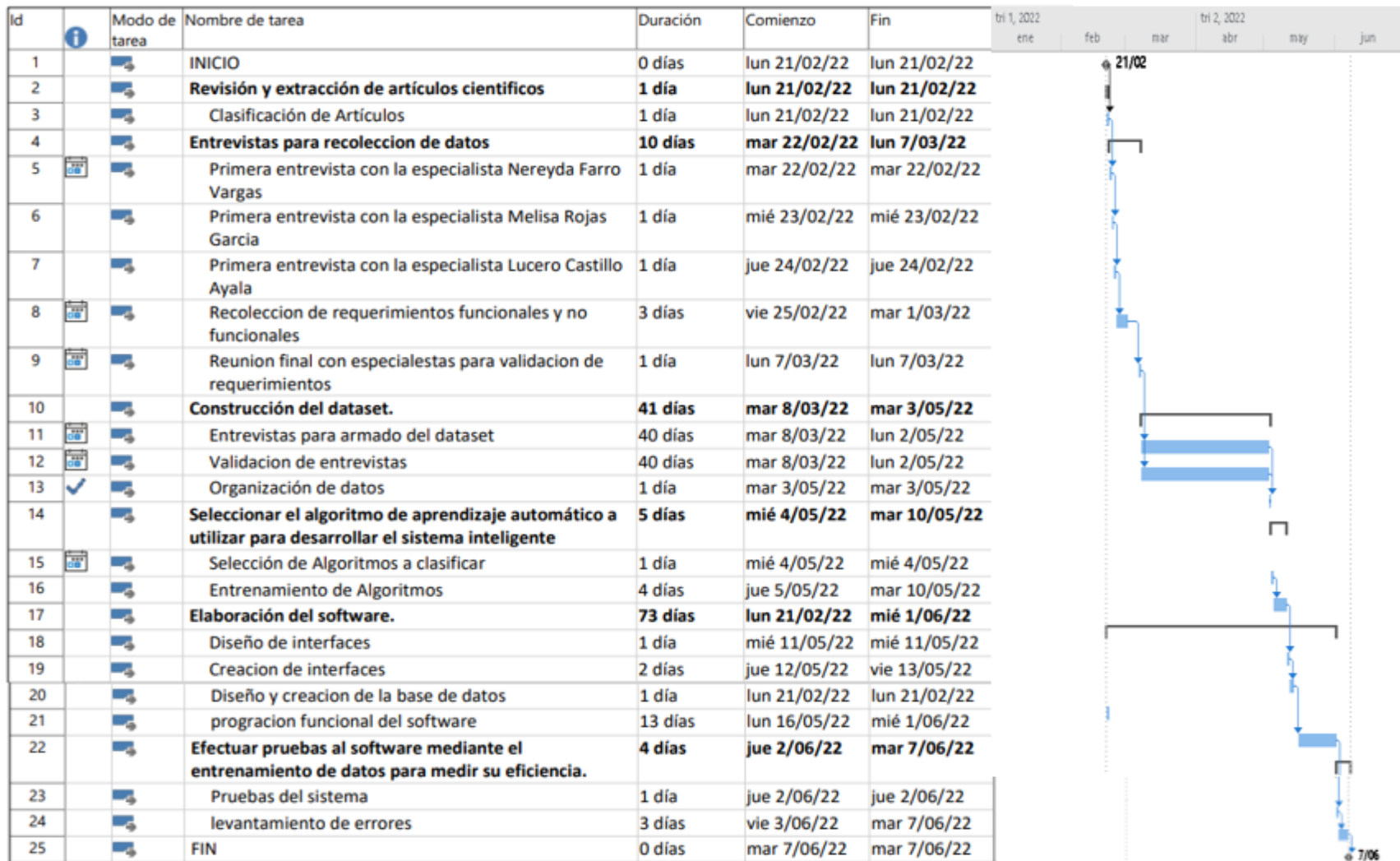


Figura 24: Diagrama de Gantt. Fuente: Elaboración propia.

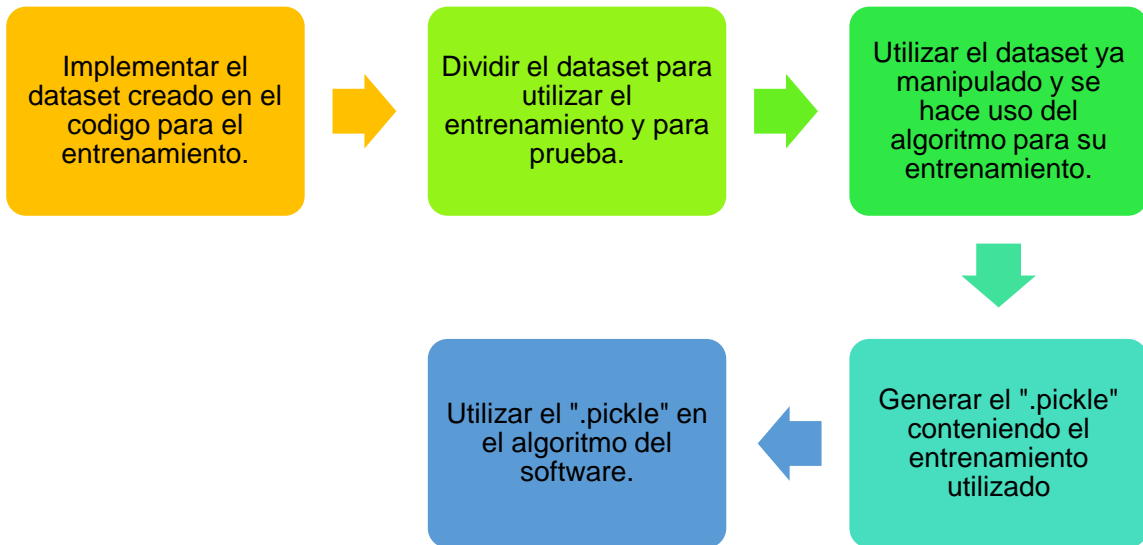


Figura 25: Etapas de desarrollo. Fuente: Elaboración propia.

En la presente imagen nos muestra al dataset y cómo está distribuido sus atributos en gráficos de barras.

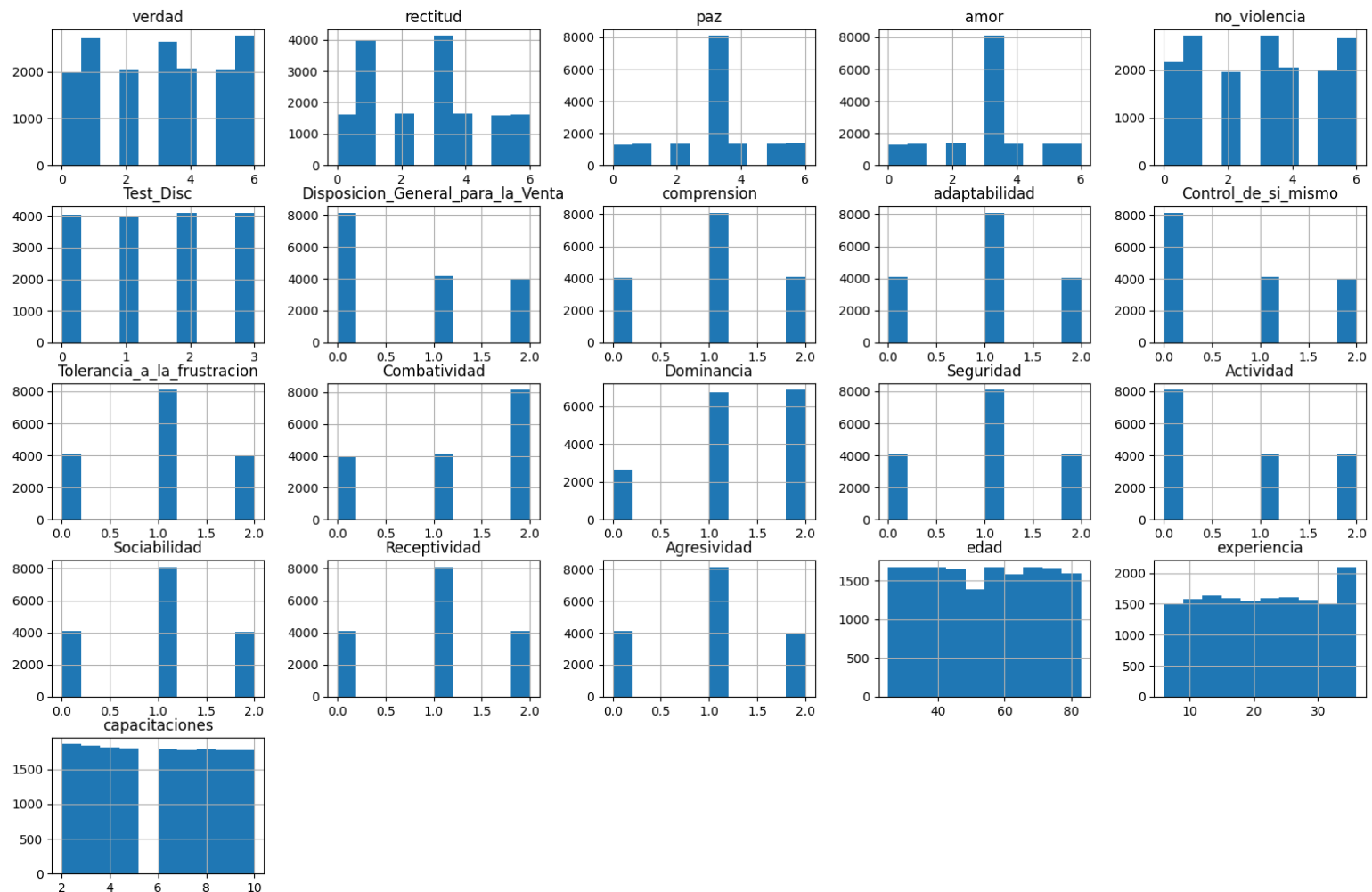


Figura 26: Grafica de barras del dataset. Fuente: Elaboración propia.

Para determinar si la persona es idónea al puesto de gerencia en base a los atributos del dataset, se evaluó que la variable “clasificador” determina que personas son aptas y no aptas al puesto de gerencia, seguidamente se empezó en la creación de la base de datos para poder permitir guardar a toda la lista de postulantes y así posteriormente poder tener un control de las personas interesadas al puesto.

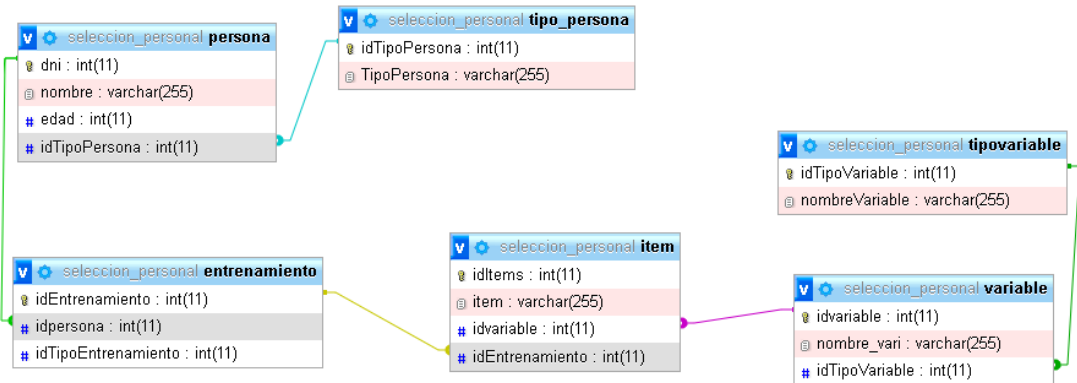


Figura 27: Diseño de base en Xampp. Fuente: Elaboración propia

De este modo ya habiendo seleccionado el algoritmo y dataset, se trabajó en el lenguaje de programación Python, donde se plasmó los procesos a seguir del algoritmo Naive Bayes, creando un sistema que permitió analizar y procesar el dataset seleccionado, que a su vez permitió hallar las variables de operalización de esta investigación.

En el código aplicado fue necesario la instalación e importaron de las librerías LabelEncoder, SelectKbest, GaussianNB, Make_pipeline, Train_test_split, BaseEstimator de Sklearn, Pandas, Numpy, Matplotlib, Pickle, seaborn. lo cual nos permitió realizar dicho sistema para la evaluación de los perfiles por competencia laboral de un puesto gerencial.

```

8 import pandas as pd
9 import numpy as np
10 import sklearn
11 import pickle
12 from sklearn.preprocessing import LabelEncoder
13 from sklearn.model_selection import train_test_split

```

```
14 from sklearn.metrics import confusion_matrix
15 from sklearn.preprocessing import StandardScaler
16 from imblearn.over_sampling import SMOTE
17 from sklearn import linear_model
18 from sklearn.metrics import precision_score
19 from sklearn.metrics import accuracy_score
20 from sklearn.metrics import recall_score
```

Dentro del uso de las librerías se utilizó “**Pandas**” para la manipulación del dataset, la librería “**Numpy**” para el cálculo numérico y análisis de los datos cargados en el .csv, La librería “**Matplotlib**” y “**Seaborn**” nos permiten generar gráficos en base a la manipulación del dataset, otra librería es “**Pickle**” nos ayuda a generar archivos memoria de tipo ‘.pickle’ para usarlo posteriormente. Además, la librería “**Sklearn**” permite el análisis de aprendizaje automático con el uso de “**linear_model**”.

Haciendo uso de la librería ‘pandas’ se integró el dataset a manejar para su procesamiento para ello se hace uso de la siguiente línea de código.

```
22 #-----importación del dataset a utilizar
24 dataframe = pd.read_csv(dataset)
```

Es importante resaltar que antes de integrarlo al algoritmo de Naive Bayes se tiene que transformar los datos de tipo cadena a números para ello hacemos uso.

```
27 #-----extracción de datos categóricos y datos numéricos del
dataframe
28 dataframe_cat = dataframe.select_dtypes(include=[object])
29 dataframe_num = dataframe.select_dtypes(include=np.number)

43 #-----conversión de variables categóricas (cadenas) en números
44 le = LabelEncoder()
45 dataframe_cat = dataframe_cat.apply(le.fit_transform)
```

Con el fin de poder utilizar todo el dataset se concatenó los datos categóricos ya convertidos en numéricos con los datos que ya son numéricos, haciendo uso de la siguiente línea de código:

```
51 #-----  
concatenar Los datos numericos y categoricos ya convertidos en nume  
ricos  
52 dataframe = pd.concat([dataframe_cat,dataframe_num],axis=1)
```

Posteriormente para una mejor manipulación de datos se hace uso de la siguiente línea de código para reordenar los datos de forma aleatoria que se encuentra en el dataset:

```
65 #reordenar el dataset  
66 dataframe = sklearn.utils.shuffle(dataframe)
```

Para la construcción del modelo a trabajar con el algoritmo de clasificación se separó los datos con las características siendo así “y” una variable que guarda todas las características del dataset y “X” solo la columna del clasificador que vendrían a ser los aptos y no aptos.

```
68 #-----División de datos La Luego la manipulación de Los datos de  
entrenamiento y prueba  
69 y=dataframe['Clasificador']  
70 X=dataframe.drop(['Clasificador'], axis=1)
```

Se realizó un sobremuestreo haciendo uso de la librería SMOTE donde el dataset contenía 16236 datos de los cuales 6651 son personas aptas mientras que la cantidad de personas no aptas es de 9585, por ende se utilizó la nivelación de datos obteniendo 19170 de los cuales 9585 son tanto aptos como no aptos.

```
74 #-----Inicialización el método de sobremuestreo SMOTE  
75 sm = SMOTE(random_state=43)  
76 X_sm, y_sm = sm.fit_resample(X,y)
```

Se dividió en una proporción de 80% para el entrenamiento y 20% para las pruebas. Esta proporción fue elegida, debido a que se le asigna una mayor cantidad de datos a entrenamiento para obtener mejores resultados en la prueba y, bajo la ley de Pareto que indica que, en una relación funcional, aproximadamente el 80% de las consecuencias proviene del 20% de las causas.

