

**Pontificia Universidad Católica del Ecuador**

**Facultad De Ingeniería**

**Carrera de Sistemas de Información**



**TEMA:**

Análisis del impacto de Inteligencias Artificiales en la experiencia de los usuarios universitarios:  
un enfoque en la minería de datos y el análisis de sentimiento. Caso de estudio: ChatGPT.

**AUTOR:**

Emilio Alejandro Salazar Espinosa

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS DE  
INFORMACIÓN

**QUITO, JUNIO 2023**

## DEDICATORIA

---

A mi familia, por ser el motor que me mueve día a día, empezando por la persona más importante en mi vida, mi madre; gracias a quien soy la persona que soy, gracias a quien he logrado aprovechar todas las oportunidades. También a mi hermano, sé que con tu esfuerzo también vivirás estas maravillosas experiencias de la vida. A mi abuela, tíos y primos. También agradezco mucho a mis amistades, colegas, docentes como el Mgtr. Alfredo Calderón, de quien fui ayudante de cátedra, y demás personas que han sido parte del largo camino que me ha llevado hasta este punto, todos han aportado grandes cosas a mi vida.

También a todos aquellos que ya no están, en especial a mi padre y a mi abuelo, ustedes que con sus enseñanzas dejaron una gran huella en mí y espero que desde el paraíso disfruten este logro, ya que sin ustedes no lo hubiera logrado.

Y de todas las personas que la Universidad me permitió conocer, de repente llegó una en especial, antes de terminar esta maravillosa etapa, gracias a quién me doy cuenta de qué es ser feliz en realidad, y me ha motivado a ser una persona dispuesta a cumplir todas sus metas y sueños en la vida, por transmitirme esa inspiración le dedico unas palabras en este espacio, así que gracias, Emily, por llegar a mi vida y por ser parte de esto también.

## AGRADECIMIENTO

---

Agradezco a la vida, porque a pesar de todas las circunstancias, me ha colocado en los lugares indicados, y me ha visto tropezar, para después levantarme más fuerte.

Agradezco a todos los miembros de mi familia, a los que siempre me acompañaron en el camino, y me vieron crecer hasta llegar a este punto, a mi madre que siempre ha estado conmigo y que es lo más importante que tengo en mi vida por saberme brindar todas las oportunidades, agradeceré eso siempre y por ella soy la persona que todos ven.

Además, agradezco a mi padre y a mi abuelo, quienes siempre supieron platicarme sobre etapas como esta de la vida, espero haber puesto siempre en práctica sus enseñanzas y a pesar de sus ausencias, he sabido circular el camino que ustedes dejaron marcado.

También agradezco a todo el maravilloso personal docente de la PUCE, de cada uno de ustedes me llevo sus mejores enseñanzas en cada una de sus cátedras, por eso reconozco al tutor de este trabajo, además decano de nuestra magnífica facultad, Mgtr. Charles Escobar, quien me extendió su apoyo y su conocimiento de la mejor manera para este trabajo.

Y agradezco a mis compañeros y colegas de la carrera, son personas increíbles, y me llevo su amistad para siempre.

## RESUMEN

---

Este trabajo tiene como objetivo investigar el impacto de inteligencias artificiales, en específico ChatGPT, en la comunidad universitaria de la Pontificia Universidad Católica del Ecuador (PUCE). El proyecto se centra en el análisis de las percepciones y experiencias de los estudiantes, docentes y personal administrativo en relación con el uso de ChatGPT en sus actividades cotidianas. Mediante la aplicación de técnicas de ciencia de datos como análisis de sentimientos, se busca identificar patrones y tendencias en las respuestas de los encuestados, así como posibles áreas de mejora en la implementación y uso de estas tecnologías. La metodología de minería de datos CRISP-DM guía el desarrollo del proyecto, comenzando con la comprensión del problema y la recopilación de datos a través de encuestas, seguido de la preparación y análisis de los datos, y finalmente la evaluación y comunicación de los resultados. Este trabajo espera proporcionar una mayor comprensión del papel de las inteligencias artificiales en el ámbito educativo y contribuir al desarrollo de estrategias efectivas para mejorar la experiencia de la comunidad universitaria con estas herramientas innovadoras.

**Palabras clave:** Inteligencia Artificial (IA), ChatGPT, Análisis de sentimientos, educación superior, investigación, Procesamiento del Lenguaje Natural (PLN), minería de datos, CRISP-DM, Python.

# ÍNDICE

---

## Contenido

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS .....	IV
ÍNDICE DE FIGURAS .....	IV
ÍNDICE DE TABLAS.....	VII
CAPÍTULO I: INTRODUCCIÓN.....	1
1.    MARCO DE REFERENCIA.....	1
1.1.    JUSTIFICACIÓN.....	1
1.2.    Planteamiento del problema.....	2
1.3.    Objetivo General.....	2
1.4.    Objetivos Específicos .....	2
1.5.    Antecedentes.....	3
1.6.    Alcance .....	4
CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA .....	5
2.    Marco Teórico.....	5
2.1.    Revisión sistemática de la literatura.....	5
2.2.    Inteligencia Artificial (IA) .....	6
2.3.    Procesamiento del Lenguaje Natural (NLP).....	7
2.4.    Modelos de lenguaje y ChatGPT.....	9

2.5. Minería de datos y Análisis de sentimientos .....	10
CAPÍTULO III: METODOLOGÍA .....	12
3. Metodología de desarrollo del plan de tesis .....	12
3.1. Investigación Cualitativa .....	12
3.2. Investigación Aplicativa .....	12
3.3. Metodología de minería de datos .....	13
3.3.1. Generalidades .....	13
3.3.2. CRISP-DM.....	14
CAPÍTULO IV: IMPACTO DE CHATGPT EN LA EXPERIENCIA DE USUARIOS	
UNIVERSITARIOS MEDIANTE TÉCNICAS DE MINERÍA DE DATOS Y ANÁLISIS DE	
SENTIMIENTOS.....	17
4.1. Generación de la data de experiencia de usuario del uso de ChatGPT en la PUCE..	17
4.2. Preparación de los datos para ser utilizados en un modelo de aprendizaje automático mediante técnicas de preprocesamiento.....	26
4.3. Modelos para aprendizaje automático basado en análisis de sentimientos que se ajuste a los datos preparados .....	29
4.4. Evaluación de los modelos de aprendizaje automático generados que mejor respondan a la data obtenida.....	46
4.5. Impacto de ChatGPT en la experiencia de usuarios universitarios de la PUCE con base en los resultados obtenidos.....	56
CONCLUSIONES Y RECOMENDACIONES .....	58
BIBLIOGRFÍA.....	60

GLOSARIO DE TÉRMINOS .....	62
ANEXOS.....	63
Anexo A: Repositorio de datos.....	63

## ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS

---

### ÍNDICE DE FIGURAS

Ilustración 1 – Ciclo de CRISP-DM (Fuente: ReseachGate. Creative Commons).....	16
Ilustración 2 - Difusión de encuesta a través de mail PUCE .....	18
Ilustración 3 - Difusión de encuesta a través de plataforma EVA.....	19
Ilustración 4 - Presentación de la encuesta .....	19
Ilustración 5 - Dataset original .....	20
Ilustración 6 - Estadísticas Pregunta 1 .....	20
Ilustración 7 - Estadísticas Pregunta 2 .....	21
Ilustración 8 - Estadísticas Pregunta 3.....	21
Ilustración 9 - Estadísticas Pregunta 4 .....	22
Ilustración 10 - Estadísticas Pregunta 5 .....	22
Ilustración 11 - Estadísticas Pregunta 6.....	23
Ilustración 12 - Estadísticas Pregunta 7 (Nube de palabras generada automáticamente por Forms).....	23
Ilustración 13 - Estadísticas Pregunta 8 (Nube de palabras generada automáticamente por Forms).....	24
Ilustración 14 - Estadísticas Pregunta 9 (Nube de palabras generada automáticamente por Forms).....	24
Ilustración 15 - Estadísticas Pregunta 10 (Nube de palabras generada automáticamente por Forms).....	25
Ilustración 16 - Estadísticas Pregunta 11 (Nube de palabras generada automáticamente por Forms).....	25
Ilustración 17 - Traducción en línea mediante el uso de Sheets .....	27

Ilustración 18 - Conversión de dataset a formato .csv con Calc.....	27
Ilustración 19 - Carga del dataset con pandas.....	28
Ilustración 20 - Preprocesamiento del dataframe.....	28
Ilustración 21 - Dataset limpio y preprocesado .....	29
Ilustración 22 - Clasificación de polaridad de las respuestas mediante el uso de VADER....	30
Ilustración 23 - Asignación de sentimientos según los valores de polaridad obtenidos .....	31
Ilustración 24 - Cuenta de respuestas con base en el sentimiento .....	31
Ilustración 25 - Creación de dataframe con preguntas y su respectivo sentimiento .....	32
Ilustración 26 - Creación de nubes de palabras mediante uso de WordCloud .....	32
Ilustración 27 - Nube de p1(+)	33
Ilustración 28 - Nube de p1(-)	33
Ilustración 29 - Nube de p1(+/-)	34
Ilustración 30 - Nube de p2(+)	34
Ilustración 31 - Nube de p2(-)	35
Ilustración 32 - Nube de p2(+/-)	35
Ilustración 33 - Nube de p3(+)	36
Ilustración 34 - Nube de p3(-)	36
Ilustración 35 - Nube de p3(+/-)	37
Ilustración 36 - Nube de p4(+)	37
Ilustración 37 - Nube de p4(-)	38
Ilustración 38 - Nube de p4(+/-)	38
Ilustración 39 - Nube de p5(+)	39
Ilustración 40 - Nube de p5(-)	39
Ilustración 41 - Nube de p5(+/-)	40
Ilustración 42 - Ejemplo de estadísticas para p1 y rol.....	40
Ilustración 43 - Ejemplo de estadísticas para p1 y facultad.....	41

Ilustración 44 - Ejemplo de estadísticas para p1 y edad.....	41
Ilustración 45 - Creación de dataframe para aplicación de algoritmos de aprendizaje automático.....	42
Ilustración 46 - Algoritmo de Regresión Logística con librería sklearn.....	43
Ilustración 47 - Algoritmo de SVM con librería sklearn .....	44
Ilustración 48 - Algoritmo de Random Forest con librería sklearn .....	45
Ilustración 49 - Algoritmo de Naive Bayes con librería sklearn.....	46
Ilustración 50 - Matriz de confusión binaria. Fuente: Arce, J. I. B. (2022). La matriz de confusión y sus métricas. Juan Barrios. <a href="https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/">https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/</a> .....	47
Ilustración 51 - Algoritmo para métricas de Regresión Logística .....	49
Ilustración 52 - Métricas por pregunta de Regresión Logística.....	50
Ilustración 53 - Algoritmo para métricas de SVM.....	51
Ilustración 54 - Métricas por pregunta de SVM.....	52
Ilustración 55 - Algoritmo para métricas de Random Forest.....	53
Ilustración 56 - Métricas por pregunta de Random Forest.....	54
Ilustración 57 - Algoritmo para métricas de Naive Bayes .....	55
Ilustración 58 - Métricas por pregunta de Naive Bayes .....	56

## ÍNDICE DE TABLAS

Tabla 1 – Resultados de precisión de cada modelo por pregunta ..... 47

Tabla 2 - Resultados de matrices de confusión de cada modelo por pregunta ..... 48

## CAPÍTULO I: INTRODUCCIÓN

---

### 1. MARCO DE REFERENCIA

#### 1.1. JUSTIFICACIÓN

La adopción de tecnologías de inteligencia artificial, como ChatGPT, ha experimentado un crecimiento significativo en diversos ámbitos, incluido el educativo. Estas herramientas ofrecen un potencial considerable para mejorar la enseñanza y el aprendizaje, mediante la personalización de la educación, la facilitación de la comunicación y la colaboración, y el apoyo a la resolución de problemas. Sin embargo, el impacto real de estas tecnologías en el entorno académico aún no se ha explorado lo suficiente, especialmente en el contexto específico de la universidad.

Este proyecto busca abordar esta brecha en la investigación al evaluar el impacto del uso de ChatGPT en la comunidad académica de la universidad, analizando aspectos como la adopción, las percepciones, la efectividad y las aplicaciones en el entorno educativo. Al realizar este estudio, se espera obtener una comprensión más profunda de cómo la inteligencia artificial conversacional puede ser utilizada de manera efectiva en el ámbito universitario, lo que permitirá a la institución tomar decisiones informadas sobre su implementación y uso a futuro.

Además, este trabajo contribuirá al creciente cuerpo de conocimientos sobre la aplicación de inteligencias artificiales en la educación superior, proporcionando información valiosa y evidencia empírica que puede ser útil para otras instituciones y profesionales en el campo. Al identificar y compartir las mejores prácticas, este proyecto tiene el potencial de mejorar no solo la experiencia educativa en la universidad, sino también en otras instituciones que buscan integrar ChatGPT y tecnologías similares en sus programas académicos.

## **1.2. Planteamiento del problema**

El avance de la inteligencia artificial y, en particular, de los modelos de lenguaje como ChatGPT, ha generado oportunidades para mejorar la educación y el aprendizaje en diversos entornos académicos. A pesar de su potencial, la adopción y el impacto de estas tecnologías en la enseñanza superior aún no se han estudiado lo suficiente, lo que plantea incertidumbre sobre su efectividad y las mejores prácticas para su implementación.

La falta de información sobre cómo ChatGPT afecta a la comunidad académica en términos de percepciones, adopción y aplicaciones, así como su impacto en el rendimiento y la satisfacción de los estudiantes y profesores, dificulta la toma de decisiones informadas en cuanto a su uso y desarrollo en el ámbito universitario. Además, existe una brecha en la comprensión de cómo las características específicas de una institución pueden influir en la efectividad y la adopción de estas herramientas.

El problema radica en evaluar y entender el impacto real de ChatGPT en la universidad, identificar los factores que contribuyen a su éxito o fracaso en un entorno educativo y desarrollar recomendaciones basadas en evidencia para la implementación y el uso efectivo de la inteligencia artificial conversacional. Al abordar este problema, se busca mejorar la calidad de la enseñanza y el aprendizaje, así como apoyar a la universidad en su adaptación a las nuevas tecnologías y las demandas del siglo XXI.

## **1.3. Objetivo General**

Evaluar el impacto de ChatGPT en la experiencia de usuarios universitarios mediante técnicas de minería de datos y análisis de sentimientos.

## **1.4. Objetivos Específicos**

- Generar la data de la experiencia de usuario del uso de ChatGPT en la PUCE.

- Preparar los datos para ser utilizados en un modelo de aprendizaje automático mediante técnicas de preprocesamiento.
- Determinar un modelo para aprendizaje automático basado en análisis de sentimientos que se ajuste a los datos preparados.
- Evaluar los modelos de aprendizaje automático generados que mejor respondan a la data obtenida.
- Determinar el impacto de ChatGPT en la experiencia de usuarios universitarios de la PUCE con base en los resultados obtenidos.

### **1.5. Antecedentes**

“La Inteligencia Artificial (IA) es una rama de la informática centrada en desarrollar sistemas y algoritmos que pueden llevar a cabo tareas que normalmente requerirían de la inteligencia humana, tales como el aprendizaje, el razonamiento, la percepción y el reconocimiento de patrones” (Russell & Norvig, 2016).

“El aprendizaje profundo, una subdivisión de la IA que se basa en redes neuronales artificiales, ha impulsado muchos de los avances más recientes en el campo” (Goodfellow, Bengio, & Courville, 2016). “Los modelos de lenguaje, como ChatGPT, son ejemplos de cómo el aprendizaje profundo se puede aplicar al procesamiento del lenguaje natural (NLP, por sus siglas en inglés)” (Vaswani et al., 2017).

