Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Carrera de Computación

Ajuste del modelo Gemma para la descripción de tablas de datos en formato Markdown.

Tuning the Gemma model for describing data tables in Markdown format

Trabajo de Integración Curricular, previo a la obtención del título de Ingeniero en Ciencias de la Computación.

AUTOR:

Patricio Oswaldo Paredes Chamba

DIRECTOR:

Ing. Oscar Miguel Cumbicus Pineda, Mg.Sc.

Loja – Ecuador

2024

Certificación

Loja, 24 de Septiembre de 2024

Ing. Oscar Miguel Cumbicus Pineda, Mg.Sc.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Integración Curricular denominando: **Ajuste del modelo Gemma para la descripción de tablas de datos en formato Markdown**, previo a la obtención del título de **Ingeniero en Ciencias de la Computación**, de la autoría del estudiante **Patricio Oswaldo Paredes Chamba**, con **cédula de identidad** Nro. **1150011805**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo a presentación del mismo para su respectiva sustentación y defensa.

Firmado electrónicamente por:
OSCAR MIGUEL
CUMBICUS PINEDA

Ing. Oscar Miguel Cumbicus Pineda, Mg.Sc.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Autoría

Yo, **Patricio Oswaldo Paredes Chamba**, declaro ser autor del presente Trabajo de Integración Curricular y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Integración Curricular, en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma: 

Cédula de identidad: 1150011805

Fecha: 24 de septiembre de 2024

Correo electrónico: patricio.paredes@unl.edu.ec

Teléfono: (+593) 986665365

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Integración Curricular

Yo, **Patricio Oswaldo Paredes Chamba**, declaro ser el autor del Trabajo de Integración Curricular denominado: **Ajuste del modelo Gemma para la descripción de tablas de datos en formato Markdown**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Integración Curricular que realice un tercero.

Para constancia de esta autorización, suscribo, en la ciudad de Loja, a los veinticuatro días del mes de septiembre del dos mil veinticuatro.

Firma: Firmado electrónicamente por: PATRICIO OSWALDO PAREDES CHAMBA

Autor: Patricio Oswaldo Paredes Chamba

Cédula de identidad: 1150011805

Dirección: Loja (El Valle, Belén)

Correo electrónico: patricio.paredes@unl.edu.ec

Teléfono: (+593) 986665365

DATOS COMPLEMENTARIOS:

Director del Trabajo de Integración Curricular: Ing. Oscar Miguel Cumbicus Pineda Mg.Sc.

Dedicatoria

A mis padres, abuelos y hermanos por haberme brindado su apoyo incondicional en cada etapa de mi vida, siendo además una fuente de inspiración.

Patricio Oswaldo Paredes Chamba.

Agradecimiento

Agradezco a mis padres, hermanos, amigos y compañeros por todo el apoyo recibido, no solo durante el desarrollo de este Trabajo de Integración Curricular, sino en todo el proceso educativo dentro de la carrera. Asimismo, extiendo un sincero agradecimiento a mi director, Ing. Oscar Cumbicus, quien me brindó ayuda con sus conocimientos y su predisposición para la culminación exitosa del Trabajo de Integración Curricular.

Patricio Oswaldo Paredes Chamba

Índice de contenidos:

Índice de tablas:

Índice de figuras:

Índice de anexos:

1. Título

Ajuste del modelo Gemma para la descripción de tablas de datos en formato Markdown.

Tuning the Gemma model for describing data tables in Markdown format.