Cabe señalar que estos datos para entrenamiento y para prueba, se utilizará con la ayuda de ‘train_test_split’ que permite dividir el dataset.

```

78 #-----
Se separa los datos siendo "train" en entrenamiento y "test"
prueba para probar el algoritmo
79
X_train, X_test,y_train,y_test = train_test_split(X_sm,y_sm, test_s
ize=0.2, stratify=y_sm, random_state=0)

```

En base a la fórmula de Python explicada en el apartado de teorías relacionadas se hace uso de la librería ya importada “**Sklearn**” podemos implementar el algoritmo para la creación del modelo de Naive Bayes con “**GaussianNB()**” y así permitir hacer el entrenamiento con el dataset de entrenamiento para el modelo y realizar las predicciones en base al dataset de prueba.

```

82 #Se define el algoritmo a utilizar
83 gnb = GaussianNB()
84
85 #Entreno el modelo
86 algoritmo.fit(X_train,y_train)

```

Después haciendo uso la librería pickle guardamos el entrenamiento realizado para usarlo en la interfaz que nos permitirá evaluar a la persona.

```

94 #Se crea un archivo .pickle para guardar el modelo creado
95 pickle.dump(algoritmo, open("modelo de entrenamiento.Pickle",
"wb"))

```

Para proceder a la elección del candidato se utilizó el archivo “**.pickle**” que permitió obtener el entrenamiento previamente hecho como se muestra en la siguiente línea de código:

```

159 #obtenemos el entrenamiento posteriormente guardado en un
".pickle"
160 pipeline = pickle.load(open("modelo de
entrenamiento.Pickle", "rb"))

```

En base a la predicción obtenida del “**.pickle**” se procede a hacer la predicción creando un array que contiene los datos de la persona a

predecir para que posteriormente obtener el resultado de que si la persona es apta o no al puesto.

```
169 #guardamos en un array los resultados de cada item del  
postulante obtenidos de la fila
```

```
170 n1=np.array([
```

```
[dat[4],dat[5],dat[6],dat[7],dat[8],dat[9],dat[10],dat[11],dat[12],  
dat[13],dat[14],dat[15],dat[16],dat[17],dat[18],dat[19],dat[20],dat  
[21],dat[2],dat[3],dat[22]])
```

```
172 # obtenemos el resultado del entrenamiento
```

```
173 r1=pipeline.predict(n1)
```

Posteriormente al dataset manipulado se obtuvo el registro electrónico donde obtenemos la matriz de confusión, la precisión, la exactitud y exhaustividad del entrenamiento mostrado en la siguiente imagen:

<pre>[[1898 19] [197 1720]] Precision del modelo : 0.9890741805635422 exactitud del modelo : 0.9436619718309859 exhaustividad o rcall del modelo : 0.8972352634324465</pre>	Entrenamiento 1
<pre>[[1904 13] [209 1708]] Precision del modelo : 0.9924462521789658 exactitud del modelo : 0.9420970266040689 exhaustividad o rcall del modelo : 0.8909754825247783</pre>	Entrenamiento 2
<pre>[[1899 18] [190 1727]] Precision del modelo : 0.9896848137535816 exactitud del modelo : 0.9457485654668754 exhaustividad o rcall del modelo : 0.900886802295253</pre>	Entrenamiento 3

Figura 28: Resultados de Entrenamientos del software. Fuente: Elaboración propia.

Tabla 34:

Matriz de confusión de Entrenamiento 01

		Predicción	
		Positivos	Negativos
Observaci	Positivos	VP: 1598	FN: 19
	Negativos	FP: 198	VN: 1720

Nota: Elaboración propia.

Tabla 35:

Matriz de confusión de Entrenamiento 02.

		Predicción	
		Positivos	Negativos
Observaci	Positivos	VP: 1904	FN: 13
	Negativos	FP: 209	VN: 1708

Nota: Elaboración propia.

Tabla 36:

Matriz de confusión de Entrenamiento 03

		Predicción	
		Positivos	Negativos
Observaci	Positivos	VP: 1899	FN: 18
	Negativos	FP: 190	VN: 1727

Nota: Elaboración propia.

Así mismo teniendo los datos electrónicos se obtuvo los resultados que se plasmaron en el instrumento de recolección de datos encontrados en

el anexo 2, teniendo como parámetros el consumo de CPU, consumo de memoria RAM y tiempo de ejecución; así mismo también se halló la precisión, la exhaustividad y la exactitud con el fin de comprobar la eficiencia del sistema.

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones.

- Según la información obtenida de las investigaciones y artículos científicos de esta tesis, se eligió cinco algoritmos de clasificación como son Random Forest, Naive Bayes, Regresión logística, SVM y KNN donde se evaluó cada uno de ellas en base a su exactitud, tiempo de ejecución y uso de CPU para su implementación en el software, escogiendo a Naive Bayes como el mejor de los cinco por sus óptimos resultados en la comparación de sus métricas.
- Se construyó el software de selección de personal en el lenguaje de Python y con la ayuda del algoritmo se implementó en el software utilizando los atributos del dataset creado siendo estos todos los atributos necesarios para la clasificación,
- Haciendo el entrenamiento con el algoritmo de clasificación elegido en el desarrollo del software se utilizó la librería “.pickle” para guardar el entrenamiento y así teniendo este archivo poder usarlo posteriormente en la interfaz del usuario dándonos como resultado un grupo de las personas aptas o no aptas el puesto.
- Se procedió a entrenar el software abarcando todos los indicadores teniendo resultado óptimos en lo que es precisión, exhaustividad y exactitud, y comparando con otras investigaciones se logró visualizar que el software es eficiente en predecir al candidato cumpliendo con el objetivo plasmado inicialmente.

4.2. Recomendaciones.

- El software debe ser administrado y utilizado por un experto y con experiencia en el área de selección de personal, quien será el encargado de dar un juicio más extenso e interpretativo de los resultados que muestre el sistema sobre el cargo de gerencia de ventas que necesite contratar, del mismo modo podrá hacer uso de los resultados que se encuentran almacenados en la base de datos del software, el encargado debe conservar la información de los resultados de manera reservada, se recomienda que se realice todo el proceso con la finalidad de que el registro de los datos de los postulantes y la evaluación de los test este completo.
- El sistema está dispuesto para recibir futuras mejoras, añadiendo más aplicaciones de acuerdo a las necesidades que el usuario necesite en la selección de personal. Contando con cambios que permitan tener un impacto en los recursos y tiempo de respuesta del software.
- Este trabajo de tesis fue desarrollado con el fin de crear un modelo de software que pueda utilizarse e implementarse por un usuario capacitado en selección de personal utilizando cambios y adecuaciones, por ende es de mucha importancia hacer una evaluación inicial y utilizar indicadores adecuados para poder llegar a realizar un filtro de una gran cantidad de postulantes.

REFERENCIAS.

- Alduayj, S. S., & Smith, P. (Mayo de 2019). Sentiment Classification and Prediction of Job Interview Performance. *2nd International Conference on Computer Applications & Information Security (ICCAIS)*. doi:10.1109/CAIS.2019.8769559
- Bachtiar, F. A., Pradana, F., & Yudiari, R. D. (2019). Employee Recruitment Recommendation Using Profile Matching and Naïve Bayes. *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*., pp. 94-99. Retrieved from <https://ieeexplore.ieee.org/document/8985988>
- Cala Estupiñan, J. L. (2015). Metodología para la aplicación de la lógica 'fuzzy' en la asignación y cuantificación de riesgos en proyectos de infraestructura social bajo la modalidad de APP. *Universidad de los Andes Colombia*, pp. 1-13. Obtenido de <https://repositorio.uniandes.edu.co/handle/1992/17755>
- Cardona Mazariegos, G. M., & Olla de Leon, C. A. (2011). *EL ENEAGRAMA Y SU APLICACIÓN A PROCESOS DE SELECCIÓN DE PERSONAL A NIVEL ADMINISTRATIVO EN DUNBAR INTERNATIONAL*. Guatemala. doi:<http://www.repositorio.usac.edu.gt/11691/>
- CEBALLOS, B., Lamata Jiménez, M. T., PELTA, D., & SANCHEZ, J. M. (2013). EL MÉTODO TOPSIS RELATIVO VS. ABSOLUTO. *Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA*, pp. 181-192.
- chen, c.-t., cheng, h.-l., & hung, w.-z. (2016). A Two-Phase Decision-Making Method for Handling Personnel Selection Problem. *IEEE*, pp. 1021-1026. doi:978-1-5090-4093-3/16/\$31.00
- Chen, C.-T., Cheng, H.-L., & Hung, W.-Z. (2016). A two-phase decision-making method for handling personnel selection problem. *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 1021-1026. doi:<https://doi.org/10.1109/fskd.2016.7603319>

- CHENG, G., & TONG, X. (2018). FUZZY CLUSTERING MULTIPLE KERNEL SUPPORT VECTOR MACHINE. *2018 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, pp. 7 - 12. Obtenido de <https://ieeexplore.ieee.org/document/8521307>
- ECPA - Les Editions du Centre de Psychologie. (2016). *IPV. Inventario de Personalidad para vendedores* (Novena ed.). Francia: TEA Ediciones.
- Fawcett, T. (25 de Agosto de 2016). *Silicon Valley Data Science*. Obtenido de Silicon Valley Data Science: <https://www.svds.com/learning-imbalanced-classes/>
- Feng-Jen , Y. (2019). An Extended Idea about Decision Trees. *IEEE*, pp. 349-354. doi:10.1109/CSCI49370.2019.00068
- Gil Gaytan, O. L., & Nuñez Partido , A. (2017). Rasgos de personalidad de exportadores mexicanos con éxito. *Revista Academia & Negocios*, 23 - 34. Obtenido de <https://www.redalyc.org/articulo.oa?id=560863102003>
- Gironés Roig, J., Casas Roma, J., Minguillón Alfonso, J., & Caihuelas Quiles, R. (2017). *Minería de datos - Modelos y algoritmos*. Barcelona, España: Editorial UOC.
- Horat, S. I., Kara, K. C., & Karakaş, A. (Noviembre de 2019). Job Pre-Interview System with Artificial Intelligence. *1st International Informatics and Software Engineering Conference (UBMYK)*. doi:10.1109/UBMYK48245.2019.8965497
- Kang, M., & Pecht , M. (2018). *Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things*. USA: Editorial: Wiley-IEEE Press.
- Kim, J., Lee, J., Lee, C., Park, E., Kim, J., & Kim, H. (2013). Optimal Feature Selection for Pedestrian Detection based on Logistic Regression Analysis. *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 239-242. Retrieved from <https://ieeexplore.ieee.org/document/6721800/authors#authors>

- Korkmaz, O. (2019). PERSONNEL SELECTION METHOD BASED ON TOPSIS MULTI-CRITERIA DECISION MAKING METHOD. *Uluslararası İktisadi ve İdari İncelemeler Dergisi*, pp. 284-287. doi:<https://doi.org/10.18092/ulikidince.468486>
- Kristiawan, N., Edy, N., Purwanto, Muljono, Ahmad, Z. F., Affandy, & Ruri, S. B. (2019). Improving Random Forest Method to Detect. *2019 International Conference on Information and Communications Technology (ICOIACT)*, pp. 514 - 518. Retrieved from <https://ieeexplore.ieee.org/document/8938451>
- Merchan Blanco, K., & Lamadrid Bedoya, R. (2019). *Diseño de una Metodología de Selección de Personal Obrero en la Empresa Constructora LyM S.A.S.* Barranquilla, Colombia.
- Nimsarkar, V. (27 de 10 de 2020). *HR Analytics Dataset*. Obtenido de Kaggle: <https://www.kaggle.com/viveknimsarkar/hr-analytics-dataset?select=HR.csv>
- Onuralp Yigit, I., & Shourabizadeh, H. (2017). An Approach for Predicting Employee Churn by using data mining. *IEEE*, pp. 1 - 4. doi:10.1109/IDAP.2017.8090324
- Padilla Jordán, P. J. (2014). "RELACIÓN ENTRE LOS RESULTADOS DE LA PRUEBA DE INVENTARIO DE PERSONALIDAD PARA VENDEDORES (IPV) Y EL DESEMPEÑO LABORAL DE UN GRUPO DE VENDEDORES QUE TRABAJAN PARA UNA EMPRESA DEDICADA A COMERCIALIZAR LOS PRODUCTOS Y SERVICIOS DE TELECOMUNICACIONES.
- Pérez Pérez, O. d. (2021). Aprendizaje Basado en Resolución de Problemas (ABRP) como mediación didáctica para el desarrollo del razonamiento clínico y humanístico del residente de. *RIAICES*, 47-54. doi:DOI: <https://doi.org10.17811/ria.2.1.2013.47-54>
- Rabcan, J., Vaclavkova, M., & Blasko, R. (2017). Selection of appropriate candidates for a type position using C4.5 decision tree. *International*

- Conference on Information and Digital Technologies (IDT)*, pp. 332-338.
doi:<https://doi.org/10.1109/dt.2017.8024318>
- Rahayu, S., Umar, R., & Sunardi. (2020). Perbandingan Metode Profile Matching Dengan Metode SMART Untuk. *REVISTA RESTI*, pp. 311-318.
- Rahmi, B. (2019). Personnel selection through Fuzzy ELECTRE I method. *INTERNATIONAL YOUTH SCIENCE FORUM "LITTERIS ET ARTIBUS", 24-26 NOVEMBER 2016, LVIV, UKRAINE*, pp. 284-287. Obtenido de https://www.researchgate.net/publication/332269189_Personnel_selection_through_Fuzzy_ELECTRE_I_method
- Senel, B., Senel, M., & Aydemir, G. (2018). Use And Comparison of Topis And Electre Methods In Personnel Selection. *ITM Web of Conferences*, pp. 1-10. doi:<https://doi.org/10.1051/itmconf/20182201021>
- Waheed, A. P., Moshirpour, M., Moshirpour, M., Rokne, J., & Alhajj, R. (2017). Effective Personnel Selection and Team Building Using Intelligent Data Analytics. *Studies in Big Data*, pp. 61-69. doi:https://doi.org/10.1007/978-3-319-60255-4_5
- Wang, C. J., Jin, X., & Zheng, Z. (Junio de 2021). Analysis of Systematic Reform of Future Teaching in the Age of Artificial Intelligence. *2nd International Conference on Artificial Intelligence and Education (ICAIE)*. doi:10.1109/ICAIE53562.2021.00154
- Wikan, K. J., & Kemas, M. L. (2020). Optimization of Decision Tree Algorithm in Text Classification of Job Applicants Using Particle Swarm Optimization. *3rd International Conference on Information and Communications Technology (ICOIACT)*. doi:10.1109/ICOIACT50329.2020.9332101
- Yang, F.-J. (2018). An Implementation of Naive Bayes Classifier. *IEEE International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 301-306. doi:10.1109/CSCI46756.2018.00065
- Yang, F.-J. (2019). An Extended Idea about Decision Trees. *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 301-306. doi:<https://doi.org/10.1109/csci49370.2019.00068>

Yusuf, M., & Lhaksana, K. M. (25 de Noviembre de 2020). An Automated Interview Grading System in Talent Recruitment using SVM. *3rd International Conference on Information and Communications Technology (ICOI/ACT)*. doi:10.1109/ICOI/ACT50329.2020.9332109

ANEXOS.