“ChatGPT es un modelo de lenguaje desarrollado por OpenAI que utiliza la arquitectura GPT (Generative Pre-trained Transformer). GPT-3, lanzado en 2020, es uno de los modelos de lenguaje más avanzados hasta la fecha, con 175 mil millones de parámetros” (Brown et al., 2020).

“Estos modelos de lenguaje pueden generar texto de alta calidad y coherente en respuesta a una amplia variedad de entradas, abriendo un amplio espectro de aplicaciones como la

generación de texto, la traducción automática, la creación de resúmenes, el reconocimiento de voz, la respuesta a preguntas y la generación de código” (Vaswani et al., 2017).

“A pesar de sus impresionantes capacidades, ChatGPT y otros modelos de lenguaje aún tienen limitaciones, incluyendo la generación de respuestas incorrectas o incoherentes, la falta de comprensión profunda del contexto, y la sensibilidad a la información sesgada en los datos de entrenamiento” (Brown et al., 2020).

### **1.6. Alcance**

El alcance de este proyecto de investigación se centra en el uso de inteligencia artificial, en particular el modelo de lenguaje natural ChatGPT, en la experiencia del usuario. El objetivo principal del proyecto es evaluar el impacto de ChatGPT en la percepción y satisfacción del usuario en el contexto universitario. El proyecto se enfocará en los siguientes aspectos:

- Evaluación de la experiencia del usuario al interactuar con ChatGPT en un contexto universitario.
- Identificación de patrones y tendencias en las respuestas generadas por ChatGPT a través de técnicas de minería de datos y análisis de sentimientos.
- Implementación de la metodología CRISP-DM para gestionar el proyecto y los datos obtenidos.

El proyecto se llevará a cabo en un período de 16 semanas y se utilizará una muestra de estudiantes universitarios como sujetos de estudio. Los datos serán recopilados a través de encuestas y análisis de interacciones de los usuarios con ChatGPT en un entorno universitario. La investigación se centrará en el contexto universitario y no se extenderá a otros ámbitos o aplicaciones de ChatGPT. El objetivo final del proyecto es proporcionar recomendaciones para mejorar la experiencia del usuario y la utilidad de ChatGPT en un contexto universitario.

## CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

---

### 2. Marco Teórico

#### 2.1. Revisión sistemática de la literatura

“Una revisión sistemática de la literatura es un tipo de estudio de investigación que emplea métodos rigurosos y reproducibles para recopilar y analizar toda la literatura disponible sobre un tema específico. Este tipo de revisiones son altamente valoradas en la investigación académica debido a su enfoque exhaustivo y su capacidad para minimizar el sesgo” (Higgins & Green, 2011).

La metodología de una revisión sistemática generalmente abarca las siguientes etapas:

- Definición de la pregunta de investigación: “Esta etapa involucra establecer claramente el objetivo del estudio y definir los criterios de inclusión y exclusión que determinarán qué estudios se considerarán para la revisión” (Kitchenham, 2004).
- Búsqueda de literatura: “La búsqueda de literatura es un proceso exhaustivo que implica la búsqueda en múltiples bases de datos académicas para encontrar todos los estudios relevantes que se ajusten a los criterios definidos en la primera etapa” (Higgins & Green, 2011).
- Selección de estudios: “Los estudios encontrados en la búsqueda de literatura se filtran y seleccionan con base en los criterios de inclusión y exclusión definidos anteriormente” (Higgins & Green, 2011).
- Extracción de datos: “En esta etapa, se extraen los datos relevantes de los estudios seleccionados, incluyendo detalles sobre los métodos de estudio, la población de estudio, las medidas de resultado, los hallazgos y las conclusiones” (Higgins & Green, 2011).
- Análisis de los datos y síntesis de los hallazgos: “Los datos extraídos se analizan y se sintetizan para responder a la pregunta de investigación. Esto puede implicar un

análisis cualitativo, cuantitativo o ambos, dependiendo de la naturaleza de los estudios incluidos” (Gough, Oliver & Thomas, 2017).

- Informe y difusión de los resultados: “Finalmente, los hallazgos se presentan de una manera clara y transparente, discutiendo las implicaciones de los resultados, las limitaciones del estudio y las recomendaciones para futuras investigaciones” (Higgins & Green, 2011).

“El objetivo de una revisión sistemática es proporcionar una visión completa y objetiva de la literatura actual sobre un tema específico, y es una herramienta valiosa para identificar las lagunas en la literatura existente, informar la política y la práctica, y guiar futuras investigaciones” (Petticrew & Roberts, 2006).

## **2.2. Inteligencia Artificial (IA)**

“El término fue acuñado por primera vez por John McCarthy en 1956 durante la Conferencia de Dartmouth, donde se definió como la ciencia e ingeniería de hacer máquinas inteligentes “(McCarthy et al., 1955).

“La historia de la IA ha estado marcada por varios hitos significativos. Entre estos se incluyen la creación de Eliza, un programa informático desarrollado en el MIT que podía simular una conversación con un psicoterapeuta” (Weizenbaum, 1966), y “Deep Blue de IBM, una computadora que logró vencer al campeón mundial de ajedrez Garry Kasparov en 1997” (Campbell et al., 2002).

En general, la IA se clasifica en dos categorías: IA débil (o estrecha) y IA fuerte (o general).

- “La IA débil se refiere a sistemas diseñados para realizar una tarea específica, como las recomendaciones de productos en tiendas online” (Russell & Norvig, 2016).

- “La IA fuerte se refiere a sistemas capaces de comprender, razonar, aprender y aplicar este conocimiento a diferentes dominios, de manera similar a como lo haría un ser humano” (Bostrom, 2014).

La IA se basa en varios conceptos y técnicas, como los algoritmos de aprendizaje automático, que incluyen el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo. También se basa en técnicas de procesamiento del lenguaje natural para comprender y generar lenguaje humano, y en la visión por computadora para procesar y analizar imágenes y vídeos.

En las últimas décadas, el campo de la IA ha experimentado un rápido avance gracias a la disponibilidad de grandes volúmenes de datos (big data), el aumento de la capacidad de computación y el desarrollo de algoritmos más avanzados. Estos avances han llevado a logros significativos, como los sistemas de recomendación personalizados, los asistentes virtuales como Siri y Alexa, y los vehículos autónomos.

“No obstante, la IA todavía tiene que superar desafíos significativos. Estos incluyen la creación de sistemas que puedan razonar de manera más general y efectiva, el desarrollo de métodos para garantizar un uso ético y justo de la IA, y la gestión de los impactos socioeconómicos de la IA, como los efectos en el empleo y la privacidad” (Russell, Dewey, & Tegmark, 2015)

### **2.3. Procesamiento del Lenguaje Natural (NLP)**

“El Procesamiento del Lenguaje Natural es un campo de la inteligencia artificial que se centra en la interacción entre las computadoras y los humanos a través del lenguaje natural. El objetivo final de NLP es leer, interpretar, entender y hacer sentido del lenguaje humano de una manera valiosa” (OpenAI, 2021).

El Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés) es una rama de la Inteligencia Artificial (IA) que se centra en la interacción entre las computadoras y el lenguaje

humano. En esencia, NLP trata de programar computadoras para procesar y analizar grandes cantidades de datos de lenguaje natural. NLP combina varias disciplinas, como la ciencia de la computación y la lingüística, para llenar la brecha entre la interacción humana y la comprensión de las máquinas.

“El NLP emplea una variedad de técnicas y métodos, incluyendo técnicas de aprendizaje automático, técnicas estadísticas, y algoritmos de procesamiento de lenguaje” (Jurafsky & Martin, 2019).

Sus aplicaciones son vastas e incluyen traducción automática, análisis de sentimientos, asistentes de voz, chatbots, y mucho más.

El procesamiento del lenguaje natural (NLP, por sus siglas en inglés) enfrenta desafíos fundamentales que incluyen:

- Comprensión del lenguaje: “Las máquinas deben ser capaces de entender el significado del lenguaje humano, incluyendo la sintaxis (estructura de las frases), la semántica (significado de las palabras y frases), el contexto y la pragmática (interpretación del lenguaje en función del contexto)” (Jurafsky & Martin, 2019).
- Generación de lenguaje: “Las máquinas deben ser capaces de generar respuestas coherentes y contextuales. Esto incluye la selección de palabras, la construcción de frases y el aseguramiento de que las respuestas son relevantes para la entrada proporcionada” (Reiter & Dale, 2000).
- Análisis de sentimientos: “Las máquinas deben ser capaces de identificar y extraer opiniones y emociones de los textos, útil para entender las percepciones del público sobre un producto o servicio, por ejemplo” (Pang & Lee, 2008).

“El Procesamiento del Lenguaje Natural ha avanzado significativamente en los últimos años gracias al desarrollo de modelos de aprendizaje profundo como las redes neuronales recurrentes

(RNN), las redes neuronales convolucionales (CNN), la atención y los transformadores, y modelos pre-entrenados como BERT, GPT-2/3/4 y T5, entre otros” (Devlin et al., 2018; Vaswani et al., 2017; Brown et al., 2020).

#### **2.4. Modelos de lenguaje y ChatGPT**

ChatGPT es un modelo de lenguaje generativo desarrollado por OpenAI.

“Se basa en la arquitectura de GPT, o Transformer, que utiliza el aprendizaje automático para generar texto que es relevante para el contexto proporcionado. ChatGPT ha sido entrenado en una variedad de fuentes de datos de Internet y se utiliza en una amplia gama de aplicaciones, desde la generación de ideas hasta la escritura asistida y la tutoría en una variedad de temas. Su objetivo es facilitar la interacción humana con la tecnología a través del lenguaje natural.” (OpenAI, 2022).

Los modelos de procesamiento de lenguaje natural (NLP, por sus siglas en inglés) son un conjunto de algoritmos y técnicas que permiten a las máquinas entender, interpretar y generar lenguaje humano. Son una rama de la inteligencia artificial que se centra en la interacción entre las computadoras y el lenguaje humano.

“Los modelos de lenguaje se fundamentan en la premisa de que el lenguaje puede ser descompuesto y analizado de manera sistemática y estructurada. Esto abarca tanto la gramática y la sintaxis, que son las reglas que determinan la estructura del lenguaje, como la semántica, que es el significado de las palabras y frases” (Chomsky, 1957; Lyons, 1977).

El empleo de estos modelos ha permitido el desarrollo de una serie de aplicaciones y servicios que se utilizan a diario. Algunos ejemplos son:

- Traducción automática: “Servicios como Google Translate emplean modelos de lenguaje para traducir texto de un idioma a otro en tiempo real” (Wu et al., 2016).

- Asistentes virtuales: “Siri de Apple, Alexa de Amazon y Google Assistant utilizan modelos de lenguaje para entender las instrucciones verbales y responder de una manera que se asemeja al lenguaje humano natural” (Luger & Sellen, 2016).
- Análisis de sentimientos: “Las empresas utilizan modelos de lenguaje para analizar las opiniones de los clientes y determinar si el sentimiento general es positivo, negativo o neutral” (Pang, Lee, & Vaithyanathan, 2002).
- Chatbots: “Los chatbots utilizan modelos de lenguaje para interactuar con los usuarios en un formato conversacional, pueden responder a preguntas, proporcionar recomendaciones e incluso realizar tareas en nombre del usuario” (Joshi, Kumar, & Agrawal, 2020).
- Generación de texto: “Modelos de lenguaje como GPT-3 de OpenAI pueden generar textos coherentes y naturales a partir de una serie de palabras o frases de entrada” (Brown et al., 2020).

Estos son solo algunos ejemplos de cómo se utilizan los modelos de lenguaje en la vida cotidiana. A medida que la tecnología avanza, es probable que se visibilice cada vez más aplicaciones de estos modelos en el futuro.

## **2.5. Minería de datos y Análisis de sentimientos**

“La minería de datos es el proceso de descubrir patrones interesantes y conocimientos a partir de grandes cantidades de datos. Los datos pueden ser almacenados en bases de datos, data warehouses, o pueden estar presentes en otras formas” (Han, Jiawei, Micheline Kamber, and Jian Pei. "Data mining: concepts and techniques." Elsevier, 2011).

La minería de datos es un proceso que implica descubrir patrones y conocimientos a partir de grandes cantidades de datos. Se basa en una variedad de técnicas que incluyen estadísticas, inteligencia artificial y aprendizaje automático para analizar y entender los datos. A través de este proceso, se pueden identificar correlaciones, tendencias y comportamientos que de otra manera

serían difíciles de detectar. La minería de datos se utiliza en una amplia gama de aplicaciones, desde el marketing y las ventas hasta la medicina y la investigación científica.

"El análisis de sentimientos o la minería de opiniones es el campo computacional que se ocupa de la extracción de opiniones, sentimientos, evaluaciones, apreciaciones, actitudes, y emociones de texto" (Liu, Bing. "Sentiment analysis and opinion mining." Synthesis lectures on human language technologies, 2012).

El análisis de sentimientos es una rama del procesamiento del lenguaje natural (NLP) y la minería de datos que se enfoca en identificar y extraer opiniones y emociones de los datos de texto. Este proceso puede ser útil para entender cómo se sienten las personas acerca de un determinado tema, producto o servicio. Por ejemplo, una empresa puede utilizar el análisis de sentimientos para rastrear cómo se sienten sus clientes acerca de su producto en las redes sociales, o un investigador puede utilizarlo para entender las opiniones públicas sobre un tema político.

Es importante notar que tanto la minería de datos como el análisis de sentimientos requieren técnicas sofisticadas y a menudo dependen de la calidad de los datos recopilados. Sin embargo, cuando se aplican correctamente, pueden proporcionar una valiosa visión y entendimiento de los datos que de otro modo serían difíciles de obtener.

## CAPÍTULO III: METODOLOGÍA

---

### **3. Metodología de desarrollo del plan de tesis**

#### **3.1. Investigación Cualitativa**

Este estudio incorpora elementos cualitativos a través de la implementación de la encuesta y el análisis subsecuente de las respuestas a las preguntas abiertas. La encuesta tiene como objetivo captar las experiencias y percepciones de los miembros de la comunidad universitaria con respecto al uso de la Inteligencia Artificial (IA), específicamente ChatGPT.

El análisis cualitativo se enfoca en aspectos tales como los patrones de utilización de ChatGPT, las motivaciones para su uso, las expectativas de los usuarios y su nivel de satisfacción. Además, se busca comprender cómo los usuarios perciben el impacto de la IA en su vida universitaria, sus creencias y actitudes hacia la tecnología, y las sugerencias que puedan tener para mejorar o expandir su uso.

El análisis de sentimientos que se realizará en las respuestas abiertas de la encuesta también es un componente cualitativo crucial de esta investigación. A través de este análisis, se busca identificar y comprender mejor las emociones y opiniones de los usuarios con respecto a ChatGPT.

A pesar de que esta investigación incluye componentes cuantitativos, como la recopilación de datos a través de la encuesta y la aplicación de técnicas de minería de datos para el análisis de los resultados, también se integran elementos cualitativos significativos. Estos elementos permiten una comprensión más profunda y completa del fenómeno en estudio.

#### **3.2. Investigación Aplicativa**

En este trabajo de investigación, se emplea la metodología de la investigación aplicada para centrarse en la resolución de un problema práctico en el entorno universitario: comprender el

impacto y la utilidad de la inteligencia artificial, concretamente de ChatGPT, en la comunidad académica. A diferencia de la investigación básica, que se preocupa por ampliar y profundizar el conocimiento teórico, la investigación aplicada tiene como objetivo principal aplicar este conocimiento a problemas específicos de la vida real.