2. Resumen

El uso de LLM (Large Language Models) como Gemma se ha extendido para diversas tareas. Una poco explorada es el procesamiento de estructuras como las tablas de datos, ya que estas cuentan con una gran diversidad, tanto en su temática, presentación y extensión de contenido lo cual dificulta su procesamiento. Existe antecedentes de modelos para distintas tareas. Teniendo como entrada, tablas de datos en distintos formatos, una de estas tareas es la descripción de todo el contenido de las tablas de datos. El objetivo de este Trabajo de Integración Curricular se planteó ajustar el modelo Gemma para la descripción de tablas de datos de un máximo de 3 filas por 3 columnas las cuales estuvieron en un formato Markdown. La metodología utilizada se basó en CRISP-ML(Q), haciendo uso de las fases comprensión de datos, ingeniería de datos, ingeniería de modelos y evaluación del modelo. En la primera fase se creó datos sintéticos de tablas de datos en Markdown y la descripción de la información contenida en estas. En la segunda fase se seleccionó los datos válidos, se realizó una revisión y corrección manual de datos válidos de manera parcial. La tercera fase permitió el ajuste del modelo Gemma y la fase final evaluó los experimentos realizados, midiendo la métrica BLEU y realizando una evaluación humana. El modelo obtenido alcanzó en el conjunto de test una puntuación BLEU de 74,009, al realizar una revisión humana se obtuvo una puntuación de 4,625 en la escala Likert, concluyendo que el modelo ajustado logra generar descripciones de buena calidad de tablas de datos en formato Markdown y al realizar un test A/B comparando BLEU con la revisión humana se constató que las descripciones generadas con una alta puntuación BLEU también contaron con un puntaje alto en la evaluación humana.

Palabras clave: *LLM, Gemma, PEFT, Datos sintéticos, Markdown.*

Abstract

The use of LLM (Large Language Models) like Gemma has been extended for various tasks. One that has been little explored is the processing of structures such as data tables since they have great diversity, both in their subject matter, presentation and content extension, which makes their processing difficult. There is a history of models for different tasks. Having data tables in different formats as input, one of these tasks is the description of all the content of the data tables. The objective of this Curricular Integration Work was to adjust the Gemma model for the description of data tables of a maximum of 3 rows by 3 columns which were in a Markdown format. The methodology used was based on CRISP-ML(Q), making use of the data understanding, data engineering, model engineering and model evaluation phases. In the first phase, synthetic data was created from data tables in Markdown and the description of the information contained in them. In the second phase, valid data was selected, a manual review and correction of partially valid data was carried out. The third phase allowed the adjustment of the Gemma model and the final phase evaluated the experiments carried out, measuring the BLEU metric and performing a human evaluation. The model obtained reached a BLEU score of 74.009 in the test set. When performing a human review, a score of 4.625 was obtained on the Likert scale, concluding that the adjusted model manages to generate good quality descriptions of data tables in Markdown format and when performing an A/B test comparing BLEU with human review, it was found that the descriptions generated with a high BLEU score also had a high score in human evaluation.

Keywords: *LLM, Gemma, PEFT, Synthetic data, Markdown.*

3. Introducción

En la actualidad el desarrollo y uso de modelos deep learning que utilizan arquitecturas basadas en Transformers como son los LLM (Large Language Models) y han experimentado un crecimiento acelerado en su investigación como para su desarrollo, han llegado a plantearse en campos en que existe una escasez de profesionales disponibles [1]. Dentro de la Universidad Nacional de Loja existe el proyecto de vinculación denominado “Inclusión lectora de los estudiantes con discapacidad visual de la Universidad Nacional de Loja mediante innovación tecnológica.”[2], con el fin de su desarrollo, aparecieron muchos más, dentro de los cuales está el “lograr la descripción de tablas de datos”. Explorando una solución para este problema con su encargado y experto en inteligencia artificial, el Ing. Oscar Cumbicus, se determinó la necesidad de un modelo de inteligencia artificial capaz de aceptar una tabla de datos en un formato específico y generar una descripción de la misma (véase **Anexo 1**).