Anexo 01: Resolución N° 0447-2021 de aprobación de tema de tesis.



UNIVERSIDAD
SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N°0447-2021/FIAU-USS

Pimentel, 28 de mayo de 2021

VISTO:

El Acta de reunión N°1305-2021 del Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS remitida mediante oficio N°0227-2021/FIAU-IS-USS de fecha 19 de mayo de 2021, y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y tesis son aprobados por el Comité de Investigación y derivados a la Facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El período de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C."

Que, según documentos de Vistos el Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS acuerdan aprobar los temas de las Tesis a cargo de los estudiantes del curso de Investigación II que se detallan en el anexo de la presente Resolución.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO 1°: APROBAR, el tema de la Tesis perteneciente a la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de los estudiantes del Programa de estudios de INGENIERÍA DE SISTEMAS según se detalla en el anexo de la presente Resolución.

ARTÍCULO 2°: ESTABLECER, que la inscripción del Tema de la Tesis se realice a partir de emitida la presente resolución y tendrá una vigencia de dos (02) años.

ARTÍCULO 3°: DEJAR SIN EFECTO, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE



Cc: Interesado, Archivo

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N°0447-2021/FIAU-USS

Pimentel, 28 de mayo de 2021

ANEXO

N°	AUTOR (ES)	TÍTULO APROBADO
1	ALMENDRAS FLORES RONALD ELADIO	COMPARACIÓN DE PROTOCOLOS DE COMUNICACIÓN DE DISPOSITIVOS IOT PARA EL ASEGURAMIENTO DE DATOS EN EL ÁMBITO DE LA SALUD
2	BUSTAMANTE GUERRERO FERNANDO LUCERO CHUCAS LUIS ALBERTO	EVALUACION DE LA USABILIDAD DE UNA APLICACIÓN DE GESTIÓN DE PROYECTOS PARA UNA EMPRESA DE ASESORÍA TÉCNICA BASADA EN LA NORMA ISO/IEC 25010
3	CABANILLAS TORRES ALVARO PAUL HARBERT FARRO VARGAS LUIS ADEMAR	DESARROLLO DE UN SISTEMA INTELIGENTE PARA LA EVALUACIÓN DE LOS PERFILES POR COMPETENCIA LABORAL DE UN PUESTO GERENCIAL
4	CABRERA GUEVARA EDSON OMAR ROQUE IZQUIERDO ADELMO	DESARROLLO DE UNA METODOLOGÍA ÁGIL AD HOC PARA LA CREACIÓN DE APLICACIONES MÓVILES EN PEQUEÑAS EMPRESAS. CASO DE ESTUDIO: AMÉRICA PERÚ S.A.C, LIMA.
5	CARRASCO MARTINEZ LUIS MARLON	EVALUACIÓN DE METODOS DE PREVENCIÓN DE INTRUSOS PARA BLOQUEAR Y NOTIFICAR ATAQUES EN LA ZONA DESMILITARIZADA DE LA RED.
6	CARRERA SANCHEZ JOSE ROSVEL	DESARROLLO DE LA ARQUITECTURA EMPRESARIAL BASADO EN MARCOS DE TRABAJO PARA EL ALINEAMIENTO ESTRATÉGICO DE TECNOLOGÍAS DE INFORMACIÓN. CASO DE ESTUDIO: COLEGIO RAMÓN CASTILLA - BAGUA GRANDE
7	CASTRO FERNANDEZ LEVI RONALD	ANÁLISIS COMPARATIVO DE ALGORITMOS DE APRENDIZAJE AUTOMÁTICO PARA IDENTIFICAR ATAQUES DE INYECCIÓN SQL A BASE DE DATOS EN APLICACIONES WEB
8	CELIS SANCHEZ SUJEILY PEREZ ROJAS FRANKLIN ALEXI	MODELO DE EVALUACIÓN DE PROCESOS TI CON BASE EN EL MARCO DE REFERENCIA COBIT5 PAM - CASO DE ESTUDIO SIEMPRESOFT E.I.R.L. CHICLAYO
9	CONDOR ROJAS JHAIR	DESARROLLO DE UNA METODOLOGÍA DE GESTIÓN DE RIESGOS DE CIBERSEGURIDAD PARA UNA MUNICIPALIDAD PERUANA. CASO DE ESTUDIO: MUNICIPALIDAD DE CHICLAYO
10	DE LOS SANTOS GUERRERO KARINA CAROLINA	DISEÑO DE PROCESO DE GOBIERNO DE TI PARA CONTROLAR LA GESTIÓN DE SEGURIDAD DE LA INFORMACIÓN BASADO EN COBIT 2019. CASO DE ESTUDIO: MUNICIPALIDAD DISTRITAL DE LA VICTORIA, 2021
11	FERNANDEZ SALDAÑA LEONCIO IDROGO CORNEJO LEONCIO	DESARROLLO DE UN MÉTODO DE CLASIFICACIÓN AUTOMÁTICA DE CITRUS AURANTIFOLIA USANDO PROCESAMIENTO DIGITAL DE IMÁGENES
12	FRANCO RODRIGUEZ ANTHONY WALTER	EVALUACIÓN DEL DESEMPEÑO DE LA CALIDAD DE SERVICIO EN LA IMPLEMENTACIÓN DE MEJORA EN UNA RED METRO ETHERNET DE UNA MUNICIPALIDAD DISTRITAL PERUANA
13	GASTELO FERNANDEZ EDIN JONNY RODRIGUEZ FLORES ALFREDO HIPOLITO	DESARROLLO DE UN MODELO DE GESTIÓN DE RIESGOS BASADO EN LA METODOLOGÍA MAGERIT PARA MINIMIZAR LOS RIESGOS DE ADQUISICIÓN Y USO DE TI EN UNA MUNICIPALIDAD DE PERÚ. CASO DE ESTUDIO: MUNICIPALIDAD DISTRITAL DE CUPISNIQUE.
14	JUAREZ ACOSTA BENITO JESUS TORRES PEREZ ISABEL	EVALUACIÓN DEL NIVEL DE USABILIDAD BASADO EN ISO/IEC 9126 DE UN PRODUCTO SOFTWARE GENERADO POR LA HERRAMIENTA DE DESARROLLO GUIADO POR MODELOS BONITASOFT.
15	LOPEZ BARRETO ALDO HASSAN	COMPARACIÓN DE PROTOCOLOS DE REDES PRIVADAS VIRTUALES PARA AUMENTAR LA SEGURIDAD EN EL CONTEXTO DEL TELETRABAJO

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N°0447-2021/FIAU-USS

Pimentel, 28 de mayo de 2021

N°	AUTOR (ES)	TÍTULO APROBADO
16	LUQUE CONDORI BASILIO	EVALUACIÓN DE LA CALIDAD DE APLICACIONES GENERADAS POR BPM PARA LA GESTIÓN DE TRÁMITES EN UNA GERENCIA REGIONAL DE TRANSPORTE PERUANO
17	PALOMINO PALOMINO CLAUDIO JESUS	ANÁLISIS COMPARATIVO DE HERRAMIENTAS DE INTELIGENCIA DE NEGOCIOS ORIENTADOS A LAS PYMES
18	PEREZ SILVA EDWIN WILDOR	RECONOCIMIENTO DE PLACAS VEHICULARES MEDIANTE VISION COMPUTACIONAL PARA MEJORAR EL ACCESO A UN PARQUEADERO
19	ROJAS CAHUATA GUILLERMO	AUTOMATIZACION DE PROCESOS CONSIDERANDO LA METODOLOGIA BPM EN EL CENTRO DE CONTROL EN UNA ENTIDAD BANCARIA.
20	SANTISTEBAN VALDERA ELVIS ALBERTO VILLANUEVA CASTRO JOSE SEBASTIAN	DESARROLLO DE UN MODELO DE GESTIÓN DE INCIDENCIAS BASADO EN MARCOS DE REFERENCIA DE BUENAS PRÁCTICAS PARA MEJORAR EL SERVICIO DE TI. CASO DE ESTUDIO MUNICIPALIDAD DISTRITAL DE MÓRROPE - LAMBAYEQUE
21	VILCHEZ SILVA OMAR JHONATHAN	ANÁLISIS COMPARATIVO DE ALGORITMO DE VITERBI Y BAUM-WELCH DE RECONOCIMIENTO DE VOZ PARA LA IDENTIFICACIÓN DE PERSONAS

Anexo 02: Formatos de instrumentos de recolección de datos de la presente investigación.

Rendimiento del clasificador					
Precisión					
Precisión	Algoritmo RF	Fecha	Matriz de confusión		Valor
			Positivos	Negativos	
	Entrenamiento 1	29-04-2022	VP: 1898	FN: 19	0.989
	Entrenamiento 2	29-04-2022	VP: 1904	FN: 13	0.992
Entrenamiento 3	29-04-2022	VP: 1899	FN: 18	0.989	
Exactitud					
Exactitud	Algoritmo RF	Fecha	Matriz de confusión		Valor
			Positivos	Negativos	
	Entrenamiento 1	29-04-2022	VP: 1898	FN: 197	0.943
	Entrenamiento 2	29-04-2022	VP: 1904	FN: 209	0.942
Entrenamiento 3	29-04-2022	VP: 1899	FN: 190	0.945	
Exhaustividad					
R-Call o Exhaustividad	Algoritmo RF	Fecha	Matriz de confusión		Valor
			Positivos	Negativos	
	Entrenamiento 1	29-04-2022	VP: 1898	FN: 197	0.897
	Entrenamiento 2	29-04-2022	VP: 1904	FN: 209	0.890
Entrenamiento 3	29-04-2022	VP: 1899	FN: 190	0.900	

CONSUMO DE RECURSOS			
CPU			
	Algoritmo RF	Fecha	Valor
Consumo de CPU	Entrenamiento 1	29-04-2022	11.20
	Entrenamiento 2	29-04-2022	11.15
	Entrenamiento 3	29-04-2022	11.02
MEMORIA			
	Algoritmo RF	Fecha	Valor
Consumo de memoria RAM	Entrenamiento 1	29-04-2022	629.3 Mb
	Entrenamiento 2	29-04-2022	653.14 Mb
	Entrenamiento 3	29-04-2022	550.4 Mb
TIEMPO DE RESPUESTA			
	Algoritmo RF	Fecha	Valor
Tiempo de respuesta.	Entrenamiento 1	29-04-2022	9.10
	Entrenamiento 2	29-04-2022	12.40
	Entrenamiento 3	29-04-2022	8.26

Anexo 03: Consentimiento informado de especialistas para el desarrollo de la presente investigación.

ACEPTACIÓN DE VALIDACIÓN DE EXPERTOS DEL DATASET

Chiclayo 02 de marzo de 2022

Quien suscribe:

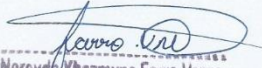
Srita.

Farro Vargas Nereyda Yhazmyne

AUTORIZA: Permiso para recojo de información pertinente en función del proyecto de investigación, denominado: "DESARROLLO DE UN SISTEMA INTELIGENTE PARA LA EVALUACIÓN DE LOS PERFILES POR COMPETENCIA LABORAL DE UN PUESTO GERENCIAL".

Por el presente, la que suscribe, Srita Farro Vargas Nereyda Yhazmyne Psicóloga Organizacional, AUTORIZO a los alumnos Cabanillas Torres Alvaro Paul Harbert, identificado con DNI N.º 71775833 y Farro Vargas Luis Ademar identificado con DNI N.º 75557753, pertenecientes a la Escuela Profesional de Ingeniería de Sistemas y Autores de dicha información que conforma la entrevista realizada para el recojo de información del DATASET, para efectos exclusivamente académicos de la elaboración de tesis enunciada líneas arriba de quien solicita garantice la absoluta confidencialidad de la información solicitada.

Atentamente.



Nereyda Yhazmyne Farro Vargas
Lic. en Psicología Organizacional
C.P.S.P. N° 38068

Farro Vargas Nereyda Yhazmyne
DNI N.º 75195119
Psicóloga Organizacional

ACEPTACIÓN DE VALIDACIÓN DE EXPERTOS DEL DATASET

Chiclayo 02 de marzo de 2022

Quien suscribe:

Srta.

Rojas García Rosa Melissa

AUTORIZA: Permiso para recojo de información pertinente en función del proyecto de investigación, denominado: "DESARROLLO DE UN SISTEMA INTELIGENTE PARA LA EVALUACIÓN DE LOS PERFILES POR COMPETENCIA LABORAL DE UN PUESTO GERENCIAL".

Por el presente, la que suscribe, Srta. Rojas García Rosa Melissa Psicóloga Organizacional, AUTORIZO a los alumnos Cabanillas Torres Alvaro Paul Harbert, identificado con DNI N.º 71775833 y Farro Vargas Luis Ademar identificado con DNI N.º 75557753, pertenecientes a la Escuela Profesional de Ingeniería de Sistemas y Autores de dicha información que conforma la entrevista realizada para el recojo de información del DATASET, para efectos exclusivamente académicos de la elaboración de tesis enunciada líneas arriba de quien solicita garantice la absoluta confidencialidad de la información solicitada.

Atentamente.



Lic. Rosa Melissa Rojas García
PSICÓLOGA ORGANIZACIONAL
C.P.S.P. N°42781

Rojas Garcia Rosa Melissa
DNI N.º 45682903
Psicóloga Organizacional

ACEPTACIÓN DE VALIDACIÓN DE EXPERTOS DEL DATASET

Chiclayo 02 de marzo de 2022

Quien suscribe:

Srita.

Castillo Ayala Lucero Verónica.

AUTORIZA: Permiso para recojo de información pertinente en función del proyecto de investigación, denominado: "DESARROLLO DE UN SISTEMA INTELIGENTE PARA LA EVALUACIÓN DE LOS PERFILES POR COMPETENCIA LABORAL DE UN PUESTO GERENCIAL".

Por el presente, la que suscribe, Srita Castillo Ayala Lucero Verónica. Psicóloga Organizacional, AUTORIZO a los alumnos Cabanillas Torres Alvaro Paul Harbert, identificado con DNI N.º 71775833 y Farro Vargas Luis Ademar identificado con DNI N.º 75557753, pertenecientes a la Escuela Profesional de Ingeniería de Sistemas y Autores de dicha información que conforma la entrevista realizada para el recojo de información del DATASET, para efectos exclusivamente académicos de la elaboración de tesis enunciada líneas arriba de quien solicita garantice la absoluta confidencialidad de la información solicitada.

Atentamente.



Castillo Ayala Lucero Verónica
DNI N.º 72281813
Psicóloga Organizacional

Anexo 04: Formato del test de Disc.

DISC.xlsx - Excel

Inicio ses. Compartir

Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisar Vista Ayuda ¿Qué desea hacer?