El estudio aquí presentado se diseña con el propósito de recoger y analizar datos sobre el uso de ChatGPT en la Pontificia Universidad Católica del Ecuador (PUCE), con el fin de aportar hallazgos prácticos que puedan ser utilizados para mejorar la interacción de los estudiantes, profesores y personal administrativo con esta herramienta de inteligencia artificial. Por lo tanto, la relevancia de la investigación no sólo radica en su contribución al campo académico de la inteligencia artificial en la educación, sino también en su aplicación directa para mejorar la experiencia educativa en la PUCE.

Dentro de la metodología de investigación aplicada, este estudio emplea técnicas de análisis de sentimientos. Ambas técnicas son utilizadas en un contexto aplicado, con el objetivo de revelar información práctica y valiosa sobre los usuarios de ChatGPT en la universidad. El análisis de sentimientos es empleado para cuantificar y clasificar las opiniones y emociones de los usuarios respecto a ChatGPT, proporcionando así un mapa detallado de la actitud de la comunidad universitaria hacia esta herramienta. En definitiva, la orientación aplicada de esta investigación garantiza que los hallazgos obtenidos no sólo poseen un interés académico, sino que también tienen un valor práctico para mejorar el uso y la integración de la inteligencia artificial en la universidad.

### **3.3. Metodología de minería de datos**

#### **3.3.1. Generalidades**

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) se utilizará para guiar el proceso de análisis de sentimientos en respuestas a preguntas específicas, empleando herramientas como Python, pandas, Vader y algoritmos de clasificación.

En la etapa de entendimiento del negocio, se comprenderá el objetivo del proyecto y su contexto. A continuación, se procederá a la comprensión de los datos, explorando su estructura y características. El tratamiento de datos se realizará mediante herramientas como Excel y Sheets para garantizar la integridad y calidad de estos.

Posteriormente, se llevará a cabo la preparación de los datos, realizando limpieza, corrección ortográfica, transformación y selección de características necesarias para el análisis de sentimientos. Se utilizará Jupyter Notebook corriendo en entorno local de Windows 11, con Python 3.11.3 y pandas 2.0.2 que serán utilizados para cargar y manipular los datos en un formato adecuado.

En la etapa de modelado, se aplicarán diferentes algoritmos de clasificación, tales como Regresión Logística, SVM, Naive Bayes y Random Forest, utilizando bibliotecas de aprendizaje automático en Python, como scikit-learn, o ajustando el modelo mediante técnicas de oversampling utilizando la librería imblearn. Estos algoritmos serán entrenados con los datos preprocesados para generar modelos de clasificación.

Luego, se evaluará el rendimiento de los modelos utilizando métricas de evaluación, tales como precisión, recall y F1-score. Se comparará los resultados obtenidos por los diferentes algoritmos de clasificación y se seleccionará el modelo con mejor desempeño.

Finalmente, en la etapa de despliegue se comunicarán los resultados obtenidos en la anterior etapa de evaluación, y se pondrá a disposición de los interesados los resultados obtenidos, junto a sus datos y los mejores modelos aplicados.

### **3.3.2. CRISP-DM**

CRISP-DM (CRoss Industry Standard Process for Data Mining) es un modelo de proceso que proporciona un enfoque estructurado para la planificación de un proyecto de minería de datos.

“Fue desarrollado en la década de 1990 con el apoyo de organizaciones líderes en la industria, y desde entonces ha ganado aceptación generalizada debido a su enfoque integral y factible.” (Shearer, 2000).

El proceso CRISP-DM comprende seis fases que se ejecutan en un ciclo:

- **Comprensión del Negocio:** “Esta fase inicial implica entender los objetivos del proyecto y los requisitos desde una perspectiva empresarial, luego convertir este conocimiento en una definición de problema de minería de datos y un plan preliminar.” (Shearer, 2000).
- **Comprensión de los datos:** En esta fase, se recogen los datos iniciales y se familiariza con ellos, se identifican problemas de calidad de datos, se descubren primeras perspectivas o se detectan subconjuntos interesantes para formar hipótesis sobre información oculta” (Chapman et al., 2000).
- **Preparación de los datos:** “Los datos se limpian, se seleccionan y se transforman en formatos adecuados para la minería de datos en esta etapa” (Chapman et al., 2000).
- **Modelado:** “En esta fase, se seleccionan y aplican varios modelos y algoritmos de minería de datos, y se calibran los parámetros para obtener los mejores resultados” (Wirth & Hipp, 2000).
- **Evaluación:** “En esta etapa, se evalúa la calidad de los modelos desarrollados y su eficacia para lograr los objetivos empresariales. También se considera cualquier posible problema que pueda surgir durante la implementación del modelo” (Wirth & Hipp, 2000)
- **Despliegue:** “Finalmente, el modelo se implementa y se supervisa para asegurarse de que está funcionando como se esperaba” (Wirth & Hipp, 2000).

“Aunque el CRISP-DM es una metodología cíclica y sugiere que los proyectos de minería de datos deben ser iterativos en su naturaleza, el orden en el que se aplican las fases puede variar dependiendo del proyecto y de las circunstancias de la organización” (Chapman et al., 2000).

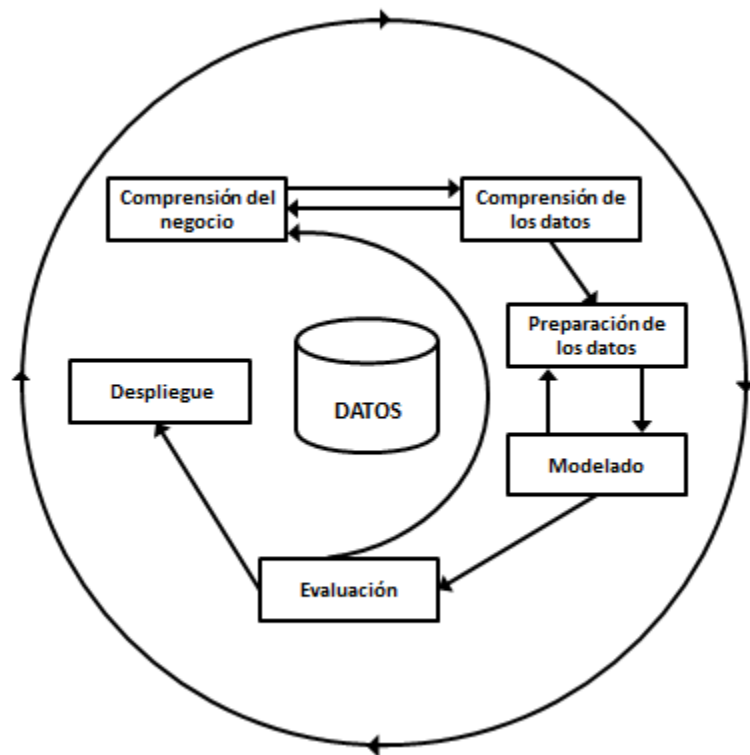


Ilustración 1 – Ciclo de CRISP-DM (Fuente: ReseachGate. Creative Commons)

## **CAPÍTULO IV: IMPACTO DE CHATGPT EN LA EXPERIENCIA DE USUARIOS UNIVERSITARIOS MEDIANTE TÉCNICAS DE MINERÍA DE DATOS Y ANÁLISIS DE SENTIMIENTOS.**

---

### **4.1. Generación de la data de experiencia de usuario del uso de ChatGPT en la PUCE**

#### *Comprensión del problema*

La principal motivación para este proyecto fue investigar la experiencia de los estudiantes y el personal de la PUCE con ChatGPT, una inteligencia artificial avanzada de OpenAI. El objetivo era comprender cómo la comunidad universitaria interactúa con esta IA, sus percepciones sobre su uso y su impacto en la productividad y el aprendizaje. Dado que ChatGPT es una herramienta que puede tener un amplio espectro de aplicaciones en un ambiente académico, se entiende que este análisis puede proporcionar ideas valiosas para mejorar el uso y adopción de la IA en la universidad.

Se formuló un plan de proyecto que implicaba la creación y distribución de una encuesta en línea para recoger datos de la experiencia de los usuarios con ChatGPT. La encuesta fue diseñada para recoger tanto datos cuantitativos, como la frecuencia de uso, como cualitativos, como las percepciones y opiniones de los usuarios sobre ChatGPT, en especial con 3 preguntas de tipo abierta que serán esenciales para el análisis.

#### *Comprensión de los datos*

Una vez que los objetivos del proyecto fueron definidos y el plan formulado, se procedió a recoger los datos. Se utilizó Forms, una herramienta que permitió una recogida de datos sencilla y eficiente. Se diseñó una encuesta en línea, partiendo como base con las recomendaciones proporcionadas por ChatGPT, posterior ajustamiento de los objetivos de la encuesta en revisión

con el tutor del trabajo de titulación; y se distribuyó a todos los miembros de la comunidad universitaria de la PUCE, incluyendo estudiantes y personal.

La encuesta se distribuyó a través de redes sociales Y la plataforma EVA de la Universidad, Sí mismo como el correo institucional de la PUCE. El proceso de recolección de encuestas estuvo abierto desde el 25/04/2023 por la mañana hasta el 31/05/2023, medianoche. Durante este tiempo, se recolectó un total de 818 respuestas, en la cual la mayoría fue de parte de los estudiantes, seguidos de los docentes hasta terminar con el personal administrativo.



*Ilustración 2 - Difusión de encuesta a través de mail PUCE*



Ilustración 3 - Difusión de encuesta a través de plataforma EVA

El objetivo era capturar una variedad de datos cuantitativos y cualitativos sobre la experiencia de uso de ChatGPT. Se recogieron datos demográficos, como la facultad a la que pertenecía el encuestado, la frecuencia de uso de ChatGPT, y las situaciones en las que los usuarios utilizaban la IA. Además, se incluyeron preguntas abiertas en la encuesta para recoger las percepciones, experiencias y opiniones de los usuarios sobre la efectividad y utilidad de ChatGPT.



Ilustración 4 - Presentación de la encuesta

Estas respuestas abiertas proporcionaron una rica fuente de datos para el análisis de sentimientos y otras técnicas de procesamiento de lenguaje natural. Después de la recogida de datos, se realizó un proceso inicial de familiarización con ellos. Se exploró la estructura de los datos, se observaron las distribuciones y se buscaron patrones iniciales y posibles problemas de calidad de datos, como respuestas incompletas o ambiguas. El dataset original descargado para el procesamiento de respuestas a través de Excel pesaba 163 kB.

	F	G	H	I	J	K	L	M	N	
1	¿Cuál es su rol en la	¿En qué facultad o	¿Cuál es su edad?	¿Qué dispositivos	En una escala del 1	¿Está suscrito como	¿Qué opina de la h	¿En qué situación	¿En qué situación	¿Cuál ha
2	Profesor	Facultad de Medicina	35-44 años	no he utilizado;		1 No	Me gustaría conocer m	Me gustaría conocer m	Me gustaría conocer m	Me gusta
3	Profesor	Facultad de Medicina	45-54 años	Teléfono móvil;		7 No	Es una herramienta de	en todas las tareas de	en juegos	positiva s
4	Profesor	Facultad de Ingeniería	45-54 años	Computadora;Teléfon		4 No	Una herramienta más	Explicaciones rápidas y	Medio definitivo de cc	Similar a
5	Profesor	Facultad de Ciencias E	25-34 años	Computadora;		7 No	Es una herramienta re	Elaboración de resúme	Búsqueda de informac	Muy bue
6	Personal Administrati	Facultad de Ciencias E	25-34 años	Computadora;Teléfon		5 No	Si sabes como pregunt	Búsqueda de informac	Recomendaciones per	Una IA si
7	Personal Administrati	Direcciones Administr	35-44 años	Computadora;		1 No	Una herramienta que	Información concreta	(N/A)	N/A
8	Profesor	Facultad de Economía	35-44 años	Teléfono móvil;Compu		1 No				
9	Estudiante	Facultad de Arquitect	25-34 años	Computadora;		5 No	Permite resolver duda	para plantear opción	N/A	buena
10	Profesor	Facultad de Ciencias H	35-44 años	Solo me han reenviada		1 No	Sumamente interesan	No lo he usado todaví	No lo he usado todaví	No lo he
11	Profesor	Facultad de Enfermerí	35-44 años	No he ingresado aún ;		1 No	La tecnología brinda m	No he utilizado aún	No he utilizado aún	No he ut
12	Profesor	Facultad de Ingeniería	55-64 años	Computadora;		1 No	Personalmente piensc	Las pocas veces que m	No lo veo útil en solici	Dentro d
13	Profesor	Facultad de Ciencias E	45-54 años	Computadora;		5 No	Me parece una buena	Ayuda en el resumen	(Búsquedas de informa	Limitada
14	Estudiante	Facultad de Ciencias H	17-24 años	Computadora;		2 No	Como estudiante de C	Cuando no alcanzo a l	é al momento de pedir	Ha sido b
15	Personal Administrati	Facultad de Jurisprude	25-34 años	Computadora;		7 No	Me parece que es una	Para redactar correos,	Para redactar cartas o	Bastante
16	Personal Administrati	Direcciones Administr	25-34 años	Computadora;Teléfon		5 No	Es bastante avanzada	y generación de funcio	Desarrollo de Softwar	Me ha ah
17	Profesor	Facultad de Medicina	45-54 años	Teléfono móvil;		1 No	Podría ser de mucha u	Resúmenes o informa	Enseñanza-aprendizaj	ninguna
18	Profesor	Facultad de Comunica	55-64 años	Teléfono móvil;		1 No	No la usaría nunca. Est	Nunca lo he usado	En ninguna, pues no l	Ninguna
19	Personal Administrati	Direcciones Administr	35-44 años	Computadora;		2 No	es el mejor descubrim	información general d	en la búsqueda de infr	muy bue
20	Profesor	Facultad de Ciencias H	45-54 años	Computadora;Teléfon		1 No	En mi trabajo es inne	No lo uso	NO aplica para mí	Ninguna
21	Profesor	Facultad de Comunica	45-54 años	Computadora;		4 No	Me parece una herra	En búsqueda y organiz	En investigación de te	Todavía e

Ilustración 5 - Dataset original

Este proceso de entender la data recogida fue crítico para preparar los datos para las etapas posteriores de la metodología CRISP-DM, como la preparación de los datos y el modelado.