El tratamiento de datos tabulares es un reto dentro del deep learning, al existir diferentes enfoques y pruebas realizadas con datasets con propósitos propios, es complejo determinar una comparación efectiva entre los enfoques propuestos para esta problemática, aunque con el modelo con arquitecturas Transformers se demostró resultados relevantes [3]. Al tratarse de modelos LLM de procesamiento de lenguaje natural, entre los contemporáneos un utilizado en el desarrollo de datasets es: Markdwon, el cual resulta de gran utilidad debido a sus características como flexibilidad y bajo nivel de complejidad [4]. De entre los modelos LLM existentes el modelo Gemma resalta en compresión del lenguaje y rendimiento de generación siendo superior en términos de Razonamiento en comparación con otros modelos de similares tamaños como: LlaMA2 y Mistral [5]. Los modelos LLM como lo pueden ser los de la familia Gemma, enfrentan desafíos al tratar de realizar tareas específicas como generar descripciones de tablas de datos, esto es debido a la gran diversidad que existe en la presentación de las mismas, pueden tener variaciones de: tamaño, celdas combinadas, títulos en la parte superior e izquierda, entre otros. Requiriendo datasets con un tamaño considerable, esto según los tipos de tablas que se desee. Dentro de los modelos actuales, los LLM de la familia Gemma han sido entrenados con una gran cantidad de datos de distintas fuentes, en especial recursos en el idioma inglés [1]. Sin embargo, no están ajustados para la descripción de tablas de datos. Dado un ajuste con el fin de una tarea específica. El objetivo es obtener descripciones de tablas de datos de un tamaño máximo de tres filas, tres columnas y en formato Markdown de alta calidad medido con la métrica BLEU. De un interesante modelo Gemma ajustado, surgió la siguiente pregunta de investigación: “¿Qué puntuación BLEU se podría obtener al ajustar el modelo Gemma para la tarea específica de generar descripciones de tablas de datos, las cuales estarán en un formato Markdown, con un tamaño determinado

máximo de 3*3, sin tomar en cuenta signos poco usados como los presentes en fórmulas matemáticas?". El objetivo general de este TIC es: "Obtener la puntuación BLEU al ajustar el modelo Gemma para la tarea específica de generar descripciones de tablas de datos, las cuales estarán en un formato Markdown, con un tamaño determinado máximo de 3*3, sin tomar en cuenta signos poco usados como los presentes en fórmulas matemáticas", se propone cumplir basado en los objetivos específicos propuestos: "Ajustar el modelo Gemma para que sea capaz de describir tablas de datos, estando estas en un formato Markdown, un tamaño determinado con un máximo de 3*3, excluyendo signos de uso mínimo como los presentes en fórmulas matemáticas, mediante una metodología basada en CRISP-ML(Q)." y "Evaluar mediante la métrica BLEU los distintos modelos obtenidos al usar una variación de hiperparámetros y así obtener la mejor opción que logre describir tablas de datos en formato Markdown. ".

Teniendo como limitaciones el breve periodo de tiempo académico y la disponibilidad del uso de recursos computacionales (GPU), fue otro de los retos debido a la complejidad en cada experimento de ajuste del modelo, se trabajó con "Google Colab de pago", fue corto el lapso total de disponibilidad de recursos de hardware, aun en cuenta de pago y la poca información de ajustes SFT en el modelo Gemma.

En los trabajos relacionados tratan esta problemática como "table to text" esta engloba el poder de generar la descripción de celdas específicas de una tabla o generarla en base a un contexto, entre otros. En cuanto al dataset utilizado, cada uno tiene un formato específico para la tabla de datos, además que los datos se encuentran en el idioma inglés, en cuanto al ajuste se hace uso de modelos de menor tamaño que realizan un ajuste total, requiriendo la disponibilidad de clústeres de GPUs. El trabajo está enfocado a generar una descripción de toda la información contenida en la tabla; la creación de un dataset de tablas de datos en un formato específico (Markdown) con sus descripciones de referencia en español, con temáticas y un límite de dimensiones claro; el uso del modelo Gemma 2B, aparte de realizar un ajuste mediante un método que permite el ajuste en ambientes con pocos recursos computacionales como lo es PEFT con LoRa.