C23

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	
3																																	
4	EJEMPLO																																
5	MAS MENOS		B		MAS MENOS																												
6	Elefante	X			Azul																												
7	León				Rojo																												
8	Jirafa		X		Amarillo			X																									
9	Cocodrilo				Verde			X																									
10	En cada uno de los 28 grupos de palabras, escoja la palabra que más lo(a) represente y márkela en la columna MAS y escoja una palabra que menos lo(a) represente y márkela en la columna MENOS.																																
11																																	
12																																	
13																																	
14	1	MAS	MENOS	8	MAS	MENOS	15	MAS	MENOS	22	MAS	MENOS																					
15	Entusiasta			Extrovertido(a)			Popular			Impulsivo(a)																							
16	Rápido(a)	X		Precavido(a)			Reflexivo(a)			Cuida los Detalles																							
17	Lógico(a)		X	Constante			Tenaz			Energico(a)																							
18	Apacible			Impaciente			Calzado(a)			Tranquilo(a)																							
19																																	
20	2	MAS	MENOS	9	MAS	MENOS	16	MAS	MENOS	23	MAS	MENOS																					
21	Cauteloso(a)			Discreto(a)			Analtico(a)			Sociable																							
22	Decidido(a)	X		Compaciente			Audaz			Sistemático(a)																							
23	Receptivo(a)		X	Encantador(a)			Leal			Vigoroso(a)																							
24	Bondadoso(a)			Insistente			Promotor(a)			Tolerante																							
25																																	
26	3	MAS	MENOS	10	MAS	MENOS	17	MAS	MENOS	24	MAS	MENOS																					
27	Amigable			Valeroso(a)			Sociable			Cautivador(a)																							
28	Preciso(a)			Anima a los demás			Paciente			Contento(a)																							
29	Franco(a)			Pacifico(a)			Autosuficiente			Exigente																							
30	Tranquilo(a)			Perfeccionista			Certero(a)			Apegado(a) a las normas																							
31																																	
32	4	MAS	MENOS	11	MAS	MENOS	18	MAS	MENOS	25	MAS	MENOS																					
33	Elocuente			Reservado(a)			Adaptable			Le agrada discutir																							
34	Controlado(a)			Atento(a)			Resuelto(a)			Metódico(a)																							
35	Tolerante			Osado(a)			Prevenido(a)			Comedido(a)																							
36	Decisivo(a)			Alegre			Vivaz			Desenvuelto(a)																							
37																																	
38	5	MAS	MENOS	12	MAS	MENOS	19	MAS	MENOS	26	MAS	MENOS																					
39	Atrevido(a)			Estimulante			Agresivo(a)			Jovial																							
40	Concienzudo(a)			Gentil			Impetuoso(a)			Preciso(a)																							
41	Comunicativo(a)			Perceptivo(a)			Amistoso(a)			Directo(a)																							
42	Moderado(a)			Independiente			Discerniente			Ecuánime																							
43																																	
44	6	MAS	MENOS	13	MAS	MENOS	20	MAS	MENOS	27	MAS	MENOS																					
45	Ameno(a)			Competitivo(a)			De trato Fácil			Inquieto(a)																							
46	Ingenioso(a)			Considerado(a)			Compasivo(a)			Amable																							
47	Investigador(a)			Alegre			Cauto(a)			Elocuente																							
48	Acepta Riesgos			Sagaz			Habla Directo			Cuidadoso(a)																							
49																																	
50	7	MAS	MENOS	14	MAS	MENOS	21	MAS	MENOS	28	MAS	MENOS																					
51	Expresivo(a)			Meticuloso(a)			Evaluador(a)			Prudente																							
52	Cuidadoso(a)	X		Obediente			Generoso(a)			Pionero(a)																							
53	Dominante			Ideas Firmes			Animado(a)			Espontáneo(a)																							
54	Sensible		X	Alentador(a)			Persistente			Colaborador																							
55																																	
56																																	

PRUEBA CORRECCIÓN Tablas

Listo Accesibilidad: es necesario investigar 85%

18 de mayo de 2022

INTENSIDAD	D	I	S	C	SEGMENTO
28	+28	+28	+28	+28	
27	+12	+10	+10	+11	7
26					
25	+9	+7	+8	+9	
24	+8	+6	+7	+8	
23	+7	+5	+6	+7	6
22	+6	+4	+4	+6	
21	+5	+4	+3	+5	
20	+4	+3	+2	+4	
19	+3		+1		5
18	+2	+2		+3	
17	+2		0		
16	+1	+1	-1	+2	
15	0	0	-2	+1	4
14					
13	0	-1	-3	0	
12	-1	-1	-4		
11	-2	-2	-5	-1	3
10	-2		-5		
9	-3	-3	-6	-2	
8	-4	-4	-7	-3	
7	-5	-5	-8	-4	2
6	-6	-6	-9	-5	
5	-7	-7	-10	-5	
4	-8	-8	-11	-6	
3	-8	-8	-11	-6	1
2	-11	-11	-13	-11	
1	-28	-28	-28	-28	
	-1	-1	5	-3	
	3	4	6	2	

Patrón del Especialista

Emociones: Moderación calculada; afán de servir, de adaptarse a los demás.

Meta: Conservar el "status quo", controlar el ambiente.

Juzga a los demás por: Las normas de amistad, después por su capacidad.

Influye en los demás mediante: Su constancia en el desempeño; por su afán de servir, de adaptarse a las necesidades de los demás.

Su valor para la organización: Planifica a corto plazo; es predecible, es congruente; mantiene un ritmo uniforme y seguro.

Abusa de: La modestia; su miedo a correr riesgos; su resistencia pasiva hacia las innovaciones.

FALSO 0 alternativas "Más"
 FALSO 0 alternativas "Menos"
 FALSO 0 alternativas "Más"
 FALSO 0 alternativas "Menos"

DISC.xlsx - Excel

Inicio ses. Compartir

Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisar Vista Ayuda ¿Qué desea hacer?

R39

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
34																	-1	-1	5	-3										
35					X				X								Número de segmento / Patrón de Perfil Clásico													
36			X								X						3	4	6	2										
37																	Patrón del Especialista													
38																														
39																														
40																														
41																														
42																														
43																														
44																														
45																														
46																														
47																														
48																														
49																														
50																														
51																														
52																														
53																														
54																														
55																														
56																														
57																														
58																														
59																														
60																														
61																														
62																														
63																														
64																														
65																														
66																														
67																														
68																														
69																														
70																														
71																														
72																														
73																														
74																														
75																														
76																														
77																														
78																														
79																														
80																														
81																														
82																														
83																														
84																														
85																														

Emociones: Moderación calculada; afán de servir, de adaptarse a los demás.
Meta: Conservar el "status quo", controlar el ambiente.
Juzga a los demás por: Las normas de amistad, después por su capacidad.
Influye en los demás mediante: Su constancia en el desempeño; por su afán de servir, de adaptarse a las necesidades de los demás.
Su valor para la organización: Planifica a corto plazo; es predecible, es congruente; mantiene un ritmo uniforme y seguro.
Abusa de: La modestia; su miedo a correr riesgos; su resistencia pasiva hacia las innovaciones.
Bajo presión: Se adapta a quienes tienen autoridad y a lo que opina el grupo.
Teme: Los cambios; la desorganización.
Sería más eficaz si: Compartiera más sus ideas; aumentara su confianza en sí mismo basándose en la retroalimentación que recibe; utilizara métodos más sencillos y directos.

El Especialista se "lleva bien" con los demás. Por su actitud moderada y controlada y por su comportamiento modesto, puede trabajar en armonía con diversos estilos de conducta. El Especialista es considerado paciente y siempre está dispuesto a ayudar a quienes considera sus amigos. De hecho, tiende a desarrollar en el trabajo una estrecha relación con un grupo relativamente reducido de compañeros.

Se esfuerza por conservar pautas de comportamiento conocidos y predecibles. El Especialista, al ser bastante eficiente en áreas especializadas, planea su trabajo, lo enfoca de manera clara y directa y consigue una notoria constancia en su desempeño. El reconocimiento que recibe de los demás le ayuda a conservar este nivel.

El Especialista es lento para adaptarse a los cambios. Una preparación previa le concede el tiempo que requiere para cambiar sus procedimientos y conservar su nivel de rendimiento. El Especialista puede necesitar ayuda al inicio de un nuevo proyecto y para desarrollar métodos prácticos y sencillos para cubrir plazos establecidos. Suele dejar a un lado los proyectos terminados para posteriormente concluirlos. Un pequeño consejo: tira algunas de esas carpetas viejas de su archivo!

PRUEBA CORRECCIÓN Tablas

Listo Accesibilidad: es necesario investigar 100%

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z							
28					PATRON DEL ALENTADOR.																												
29					E1	Acepta la agresión, tiende a aparentar dar poca importancia a la necesidad que tiene de afecto.																											
30					M1	Controlar su ambiente o a su público.																											
31					J1	La forma en que proyecta su fuerza personal, carácter y posición social.																											
32					H1	Su encanto, dirección, intimidación, uso de recompensas.																											
33					S1	Mueve a la gente, inicia, ordena, felicita disciplina.																											
34					A1	Su enfoque de que "el fin justifica los medios".																											
35	7777	Superactivo			B1	Se vuelve manipulador, pendenciero, beligerante.																											
36	7776	Superactivo			T1	Ser demasiado blando, perder su posición social.																											
37	7775	Superactivo			SE1	Fuera más genuina su sensibilidad; estuviera más dispuesto a ayudar a otros a tener éxito en su propio desarrollo personal.																											
38	7774	Aalentador			O11	Las personas con patrón alentador saben con exactitud los resultados que quieren, pero no siempre los verbalizan de inmediato. Manifiestan																											
39	7773	Aalentador			O12	El alentador puede ser encantador en su trato con los demás. Es persuasivo para obtener ayuda cuando se le presentan detalles repetitivos y																											
40	7772	Aalentador			O13																												
41	7771	Aalentador			PATRON DEL REALIZADOR																												
42	7767	Superactivo			E2	Activo, diligente, muestra frustración.																											
43	7766	Superactivo			M2	Logros personales, en ocasiones a expensas de la meta de grupo.																											
44	7765	Superactivo			J2	El logro de resultados concretos.																											
45	7764	Aalentador			I2	La aceptación de responsabilidad por su propio trabajo.																											
46	7763	Aalentador			S2	Se propone y consigue resultados en áreas clave.																											
47	7762	Aalentador			A2	Confianza en sí mismo, absorción en el trabajo.																											
48	7761	Aalentador			B2	Se frustra e impacienta con los demás, se convierte en una persona que "lo hace todo" en vez de ser alguien que delega.																											
49	7757	Superactivo			T2	A quienes tienen niveles inferiores o competitivos de trabajo, que afectan los resultados.																											
50	7756	Superactivo			SE2	Deja de pensar en "esto o lo otro", estableciera su prioridades con mayor claridad y aceptara enfoques alternativos, estuviera dispuesto a																											
51	7755	Superactivo			O21	La motivación del Patrón Realizador surge en gran parte de su interior y de metas personales muy profundas. Este compromiso previene con sus																											
52	7754	Aalentador			O22	El Realizador demuestra un profundo interés por su trabajo y un continuo e intenso afán por conseguir lo que se propone. Tiene una alta																											
53	7753	Aalentador			O23	Si el Realizador se comunica más con los demás dejaría de pensar en "esto o lo otro", del "yo mismo lo tengo que hacer" o "quiero todo el crédito																											
54	7752	Aalentador			PATRON DEL PERFECCIONISTA																												
55	7751	Aalentador			E3	Competente para hacer bien las cosas, reservado, cauteloso.																											
56	7747	Evaluador			M3	Logros estables, predecible.																											
57	7746	Evaluador			J3	Normas precisas.																											
58	7745	Evaluador			I3	La atención al detalle y precisión.																											
59	7744	Aalentador			S3	Conciencioso, conserva las normas, control de calidad.																											
60	7743	Aalentador			A3	Los procedimientos y controles excesivos para evitar las fallas, depende demasiado de la gente, productos y procesos que le funcionaron en el																											
61	7742	Aalentador			B3	Es discreto, diplomático.																											
62	7741	Aalentador			T3	El antagonismo.																											
63	7737	Evaluador			SE3	Fuera más flexible en su papel, fuera más independiente e interdependiente, tuviera más fe en sí mismo y si se viera a sí mismo como una																											
64	7736	Evaluador			O31	El Perfeccionista es metódico y preciso en su forma de pensar y trabajar, por lo que suele seguir procedimientos ordenados tanto en su vida																											
65	7735	Evaluador			O32	El perfeccionista se puede empantanar en los detalles cuando tiene que tomar decisiones. Sabe tomar decisiones importantes, pero se le puede																											
66	7734	Aalentador			O33	El Perfeccionista se evalúa y evalúa a los demás bajo normas precisas que aseguren resultados concretos y se adhiere a procedimientos																											
67	7733	Aalentador			PATRON DEL CREATIVO																												
68	7732	Aalentador			E4	Acepta la agresión, puede contenerse al expresarse.																											
69	7731	Aalentador			M4	Dominar logros únicos.																											
70	7727	Evaluador			J4	Sus propias normas, las ideas progresivas al llevar a cabo el trabajo.																											
71	7726	Evaluador			I4	El establecimiento de un ritmo a seguir para desarrollar sistemas y enfoques innovadores.																											
72	7725	Evaluador			S4	El iniciar o diseñar cambios.																											
73	7724	Aalentador			A4	La brusquedad, la actitud crítica o condescendiente.																											
74	7723	Aalentador			B4	Se aburre fácilmente con el trabajo rutinario, cuando se le restringe se torna malhumorado, es independiente.																											
75	7722	Aalentador			T4	No poder influir, no alcanzar el nivel establecido.																											
76	7721	Aalentador			SE4	Fuera más amable, usara más tacto al comunicarse, cooperara más con el equipo, reconociera que existen sanciones.																											
77	7717	Evaluador			O41	Las personas con un Patrón Creativo muestran dos fuerzas opuestas en su comportamiento. El deseo de resultados tangibles se contrapone a																											
78	7716	Evaluador			O42	Las personas creativas previenen de manera extraordinaria el enfoque que hay que dar a un proyecto y efectúan los cambios oportunos. En																											
79	7715	Evaluador			O43	La persona creativa desea libertad para explorar y la autoridad para examinar y verificar los resultados. Puede tomar las decisiones diarias con																											
80	7714	Aalentador			PATRON DEL OBJETIVO																												
81	7713	Aalentador			E5	Puede rechazar la agresión interpersonal.																											
82	7712	Aalentador			M5	La exactitud.																											
83	7711	Aalentador			J5	Su capacidad de pensamiento analítico.																											
84	7677	Superactivo			I5	La información objetiva, los argumentos lógicos.																											
85	7676	Superactivo			S5	Define, esclarece, obtiene información, evalúa, comprueba.																											
86	7675	Superactivo			A5	El análisis.																											
87	7674	Realizador			B5	Se vuelve aprensivo.																											
88	7673	Realizador			T5	Actos irracionales, el ridículo.																											
89	7672	Realizador			SE5	Fuera más abierto, compartiera en público su perspicacia y opiniones.																											
90	7671	Realizador			O51	La capacidad de pensamiento crítico suele estar muy desarrollada en el Objetivo. Recalca la importancia de sacar conclusiones y basar las																											
91	7667	Superactivo			O52	El Objetivo prefiere trabajar con personas que, como él, prefieren mantener un ambiente laboral tranquilo. Como puede mostrarse reticente en																											
92	7666	Superactivo			O53	El Objetivo es reacio por llevar a cabo tareas "correctas" ya que puede resultar difícil tomar decisiones en situaciones ambiguas. Su tendencia a																											

Anexo 05: Formato del test de IPV.

The screenshot shows an Excel spreadsheet with the following content:

INVENTARIO PARA VENTAS

Nombre: WILL ROBERTO VELARDE MENDOZA
Edad: 33
Sexo: M
Telf.:

INSTRUCCIONES

En las páginas siguientes encontrará cierto número de situaciones y de preguntas seguidas de varias respuestas posibles.

SE LE PIDE QUE ESCOJA UNA ALTERNATIVA —Y UNA SOLA— PARA CADA PREGUNTA; la que de manera espontánea le parezca preferible.

En algunas cuestiones estará usted personalmente implicado; por ejemplo:

- Si usted pudiera elegir el tipo de vivienda de manera completamente libre, ¿qué preferiría?
A. Una casa en el campo con un gran jardín. B. Un lujoso apartamento en una gran ciudad.
C. Una casa situada en una calle tranquila de una pequeña ciudad.