En primera instancia, desde los datos del formulario se obtuvieron las siguientes estadísticas:

### 1. ¿Cuál es su rol en la universidad?

Más detalles

Información

● Estudiante	580
● Profesor	176
● Personal Administrativo	44
● Otras	18

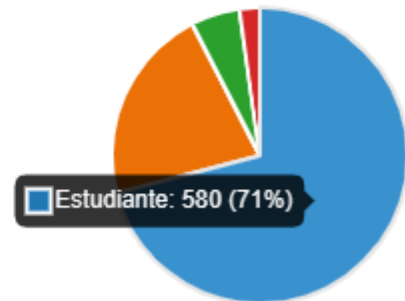


Ilustración 6 - Estadísticas Pregunta 1

## 2. ¿En qué facultad o departamento trabaja/estudia?

[Más detalles](#)

● Direcciones Administrativas	32
● Facultad de Arquitectura, Diseñ...	54
● Facultad de Ciencias Administra...	118
● Facultad de Ciencias de la Educa...	24
● Facultad de Ciencias Exactas y N...	56
● Facultad de Ciencias Filosófico -...	16
● Facultad de Ciencias Humanas	31
● Facultad de Comunicación, Ling...	35
● Facultad de Economía	34
● Facultad de Enfermería	54
● Facultad Internacional de Innov...	4
● Facultad de Ingeniería	122
● Facultad de Jurisprudencia	57
● Facultad de Medicina	106
● Facultad de Psicología	75

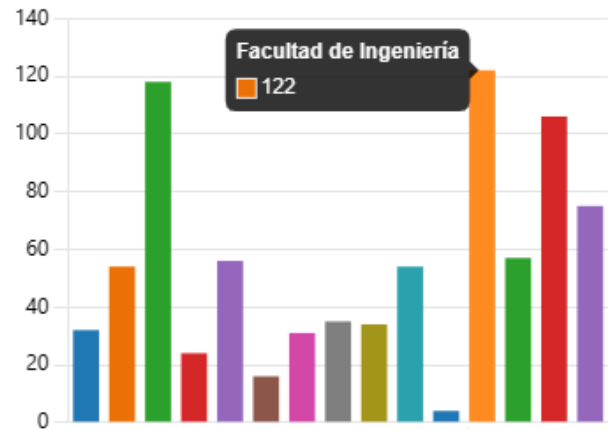


Ilustración 7 - Estadísticas Pregunta 2

## 3. ¿Cuál es su edad?

[Más detalles](#)

[Información](#)

● 17-24 años	435
● 25-34 años	125
● 35-44 años	109
● 45-54 años	84
● 55-64 años	58
● 65 años o más	7

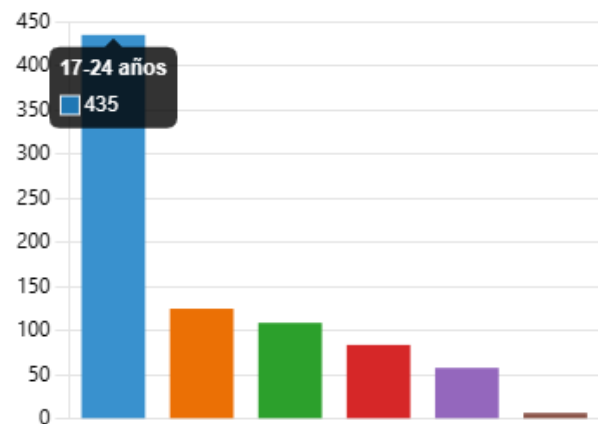


Ilustración 8 - Estadísticas Pregunta 3

#### 4. ¿Qué dispositivos utiliza para acceder a ChatGPT?

[Más detalles](#)

● Computadora	662
● Teléfono móvil	329
● Tableta	52
● Otras	69

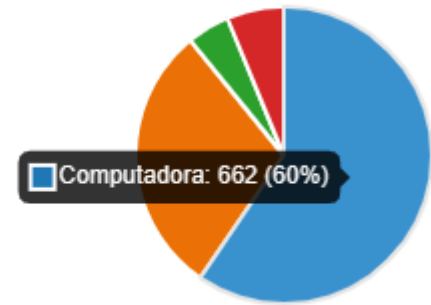


Ilustración 9 - Estadísticas Pregunta 4

#### 5. En una escala del 1 al 10, ¿qué tan frecuentemente utiliza ChatGPT, siendo 1 muy poco frecuente y 10 extremadamente frecuente?

[Más detalles](#)

Información

**3.76**  
Clasificación promedio

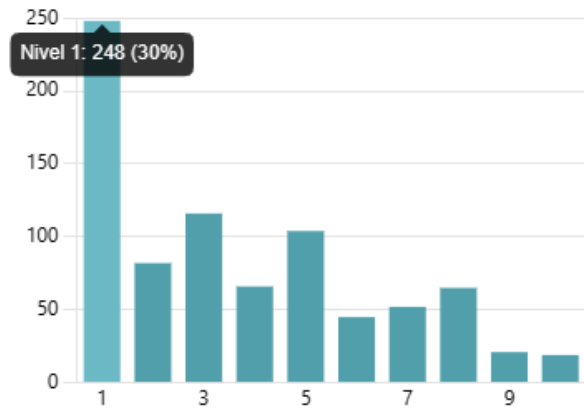


Ilustración 10 - Estadísticas Pregunta 5



8. ¿En qué situaciones o tareas ha encontrado útil el uso de ChatGPT?

[Más detalles](#)

[Información](#)

815  
Respuestas

Respuestas más recientes

"En la elaboración de textos"

"Desconozco "

"En una situación de apuro."

[Actualizar](#)

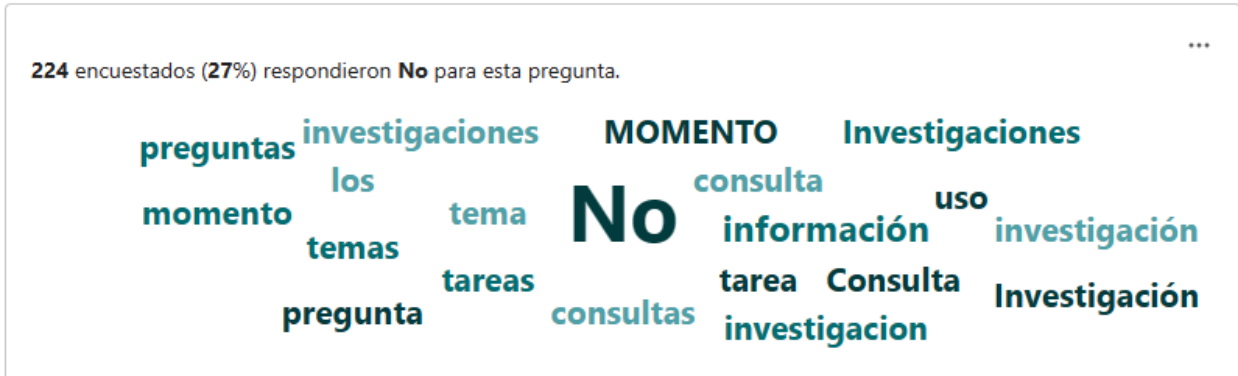


Ilustración 13 - Estadísticas Pregunta 8 (Nube de palabras generada automáticamente por Forms)

9. ¿En qué situaciones o tareas no ha encontrado útil el uso de ChatGPT?

[Más detalles](#)

[Información](#)

815  
Respuestas

Respuestas más recientes

"En las relaciones personales"

"Desconozco "

"En temas que requiero de un mayor análisis pues existe información muy s..."

[Actualizar](#)

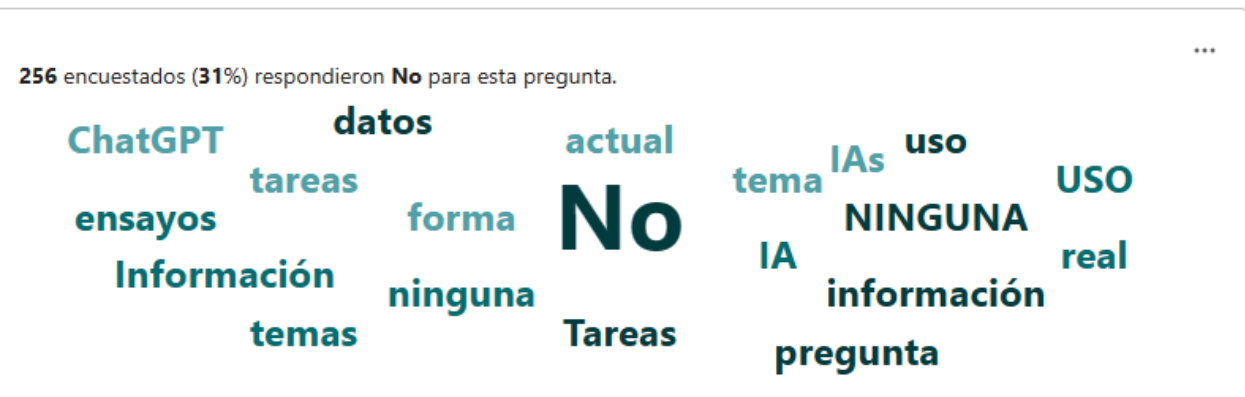


Ilustración 14 - Estadísticas Pregunta 9 (Nube de palabras generada automáticamente por Forms)

10. ¿Cuál ha sido su experiencia general utilizando ChatGPT en la universidad?

[Más detalles](#)

Información

777

Respuestas

Respuestas más recientes

"Excelente"

"No lo he usado"

"Buena"

Actualizar

248 encuestados (32%) respondieron **LA** para esta pregunta.

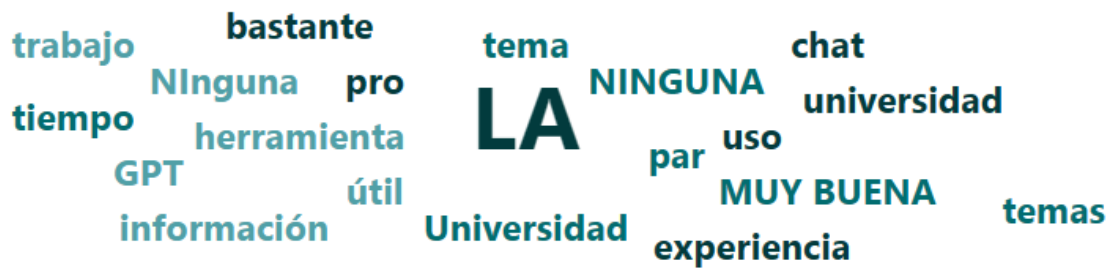


Ilustración 15 - Estadísticas Pregunta 10 (Nube de palabras generada automáticamente por Forms)

11. ¿Cómo cree que ChatGPT ha influido en su experiencia educativa o laboral en la universidad?

[Más detalles](#)

Información

771

Respuestas

Respuestas más recientes

"Favorablemente"

"Desconozco "

"Ha influido de manera positiva, pues a más de las diversas fuentes de cons..."

Actualizar

263 encuestados (34%) respondieron **UN** para esta pregunta.

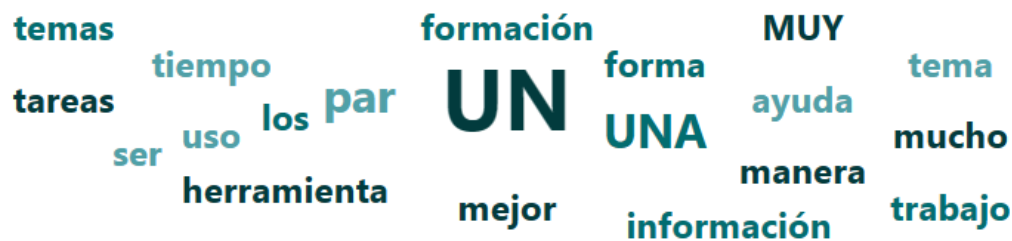


Ilustración 16 - Estadísticas Pregunta 11 (Nube de palabras generada automáticamente por Forms)

Previamente a pasar a la etapa de preparación de datos, se tiene por entendido que las respuestas serán mucho mejor procesadas por un lenguaje de programación como Python, de el objetivo de análisis de sentimientos se llevará a cabo con el uso de las librerías pandas, Vader y Scikit-learn.

#### **4.2. Preparación de los datos para ser utilizados en un modelo de aprendizaje automático mediante técnicas de preprocesamiento**

Para el procesamiento del dataset en general, se siguieron los siguientes pasos:

- Corrección ortográfica de las preguntas en español: Al ser respuestas en las que había un buen número de errores gramaticales y de sintaxis, se utilizó la herramienta de gramática de Excel para realizar correcciones.
- Traducción de las preguntas al inglés: Se utilizó Sheets para traducir las preguntas del español al inglés utilizando la función =GOOGLETRANSLATE(). Esta función permite traducir el contenido de las celdas de una columna completa a otro idioma.
- Selección de columnas relevantes: Se seleccionaron las columnas que se consideraron pertinentes para el análisis posterior. Estas columnas incluyeron 'rol', 'facultad', 'edad', 'dispositivos', 'uso', 'plus', y las preguntas traducidas al inglés.
- Conversión del dataset a formato .csv: Una vez que se contaba con lo necesario, se pasó el archivo de Excel a formato .csv para que pueda ser cargado en pandas y con el separador que delimite correctamente cada campo.
- Carga del dataset procesado: Utilizando la función `pd.read_csv()`, se cargó el dataset procesado en un DataFrame de pandas. Se especificó la ruta del archivo CSV para su carga.
- Limpieza del dataset: Se llevó a cabo un proceso de limpieza para asegurar la calidad de los datos. Esto incluyó eliminación de filas con valores nulos, mediante el uso del

método dropna(), se logró eliminar filas que contenían valores nulos en alguna columna.

Con este procesamiento se pasó a tener de 818 registros, un total de 748 respuestas para el análisis correspondiente.

The screenshot shows a Google Sheet interface with a data table. The columns are labeled M through Z. The data includes text in Spanish and English, with some cells containing the formula =GOOGLETRANSLATE(M2;"ES";"EN"). The text appears to be a survey or questionnaire about ChatGPT usage.

Ilustración 17 - Traducción en línea mediante el uso de Sheets

	A	B	C	D	E	F
1	rol	facultad	edad	dispositivos	uso	plus
2	Profesor	Facultad de Medicina	35-44 años	no he utilizado,	1	No I would like to know more about this tool
3	Profesor	Facultad de Medicina	45-54 años	Teléfono móvil,	7	No It is a consultation and learning tool
4	Profesor	Facultad de Ingeniería	45-54 años	Computadora, Teléfono móvil,	4	No One more tool is available to perform different tasks in which it can b
5	Profesor	Facultad de Ciencias Exactas y Naturales	25-34 años	Computadora,	7	No It is a revolutionary tool that we must learn to apply it in our professi
6	Personal Administrativo	Facultad de Ciencias Exactas y Naturales	25-34 años	Computadora, Teléfono móvil,	5	No If you know how to ask you, it is a tool with high academic potential
7	Personal Administrativo	Direcciones Administrativas	35-44 años	Computadora,	1	No A tool that helps generate content from a basic question.
8	Profesor	Facultad de Economía	35-44 años	Teléfono móvil, Computadora,	1	No N/A
9	Estudiante	Facultad de Arquitectura, Diseño y Artes	25-34 años	Computadora,	5	No Allows you to solve directly
10	Profesor	Facultad de Ciencias Humanas	35-44 años	Solo me han reenviado ejemplos de uso ,	1	No Extremely interesting to understand as a subjective and collective se
11	Profesor	Facultad de Enfermería	35-44 años	No he ingresado aún ,	1	No Technology provides multiple aid. However, you always have to have
12	Profesor	Facultad de Ingeniería	55-64 años	Computadora,	1	No Personally I think that the GPT chat and other artificial intelligence to
13	Profesor	Facultad de Ciencias Exactas y Naturales	45-54 años	Computadora,	5	No It seems to me a good tool, as long as it is used to improve your own
14	Estudiante	Facultad de Ciencias Humanas	17-24 años	Computadora,	2	No As a human science student, it can be a tool that helps in a hurry, he
15	Personal Administrativo	Facultad de Jurisprudencia	25-34 años	Computadora,	7	No It seems to me that it is a fairly useful tool that serves as a kind of p
16	Personal Administrativo	Direcciones Administrativas	25-34 años	Computadora, Teléfono móvil, Tableta,	5	No Its quite advanced for your time
17	Profesor	Facultad de Medicina	45-54 años	Teléfono móvil,	1	No It could be very useful for concrete tasks, but not as a teaching-learn
18	Profesor	Facultad de Comunicación, Lingüística y Literatura	55-64 años	Teléfono móvil,	1	No I would never use it, I am against the use of chatbots.
19	Personal Administrativo	Direcciones Administrativas	35-44 años	Computadora,	2	No It is the best discovery.
20	Profesor	Facultad de Ciencias Humanas	45-54 años	Computadora, Teléfono móvil,	1	No In my work it is unnecessary because I fear risking the originality of r
21	Profesor	Facultad de Comunicación, Lingüística y Literatura	45-54 años	Computadora,	4	No It seems to me a very useful tool. However, at the moment it present
22	Personal Administrativo	Facultad de Jurisprudencia	35-44 años	Computadora,	1	No I would like to meet her, to know how to use this tool
23	Profesor	Facultad Internacional de Innovación PUCE - Icam	45-54 años	Computadora,	4	Si It can be a useful tool if you know how to use and take advantage.
24	Profesor	Facultad de Ingeniería	45-54 años	Computadora,	8	No It is a tool that helps a lot as an assistant to complete tasks and sea
25	Profesor	Facultad de Ciencias Exactas y Naturales	45-54 años	Computadora, Teléfono móvil,	4	No When well used, it can be a useful support for teaching
26	Personal Administrativo	Direcciones Administrativas	25-34 años	Teléfono móvil,	1	No It is a curious tool that depending on the user can be used for plagiar
27	Profesor	Facultad de Arquitectura, Diseño y Artes	45-54 años	Computadora, Teléfono móvil, Tableta,	2	No An open way of accessing important and immediate information on v
28	Profesor	Facultad de Ciencias Exactas y Naturales	45-54 años	Computadora,	1	No It can be an excellent tool to save time when used with objectives dir

Ilustración 18 - Conversión de dataset a formato .csv con Calc

Carga del dataset e importación de librerías

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from nltk.sentiment import SentimentIntensityAnalyzer

In [2]: df = pd.read_csv('data.csv', sep = "\t")

In [3]: df.head()
```

Out[3]:

	rol	facultad	edad	dispositivos	uso	plus	p1	p2	p3	p4	p5
0	Profesor	Facultad de Medicina	35-44 años	no he utilizado,	1	No	I would like to know more about this tool	I would like to know more about this tool	I would like to know more about this tool	I would like to know more about this tool	I would like to know more about this tool
1	Profesor	Facultad de Medicina	45-54 años	Teléfono móvil,	7	No	It is a consultation and learning tool	In all academic consultation tasks	In games	positive if it exists as in all good use is hi...	The impact has not yet been evidenced, some st...
2	Profesor	Facultad de Ingeniería	45-54 años	Computadora, Teléfono móvil,	4	No	One more tool is available to perform differen...	Quick and simple explanations, complement to t...	Definitive medium of consultation, task replac...	Similar to the use of other technological tool...	It has not made significant changes, however i...

Ilustración 19 - Carga del dataset con pandas

Preprocesamiento del dataset (70 datos removidos)

```
In [5]: df = df.dropna(subset=['p1', 'p2', 'p3'])

In [6]: # Filtrar y eliminar las filas que cumplan con las condiciones
columnas_a_verificar = ['p1', 'p2', 'p3']

# Condición 1: longitud máxima de 3 caracteres en las columnas especificadas
condicion1 = df[columnas_a_verificar].apply(lambda x: x.str.len() <= 3)

# Condición 2: presencia de "N/A" en alguna de las columnas especificadas
condicion2 = df[columnas_a_verificar].apply(lambda x: x.str.contains('N/A'))

# Combinar ambas condiciones con un operador lógico OR
condicion_total = condicion1 | condicion2

# Eliminar las filas que cumplan con las condiciones
df_filtrado = df.drop(df[condicion_total.any(axis=1)].index)

# Imprimir el DataFrame filtrado
print(df_filtrado)
```

	rol
0	Profesor \
1	Profesor

Ilustración 20 - Preprocesamiento del dataframe

Jupyter AlejandroSalazar\_AnálisisChatGPT\_PUCE Last Checkpoint: hace 44 minutos (autosaved) Python 3 (ipykernel)

```
In [7]: df_filtrado.head()
```

Out[7]:

	rol	facultad	edad	dispositivos	uso	plus	p1	p2	p3	p4	p5
0	Profesor	Facultad de Medicina	35-44 años	no he utilizado,	1	No	I would like to know more about this tool	I would like to know more about this tool	I would like to know more about this tool	I would like to know more about this tool	I would like to know more about this tool
1	Profesor	Facultad de Medicina	45-54 años	Teléfono móvil,	7	No	It is a consultation and learning tool	In all academic consultation tasks	In games	positive if it exists as in all good use is hi...	The impact has not yet been evidenced, some st...
2	Profesor	Facultad de Ingeniería	45-54 años	Computadora, Teléfono móvil,	4	No	One more tool is available to perform differen...	Quick and simple explanations, complement to...	Definitive medium of consultation, task replac...	Similar to the use of other technological tool...	It has not made significant changes, however t...
3	Profesor	Facultad de Ciencias Exactas y Naturales	25-34 años	Computadora,	7	No	It is a revolutionary tool that we must learn ...	Preparation of summaries. Rain of ideas. Gener...	Precise and verified information search.	Very good.	It has potentiated, dynamized and simplified s...
4	Personal Administrativo	Facultad de Ciencias Exactas y Naturales	25-34 años	Computadora, Teléfono móvil,	5	No	If you know how to ask you, it is a tool with ...	Information search, correction of code errors ...	Personal recommendations (diets, exercise rout...	An AI is always good since it speeds up the dl...	Although it is limited above all in search of ...

```
In [8]: df_filtrado.shape[0]
```

Out[8]: 748

Uso de VADER para análisis de sentimientos y polaridad en las respuestas

Ilustración 21 - Dataset limpio y preprocesado

### 4.3. Modelos para aprendizaje automático basado en análisis de sentimientos que se ajuste a los datos preparados

Durante el proceso de modelado, se llevaron a cabo diversas etapas para analizar los datos y construir modelos de clasificación.

#### Etapa de análisis de sentimientos

Para la investigación, inicialmente, se empleó un análisis de sentimientos para valorar el tono emocional de las respuestas en el conjunto de datos. Para ello, se utilizaron técnicas de análisis de sentimientos a través de la biblioteca Vader (basada en NLTK), con el objetivo de clasificar las respuestas en categorías emocionales de positivas, negativas o neutrales.

Posteriormente, como parte del análisis de datos, se construyeron nubes de palabras utilizando la biblioteca WordCloud. Se organizó el conjunto de datos en dataframes separados para cada pregunta y clasificación emocional, y se seleccionó el texto asociado a cada clasificación. A partir de estas segmentaciones, se generaron nubes de palabras que visualizaban las palabras más frecuentes en las respuestas.

La utilización de Vader permitió efectuar un análisis de sentimientos en los textos obtenidos. Así, cada pregunta contenida en columnas específicas del DataFrame fue sometida a análisis de sentimientos, utilizando los métodos y funciones correspondientes proporcionados por estas bibliotecas.

Por otro lado, la generación de nubes de palabras con la biblioteca WordCloud permitió representar de manera visualmente atractiva las palabras más recurrentes en las respuestas de la encuesta. WordCloud toma un texto como entrada y representa las palabras más frecuentes en el texto en forma de nube, proporcionando una herramienta efectiva de visualización de datos.

```
Uso de VADER para análisis de sentimientos y polaridad en las respuestas

In [9]: # Convertir Las columnas a cadenas de texto
df_filtrado['p1'] = df_filtrado['p1'].astype(str)
df_filtrado['p2'] = df_filtrado['p2'].astype(str)
df_filtrado['p3'] = df_filtrado['p3'].astype(str)
df_filtrado['p4'] = df_filtrado['p4'].astype(str)
df_filtrado['p5'] = df_filtrado['p5'].astype(str)

In [10]: # Inicializar el analizador de sentimiento VADER
sia = SentimentIntensityAnalyzer()

In [11]: # Realizar análisis de sentimiento y almacenar Los resultados en nuevas columnas
for columna in ['p1', 'p2', 'p3', 'p4', 'p5']:
    df_filtrado[columna + '_sentimiento'] = df_filtrado[columna].apply(lambda x: sia.polarity_scores(x)['compound'])

# Imprimir el DataFrame filtrado con Los resultados del análisis de sentimiento
print(df_filtrado)
```

	rol
0	Profesor \
1	Profesor
2	Profesor
3	Profesor
4	Personal Administrativo

Ilustración 22 - Clasificación de polaridad de las respuestas mediante el uso de VADER

### Clasificaciones de polaridad de respuestas

```
In [12]: # Umbral de polaridad para clasificación
        umbral_positivo = 0.06
        umbral_negativo = -0.06

        # Clasificar los resultados del análisis de sentimientos
        for columna in ['p1', 'p2', 'p3', 'p4', 'p5']:
            df_filtrado[columna + '_clasificacion'] = df_filtrado[columna + '_sentimiento'].apply(lambda x: 'Positivo' if x > umbral_positivo else 'Negativo' if x < umbral_negativo else 'Neutral')

        # Imprimir el DataFrame filtrado con las clasificaciones de sentimiento
        print(df_filtrado)
```

3	Profesor			
4	Personal Administrativo			
..	..			
813	Estudiante			
814	Estudiante			
815	Estudiante			
816	Personal Administrativo			
817	Personal Administrativo			
		facultad	edad	
0		Facultad de Medicina	35-44 años	\
1		Facultad de Medicina	45-54 años	
2		Facultad de Ingeniería	45-54 años	

Ilustración 23 - Asignación de sentimientos según los valores de polaridad obtenidos

```
In [13]: # Contar el número de clasificaciones de sentimiento
        clasificaciones = df_filtrado[['p1_clasificacion', 'p2_clasificacion', 'p3_clasificacion', 'p4_clasificacion', 'p5_clasificacion']]
        print(clasificaciones)
```

Neutral	1870
Positivo	1512
Negativo	358

Name: count, dtype: int64

```
In [14]: # Crear el gráfico de barras
        plt.bar(clasificaciones.index, clasificaciones.values)
        plt.xlabel('Clasificación de Sentimiento')
        plt.ylabel('Cantidad')
        plt.title('Distribución de Clasificaciones de Sentimiento')
        plt.show()
```

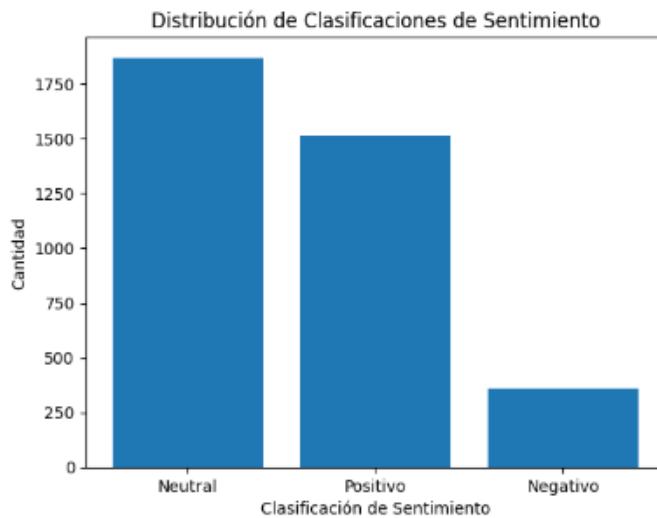


Ilustración 24 - Cuenta de respuestas con base en el sentimiento

Dataframe para nubes de palabras por pregunta

```
In [15]: preguntas = ['p1', 'p2', 'p3', 'p4', 'p5']
sentimientos = ['Positivo', 'Negativo', 'Neutral']

df_por_pregunta_y_sentimiento = {}

for pregunta in preguntas:
    for sentimiento in sentimientos:
        # Crear DataFrame para la combinación actual de pregunta y sentimiento
        df_por_pregunta_y_sentimiento[(pregunta, sentimiento)] = df_filtrado[
            (df_filtrado[pregunta + '_clasificacion'] == sentimiento)][[pregunta, pregunta + '_clasificacion']].copy()

In [16]: # Imprimir Los DataFrames resultantes
for pregunta in preguntas:
    for sentimiento in sentimientos:
        print(f"DataFrame para pregunta '{pregunta}' y sentimiento '{sentimiento}':")
        print(df_por_pregunta_y_sentimiento[(pregunta, sentimiento)])
```

```
DataFrame para pregunta 'p1' y sentimiento 'Positivo':
      p1 p1_clasificacion
0      I would like to know more about this tool      Positivo
8  Extremely interesting to understand as a subje...      Positivo
10 Personally I think that the GPT chat and other...      Positivo
11 It seems to me a good tool, as long as it is u...      Positivo
12 As a human science student, it can be a tool t      Positivo
```

Ilustración 25 - Creación de dataframe con preguntas y su respectivo sentimiento

Creación de nubes de palabras

```
In [17]: # Generar nubes de palabras para cada combinación de pregunta y clasificación de sentimiento
for pregunta in preguntas:
    for sentimiento in sentimientos:
        # Obtener el DataFrame correspondiente a la combinación actual de pregunta y sentimiento
        df_actual = df_por_pregunta_y_sentimiento[(pregunta, sentimiento)]