Otras se refieren a personas ajenas, como:

- J... Está comprando en un gran almacén cuando advierte que alguien esconde discretamente un disco bajo su blusa. ¿Que hará J.?
A. Dar cuenta del hecho al primer vendedor que encuentre. B. No hacer nada porque piensa que no es cosa suya. C. Indicar al infractor con el gesto o de palabra que ha visto lo que ha hecho.

En este caso, usted contestará lo que, según su propia opinión, es más probable que haga J...

En realidad, no hay respuestas buenas ni malas: cada uno piensa y actúa como cree conveniente en función de su carácter, de sus intereses, etc.

Para indicar sus respuestas dispone de un recuadro en blanco, escriba sus respuestas marcando la letra que coincida con la alternativa elegida.

Si quiere cambiar su respuesta regrese a la casilla en blanco a cambiar y escriba la respuesta correcta.

Responda lo más espontánea y sinceramente posible con la contestación que se le ocurra inmediatamente al leer las alternativas que se proponen. Conteste rápidamente, sin reflexionar demasiado.

No de más que una sola respuesta a cada pregunta

—Responda a todas las preguntas. Es posible que en algunos casos ninguna de las contestaciones propuestas le convenga del todo; esfuércese entonces en elegir la que crea mejor entre las alternativas que

—No hay tiempo límite, pero no emplee demasiado tiempo en cada pregunta

1 T... debe salir de viaje con una persona de la que no conoce nada. ¿Sobre cuál de los siguientes aspectos de esa persona es preferible informar a T para que el viaje resulte mejor?
A. Su estilo de vida B. Los puntos que tengan en común

CUADERNILLO PERFIL

Accesibilidad: es necesario investigar

	A	B	C	D	E	F	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	
42	1 T... debe salir de viaje con una persona de la que no conoce nada. ¿Sobre cuál de los siguientes aspectos de esa persona es preferible informar a T para que el viaje resulte mejor?																				
43	A. Su estilo de vida																			A	0
44	B. Los puntos que tengan en común																				0
45	C. Su actividad y responsabilidades profesionales																				0
46	2 Entre los siguientes tipos de vendedores de prendas confeccionadas, ¿cuál es el que tiene más probabilidades de éxito?																				
47	A. El que presente las últimas novedades																			B	0
48	B. El que, tratando de conocer el estilo de su cliente, se interese por su modo de vida																			A	0
49	C. El que posea una buena capacidad de convencer																				0
50	3 B... tiene un proyecto importante para la promoción de un nuevo producto y va a exponer su idea ante el Comité de Dirección. ¿Cuál de las siguientes cualidades le será más útil para persuadir a su auditorio?																				
51	A. Competencia técnica y un conocimiento perfecto del tema																			B	0
52	B. Capacidad para modificar sus razonamientos según la actitud del auditorio																			C	1
53	C. Facultad para mantener el orden de sus ideas a pesar de las interrupciones																				0
54	4 Se envía a G... contra su voluntad, a un país extranjero, por el cual no se siente atraído en principio, para una estancia de varias semanas. ¿Cuál será su actitud?																				
55	A. Estimar que la duración de su estancia es demasiado corta para conseguir integrarse																			B	0
56	B. Tratar de aprender la lengua para comprender mejor a este país y vivir más a gusto																				1
57	C. No tener más que los contactos estrictamente necesarios para la buena marcha de su trabajo																				0
58	5 Según su opinión, las personas que dicen siempre "su" verdad a los demás, aunque esta sea desagradable, lo hacen, en general, porque:																				
59	A. No saben controlar sus impulsos y dicen espontáneamente lo que piensan																			B	0
60	B. No les gusta la hipocresía																			C	0
61	C. Piensan que esto simplifica las relaciones																				0
62	6 Un buen amigo de C... había comenzado bastante brillantemente su carrera profesional, pero los resultados no han sido los que cabría esperar y terminó teniendo numerosos fracasos. ¿Que piensa C... acerca de esto?																				
63	A. Que las condiciones no le han sido favorables																			B	1
64	B. Que no estaba a la altura necesaria																			C	0
65	C. Que no ha utilizado bien los medios para salir adelante																				0
66	7 B... encuentra, en casa de unos amigos, a una persona que aparenta una edad muy inferior a la que realmente tiene. ¿Que opinará B...?																				
67	A. "Probablemente ha encontrado en la vida lo que le convenía"																			B	0
68	B. "Tiene buena suerte; es cuestión de naturaleza"																			A	1
69	C. "Seguramente ha tenido una vida fácil"																				0
70	8 Para expansionarse, S... decide aprender judo. Después de unos meses de entrenamiento se da cuenta de que progresa muy lentamente. ¿Cuál será su reacción?																				
71	A. Pensando que realmente no está hecho para el judo, elegirá otro deporte																			B	0
72	B. Convencido de que no está dotado para actividad buscará otra forma de actividad																				1
73	C. Continuará sus esfuerzos con la esperanza de que, seguramente, serán un día exitosos																				0
74	9 P... se acuesta una noche muy fatigado, pero no puede dormirse porque sus vecinos del piso superior han organizado una fiesta muy ruidosa. ¿Qué hará?																				
75	A. Subir y advertir a sus vecinos																			B	0
76	B. Dar con la escoba algunos golpes en el techo																			C	0
77	C. Tomar un somnífero y tratar de dormir cueste lo que cueste																				0

IPV respuestas.xlsx - Excel

Inicio ses. | ¿Qué desea hacer?

Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisar Vista Ayuda

A1

INVENTARIO DE HABILIDADES PARA LAS VENTAS

Nombre WILL ROBERTO VELARDE MENDOZA
 Edad 33 Sexo M
 Teléfono 0

P.Direct	Factores	Decatipos	
9	Disposición General para la Ver	DGV	4
11	Comprensión	I	3
12	Adaptabilidad	II	5
13	Control de sí mismo	III	8
14	Tolerancia a la frustración	IV	4
15	Combatividad	V	4
16	Dominancia	VI	6
17	Seguridad	VII	5
18	Actividad	VIII	6
19	Sociabilidad	IX	5
20	Receptividad	R	5
21	Agresividad	A	5

TEXTOS ASOCIADOS A LOS FACTORES

24	DGV	Persona con facilidad intermedia para establecer en la venta relaciones con los demás.
25	I	Es una persona con baja empatía y objetividad en sus relaciones humanas, intuitivo, así como una baja capacidad de integrar un suceso cualquiera en su contexto.
26	II	Capacidad intermedia de adaptación a situaciones y personas diferentes, puede llegar a ser flexible en sus actividades (intelectuales o de relación) y mediana capacidad de desempeñar su papel en una situación dada
27	III	Señala a un sujeto controlado, dueño de sí mismo y capaz de una buena administración de su potencial intelectual, psicológico o físico; es una persona organizada, perseverante y hábil para ocultar sus sentimientos.
28	IV	Sujeto con capacidad intermedia para soportar adecuadamente las acciones frustrantes, puede ser capaz de comprender los fracasos (aunque sean provisionales) sin personalizar demasiado las situaciones en que se ve implicado.
29	V	Alude a un sujeto capacidad intermedia de entrar en conflicto y soportar los desacuerdos;
30	VI	Señala al sujeto con mediana voluntad de dominio, de ganar o manipular a los demás, puede llegar a ser persuasivo o cautivador en su entorno.
31	VII	Persona con niveles intermedios de seguridad en sí mismo, con mediana capacidad para enfrentar situaciones nuevas o inesperadas y puede ser capaz de enfrentarse si es necesario, a los riesgos.
32	VIII	Alude al sujeto medianamente activo y dinámico en el sentido físico, llegando a tolerar tanto la agresividad como la inactividad.

CUADERNILLO **PERFIL**

Listo Accesibilidad: es necesario investigar

Anexo 06: Formato del test de Valanti.

Excel window: Informatizado Valanti.xls [Modo de compatibilidad] - Excel

Nombre:	Nathalie Vela	
Edad:	27	Sexo: F
Cargo:	COORDINADOR	
Estudios:	UNIVERSITARIOS	

mayo 18, yyyy

1	2	1	6	2	1
2	1	2	7	2	1
3	1	2	8	1	2
4	3	0	9	2	1
5	2	1			
10	2	1	21	3	0
11	2	1	22	1	2
12	2	1	23	2	1
13	2	1	24	0	3
14	1	2	25	1	2
15	1	2	26	3	0
16	1	2	27	3	0
17	2	1	28	3	0
18	3	0	29	2	1
19	2	1	30	2	1
20	2	1			

Valor	Verdad	Rectitud	Paz	Amor	No violencia
Promedio Normas nacionales, 1997 (n=730)	15,65	21,05	17,35	16,68	21,22
Desviación estándar Normas nacionales, 1997 (n=730)	4,70	4,44	6,61	5,41	7,19
Puntajes directos	17	24	14	13	22
Valores Organizacionales	50	50	50	50	50
Nathalie Vela	52,875	54,635	44,926	43,196	51,078
Interpretación del puntaje estándar	Promedio	Promedio Alto	Promedio	Promedio Bajo	Promedio
	*****	*****	***	**	*****
Distancia con la Organización	3	7	-5	-7	1

Perfil valoral, cuestionario VALANTI

Legend: --- Valores Organizacionales, — Nathalie Vela

El área física (Valor RECTITUD) es la más importante para Nathalie Vela. El área menos importante es la intuitiva (Valor AMOR).

CLAVE

Accesibilidad: No disponible

Informatizado Valanti.xls [Modo de compatibilidad] - Excel

Inicio ses. Compartir

Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisar Vista Ayuda ¿Qué desea hacer?

G2 'Muy bajo predominio del valor verdad. La parte intelectual de sus valores es completamente ignorada, sin importar cualidades como veracidad, raciocinio, curiosidad y honestidad intelectual.'

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1							Verdad	Rectitud	Paz	Amor	No violencia								
2		0 Muy bajo			0 *		0 Muy bajo pred	Le otorga muy	Muy baja valor	Muy bajo impa	Muy bajo predom	Muy bajo predominio del valor no violencia. La parte espiritual de sus valores es completamente ignorada, sin importar cualidades como							
3		24 Bajo			24 **		24 Bajo predom	Le otorga baja	Baja valoració	Bajo impacto e	Bajo predominio	del valor no violencia. La parte espiritual de sus valores es ignorada, importándole poco cualidades como amor universa							
4		34 Promedio Bajo			34 ***		34 El predominio	La importancia	La valoración	El impacto del	El valor no violencia	no es básico. La parte espiritual de valores es puesta en segundo plano, dándole menos importancia a cualidades c							
5		44 Promedio			44 ****		44 El valor verdad	El valor rectitu	La valoración	El valor amor	El valor no violencia	está dentro de lo normal. La parte espiritual de sus valores está equilibrada, dándole adecuada importancia a cualid							
6		54 Promedio Alto			54 *****		54 Valora la verd	Le da un poco	La valoración	Impacta en su	Valora la no violencia	un poco más que el resto de la gente. Resalta la parte espiritual de sus valores, dándole buena prioridad a cualida							
7		64 Alto			64 *****		64 Valora la verd	La importancia	La valoración	El amor impac	Valora la no violencia	mucho más que el resto de la gente. Resalta la parte espiritual de sus valores, dándole muy buena prioridad a cua							
8		74 Muy alto			74 *****		74 Valora la verd	Le de importar	Valora la paz	Valora el amor	Valora la no violencia	de manera extraordinaria. Resalta fuertemente la parte espiritual de sus valores, dándole total importancia a cualid							
9							52,87466804	56,63523905	44,925886	43,19625297	51,07772332								
10								El área física (Valor RECTITUD) es la más importante para Nathalie Vela.											
11										El área menos importante es la intuitiva (Valor AMOR)									
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			
23																			
24																			
25																			
26																			
27																			
28																			
29																			
30																			
31																			
32																			
33																			
34																			

CLAVE

Accesibilidad: No disponible

130%

Anexo 07: Instrumento de recolección de datos para el llenado del dataset

MODELO DE RECOJO DE DATOS

Edad							
Experiencia laboral en meses							
Capacitaciones							
Resultado de test de VALANTI:							
Verdad	<input type="radio"/> Alto	<input type="radio"/> Bajo	<input type="radio"/> Muy alto	<input type="radio"/> Muy bajo	<input type="radio"/> Promedio	<input type="radio"/> Promedio alto	<input type="radio"/> Promedio bajo
Rectitud	<input type="radio"/> Alto	<input type="radio"/> Bajo	<input type="radio"/> Muy alto	<input type="radio"/> Muy bajo	<input type="radio"/> Promedio	<input type="radio"/> Promedio alto	<input type="radio"/> Promedio bajo
Paz	<input type="radio"/> Alto	<input type="radio"/> Bajo	<input type="radio"/> Muy alto	<input type="radio"/> Muy bajo	<input type="radio"/> Promedio	<input type="radio"/> Promedio alto	<input type="radio"/> Promedio bajo
Amor	<input type="radio"/> Alto	<input type="radio"/> Bajo	<input type="radio"/> Muy alto	<input type="radio"/> Muy bajo	<input type="radio"/> Promedio	<input type="radio"/> Promedio alto	<input type="radio"/> Promedio bajo
No violencia	<input type="radio"/> Alto	<input type="radio"/> Bajo	<input type="radio"/> Muy alto	<input type="radio"/> Muy bajo	<input type="radio"/> Promedio	<input type="radio"/> Promedio alto	<input type="radio"/> Promedio bajo
Resultado de test de Disc	<input type="radio"/> Correcto		<input type="radio"/> Dominante		<input type="radio"/> Influyente		<input type="radio"/> sensato
Resultado de test de IPV:							
Disposición general para la venta	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		
Comprensión	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		
Adaptabilidad	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		
Control de sí mismo.	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		
Tolerancia a la frustración	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		
Compatibilidad	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		
Dominancia	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		
Seguridad	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		
Actividad	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		
Sociabilidad	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		
Receptividad.	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		
Agresividad	<input type="radio"/> Alto		<input type="radio"/> Bajo		<input type="radio"/> Medio		

Anexo 08: Instrumentos de recolección de datos utilizado para el armado de atributos del dataset

INSTRUMENTO DE RECOJO DE DATOS

Nombre:

Dni:

Especialidad:

Tipo de puesto de gerencia es el accesible para una persona evaluadora:

Tipos de puesto gerencial	Observación

Posibles atributos cuantitativos a utilizar en la creación del dataset:

Nombre	Tipo	Observación	Descripción

Posibles atributos cualitativos a utilizar en la creación del dataset:

Nombre	Tipo	Observación	Descripción

Test utilizados en las evaluaciones de reclutamiento:

Test	Valores utilizados	Observaciones	

Observaciones:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Anexo 09: Interfaz de Inicio del Sistema



Anexo 10: interfaz de Listado de Candidatos

MENU

INICIO

LISTADO CANDIDATOS

SELECCION DE PERSONAL

Reclutamiento de Personal

LISTADO DE CANDIDATOS




Filtrar por DNI:

	DNI	NOMBRE	FECHA	RESULTADO
1	32651984	josé	2022-11-02	no Apto
2	54986132	harbert	2022-11-02	no Apto
3	12345678	juan	2022-11-02	no Apto
4	14963781	Manuel	2022-11-02	Apto
5	32651984	josé	2022-11-02	no Apto
6	69325696	Benjamin	2022-11-02	Apto
7	12345678	juan	2022-11-02	no Apto
8	14963781	Manuel	2022-11-02	Apto
9	32651984	josé	2022-11-02	no Apto
10	69325696	Benjamin	2022-11-02	Apto
11	12345678	juan	2022-11-02	no Apto
12	14963781	Manuel	2022-11-02	Apto
13	69325696	Benjamin	2022-11-02	Apto

Anexo 11: interfaz de Selección de Personal

The screenshot displays a web application interface for personnel selection. On the left is a dark blue sidebar with a yellow 'MENU' header. The sidebar contains three menu items: 'INICIO' with a house icon, 'LISTADO CANDIDATOS' with a person icon, and 'SELECCION DE PERSONAL' with a list icon. At the bottom of the sidebar is a white box with a blue border containing an illustration of two people and a gear, with the text 'Reclutamiento de Personal' below it. The main content area has a title 'Sistema de Selección de Personal del Área Gerencial'. Below the title, there is a label 'Seleccionar Excel:' followed by a text input field and a 'Seleccionar' button. Underneath is the label 'Resultado de los Candidatos:' followed by a large empty rectangular box. To the right of this box are two buttons: 'Ejecutar' and 'limpiar'. The interface is styled with a clean, professional look using blue and yellow accents.