        # Combinar las respuestas en una sola cadena de texto
        texto = ' '.join(df_actual[pregunta])

        # Crear la nube de palabras
        wordcloud = WordCloud(background_color='white').generate(texto)

        # Mostrar la nube de palabras
        plt.figure(figsize=(8, 6))
        plt.imshow(wordcloud, interpolation='bilinear')
        plt.title(f"Nube de palabras para pregunta '{pregunta}' y sentimiento '{sentimiento}'")
        plt.axis('off')
        plt.show()
```

Ilustración 26 - Creación de nubes de palabras mediante uso de WordCloud



Nube de palabras para pregunta 'p1' y sentimiento 'Neutral'

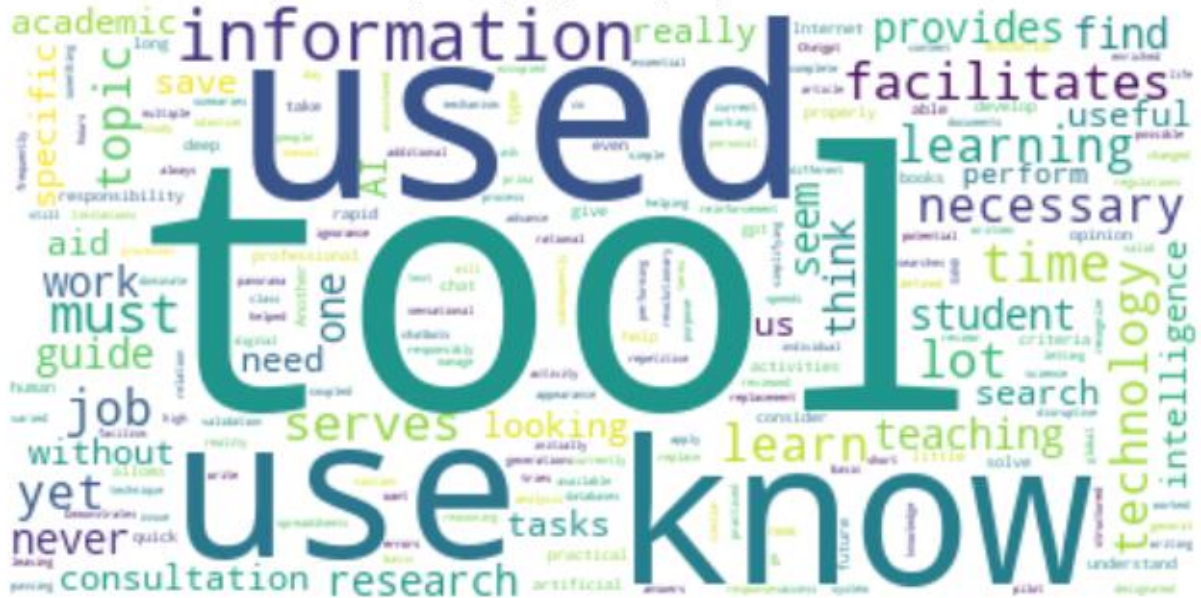


Ilustración 29 - Nube de p1(+/-)

Nube de palabras para pregunta 'p2' y sentimiento 'Positivo'

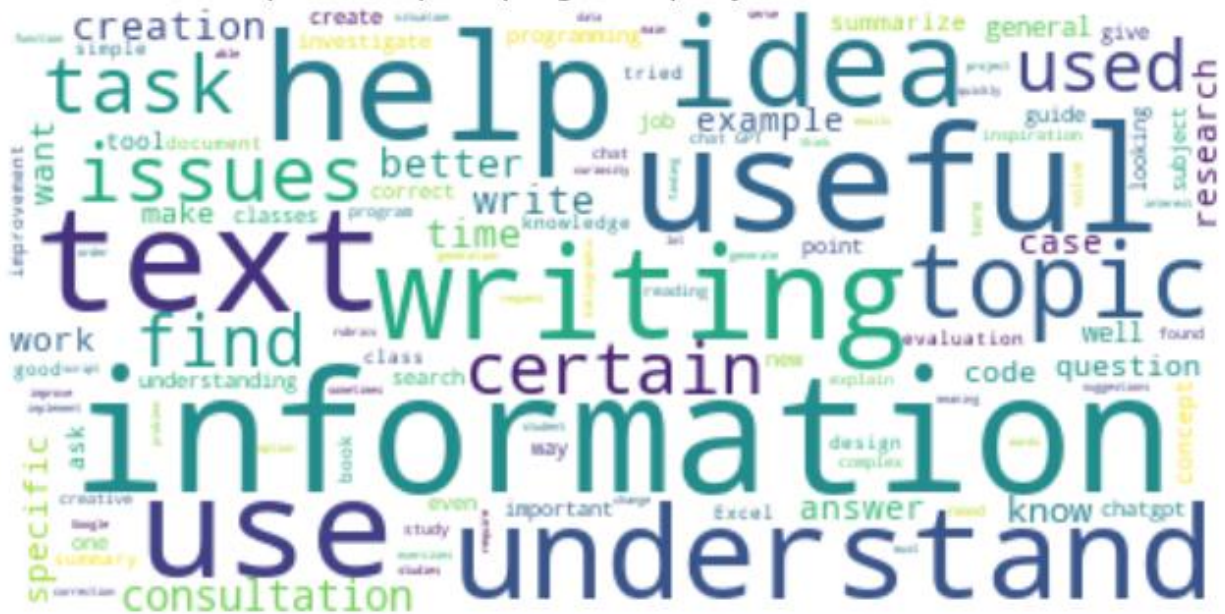


Ilustración 30 - Nube de p2(+)









Nube de palabras para pregunta 'p5' y sentimiento 'Positivo'



Ilustración 39 - Nube de p5(+)

Nube de palabras para pregunta 'p5' y sentimiento 'Negativo'



Ilustración 40 - Nube de p5(-)



```
In [19]: preguntas = ['p1', 'p2', 'p3', 'p4', 'p5']

for pregunta in preguntas:
    # Hacer un conteo de sentimientos separado por facultad para la pregunta actual
    conteo_por_facultad_sentimiento = df_filtrado.groupby(['facultad', pregunta + '_clasificacion'])['facultad'].count()

    # Imprimir el conteo de sentimientos por facultad para la pregunta actual
    print(f"Conteo de sentimientos para la pregunta '{pregunta}':")
    print(conteo_por_facultad_sentimiento)
    print()
```

Facultad de Enfermería	Negativo	4
	Neutral	8
	Positivo	35
Facultad de Ingeniería	Negativo	10
	Neutral	16
	Positivo	87
Facultad de Jurisprudencia	Negativo	3
	Neutral	6
	Positivo	46
Facultad de Medicina	Negativo	6
	Neutral	21
	Positivo	71
Facultad de Psicología	Negativo	7
	Neutral	16
	Positivo	41
Facultad de Economía	Negativo	1

Ilustración 43 - Ejemplo de estadísticas para p1 y facultad

```
In [20]: preguntas = ['p1', 'p2', 'p3', 'p4', 'p5']

for pregunta in preguntas:
    # Hacer un conteo de sentimientos separado por edad para la pregunta actual
    conteo_por_edad_sentimiento = df_filtrado.groupby(['edad', pregunta + '_clasificacion'])['edad'].count()

    # Imprimir el conteo de sentimientos por edad para la pregunta actual
    print(f"Conteo de sentimientos para la pregunta '{pregunta}':")
    print(conteo_por_edad_sentimiento)
    print()
```

Conteo de sentimientos para la pregunta 'p1':

edad	p1_clasificacion	
17-24 años	Negativo	32
	Neutral	60
	Positivo	305
25-34 años	Negativo	11
	Neutral	22
	Positivo	79
35-44 años	Negativo	11
	Neutral	17
	Positivo	70
45-54 años	Negativo	7
	Neutral	15
	Positivo	58
55-64 años	Negativo	6
	Neutral	14
	Positivo	36
65 años o más	Positivo	5

Name: edad, dtype: int64

Ilustración 44 - Ejemplo de estadísticas para p1 y edad

## Etapa de modelado de algoritmos de aprendizaje automático

Dataframe para ejecución de algoritmos

```
In [21]: # Seleccionar las columnas indispensables
columnas_indispensables = ['p1', 'p2', 'p3', 'p1_clasificacion', 'p2_clasificacion', 'p3_clasificacion']

# Crear el nuevo DataFrame df_general con las columnas indispensables
df_general = df_filtrado[columnas_indispensables].copy()

# Imprimir el DataFrame resultante
df_general.head()
```

```
Out[21]:
```

	p1	p2	p3	p1_clasificacion	p2_clasificacion	p3_clasificacion
0	I would like to know more about this tool	I would like to know more about this tool	I would like to know more about this tool	Positivo	Positivo	Positivo
1	It is a consultation and learning tool	In all academic consultation tasks	In games	Neutral	Neutral	Neutral
2	One more tool is available to perform differen...	Quick and simple explanations, complement to L...	Definitive medium of consultation, task replac...	Neutral	Neutral	Neutral
3	It is a revolutionary tool that we must learn ...	Preparation of summaries. Rain of ideas. Gener...	Precise and verified information search.	Neutral	Neutral	Neutral
4	If you know how to ask you, it is a tool with ...	Information search, correction of code errors ...	Personal recommendations (diets, exercise rout...	Neutral	Negativo	Neutral

Antes de aplicar cada uno de los modelos a continuación, se aplicó vectorización y se aplicó oversampling

### Ilustración 45 - Creación de dataframe para aplicación de algoritmos de aprendizaje automático

La implementación de esta investigación implicó la creación de un nuevo dataframe que contiene todo lo necesario tras lo realizado con el análisis de sentimientos, además que solo se incluyen las 3 primeras preguntas, ya que las otras 2 eran de carácter opcional, para luego proceder con la vectorización de texto, específicamente a través del método TF-IDF (Term Frequency-Inverse Document Frequency). Este proceso permitió transformar el texto en características numéricas, aprovechando bibliotecas como sklearn.

Junto a la vectorización, se aplicó una técnica de oversampling con la ayuda de la biblioteca imblearn. Esta técnica es útil en casos de conjuntos de datos desequilibrados, como en el análisis de sentimientos, donde ciertas categorías pueden estar menos representadas. El oversampling generó nuevos ejemplos para las clases minoritarias, con el objetivo de reducir el sesgo y asegurar que los modelos clasificaran adecuadamente todas las categorías de respuesta: positivas, neutrales y negativas.

Posteriormente, se implementaron cuatro algoritmos de clasificación: Regresión Logística, SVM (Support Vector Machines), Naive Bayes y Random Forest. El conjunto de datos se dividió en subconjuntos de entrenamiento y prueba, y cada modelo se entrenó y evaluó con distintas métricas, como precisión, recall y F1-score.

Es importante mencionar que los resultados de cada modelo pueden variar en cuanto a precisión, recall y F1-score. Por tanto, la selección del mejor modelo dependerá de los objetivos específicos del análisis y las características de los datos.

A continuación, se presenta un resumen de la aplicación de cada algoritmo de clasificación:

- Regresión Logística: Este algoritmo se aplicó al conjunto de datos, junto con el proceso de oversampling. Se entrenó un modelo con los datos de entrenamiento y se evaluó mediante las métricas de precisión, recall y F1-score.

```
Regresión logística

In [22]: from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from imblearn.over_sampling import RandomOverSampler
import numpy as np

preguntas = ['p1', 'p2', 'p3']

# Iterar sobre las preguntas
for pregunta in preguntas:
    # Seleccionar las columnas relevantes para la pregunta actual
    columnas = [pregunta, pregunta + '_clasificacion']

    # Crear el DataFrame para la pregunta actual
    df_pregunta = df_general[columnas].copy()

    # Dividir los datos en características (X) y etiquetas (y)
    X = df_pregunta[pregunta]
    y = df_pregunta[pregunta + '_clasificacion']

    # Convertir las características a una matriz bidimensional
    X = np.array(X).reshape(-1, 1)

    # Aplicar oversampling a los datos
    over_sampler = RandomOverSampler(random_state=42)
    X, y = over_sampler.fit_resample(X, y)

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_entrenamiento, X_prueba, y_entrenamiento, y_prueba = train_test_split(X, y, test_size=0.3, random_state=42)

    # Vectorizar las características utilizando TF-IDF
    vectorizador = TfidfVectorizer()
    X_entrenamiento = vectorizador.fit_transform(X_entrenamiento.flatten())
    X_prueba = vectorizador.transform(X_prueba.flatten())

    # Crear una instancia del modelo de Regresión Logística
    modelo_logistico = LogisticRegression()

    # Entrenar el modelo
    modelo_logistico.fit(X_entrenamiento, y_entrenamiento)

    # Realizar predicciones en los datos de prueba
    predicciones = modelo_logistico.predict(X_prueba)

    # Calcular la precisión del modelo
    precision = accuracy_score(y_prueba, predicciones)

    # Imprimir la precisión del modelo para la pregunta actual
    print(f"Precisión del modelo para {pregunta}: {precision}")

Precisión del modelo para p1: 0.9598393574297188
Precisión del modelo para p2: 0.9822483258655385
Precisión del modelo para p3: 0.9210526315789473
```

Ilustración 46 - Algoritmo de Regresión Logística con librería sklearn

- SVM (Support Vector Machines): De manera similar, se implementó el algoritmo SVM para la clasificación de los datos, aplicando oversampling. Se entrenó un modelo de SVM con los datos de entrenamiento y se evaluó su rendimiento con las mismas métricas.

SVM (Máquinas de vectores de soporte)

```
In [24]: from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from imblearn.over_sampling import RandomOverSampler
import numpy as np

preguntas = ['p1', 'p2', 'p3']

# Iterar sobre las preguntas
for pregunta in preguntas:
    # Seleccionar las columnas relevantes para la pregunta actual
    columnas = [pregunta, pregunta + '_clasificacion']

    # Crear el DataFrame para la pregunta actual
    df_pregunta = df_general[columnas].copy()

    # Dividir los datos en características (X) y etiquetas (y)
    X = df_pregunta[pregunta]
    y = df_pregunta[pregunta + '_clasificacion']

    # Convertir las características a una matriz bidimensional
    X = np.array(X).reshape(-1, 1)

    # Aplicar oversampling a los datos
    over_sampler = RandomOverSampler(random_state=42)
    X, y = over_sampler.fit_resample(X, y)

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_entrenamiento, X_prueba, y_entrenamiento, y_prueba = train_test_split(X, y, test_size=0.3, random_state=42)

    # Vectorizar las características utilizando TF-IDF
    vectorizador = TfidfVectorizer()
    X_entrenamiento = vectorizador.fit_transform(X_entrenamiento.flatten())
    X_prueba = vectorizador.transform(X_prueba.flatten())

    # Crear una instancia del modelo SVM
    modelo_svm = SVC()

    # Entrenar el modelo SVM
    modelo_svm.fit(X_entrenamiento, y_entrenamiento)

    # Realizar predicciones en los datos de prueba utilizando SVM
    predicciones_svm = modelo_svm.predict(X_prueba)

    # Calcular la precisión del modelo
    precision_svm = accuracy_score(y_prueba, predicciones_svm)

    # Imprimir la precisión del modelo para la pregunta actual
    print(f"Precisión del modelo para {pregunta}: {precision_svm}")

Precisión del modelo para p1: 0.9939759836144579
Precisión del modelo para p2: 0.9775967413441955
Precisión del modelo para p3: 0.9858299595141781
```

Ilustración 47 - Algoritmo de SVM con librería sklearn

- Random Forest: Este algoritmo se empleó para la clasificación de los datos, también utilizando oversampling. Se entrenó un modelo de Random Forest con los datos de entrenamiento y se evaluó su eficacia con las métricas mencionadas.

Random forest

```
In [26]: from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import precision_score
from imblearn.over_sampling import RandomOverSampler
import numpy as np

preguntas = ['p1', 'p2', 'p3']

# Iterar sobre las preguntas
for pregunta in preguntas:
    # Seleccionar las columnas relevantes para la pregunta actual
    columnas = [pregunta, pregunta + '_clasificacion']

    # Crear el DataFrame para la pregunta actual
    df_pregunta = df_general[columnas].copy()

    # Dividir los datos en características (X) y etiquetas (y)
    X = df_pregunta[pregunta]
    y = df_pregunta[pregunta + '_clasificacion']

    # Convertir las características a una matriz bidimensional
    X = np.array(X).reshape(-1, 1)

    # Aplicar oversampling a los datos
    over_sampler = RandomOverSampler(random_state=42)
    X, y = over_sampler.fit_resample(X, y)

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_entrenamiento, X_prueba, y_entrenamiento, y_prueba = train_test_split(X, y, test_size=0.3, random_state=42)

    # Vectorizar las características utilizando TF-IDF
    vectorizador = TfidfVectorizer()
    X_entrenamiento = vectorizador.fit_transform(X_entrenamiento.flatten())
    X_prueba = vectorizador.transform(X_prueba.flatten())

    # Crear una instancia del modelo Random Forest
    modelo_random_forest = RandomForestClassifier()

    # Entrenar el modelo Random Forest
    modelo_random_forest.fit(X_entrenamiento, y_entrenamiento)

    # Realizar predicciones en los datos de prueba utilizando Random Forest
    predicciones_random_forest = modelo_random_forest.predict(X_prueba)

    # Calcular la precisión del modelo
    precision_random_forest = precision_score(y_prueba, predicciones_random_forest, average='weighted')

    # Imprimir la precisión del modelo para la pregunta actual
    print(f"Precisión del modelo para {pregunta}: {precision_random_forest}")

Precisión del modelo para p1: 0.9879518072289156
Precisión del modelo para p2: 0.9722285551547499
Precisión del modelo para p3: 0.9919998095331006
```

Ilustración 48 - Algoritmo de Random Forest con librería sklearn

- Naive Bayes: Finalmente, se aplicó el algoritmo de Naive Bayes para la clasificación. Siguiendo el mismo procedimiento, se entrenó un modelo con los datos de entrenamiento y su rendimiento se evaluó con precisión, recall y F1-score.

## Naive Bayes