Anexo 12: herramientas utilizadas para el desarrollo del software.

TIPO	HERRAMIENTA	JUSTIFICACIÓN
Herramienta de Diseño de Interfaces	pyqt5 designer 	La aplicación nos permite desarrollar aplicaciones con un entorno grafico agradable para el usuario.
IDEs (Entorno de Desarrollo Integrado)	Visual Studio Code  Visual Studio Code	Es un tipo de editor de código fuente para distintos lenguajes de programación, entre ellos: Python.
Herramienta de gestión de Base de Datos	Xampp 	Es un paquete de software libre que permite en el sistema la gestión de base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

Anexo 13: Código del programa de Selección de personal en Python

```
import sys
from diseño14 import *
from PySide2 import QtCore
from PySide2.QtCore import QPropertyAnimation
from PySide2 import QtCore, QtGui, QtWidgets
import conexion
import pandas as pd
from tkinter import messagebox
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pickle

class MiApp(QMainWindow):

    def __init__(self):
        super().__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

        #eliminar barra y de titulo - opacidad
        self.setWindowFlag(Qt.FramelessWindowHint)
        self.setWindowOpacity(1)

        #SizeGrip
        self.gripSize = 10
        self.grip = QtWidgets.QSizeGrip(self)
        self.grip.resize(self.gripSize, self.gripSize)

        # mover ventana
        self.ui.frame_superior.mouseMoveEvent = self.mover_ventana

        #acceder a las paginas
        self.ui.bt_inicio.clicked.connect(lambda:
self.ui.stackedWidget.setCurrentWidget(self.ui.page))
        self.ui.bt_dos.clicked.connect(lambda:
self.ui.stackedWidget.setCurrentWidget(self.ui.page_dos))
        self.ui.bt_dos.clicked.connect(self.actualizar)
        self.ui.bt_tres.clicked.connect(lambda:
self.ui.stackedWidget.setCurrentWidget(self.ui.page_tres))

        #control barra de titulos
        self.ui.bt_minimizar.clicked.connect(self.control_bt_minimizar)
        self.ui.bt_cerrar.clicked.connect(lambda: self.close())
```

```

        #control de Los botones de La interfaz
        self.ui.boton_seleccionar.clicked.connect(self.fn_seleccionar_arc
hivo)
        self.ui.boton_ejecutar.clicked.connect(lambda:
self.entrenamiento())
        self.ui.boton_actualizar.clicked.connect(lambda:
self.actualizar())
        self.ui.boton_buscar.clicked.connect(lambda: self.buscar())

#-----
#-----

#funcionamiento del boton de buscar
def buscar(self):

    #se obtiene posicion del item
    contenido = self.ui.comboBox_filtro.currentIndex()

    #relacionamos la posicion de item del combobox a la posicion del
item de la base de datos.
    if contenido == 0:
        pos = 5
    elif contenido == 1:
        pos = 6
    elif contenido == 2:
        pos = 7
    elif contenido == 3:
        pos = 3
    elif contenido == 4:
        pos = 2
    elif contenido == 5:
        pos = 4
    elif contenido == 6:
        pos = 1

    #obtenemos la lectura de las tablas de la base de datos
    data = conexion.listar_postulantes_por_test(pos)

    #mostramos la lectura de la base de datos en un tableWidget
    columnas = list(data.columns)
    df_fila = data.to_numpy().tolist()
    x = len(columnas)
    y = len(df_fila)
    self.ui.tableWidget_listadoC.setRowCount(y)
    self.ui.tableWidget_listadoC.setColumnCount(x)
    for j in range(x):
        encabezado = QtWidgets.QTableWidgetItem(columnas[j])
        self.ui.tableWidget_listadoC.setHorizontalHeaderItem(j, encabe
zado )

```

```

        for i in range(y):
            dato = str(df_filas[i][j])
            if dato == 'nan':
                dato = ''
            self.ui.tableWidget_listadoC.setItem(i,j,
QTableWidgetItem(dato))

#funcionamiento del boton actualizar
def actualizar(self):

    #obtenemos Lectura de La base de datos
    data = conexion.listar_postulantes()

    #mostramos La Lectura de La base de datos en un tableWidget
    columnas = list(data.columns)
    df_filas = data.to_numpy().tolist()
    x = len(columnas)
    y = len(df_filas)
    self.ui.tableWidget_listadoC.setRowCount(y)
    self.ui.tableWidget_listadoC.setColumnCount(x)
    for j in range(x):
        encabezado = QtWidgets.QTableWidgetItem(columnas[j])
        self.ui.tableWidget_listadoC.setHorizontalHeaderItem(j,encabe
zado )

        for i in range(y):
            dato = str(df_filas[i][j])
            if dato == 'nan':
                dato = ''
            self.ui.tableWidget_listadoC.setItem(i,j,
QTableWidgetItem(dato))

#funcionamiento del boton de seleccionar archivo
def fn_seleccionar_archivo(self):
    #abrimos un explorador de windows para obtener y cargar el
archivo .xlsx(excel)
    fileName, _ = QFileDialog.getOpenFileName(None, 'Seleccione el
archivo excel', '', 'XLSX Files (*.xlsx)')
    #guarda la ruta en un cuadro de texto
    self.ui.lineEdit_rutaArchivo.setText(fileName)

#funcionamiento del boton ejecutar
def entrenamiento(self):

    #guardamos la ruta del archivo

```

```

path = self.ui.lineEdit_rutaArchivo.text()

#validamos si la ruta no se seleccionó
if path == "":
    messagebox.showwarning("Error de parametros", "No se
selecciono una ruta")
else:

    #validamos si no se selecciono un checkBox

    if self.ui.checkBox_ipv.isChecked()==False and
self.ui.checkBox_valanti.isChecked()==False and
self.ui.checkBox_disc.isChecked()==False:

        messagebox.showwarning("Error de parametros", "No se
selecciono test a evaluar")

    else:

        #obtenemos La ruta
        path = self.ui.lineEdit_rutaArchivo.text()
        #obtenemos Los datos del archivo en un dataframe
        df = pd.read_excel(path,
sheet_name='Hoja1', header=None)
        #obtenemos el tamaño de las columnas del dataframe
        colum = df.shape
        col = colum[0]
        cont = 0
        #creamos un dataframe para posteriormente mostrarlo en un
"tableWidget"
        dff = pd.DataFrame()
        #validamos si se seleccionaron los "checkBox"
        if self.ui.checkBox_ipv.isChecked()==True and
self.ui.checkBox_valanti.isChecked()==True and
self.ui.checkBox_disc.isChecked()==True:

            #obtenemos el entrenamiento posteriormente guardado
en un ".pickle"
            pipeline = pickle.load(open("modelo de
entrenamiento.pickle", "rb"))

            #recorremos la base de datos
            for x in range(2,col):

                cont = cont+1
                #obtenemos variables de la fila
                dat = df.iloc[x]

```

```

        #guardamos en un array los resultados de cada
        item del postulante obtenidos de la fila
        n1=np.array([[dat[4],dat[5],dat[6],dat[7],dat[8],
dat[9],dat[10],dat[11],dat[12],dat[13],dat[14],dat[15],dat[16],dat[17],da
t[18],dat[19],dat[20],dat[21],dat[2],dat[3],dat[22]]])

        #obtenemos el resultado del entrenamiento
        r1=pipeline.predict(n1)

        #guardamos el resultado en una variable
        cadena = str(r1)

        #obtenemos validacion del entrenamiento.
        datoss =
[dat[4],dat[5],dat[6],dat[7],dat[8],dat[9],dat[10],dat[11],dat[12],dat[13
],dat[14],dat[15],dat[16],dat[17],dat[18],dat[19],dat[20],dat[21],dat[2],
dat[3],dat[22]]

        vali = conexion.consulta(datoss)

        #guardamos el resultado en una cadena
        if cadena == "[1]" or vali == 1:
            respuesta = "no Apto"
        else:
            respuesta = "Apto"

        #guardamos los resultados obtenidos en la base de
datos
        conexion.insertar_postulantes(1,datoss,str(dat[1]
),respuesta,int(dat[0]))

        #guardamos la prediccion en una variable
        nueva_fila = { 'candidatos': dat[1],
'resultados': respuesta}

        #guardamos la prediccion en un dataframe
        dff = dff.append(nueva_fila, ignore_index=True)

        elif self.ui.checkBox_ipv.isChecked()==True and
self.ui.checkBox_valanti.isChecked()==True:

            #obtenemos el entrenamiento posteriormente guardado
en un ".pickle"
            pipeline = pickle.load(open("modelo de entrenamiento
de valanti-ipv.pickle", "rb"))

            #recorremos la base de datos
            for x in range(2,col):

                cont = cont+1

```

```

        #obtenemos variables de la fila
        dat = df.iloc[x]

        #guardamos en un array los resultados de cada
        item del postulante obtenidos de la fila
        n1=np.array([[dat[4],dat[5],dat[6],dat[7],dat[8],
        dat[10],dat[11],dat[12],dat[13],dat[14],dat[15],dat[16],dat[17],dat[18],d
        at[19],dat[20],dat[21],dat[2],dat[3],dat[22]]])

        #obtenemos el resultado del entrenamiento
        r1=pipeline.predict(n1)

        #obtenemos validacion del entrenamiento.
        cadena = str(r1)
        datoss =
[dat[4],dat[5],dat[6],dat[7],dat[8],dat[9],dat[10],dat[11],dat[12],dat[13]
],dat[14],dat[15],dat[16],dat[17],dat[18],dat[19],dat[20],dat[21],dat[2],
dat[3],dat[22]]

        vali = conexion.consulta(datoss)

        #guardamos el resultado en una cadena
        if cadena == "[1]" or vali == 1:

            respuesta = "no Apto"

        else:

            respuesta = "Apto"

        #guardamos los resultados obtenidos en la base de
datos
        conexion.insertar_postulantes(2,datoss,str(dat[1]
),respuesta,int(dat[0]))

        #guardamos la prediccion en una variable
nueva_fila = { 'candidatos': dat[1],
'resultados': respuesta}

        #guardamos la prediccion en un dataframe
dff = dff.append(nueva_fila, ignore_index=True)

        elif self.ui.checkBox_disc.isChecked()==True and
self.ui.checkBox_valanti.isChecked()==True:

            #self.ui.textBrowser_resultado.append("esto es
valanti y disc")

```

```

#obtenemos el entrenamiento posteriormente guardado
en un ".pickle"
pipeline = pickle.load(open("modelo de entrenamiento
de valanti-disc.pickle", "rb"))

#recorremos la base de datos
for x in range(2,col):

    cont = cont+1

    #obtenemos variables de la fila
    dat = df.iloc[x]

    #guardamos en un array los resultados de cada
    item del postulante obtenidos de la fila
    n1=np.array([[dat[4],dat[5],dat[6],dat[7],dat[8],
dat[9],dat[2],dat[3],dat[22]]])

    #obtenemos el resultado del entrenamiento
    r1=pipeline.predict(n1)

    #obtenemos validacion del entrenamiento.
    cadena = str(r1)
    datoss =
[dat[4],dat[5],dat[6],dat[7],dat[8],dat[9],dat[10],dat[11],dat[12],dat[13
],dat[14],dat[15],dat[16],dat[17],dat[18],dat[19],dat[20],dat[21],dat[2],
dat[3],dat[22]]

    vali = conexion.consulta(datoss)

    #guardamos el resultado en una cadena
    if cadena == "[1]" or vali == 1:

        respuesta = "no Apto"

    else:

        respuesta = "Apto"

    #guardamos los resultados obtenidos en la base de
datos
    conexion.insertar_postulantes(3,datoss,str(dat[1]
),respuesta,int(dat[0]))

    #guardamos la prediccion en una variable
nueva_fila = { 'candidatos': dat[0],
'resultados': respuesta}

#guardamos la prediccion en un dataframe

```



```

        dff = dff.append(nueva_fila, ignore_index=True)

        elif self.ui.checkBox_ipv.isChecked()==True and
self.ui.checkBox_disc.isChecked()==True:

            #self.ui.textBrowser_resultado.append("esto es disc y
            ipv")

            #obtenemos el entrenamiento posteriormente guardado
            en un ".pickle"
            pipeline = pickle.load(open("modelo de entrenamiento
            de disc-ipv.pickle", "rb"))

            #recorremos la base de datos
            for x in range(2,col):

                cont = cont+1

                #obtenemos variables de la fila
                dat = df.iloc[x]

                #guardamos en un array los resultados de cada
                item del postulante obtenidos de la fila
                n1=np.array([[dat[9],dat[10],dat[11],dat[12],dat[
                13],dat[14],dat[15],dat[16],dat[17],dat[18],dat[19],dat[20],dat[21],dat[2
                ],dat[3],dat[22]]])

                #obtenemos el resultado del entrenamiento
                r1=pipeline.predict(n1)

                #obtenemos validacion del entrenamiento.
                cadena = str(r1)
                datoss =
                [dat[4],dat[5],dat[6],dat[7],dat[8],dat[9],dat[10],dat[11],dat[12],dat[13
                ],dat[14],dat[15],dat[16],dat[17],dat[18],dat[19],dat[20],dat[21],dat[2],
                dat[3],dat[22]]

                vali = conexion.consulta(datoss)

                #guardamos el resultado en una cadena
                if cadena == "[1]" or vali == 1:
                    respuesta = "no Apto"
                else:
                    respuesta = "Apto"

                #guardamos los resultados obtenidos en la base de
                datos
                conexion.insertar_postulantes(4,datoss,str(dat[1]
                ),respuesta,int(dat[0]))

```

```

        #guardamos la prediccion en una variable
        nueva_fila = { 'candidatos': dat[0],
'resultados': respuesta}

        #guardamos la prediccion en un dataframe
        dff = dff.append(nueva_fila, ignore_index=True)

        #print("esto es valanti y ipv")
        elif self.ui.checkBox_valanti.isChecked()==True:
            #self.ui.textBrowser_resultado.append("esto es
valanti")

            #obtenemos el entrenamiento posteriormente guardado
en un ".pickle"
            pipeline = pickle.load(open("modelo de entrenamiento
de valanti.pickle", "rb"))

            #recorremos la base de datos
            for x in range(2,col):

                cont = cont+1

                #obtenemos variables de la fila
                dat = df.iloc[x]

                #guardamos en un array los resultados de cada
item del postulante obtenidos de la fila
                n1=np.array([[dat[4],dat[5],dat[6],dat[7],dat[8],
dat[2],dat[3],dat[22]]])

                #obtenemos el resultado del entrenamiento
                r1=pipeline.predict(n1)

                #obtenemos validacion del entrenamiento.
                cadena = str(r1)
                datoss =
[dat[4],dat[5],dat[6],dat[7],dat[8],dat[9],dat[10],dat[11],dat[12],dat[13
],dat[14],dat[15],dat[16],dat[17],dat[18],dat[19],dat[20],dat[21],dat[2],
dat[3],dat[22]]

                vali = conexion.consulta(datoss)

                #guardamos el resultado en una cadena
                if cadena == "[1]" or vali == 1:
                    respuesta = "no Apto"
                else:
                    respuesta = "Apto"