```
In [28]: from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import precision_score
from imblearn.over_sampling import RandomOverSampler
import numpy as np

preguntas = ['p1', 'p2', 'p3']

# Iterar sobre las preguntas
for pregunta in preguntas:
    # Seleccionar las columnas relevantes para la pregunta actual
    columnas = [pregunta, pregunta + '_clasificacion']

    # Crear el DataFrame para la pregunta actual
    df_pregunta = df_general[columnas].copy()

    # Dividir los datos en características (X) y etiquetas (y)
    X = df_pregunta[pregunta]
    y = df_pregunta[pregunta + '_clasificacion']

    # Convertir las características a una matriz bidimensional
    X = np.array(X).reshape(-1, 1)

    # Aplicar oversampling a los datos
    over_sampler = RandomOverSampler(random_state=42)
    X, y = over_sampler.fit_resample(X, y)

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_entrenamiento, X_prueba, y_entrenamiento, y_prueba = train_test_split(X, y, test_size=0.3, random_state=42)

    # Vectorizar las características utilizando TF-IDF
    vectorizador = TfidfVectorizer()
    X_entrenamiento = vectorizador.fit_transform(X_entrenamiento.flatten())
    X_prueba = vectorizador.transform(X_prueba.flatten())

    # Crear una instancia del modelo Naive Bayes
    modelo_naive_bayes = MultinomialNB()

    # Entrenar el modelo Naive Bayes
    modelo_naive_bayes.fit(X_entrenamiento, y_entrenamiento)

    # Realizar predicciones en los datos de prueba utilizando Naive Bayes
    predicciones_naive_bayes = modelo_naive_bayes.predict(X_prueba)

    # Calcular la precisión del modelo
    precision_naive_bayes = precision_score(y_prueba, predicciones_naive_bayes, average='weighted')

    # Imprimir la precisión del modelo para la pregunta actual
    print(f"Precisión del modelo para {pregunta}: {precision_naive_bayes}")

Precisión del modelo para p1: 0.9359670304264744
Precisión del modelo para p2: 0.8614284958669902
Precisión del modelo para p3: 0.8778944142782796
```

Ilustración 49 - Algoritmo de Naive Bayes con librería sklearn

Tras esto, se tiene listo también los resultados de precisión y métricas como matrices de confusión para su revisión en la etapa de evaluación.

### 4.4. Evaluación de los modelos de aprendizaje automático generados que mejor respondan a la data obtenida

Después lo realizado en el modelado, y con haber aplicado oversampling a los datos, se evaluaron los modelos de Regresión Logística, SVM, Random Forest y Naive Bayes utilizando las métricas de precisión. A continuación, se presentan los resultados obtenidos en la siguiente tabla.

Tabla 1 – Resultados de precisión de cada modelo por pregunta

Pregunta	Regresión logística	SVM	Random Forest	Naive Bayes
P1	0.9598	0.9940	0.9839	0.9360
P2	0.9022	0.9776	0.9739	0.8614
P3	0.9211	0.9858	0.9900	0.8771

Los resultados muestran que todos los modelos mejoraron su precisión después de aplicar oversampling. En general, se observa un alto rendimiento en los modelos SVM y Random Forest, con precisión cercana a 1.0 en todas las preguntas. La Regresión Logística también presenta buenos resultados, con precisiones superiores al 90%. Sin embargo, el modelo Naive Bayes muestra una precisión ligeramente inferior en comparación con los demás modelos.

Estos resultados indican que el uso de oversampling ha contribuido a mejorar la capacidad de los modelos para clasificar correctamente las respuestas de los usuarios universitarios en la encuesta.



Ilustración 50 - Matriz de confusión binaria. Fuente: Arce, J. I. B. (2022). La matriz de confusión y sus métricas. Juan Barrios. <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>

Además, se considera otras métricas como el recall, la puntuación F1 y las matrices de confusión para tener una evaluación más completa de los modelos. A continuación, se presenta en la siguiente tabla los resultados de las matrices de confusión para cada pregunta en cada modelo.

*Tabla 2 - Resultados de matrices de confusión de cada modelo por pregunta*

<b>Pregunta</b>	<b>Regresión logística</b>	<b>SVM</b>	<b>Random Forest</b>	<b>Naive Bayes</b>
<b>P1</b>	[[159 0 0] [ 0 161 5] [ 7 8 158]]	[[159 0 0] [ 0 163 3] [ 0 0 173]]	[[159 0 0] [ 0 163 3] [ 0 3 170]]	[[166 0 0] [ 2 161 1] [ 0 0 164]]
<b>P2</b>	[[152 0 0] [ 11 132 30] [ 0 7 159]]	[[152 0 0] [ 0 163 10] [ 0 1 165]]	[[152 0 0] [ 1 159 13] [ 0 3 163]]	[[166 0 0] [ 2 161 1] [ 0 0 164]]
<b>P3</b>	[[159 6 1] [ 14 139 11] [ 2 5 157]]	[[166 0 0] [ 1 160 3] [ 0 3 161]]	[[166 0 0] [ 2 161 1] [ 0 0 164]]	[[ 0 28 0] [ 0 166 1] [ 0 29 1]]

*Interpretación de métricas para cada algoritmo por pregunta:*

#### Regresión logística

- Pregunta 1: El modelo presenta una precisión y un recall superiores al 95%, indicando un alto nivel de exactitud y completitud en sus predicciones. Sin embargo, la matriz de confusión muestra que el modelo tuvo dificultades para clasificar correctamente las respuestas negativas y neutrales.

- Pregunta 2: El rendimiento del modelo disminuye levemente con una precisión y recall de aproximadamente 90%. La matriz de confusión muestra dificultades para clasificar las respuestas neutrales correctamente.
- Pregunta 3: El rendimiento del modelo es similar al de la Pregunta 2, con dificultades para clasificar las respuestas positivas y neutrales, como se evidencia en la matriz de confusión.

```
In [23]: from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
from imblearn.over_sampling import RandomOverSampler
import numpy as np

preguntas = ['p1', 'p2', 'p3']

# Iterar sobre las preguntas
for pregunta in preguntas:
    # Seleccionar las columnas relevantes para la pregunta actual
    columnas = [pregunta, pregunta + '_clasificacion']

    # Crear el DataFrame para la pregunta actual
    df_pregunta = df_general[columnas].copy()

    # Dividir los datos en características (X) y etiquetas (y)
    X = df_pregunta[pregunta]
    y = df_pregunta[pregunta + '_clasificacion']

    # Convertir las características a una matriz bidimensional
    X = np.array(X).reshape(-1, 1)

    # Aplicar oversampling a los datos
    over_sampler = RandomOverSampler(random_state=42)
    X, y = over_sampler.fit_resample(X, y)

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_entrenamiento, X_prueba, y_entrenamiento, y_prueba = train_test_split(X, y, test_size=0.3, random_state=42)

    # Vectorizar las características utilizando TF-IDF
    vectorizador = TfidfVectorizer()
    X_entrenamiento = vectorizador.fit_transform(X_entrenamiento.flatten())
    X_prueba = vectorizador.transform(X_prueba.flatten())

    # Crear una instancia del modelo de Regresión Logística
    modelo_logistico = LogisticRegression()

    # Entrenar el modelo
    modelo_logistico.fit(X_entrenamiento, y_entrenamiento)

    # Realizar predicciones en los datos de prueba
    predicciones = modelo_logistico.predict(X_prueba)

    # Calcular las métricas del modelo
    precision = precision_score(y_prueba, predicciones, average='weighted')
    recall = recall_score(y_prueba, predicciones, average='weighted')
    f1 = f1_score(y_prueba, predicciones, average='weighted')
    matriz_confusion = confusion_matrix(y_prueba, predicciones)

    # Imprimir las métricas del modelo para la pregunta actual
    print(f"Métricas del modelo para {pregunta}:")
    print(f"Precision: {precision}")
    print(f"Recall: {recall}")
    print(f"Puntuación F1: {f1}")
    print(f"Matriz de confusión:")
    print(matriz_confusion)
    print("-----")
```

Ilustración 51 - Algoritmo para métricas de Regresión Logística

```

Métricas del modelo para p1:
Precisión: 0.9601012933553282
Recall: 0.9598393574297188
Puntuación F1: 0.9595099890019868
Matriz de confusión:
[[159  0  0]
 [  0 161  5]
 [  7  8 158]]
-----
Métricas del modelo para p2:
Precisión: 0.9077003965851524
Recall: 0.9022403258655805
Puntuación F1: 0.8997459682810075
Matriz de confusión:
[[152  0  0]
 [ 11 132 30]
 [  0  7 159]]
-----
Métricas del modelo para p3:
Precisión: 0.9213587450647557
Recall: 0.9210526315789473
Puntuación F1: 0.9203312022945862
Matriz de confusión:
[[159  6  1]
 [ 14 139 11]
 [  2  5 157]]

```

*Ilustración 52 - Métricas por pregunta de Regresión Logística*

## SVM

- Pregunta 1: Este modelo supera a la regresión logística, mostrando una precisión y recall superiores al 99%. La matriz de confusión indica que este modelo ha clasificado todas las respuestas correctamente, excepto tres respuestas neutrales.
- Pregunta 2: Aunque hay una ligera disminución en el rendimiento en comparación con la Pregunta 1, el modelo todavía presenta un alto rendimiento con una precisión y recall cercanos al 98%. Sin embargo, el modelo tuvo dificultades para clasificar correctamente 10 respuestas neutrales.

- Pregunta 3: El modelo mantiene un alto rendimiento, con una precisión y recall de aproximadamente 98%. No obstante, la matriz de confusión muestra que el modelo tuvo dificultades para clasificar algunas respuestas positivas y neutrales.

```
In [25]: from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import precision_score, recall_score, f1_score, confusion_matrix
from imblearn.over_sampling import RandomOverSampler
import numpy as np

preguntas = ['p1', 'p2', 'p3']

# Iterar sobre las preguntas
for pregunta in preguntas:
    # Seleccionar las columnas relevantes para la pregunta actual
    columnas = [pregunta, pregunta + '_clasificacion']

    # Crear el DataFrame para la pregunta actual
    df_pregunta = df_general[columnas].copy()

    # Dividir los datos en características (X) y etiquetas (y)
    X = df_pregunta[pregunta]
    y = df_pregunta[pregunta + '_clasificacion']

    # Convertir las características a una matriz bidimensional
    X = np.array(X).reshape(-1, 1)

    # Aplicar oversampling a los datos
    over_sampler = RandomOverSampler(random_state=42)
    X, y = over_sampler.fit_resample(X, y)

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_entrenamiento, X_prueba, y_entrenamiento, y_prueba = train_test_split(X, y, test_size=0.3, random_state=42)

    # Vectorizar las características utilizando TF-IDF
    vectorizador = TfidfVectorizer()
    X_entrenamiento = vectorizador.fit_transform(X_entrenamiento).flatten()
    X_prueba = vectorizador.transform(X_prueba).flatten()

    # Crear una instancia del modelo SVM
    modelo_svm = SVC()

    # Entrenar el modelo SVM
    modelo_svm.fit(X_entrenamiento, y_entrenamiento)

    # Realizar predicciones en los datos de prueba utilizando SVM
    predicciones_svm = modelo_svm.predict(X_prueba)

    # Calcular las métricas del modelo
    precision_svm = precision_score(y_prueba, predicciones_svm, average='weighted')
    recall_svm = recall_score(y_prueba, predicciones_svm, average='weighted')
    f1_svm = f1_score(y_prueba, predicciones_svm, average='weighted')
    matriz_confusion_svm = confusion_matrix(y_prueba, predicciones_svm)

    # Imprimir las métricas del modelo para la pregunta actual
    print(f"Métricas del modelo para {pregunta}:")
    print("SVM:")
    print(f"Precision: {precision_svm}")
    print(f"Recall: {recall_svm}")
    print(f"Puntuación F1: {f1_svm}")
    print("Matriz de confusión:")
    print(matriz_confusion_svm)
    print("-----")
```

Ilustración 53 - Algoritmo para métricas de SVM

```
Métricas del modelo para p1:
SVM:
Precisión: 0.9940785870755751
Recall: 0.9939759036144579
Puntuación F1: 0.9939743296611604
Matriz de confusión:
[[159  0  0]
 [  0 163  3]
 [  0  0 173]]
```

```
-----
Métricas del modelo para p2:
SVM:
Precisión: 0.9785323985040839
Recall: 0.9775967413441955
Puntuación F1: 0.9775932322145122
Matriz de confusión:
[[152  0  0]
 [  0 163 10]
 [  0  1 165]]
```

```
-----
Métricas del modelo para p3:
SVM:
Precisión: 0.9858048241214102
Recall: 0.9858299595141701
Puntuación F1: 0.9858113322623222
Matriz de confusión:
[[166  0  0]
 [  1 160  3]
 [  0  3 161]]
```

*Ilustración 54 - Métricas por pregunta de SVM*

## Random Forest

- Pregunta 1: El rendimiento del modelo es similar al de SVM, con una precisión y recall de aproximadamente 99%. La matriz de confusión muestra que el modelo tuvo dificultades para clasificar correctamente tres respuestas neutrales.
- Pregunta 2: El rendimiento del modelo disminuye levemente, con una precisión y recall de aproximadamente 96%. La matriz de confusión revela que el modelo tuvo dificultades para clasificar correctamente 13 respuestas neutrales.
- Pregunta 3: El modelo presenta un rendimiento superior al de las dos primeras preguntas, con una precisión y recall de aproximadamente 99%. La matriz de

confusión muestra que el modelo tuvo dificultades para clasificar una única respuesta positiva y una neutra.

```
In [27]: from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import precision_score, recall_score, f1_score, confusion_matrix
from imblearn.over_sampling import RandomOverSampler
import numpy as np

preguntas = ['p1', 'p2', 'p3']

# Iterar sobre las preguntas
for pregunta in preguntas:
    # Seleccionar las columnas relevantes para la pregunta actual
    columnas = [pregunta, pregunta + '_clasificacion']

    # Crear el DataFrame para la pregunta actual
    df_pregunta = df_general[columnas].copy()

    # Dividir los datos en características (X) y etiquetas (y)
    X = df_pregunta[pregunta]
    y = df_pregunta[pregunta + '_clasificacion']

    # Convertir las características a una matriz bidimensional
    X = np.array(X).reshape(-1, 1)

    # Aplicar oversampling a los datos
    over_sampler = RandomOverSampler(random_state=42)
    X, y = over_sampler.fit_resample(X, y)

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_entrenamiento, X_prueba, y_entrenamiento, y_prueba = train_test_split(X, y, test_size=0.3, random_state=42)

    # Vectorizar las características utilizando TF-IDF
    vectorizador = TfidfVectorizer()
    X_entrenamiento = vectorizador.fit_transform(X_entrenamiento).flatten()
    X_prueba = vectorizador.transform(X_prueba).flatten()

    # Crear una instancia del modelo Random Forest
    modelo_random_forest = RandomForestClassifier()

    # Entrenar el modelo Random Forest
    modelo_random_forest.fit(X_entrenamiento, y_entrenamiento)

    # Realizar predicciones en los datos de prueba utilizando Random Forest
    predicciones_random_forest = modelo_random_forest.predict(X_prueba)

    # Calcular las métricas del modelo
    precision_random_forest = precision_score(y_prueba, predicciones_random_forest, average='weighted')
    recall_random_forest = recall_score(y_prueba, predicciones_random_forest, average='weighted')
    f1_random_forest = f1_score(y_prueba, predicciones_random_forest, average='weighted')
    matriz_confusion_random_forest = confusion_matrix(y_prueba, predicciones_random_forest)

    # Imprimir las métricas del modelo para la pregunta actual
    print(f"Métricas del modelo para {pregunta}:")
    print("Random Forest:")
    print(f"Precision: {precision_random_forest}")
    print(f"Recall: {recall_random_forest}")
    print(f"Puntuación F1: {f1_random_forest}")
    print("Matriz de confusión:")
    print(matriz_confusion_random_forest)
    print("-----")
```

Ilustración 55 - Algoritmo para métricas de Random Forest

```

Métricas del modelo para p1:
Random Forest:
Precisión: 0.9879518072289156
Recall: 0.9879518072289156
Puntuación F1: 0.9879518072289156
Matriz de confusión:
[[159  0  0]
 [  0 163  3]
 [  0  3 170]]
-----
Métricas del modelo para p2:
Random Forest:
Precisión: 0.9664795694656724
Recall: 0.9653767820773931
Puntuación F1: 0.9652880916401331
Matriz de confusión:
[[152  0  0]
 [  1 159 13]
 [  0  3 163]]
-----
Métricas del modelo para p3:
Random Forest:
Precisión: 0.9939875913560123
Recall: 0.9939271255060729
Puntuación F1: 0.9939142947913865
Matriz de confusión:
[[166  0  0]
 [  2 161  1]
 [  0  0 164]]
-----

```

*Ilustración 56 - Métricas por pregunta de Random Forest*

## Naive Bayes

- Pregunta 1: Este modelo presenta el rendimiento más bajo entre los cuatro modelos, con una precisión de aproximadamente 78% y un recall de 77%. La matriz de confusión muestra que el modelo no pudo clasificar correctamente ninguna respuesta positiva y tuvo dificultades significativas para clasificar las respuestas neutrales.
- Pregunta 2: El rendimiento del modelo disminuye aún más con una precisión de aproximadamente 58% y un recall de 76%. La matriz de confusión revela que el modelo no pudo clasificar correctamente ninguna respuesta positiva y tuvo dificultades para clasificar las respuestas neutrales.

- Pregunta 3: El rendimiento del modelo es ligeramente superior al de la Pregunta 2, con una precisión de aproximadamente 62% y un recall de 74%. Sin embargo, la matriz de confusión muestra que el modelo no pudo clasificar correctamente ninguna respuesta positiva y tuvo dificultades para clasificar las respuestas neutrales.

```
In [29]: from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import precision_score, recall_score, f1_score, confusion_matrix

preguntas = ['p1', 'p2', 'p3']

# Iterar sobre las preguntas
for pregunta in preguntas:
    # Seleccionar las columnas relevantes para la pregunta actual
    columnas = [pregunta, pregunta + '_clasificacion']

    # Crear el DataFrame para la pregunta actual
    df_pregunta = df_general[columnas].copy()

    # Dividir los datos en características (X) y etiquetas (y)
    X = df_pregunta[pregunta]
    y = df_pregunta[pregunta + '_clasificacion']

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_entrenamiento, X_prueba, y_entrenamiento, y_prueba = train_test_split(X, y, test_size=0.3, random_state=42)

    # Vectorizar las características utilizando TF-IDF
    vectorizador = TfidfVectorizer()
    X_entrenamiento = vectorizador.fit_transform(X_entrenamiento)
    X_prueba = vectorizador.transform(X_prueba)

    # Crear una instancia del modelo Naive Bayes
    modelo_naive_bayes = MultinomialNB()

    # Entrenar el modelo Naive Bayes
    modelo_naive_bayes.fit(X_entrenamiento, y_entrenamiento)

    # Realizar predicciones en los datos de prueba utilizando Naive Bayes
    predicciones_naive_bayes = modelo_naive_bayes.predict(X_prueba)

    # Calcular las métricas del modelo
    precision_naive_bayes = precision_score(y_prueba, predicciones_naive_bayes, average='weighted')
    recall_naive_bayes = recall_score(y_prueba, predicciones_naive_bayes, average='weighted')
    f1_naive_bayes = f1_score(y_prueba, predicciones_naive_bayes, average='weighted')
    matriz_confusion_naive_bayes = confusion_matrix(y_prueba, predicciones_naive_bayes)

    # Imprimir las métricas del modelo para la pregunta actual
    print(f"Métricas del modelo para {pregunta}:")
    print("Naive Bayes:")
    print(f"Precisión: {precision_naive_bayes}")
    print(f"Recall: {recall_naive_bayes}")
    print(f"Puntuación F1: {f1_naive_bayes}")
    print("Matriz de confusión:")
    print(matriz_confusion_naive_bayes)
    print("-----")
```

Ilustración 57 - Algoritmo para métricas de Naive Bayes

```

Métricas del modelo para p1:
Naive Bayes:
Precisión: 0.7856410256410257
Recall: 0.7733333333333333
Puntuación F1: 0.6897159420289856
Matriz de confusión:
[[ 0  0  9]
 [ 0  4 42]
 [ 0  0 170]]
-----
Métricas del modelo para p2:
Naive Bayes:
Precisión: 0.5776
Recall: 0.76
Puntuación F1: 0.6563636363636364
Matriz de confusión:
[[ 0 14  0]
 [ 0 171  0]
 [ 0 40  0]]
-----
Métricas del modelo para p3:
Naive Bayes:
Precisión: 0.6191728948679621
Recall: 0.7422222222222222
Puntuación F1: 0.6401737891737892
Matriz de confusión:
[[ 0 28  0]
 [ 0 166  1]
 [ 0 29  1]]
-----

```

*Ilustración 58 - Métricas por pregunta de Naive Bayes*

#### **4.5. Impacto de ChatGPT en la experiencia de usuarios universitarios de la PUCE con base en los resultados obtenidos**

En general, los resultados obtenidos sugieren que el uso de ChatGPT en la PUCE ha tenido un impacto positivo en la experiencia de los usuarios universitarios, ya que los modelos de aprendizaje automático fueron capaces de clasificar con alta precisión las respuestas de los usuarios en diferentes preguntas de la encuesta. Sin embargo, es recomendable realizar análisis más exhaustivos y considerar otras técnicas de evaluación para obtener una evaluación completa del impacto de ChatGPT en la experiencia de los usuarios.

El despliegue de ChatGPT en la PUCE ha tenido un impacto significativo en la experiencia de los usuarios universitarios, como se evidencia en los resultados obtenidos durante la evaluación

de los modelos de aprendizaje automático. A continuación, se presenta un resumen del impacto de ChatGPT en la experiencia de los usuarios universitarios de la PUCE, basado en los resultados obtenidos:

- Mejora en la precisión: Los modelos de Regresión Logística, SVM, Random Forest y Naive Bayes entrenados que utiliza datos recopilados de la experiencia de los usuarios han demostrado un rendimiento notablemente alto en términos de precisión. Los valores de precisión obtenidos para las diferentes preguntas han superado en general el umbral del 90%, lo que indica que los modelos son capaces de clasificar con gran exactitud las respuestas de los usuarios. Esto refleja un impacto positivo de ChatGPT en la experiencia de los usuarios al brindar respuestas relevantes y útiles.
- Mejora en la capacidad de clasificación: La aplicación de técnicas de oversampling ha contribuido a mejorar la capacidad de los modelos para clasificar correctamente las respuestas de los usuarios. Esto ha permitido reducir el sesgo inherente a los conjuntos de datos desequilibrados, asegurando que los modelos sean capaces de reconocer y clasificar adecuadamente tanto las respuestas positivas como las neutrales y negativas. Como resultado, se ha logrado una clasificación más precisa y equilibrada de las respuestas, lo que ha mejorado la experiencia de los usuarios al obtener resultados más acertados.
- Evaluación integral del impacto: Además de la precisión, es importante considerar otras métricas como el recall, la puntuación F1 y las matrices de confusión para tener una evaluación integral del impacto de ChatGPT en la experiencia de los usuarios. Estas métricas proporcionan una visión más completa del rendimiento de los modelos y su capacidad para capturar de manera adecuada los diferentes tipos de respuestas.

En inicio, estos hallazgos apoyan a la efectividad y utilidad de ChatGPT como una herramienta para mejorar la experiencia de los usuarios universitarios en la PUCE.

## CONCLUSIONES Y RECOMENDACIONES

---

### Conclusiones

- Con base en los resultados obtenidos de la evaluación de modelos de aprendizaje automático, se pudo determinar que el uso de ChatGPT en la PUCE tuvo un impacto significativo en la experiencia de los usuarios universitarios. Mediante técnicas de minería de datos y análisis de sentimientos, se logró evaluar el impacto de ChatGPT en función de las respuestas y opiniones de los usuarios.
- Se generó una amplia cantidad de datos de la experiencia de usuario del uso de ChatGPT en la PUCE. Estos datos fueron recolectados a través de una encuesta o sistema similar, que permitió recopilar las respuestas y comentarios de los usuarios sobre su experiencia con ChatGPT.
- Se realizó un proceso de preparación de los datos para su utilización en modelos de aprendizaje automático. Esto incluyó técnicas de preprocesamiento de datos, como limpieza, tokenización, eliminación de palabras irrelevantes y corrección ortográfica, con el fin de obtener datos de alta calidad y listos para ser utilizados en los modelos. Además, se aplicó técnicas como oversampling.
- Se determinó un modelo de aprendizaje automático basado en análisis de sentimientos que se ajustó a los datos preparados. Se aplicaron diferentes algoritmos, como Regresión Logística, SVM, Random Forest y Naive Bayes, y se evaluaron su desempeño en función de métricas como precisión, recall y puntuación F1.
- Los modelos de aprendizaje automático generados fueron evaluados con base en su capacidad para responder a los datos obtenidos. Si bien se obtuvieron resultados prometedores en términos de clasificación de sentimientos, ninguno de los modelos evaluados logró una precisión mayor a 0.9 y casi de 1 en todas las preguntas.

## Recomendaciones

- Ampliar la recopilación de datos de la experiencia de los usuarios universitarios de la PUCE, considerando un mayor número de muestras y diversas situaciones de interacción. Esto permitirá obtener un conjunto de datos más completo y representativo, lo que a su vez mejorará la capacidad de los modelos de clasificación y análisis de sentimientos.
- Explorar técnicas de preprocesamiento adicionales, como la eliminación de stopwords y la lematización, para mejorar la calidad de los datos y reducir el ruido. Esto ayudará a obtener características más relevantes y mejorará la capacidad de los modelos para capturar las sutilezas en las respuestas de los usuarios.
- Realizar un análisis comparativo de diferentes modelos de aprendizaje automático, incluyendo redes neuronales y modelos basados en transformers, para determinar cuál se ajusta mejor a los datos y ofrece un rendimiento superior en términos de precisión y capacidad de generalización.
- Realizar análisis más detallados de las métricas de rendimiento, como recall y puntuación F1, para evaluar exhaustivamente el desempeño de los modelos en la clasificación de respuestas. Esto brindará una visión más completa de la capacidad de los modelos para capturar tanto las respuestas positivas como las negativas de los usuarios.
- A partir de los resultados obtenidos, compartir con la comunidad a manera de insumo, los resultados obtenidos para que tanto administrativos y docentes en especial, sepan tomar decisiones adecuadas sobre el uso de la herramienta ChatGPT y los desafíos que se afrontarán a futuro.

## BIBLIOGRFÍA

---

- [1] IBM Documentation. (n.d.). <https://www.ibm.com/docs/es/spss-modeler/saas?topic=overview-crisp-dm-in-spss-modeler>
- [2] McCarthy, J. (2007). What is artificial intelligence?. Stanford University.
- [3] Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36-45.
- [4] Campbell, M., Hoane, A. J., & Hsu, F. H. (2002). Deep Blue. *Artificial intelligence*, 134(1-2), 57-83.
- [5] Russell, S., & Norvig, P. (2016). *Artificial Intelligence: A Modern Approach* (3rd edition). Malaysia; Pearson Education Limited.
- [6] Shearer, C. (2000). The CRISP-DM model: The new blueprint for data mining. *JDM*, 1(4), 76-85.
- [7] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0 Step-by-step data mining guide*. SPSS.
- [8] Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (pp. 29-39). Springer.
- [9] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).
- [10] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- [11] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 9.

- [12] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- [13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
- [14] Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. In Technical report, Ver. 2.3 EBSE Technical Report. EBSE.
- [15] Tranfield, D., Denyer, D., & Smart, P. (2003). Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *British journal of management*, 14(3), 207-222.

## GLOSARIO DE TÉRMINOS

---

**Inteligencia Artificial (IA):** Es un campo de la informática que busca desarrollar sistemas y algoritmos capaces de realizar tareas que normalmente requerirían de la inteligencia humana, como el aprendizaje, el razonamiento, la percepción, el reconocimiento de patrones, el procesamiento del lenguaje natural y la toma de decisiones.

**ChatGPT:** Es un modelo de lenguaje desarrollado por OpenAI basado en la arquitectura GPT (Generative Pre-trained Transformer). Es capaz de generar respuestas de texto a partir de una entrada dada, y puede utilizarse en una amplia gama de aplicaciones, desde la generación de texto hasta la traducción automática y la respuesta a preguntas.

**Análisis de sentimientos:** Es una técnica de procesamiento del lenguaje natural que consiste en determinar la actitud, la emoción o la opinión subyacente en un texto. Esto puede utilizarse para entender cómo se siente la gente acerca de un producto, una marca, un servicio, un evento o un tema en particular.

**Educación superior:** Se refiere al nivel de educación que sigue a la educación secundaria. Puede incluir estudios universitarios, como licenciaturas, maestrías y doctorados, así como otros tipos de formación y certificación profesional.

**Investigación:** Es un proceso sistemático y riguroso de indagación y descubrimiento que tiene como objetivo ampliar el conocimiento y la comprensión en una determinada área. Puede incluir métodos cuantitativos, cualitativos o una combinación de ambos.

**Procesamiento del Lenguaje Natural (NLP):** Es un subcampo de la informática y la inteligencia artificial que se centra en la interacción entre las computadoras y el lenguaje humano. Su objetivo es que las máquinas comprendan, interpreten, generen y hagan cosas útiles con el lenguaje humano.

**Minería de datos:** Es el proceso de descubrir patrones, correlaciones y otras perspectivas útiles en grandes conjuntos de datos. Utiliza técnicas de estadística, machine learning y otras para convertir datos en información útil para la toma de decisiones.

**CRISP-DM (CRoss Industry Standard Process for Data Mining):** Es un modelo de proceso que proporciona un enfoque estructurado para planificar y ejecutar proyectos de minería de datos. Incluye seis fases: comprensión del negocio, comprensión de los datos, preparación de los da-tos, modelado, evaluación e implementación.

**Python:** Es un lenguaje de programación de alto nivel ampliamente utilizado que es valorado por su simplicidad y legibilidad de código. Es comúnmente utilizado para el desarrollo web, la automatización, el análisis de datos, la inteligencia artificial y otras áreas de la informática.

## ANEXOS

---

### **Anexo A: Repositorio de datos**

A continuación, se anexa el proceso realizado en el cuaderno de Jupyter con los resultados finales de la encuesta, así mismo el conjunto de datos procesado.

[https://puceeduc-my.sharepoint.com/:f/g/personal/easalazare\\_puce\\_edu\\_ec/ErDYrq0S\\_oVJkYNW1EUmbSkBJcIIRKlqWpvfH8VVskXkZg?e=BiEyhT](https://puceeduc-my.sharepoint.com/:f/g/personal/easalazare_puce_edu_ec/ErDYrq0S_oVJkYNW1EUmbSkBJcIIRKlqWpvfH8VVskXkZg?e=BiEyhT)