```

```

#guardamos Los resultados obtenidos en La base de
datos
conexion.insertar_postulantes(5,datoss,str(dat[1]
),respuesta,int(dat[0]))

#guardamos La prediccion en una variable
nueva_fila = { 'candidatos': dat[0],
'resultados': respuesta}

#guardamos La prediccion en un dataframe
dff = dff.append(nueva_fila, ignore_index=True)

#print("esto en valanti")
elif self.ui.checkBox_disc.isChecked()==True:
#self.ui.textBrowser_resultado.append("esto es disc")

#obtenemos el entrenamiento posteriormente guardado
en un ".pickle"
pipeline = pickle.load(open("modelo de entrenamiento
de disc.pickle", "rb"))

#recorremos La base de datos
for x in range(2,col):

    cont = cont+1

#obtenemos variables de la fila
dat = df.iloc[x]

#guardamos en un array Los resultados de cada
item del postulante obtenidos de La fila
n1=np.array([[dat[9],dat[2],dat[3],dat[22]]])

#obtenemos el resultado del entrenamiento
r1=pipeline.predict(n1)

#obtenemos validacion del entrenamiento.
cadena = str(r1)
datoss =
[dat[4],dat[5],dat[6],dat[7],dat[8],dat[9],dat[10],dat[11],dat[12],dat[13
],dat[14],dat[15],dat[16],dat[17],dat[18],dat[19],dat[20],dat[21],dat[2],
dat[3],dat[22]]

vali = conexion.consulta(datoss)

#guardamos el resultado en una cadena
if cadena == "[1]" or vali == 1:
    respuesta = "no Apto"
else:

```

```

        respuesta = "Apto"

        #guardamos los resultados obtenidos en la base de
datos
        conexion.insertar_postulantes(6,datoss,str(dat[1]
),respuesta,int(dat[0]))

        #guardamos la prediccion en una variable
nueva_fila = { 'candidatos': dat[0],
'resultados': respuesta}

        #guardamos la prediccion en un dataframe
dff = dff.append(nueva_fila, ignore_index=True)

        #print("esto en disc")
elif self.ui.checkBox_ipv.isChecked()==True:
        #self.ui.textBrowser_resultado.append("esto es ipv")

        #obtenemos el entrenamiento posteriormente guardado
en un ".pickle"
        pipeline = pickle.load(open("modelo de entrenamiento
de ipv.pickle", "rb"))

        #recorremos la base de datos
for x in range(2,col):

        cont = cont+1

        #obtenemos variables de la fila
dat = df.iloc[x]

        #guardamos en un array los resultados de cada
item del postulante obtenidos de la fila
n1=np.array([[dat[10],dat[11],dat[12],dat[13],dat
[14],dat[15],dat[16],dat[17],dat[18],dat[19],dat[20],dat[21],dat[2
],dat[3
],dat[22]])

        #obtenemos el resultado del entrenamiento
r1=pipeline.predict(n1)

        #obtenemos validacion del entrenamiento.
cadena = str(r1)
datoss =
[dat[4],dat[5],dat[6],dat[7],dat[8],dat[9],dat[10],dat[11],dat[12],dat[13
],dat[14],dat[15],dat[16],dat[17],dat[18],dat[19],dat[20],dat[21],dat[2
],dat[3],dat[22]]

        vali = conexion.consulta(datoss)

```

```

        #guardamos el resultado en una cadena
        if cadena == "[1]" or vali == 1:
            respuesta = "no Apto"
        else:
            respuesta = "Apto"

        #guardamos los resultados obtenidos en la base de
datos
        conexion.insertar_postulantes(7, datoss, str(dat[1]
), respuesta, int(dat[0]))

        #guardamos la prediccion en una variable
nueva_fila = { 'candidatos': dat[0],
'resultados': respuesta}

        #guardamos la prediccion en un dataframe
dff = dff.append(nueva_fila, ignore_index=True)

        #mostramos el resultado obtenido para mostrarlo en un
tableWidget
        columnas = list(dff.columns)
        df_fila = dff.to_numpy().tolist()
        x = len(columnas)
        y = len(df_fila)
        self.ui.tableWidget.setRowCount(y)
        self.ui.tableWidget.setColumnCount(x)

        for j in range(x):
            encabezado = QtWidgets.QTableWidgetItem(columnas[j])
            self.ui.tableWidget.setHorizontalHeaderItem(j, encabezado )

            for i in range(y):
                dato = str(df_fila[i][j])
                if dato == 'nan':
                    dato = ''
                self.ui.tableWidget.setItem(i, j,
QtTableWidgetItem(dato))

        #funcionamiento del boton minimizar
def control_bt_minimizar(self):
    self.showMinimized()

        #funcionamiento del boton del menu
def mover_menu(self):
    if True:
        width = self.ui.frame_lateral.width()
        normal = 0
        if width==0:
            extender = 200

```

```

        else:
            extender = normal
            self.animacion = QPropertyAnimation(self.ui.frame_lateral,
b'minimumWidth')
            self.animacion.setDuration(300)
            self.animacion.setStartValue(width)
            self.animacion.setEndValue(extender)
            self.animacion.setEasingCurve(QtCore.QEasingCurve.InOutQuart)
            self.animacion.start()

    ## tamaño el grip SizeGrip
    def resizeEvent(self, event):
        rect = self.rect()
        self.grip.move(rect.right() - self.gripSize, rect.bottom() -
self.gripSize)

    # mover ventana
    def mousePressEvent(self, event):
        self.clickPosition = event.globalPos()

    def mover_ventana(self, event):
        if self.isMaximized() == False:
            if event.buttons() == Qt.LeftButton:
                self.move(self.pos() + event.globalPos() -
self.clickPosition)
                self.clickPosition = event.globalPos()
                event.accept()

            if event.globalPos().y() <=20:
                self.showMaximized()
            else:
                self.showNormal()

if __name__ == "__main__":
    app = QApplication(sys.argv)
    mi_app = MiApp()
    mi_app.show()
    sys.exit(app.exec_())

```

Anexo 14: Código para la conexión con la base de datos.

```
from traceback import print_tb
import pyodbc
import pandas as pd

#se realiza conexion con la base de datos
server='localhost'
db='seleccion_personal'
usuario='admin'
contrasena='admin'
try:
    #conexion = pyodbc.connect('DRIVER={ODBC Driver 17 for Sql server};
SERVER='+server+'; DATABASE='+db+'UID='+usuario+'; PWD='+contrasena)
    conexion = pyodbc.connect('DRIVER={MySQL ODBC 8.0 ANSI Driver};
SERVER='+server+'; DATABASE='+db+';UID='+usuario+'; PWD='+contrasena)
    #print('conexion exitosa')
except:
    #print('error al conectar con la base de datos')
    pass

#funcion para listar postulantes
def listar_postulantes():

    #se crea un cursor para hacer consulta de la base de datos
    consulta = conexion.cursor()

    #se crea la consulta.
    consulta.execute("SELECT p.dni, p.nombre,t.nombre_TipoE as
'Entrenamiento realizado', i.item as 'resultado' FROM persona p INNER
JOIN entrenamiento e on p.dni = e.idpersona INNER JOIN item i on
i.idEntrenamiento = e.idEntrenamiento INNER JOIN tipo_entrenamiento t on
t.idTipoEntrenamiento = e.idTipoEntrenamiento WHERE i.idvariable = 21;")

    #obtenemos la respuesta de la consulta.
    respuesta = consulta.fetchall()

    #creamos un dataframe con una columna que contiene los titulos
    df = pd.DataFrame(columns=['dni','nombre','Entrenamiento
realizado','resultado'])

    #creamos un while para guardar el resultado de la consulta en el
dataframe
    while respuesta:
        dni = respuesta[0]
        nombre = respuesta[1]
        entre_real = respuesta[2]
```

```

        resultado = respuesta[3]
        df = df.append({'dni': str(dni), 'nombre':str(nombre),
'Entrenamiento realizado':str(entre_reali),'resultado': str(resultado)},
ignore_index=True)
        respuesta = consulta.fetchone()

#devolvemos el dataframe.
return df

#listar los postulantes por tipo de test
def listar_postulantes_por_test(tipoTest):

#se crea un cursor para hacer consulta de la base de datos
consulta = conexion.cursor()

#se crea la consulta.
consulta.execute("SELECT p.dni, p.nombre,t.nombre_TipoE as
'Entrenamiento realizado', i.item as 'resultado' FROM persona p INNER
JOIN entrenamiento e on p.dni = e.idpersona INNER JOIN item i on
i.idEntrenamiento = e.idEntrenamiento INNER JOIN tipo_entrenamiento t on
t.idTipoEntrenamiento = e.idTipoEntrenamiento WHERE i.idvariable = 21 and
t.idTipoEntrenamiento = "+str(tipoTest)+"");

#obtenemos la respuesta de la consulta.
respuesta = consulta.fetchone()

#creamos un dataframe con una columna que contiene los títulos
df = pd.DataFrame(columns=['dni', 'nombre', 'Entrenamiento
realizado', 'resultado'])

#creamos un while para guardar el resultado de la consulta en el
dataframe
while respuesta:
    #print(respuesta)
    dni = respuesta[0]
    nombre = respuesta[1]
    entre_reali = respuesta[2]
    resultado = respuesta[3]
    df = df.append({'dni': str(dni), 'nombre':str(nombre),
'Entrenamiento realizado':str(entre_reali),'resultado': str(resultado)},
ignore_index=True)
    respuesta = consulta.fetchone()

#devolvemos el dataframe.
return df

#funcion para insertar los postulantes
def insertar_postulantes(tipodato,datos,nombre,resultado,idper):

```



```

#consultar dni si existe si no crear otro
consulta = conexion.cursor()

#se crea la consulta.
consulta.execute("select dni from persona where dni="+str(idper)+"");

#obtenemos la respuesta de la consulta.
respuesta = consulta.fetchone()

#validamos la respuesta de la consulta
inter = 0
if respuesta is not None:
    inter = 1
else:
    #insertamos los datos a la base de datos.
    consulta0 = conexion.cursor()
    consultasql0= "INSERT INTO persona
(dni,nombre,edad,idTipoPersona) values (?,?,,?);"
    consulta0.execute(consultasql0,idper,str(nombre),datos[18],2)
    consulta0.commit()

    #se hace una consulta a la base de datos para obtener el ultimo id
de la tabla entrenamiento
    consultae = conexion.cursor()
    consultae.execute("SELECT identrenamiento FROM entrenamiento ORDER by
identrenamiento DESC LIMIT 1;")
    respuestaEn = consultae.fetchone()

    #aumentamos el id +1
    if respuestaEn is not None:
        respuestaE = respuestaEn[0]+1
    else:
        respuestaE = 1

#validamos si el tipo de dato es de los 3 test
if tipodato == 1:

    #se hace una inserción a la tabla "entrenamiento" con los datos
personales del postulante
    consulta1 = conexion.cursor()
    consultasql1 = "INSERT INTO entrenamiento
(idEntrenamiento,idPersona,idTipoEntrenamiento) values (?,?,,?);"
    consulta1.execute(consultasql1,respuestaE,idper,1)
    consulta1.commit()

    #se hace una inserción a la tabla "item" conteniendo los datos de
los test del postulante.
    consulta2 = conexion.cursor()

```

```

consultasql2 = "INSERT INTO item
(item,idEntrenamiento,idvariable) values (?,?,?);"
consulta2.execute(consultasql2,str(datos[0]),respuestaE,1)
consulta2.execute(consultasql2,str(datos[1]),respuestaE,2)
consulta2.execute(consultasql2,str(datos[2]),respuestaE,3)
consulta2.execute(consultasql2,str(datos[3]),respuestaE,4)
consulta2.execute(consultasql2,str(datos[4]),respuestaE,5)
consulta2.execute(consultasql2,str(datos[5]),respuestaE,6)
consulta2.execute(consultasql2,str(datos[6]),respuestaE,7)
consulta2.execute(consultasql2,str(datos[7]),respuestaE,8)
consulta2.execute(consultasql2,str(datos[8]),respuestaE,9)
consulta2.execute(consultasql2,str(datos[9]),respuestaE,10)
consulta2.execute(consultasql2,str(datos[10]),respuestaE,11)
consulta2.execute(consultasql2,str(datos[11]),respuestaE,12)
consulta2.execute(consultasql2,str(datos[12]),respuestaE,13)
consulta2.execute(consultasql2,str(datos[13]),respuestaE,14)
consulta2.execute(consultasql2,str(datos[14]),respuestaE,15)
consulta2.execute(consultasql2,str(datos[15]),respuestaE,16)
consulta2.execute(consultasql2,str(datos[16]),respuestaE,17)
consulta2.execute(consultasql2,str(datos[17]),respuestaE,18)
consulta2.execute(consultasql2,str(datos[19]),respuestaE,19)
consulta2.execute(consultasql2,str(datos[20]),respuestaE,20)
consulta2.execute(consultasql2,str(resultado),respuestaE,21)
consulta2.commit()

#validamos si el tipo de dato es de valanti e IPV
elif tipodato == 2:

    #se hace una inserción a la tabla "entrenamiento" con los datos
    personales del postulante
    consulta3 = conexion.cursor()
    consultasql3 = "INSERT INTO entrenamiento
(idEntrenamiento,idPersona,idTipoEntrenamiento) values (?,?,?);"
    consulta3.execute(consultasql3,respuestaE,idper,2)
    consulta3.commit()

    #se hace una inserción a la tabla "item" conteniendo los datos de
    los test del postulante.
    consulta4 = conexion.cursor()
    consultasql4 = "INSERT INTO item
(item,idEntrenamiento,idvariable) values (?,?,?);"
    consulta4.execute(consultasql4,str(datos[0]),respuestaE,1)
    consulta4.execute(consultasql4,str(datos[1]),respuestaE,2)
    consulta4.execute(consultasql4,str(datos[2]),respuestaE,3)
    consulta4.execute(consultasql4,str(datos[3]),respuestaE,4)
    consulta4.execute(consultasql4,str(datos[4]),respuestaE,5)
    consulta4.execute(consultasql4,str(datos[6]),respuestaE,7)
    consulta4.execute(consultasql4,str(datos[7]),respuestaE,8)
    consulta4.execute(consultasql4,str(datos[8]),respuestaE,9)

```

```

consulta4.execute(consultasql4, str(datos[9]), respuestaE, 10)
consulta4.execute(consultasql4, str(datos[10]), respuestaE, 11)
consulta4.execute(consultasql4, str(datos[11]), respuestaE, 12)
consulta4.execute(consultasql4, str(datos[12]), respuestaE, 13)
consulta4.execute(consultasql4, str(datos[13]), respuestaE, 14)
consulta4.execute(consultasql4, str(datos[14]), respuestaE, 15)
consulta4.execute(consultasql4, str(datos[15]), respuestaE, 16)
consulta4.execute(consultasql4, str(datos[16]), respuestaE, 17)
consulta4.execute(consultasql4, str(datos[17]), respuestaE, 18)
consulta4.execute(consultasql4, str(datos[19]), respuestaE, 19)
consulta4.execute(consultasql4, str(datos[20]), respuestaE, 20)
consulta4.execute(consultasql4, str(resultado), respuestaE, 21)
consulta4.commit()
#consulta4.close()

```

```

#validamos si el tipo de dato es de valanti y Disc
elif tipodato == 3:

```

#se hace una inserción a la tabla "entrenamiento" con Los datos personales del postulante

```

consulta5 = conexion.cursor()
consultasql5 = "INSERT INTO entrenamiento
(idEntrenamiento,idPersona,idTipoEntrenamiento) values (?,?,?);"
consulta5.execute(consultasql5, respuestaE, idper, 3)
consulta5.commit()

```

#se hace una inserción a la tabla "item" conteniendo Los datos de Los test del postulante.

```

consulta6 = conexion.cursor()
consultasql6 = "INSERT INTO item
(item,idEntrenamiento,idvariable) values (?,?,?);"
consulta6.execute(consultasql6, str(datos[0]), respuestaE, 1)
consulta6.execute(consultasql6, str(datos[1]), respuestaE, 2)
consulta6.execute(consultasql6, str(datos[2]), respuestaE, 3)
consulta6.execute(consultasql6, str(datos[3]), respuestaE, 4)
consulta6.execute(consultasql6, str(datos[4]), respuestaE, 5)
consulta6.execute(consultasql6, str(datos[5]), respuestaE, 6)
consulta6.execute(consultasql6, str(datos[19]), respuestaE, 19)
consulta6.execute(consultasql6, str(datos[20]), respuestaE, 20)
consulta6.execute(consultasql6, str(resultado), respuestaE, 21)
consulta6.commit()
#consulta6.close()

```

```

#validamos si el tipo de dato es de Disc e IPV
elif tipodato == 4:

```

```

#se hace una inserción a La tabla "entrenamiento" con Los datos
personales del postulante
consulta7 = conexion.cursor()
consultasql7 = "INSERT INTO entrenamiento
(idEntrenamiento,idPersona,idTipoEntrenamiento) values (?,?,?);"
consulta7.execute(consultasql7,respuestaE,idper,4)
consulta7.commit()

#se hace una inserción a La tabla "item" conteniendo Los datos de
Los test del postulante.
consulta8 = conexion.cursor()
consultasql8 = "INSERT INTO item
(item,idEntrenamiento,idvariable) values (?,?,?);"
consulta8.execute(consultasql8,str(datos[5]),respuestaE,6)
consulta8.execute(consultasql8,str(datos[6]),respuestaE,7)
consulta8.execute(consultasql8,str(datos[7]),respuestaE,8)
consulta8.execute(consultasql8,str(datos[8]),respuestaE,9)
consulta8.execute(consultasql8,str(datos[9]),respuestaE,10)
consulta8.execute(consultasql8,str(datos[10]),respuestaE,11)
consulta8.execute(consultasql8,str(datos[11]),respuestaE,12)
consulta8.execute(consultasql8,str(datos[12]),respuestaE,13)
consulta8.execute(consultasql8,str(datos[13]),respuestaE,14)
consulta8.execute(consultasql8,str(datos[14]),respuestaE,15)
consulta8.execute(consultasql8,str(datos[15]),respuestaE,16)
consulta8.execute(consultasql8,str(datos[16]),respuestaE,17)
consulta8.execute(consultasql8,str(datos[17]),respuestaE,18)
consulta8.execute(consultasql8,str(datos[19]),respuestaE,19)
consulta8.execute(consultasql8,str(datos[20]),respuestaE,20)
consulta8.execute(consultasql8,str(resultado),respuestaE,21)
consulta8.commit()
#consulta8.close()

#validamos si el tipo de dato es de Valanti
elif tipodato == 5:

#se hace una inserción a La tabla "entrenamiento" con Los datos
personales del postulante
consulta9 = conexion.cursor()
consultasql9 = "INSERT INTO entrenamiento
(idEntrenamiento,idPersona,idTipoEntrenamiento) values (?,?,?);"
consulta9.execute(consultasql9,respuestaE,idper,5)
consulta9.commit()

#se hace una inserción a La tabla "item" conteniendo Los datos de
Los test del postulante.
consulta10 = conexion.cursor()
consultasql10 = "INSERT INTO item
(item,idEntrenamiento,idvariable) values (?,?,?);"

```

```

consulta10.execute(consultasql10,str(datos[0]),respuestaE,1)
consulta10.execute(consultasql10,str(datos[1]),respuestaE,2)
consulta10.execute(consultasql10,str(datos[2]),respuestaE,3)
consulta10.execute(consultasql10,str(datos[3]),respuestaE,4)
consulta10.execute(consultasql10,str(datos[4]),respuestaE,5)
consulta10.execute(consultasql10,str(datos[19]),respuestaE,19)
consulta10.execute(consultasql10,str(datos[20]),respuestaE,20)
consulta10.execute(consultasql10,str(resultado),respuestaE,21)
consulta10.commit()

#validamos si el tipo de dato es de Disc
elif tipodato == 6:

    #se hace una inserción a La tabla "entrenamiento" con Los datos
    personales del postulante
    consulta11 = conexion.cursor()
    consultasql11 = "INSERT INTO entrenamiento
(idEntrenamiento,idPersona,idTipoEntrenamiento) values (?,?,?);"
    consulta11.execute(consultasql11,respuestaE,idper,6)
    consulta11.commit()

    #se hace una inserción a La tabla "item" conteniendo Los datos de
    Los test del postulante.
    consulta12 = conexion.cursor()
    consultasql12 = "INSERT INTO item
(item,idEntrenamiento,idvariable) values (?,?,?);"
    consulta12.execute(consultasql12,str(datos[5]),respuestaE,6)
    consulta12.execute(consultasql12,str(datos[19]),respuestaE,19)
    consulta12.execute(consultasql12,str(datos[20]),respuestaE,20)
    consulta12.execute(consultasql12,str(resultado),respuestaE,21)
    consulta12.commit()

#validamos si el tipo de dato es de IPV
elif tipodato == 7:

    #se hace una inserción a La tabla "entrenamiento" con Los datos
    personales del postulant
    consulta13 = conexion.cursor()
    consultasql13 = "INSERT INTO entrenamiento
(idEntrenamiento,idPersona,idTipoEntrenamiento) values (?,?,?);"
    consulta13.execute(consultasql13,respuestaE,idper,7)
    consulta13.commit()

    #se hace una inserción a La tabla "item" conteniendo Los datos de
    Los test del postulante.
    consulta14 = conexion.cursor()

```

```
consultasql14 = "INSERT INTO item
(item,idEntrenamiento,idvariable) values (?,?,?);"
consulta14.execute(consultasql14,str(datos[6]),respuestaE,7)
consulta14.execute(consultasql14,str(datos[7]),respuestaE,8)
consulta14.execute(consultasql14,str(datos[8]),respuestaE,9)
consulta14.execute(consultasql14,str(datos[9]),respuestaE,10)
consulta14.execute(consultasql14,str(datos[10]),respuestaE,11)
consulta14.execute(consultasql14,str(datos[11]),respuestaE,12)
consulta14.execute(consultasql14,str(datos[12]),respuestaE,13)
consulta14.execute(consultasql14,str(datos[13]),respuestaE,14)
consulta14.execute(consultasql14,str(datos[14]),respuestaE,15)
consulta14.execute(consultasql14,str(datos[15]),respuestaE,16)
consulta14.execute(consultasql14,str(datos[16]),respuestaE,17)
consulta14.execute(consultasql14,str(datos[17]),respuestaE,18)
consulta14.execute(consultasql14,str(datos[19]),respuestaE,19)
consulta14.execute(consultasql14,str(datos[20]),respuestaE,20)
consulta14.execute(consultasql14,str(resultado),respuestaE,21)
consulta14.commit()
```

Anexo 15: Código para la selección del Algoritmo Naive Bayes.

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
from sklearn.metrics import recall_score

dataset = "HR2.csv"
dataframe = pd.read_csv(dataset)
print(dataframe.head())

dataframe_cat = dataframe.select_dtypes(include=[object])
dataframe_num = dataframe.select_dtypes(include=np.number)

#----verificar si los datos de contienen datos null
print("")
print("Datos de tipo null en las columnas de tipo cadena")
print("")
print(dataframe_cat.isnull().sum())
print("")
print("Datos de tipo null en las columnas de tipo numerico")
print("")
print(dataframe_num.isnull().sum())

#----conversion de variables categoricas (cadenas) en numeros
le = LabelEncoder()
dataframe_cat = dataframe_cat.apply(le.fit_transform)
print("")
print("datos de tipo cadena convertidos en numericos")
print("")
print(dataframe_cat.head(15))

#-----
concatenar los datos numericos y categoricos ya convertidos en numericos
dataframe = pd.concat([dataframe_cat,dataframe_num],axis=1)
print("")
print("listado de los datos ya convertidos")
print("")
print(dataframe.head())

X_train, X_test = train_test_split(dataframe, test_size=0.2, random_state=100)
y_train =X_train["left"]
y_test = X_test["left"]
```

```
# Instantiate the classifier
gnb = GaussianNB()
# predicción en el conjunto de prueba
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)

# Calcular la exactitud del modelo
print("Exactitud:", accuracy_score(y_test, y_pred))

print("recall:", recall_score(y_test,y_pred,pos_label='positive',average='micro'))
```


Anexo 16: Código para la selección del Algoritmo Regresión Logística

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
from sklearn.metrics import recall_score

dataset = "HR2.csv"
dataframe = pd.read_csv(dataset)
print(dataframe.head())

dataframe_cat = dataframe.select_dtypes(include=[object])
dataframe_num = dataframe.select_dtypes(include=np.number)

#----verificar si los datos de contienen datos null
print("")
print("Datos de tipo null en las columnas de tipo cadena")
print("")
print(dataframe_cat.isnull().sum())
print("")
print("Datos de tipo null en las columnas de tipo numerico")
print("")
print(dataframe_num.isnull().sum())

#----conversion de variables categoricas (cadenas) en numeros
le = LabelEncoder()
dataframe_cat = dataframe_cat.apply(le.fit_transform)
print("")
print("datos de tipo cadena convertidos en numericos")
print("")
print(dataframe_cat.head(15))

#-----
concatenar los datos numericos y categoricos ya convertidos en numericos
dataframe = pd.concat([dataframe_cat,dataframe_num],axis=1)
print("")
print("listado de los datos ya convertidos")
print("")
print(dataframe.head())

X_train, X_test = train_test_split(dataframe, test_size=0.2, random_state=100)
```

```
y_train =X_train["clasificador"]
y_test = X_test["clasificador"]

from sklearn import linear_model

model = linear_model.LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

# Calcular la Exactitud del modelo
print("Exactitud :", accuracy_score(y_test, y_pred))

print("recall:", recall_score(y_test,y_pred,pos_label='positive',average='micro'))
```

Anexo 17: Código para la selección del Algoritmo Random Forest

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

import time
inicio = time.time()

dataset = "HR.csv"
dataframe = pd.read_csv(dataset)
print(dataframe.head())

dataframe_cat = dataframe.select_dtypes(include=[object])
dataframe_num = dataframe.select_dtypes(include=np.number)

#----verificar si los datos de contienen datos null
print("")
print("Datos de tipo null en las columnas de tipo cadena")
print("")
print(dataframe_cat.isnull().sum())
print("")
print("Datos de tipo null en las columnas de tipo numerico")
print("")
print(dataframe_num.isnull().sum())

#----conversion de variables categoricas (cadenas) en numeros
le = LabelEncoder()
dataframe_cat = dataframe_cat.apply(le.fit_transform)
print("")
print("datos de tipo cadena convertidos en numericos")
print("")
print(dataframe_cat.head(15))

#-----
concatenar los datos numericos y categoricos ya convertidos en numericos
dataframe = pd.concat([dataframe_cat,dataframe_num],axis=1)
print("")
print("listado de los datos ya convertidos")
print("")
print(dataframe.head())

#separacion de modelos para entrenamiento y prueba
```

```
X_train, X_test = train_test_split(dataframe, test_size=0.2, random_state
=0)
y_train =X_train["left"]
y_test = X_test["left"]

# crear el clasificador
clasificador = RandomForestClassifier(n_estimators=100)

# Entrenar el modelo usando el conjunto de entranamiento
clasificador.fit(X_train, y_train)

# predicción en el conjunto de prueba
y_pred = clasificador.predict(X_test)

# Calcular la exactitud del modelo
print("Exactitud:", accuracy_score(y_test, y_pred))

fin = time.time()
print("Total de tiempo de ejecución",fin-inicio)
```

Anexo 18: Código para la selección del algoritmo Support Vector Machines (SVM)

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score

import time
inicio = time.time()
dataset = "HR.csv"
dataframe = pd.read_csv(dataset)
print(dataframe.head())

dataframe_cat = dataframe.select_dtypes(include=[object])
dataframe_num = dataframe.select_dtypes(include=np.number)

#-----verificar si los datos de contienen datos null
print("")
print("Datos de tipo null en las columnas de tipo cadena")
print("")
print(dataframe_cat.isnull().sum())
print("")
print("Datos de tipo null en las columnas de tipo numerico")
print("")
print(dataframe_num.isnull().sum())

#-----conversion de variables categoricas (cadenas) en numeros
le = LabelEncoder()
dataframe_cat = dataframe_cat.apply(le.fit_transform)
print("")
print("datos de tipo cadena convertidos en numericos")
print("")
print(dataframe_cat.head(15))

#-----
concatenar los datos numericos y categoricos ya convertidos en numericos
dataframe = pd.concat([dataframe_cat,dataframe_num],axis=1)
print("")
print("listado de los datos ya convertidos")
print("")
print(dataframe.head())

#separacion de modelos para entrenamiento y prueba
```

```
X_train, X_test = train_test_split(dataframe, test_size=0.2, random_state
=0)
y_train =X_train["left"]
y_test = X_test["left"]

# crear el clasificador
classifier = svm.SVC(gamma = "auto", kernel = "rbf")

# Entrenar el modelo usando el conjunto de entranamiento
classifier.fit(X_train, y_train)

# predicción en el conjunto de prueba
y_pred = classifier.predict(X_test)

# Calcular la Exactitud del modelo
print("Exactitud:", accuracy_score(y_test, y_pred))

fin = time.time()
print("Total de tiempo de ejecución",fin-inicio)
```

Anexo 19: Código para la selección del algoritmo k-Nearest Neighbor (KNN)

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

import time
inicio = time.time()

dataset = "HR.csv"
dataframe = pd.read_csv(dataset)
print(dataframe.head())

dataframe_cat = dataframe.select_dtypes(include=[object])
dataframe_num = dataframe.select_dtypes(include=np.number)

#-----verificar si los datos de contienen datos null
print("")
print("Datos de tipo null en las columnas de tipo cadena")
print("")
print(dataframe_cat.isnull().sum())
print("")
print("Datos de tipo null en las columnas de tipo numerico")
print("")
print(dataframe_num.isnull().sum())

#-----conversion de variables categoricas (cadenas) en numeros
le = LabelEncoder()
dataframe_cat = dataframe_cat.apply(le.fit_transform)
print("")
print("datos de tipo cadena convertidos en numéricos")
print("")
print(dataframe_cat.head(15))

#-----
concatenar los datos numericos y categoricos ya convertidos en numericos
dataframe = pd.concat([dataframe_cat,dataframe_num],axis=1)
print("")
print("listado de los datos ya convertidos")
print("")
print(dataframe.head())

#separacion de modelos para entrenamiento y prueba
```

```
X_train, X_test = train_test_split(dataframe, test_size=0.2, random_state
=0)
y_train =X_train["left"]
y_test = X_test["left"]

# crear el clasificador
n_neighbors = 1
clasificador = KNeighborsClassifier(n_neighbors)

# Entrenar el modelo usando el conjunto de entranamiento
clasificador.fit(X_train, y_train)

# predicción en el conjunto de prueba
y_pred = clasificador.predict(X_test)

# Calcular la exactitud del modelo
print("exactitud:", accuracy_score(y_test, y_pred))

fin = time.time()
print("Total de tiempo de ejecución",fin-inicio)
```