El presente TIC tiene varios beneficios. Como primer aporte está el dataset creado tanto en su primera versión como la segunda con mayor cantidad de datos y temáticas, contiene tablas de datos de hasta 3 filas por 3 columnas, en el idioma español con sus respectivas descripciones. Se demostró modelos LLM como Gemma pueden procesar información en el formato Markdown. Lograr medir la salida del modelo Gemma con la métrica BLEU y determinar su utilidad, en comparación con la evaluación humana.

4. Marco Teórico

4.1. Antecedentes.

En el Ecuador existen 55478 personas con discapacidad visual [6], las cuales tienen necesidades de ocupar sistemas adaptados para inclusión, varios de los sistemas son de pago y son para usos genéricos. Dentro de la Universidad Nacional de Loja, en el año 2022, se matricularon 26 estudiantes con discapacidad visual [7], los cuales han tenido diversas dificultades entre las cuales se destaca, la falta de inclusión en la lectura de textos académicos, por lo que la carrera de Computación se encuentra desarrollando un proyecto que vaya enfocado a mejorar la accesibilidad de la lectura para estas personas, el nombre del proyecto es: "Inclusión lectora de los estudiantes con discapacidad visual de la Universidad Nacional de Loja mediante innovación tecnológica."[2]. Dentro de este proyecto se han presentado diversos desafíos, los cuales se ha concluido que el uso de modelos de inteligencia artificial sería una buena herramienta para resolverlos, de entre problemas detectados uno fue abordado en el presente TIC, la dificultad que tienen los modelos para la generación de descripciones de tablas de datos (véase **Anexo 1**).

El uso de modelos de inteligencia artificial en la descripción de datos estructurados como son las tablas, tiene una complejidad alta, pues en trabajos previos que abarcan una problemática similar [8][9][10][11][12][13] los modelos no han logrado un rendimiento óptimo en la solución del problema, los modelos actuales que tendrían el potencial de resolver esta problemática, cuentan con varios desafíos, pues estos modelos no cumplen con el enfoque requerido para la problemática, siendo los más cercanos a los mismos aquellos modelos ajustados con data sets como ToTTo [14], que cuentan con una pequeña descripción de las tablas, pero como resumen de la misma o respuestas acorde a un contexto, además que estas descripciones no abarcan toda la información de la tabla de datos, u estructuras tabulares semejantes. Otros desafíos presentes es que los modelos que han sido entrenados para este propósito no se encuentran empaquetados para el consumo del modelo, proporcionando el código ocupado con diversos desafíos, entre ellos: data sets utilizados en mal estado, librerías desactualizadas pues varios de los mismos se encuentran en TensorFlow versión 1.0, ejemplo de algunos de esos problemas es el repositorio encontrado[1] de [11] un trabajo relacionado al TIC. Este tipo de problemas causan varios conflictos de compatibilidad entre librerías, esto tomando en cuenta que la entrada del modelo ya son datos con una estructura determinada, que varía según el artículo tratado como en los trabajos relacionados (véase la **Sección 4.3**). Por lo que el ajuste del modelo Gemma es una opción para la generación de descripciones de tablas de entrada en un formato Markdown, con ayuda de este formato se lograría limitar el tipo de tablas que es posible aceptar, ya que este no permite tablas con filas o columnas

[1] Repositorio en Git de Few-Shot-Table-to-Text-Generation

combinadas. En el caso del modelo Gemma como tal no hay antecedentes limitados en lo que refiere a tareas específicas como lo es la generación de descripciones de tablas de datos en formato Markdown, pues se trata de un modelo recientemente liberado por Google que se ha centrado más en realizar comparativas con otros modelos, en cuestiones como el número de tokens, su capacidad en general [5].

4.2. Fundamentación Teórica

4.2.1. Inteligencia Artificial

La IA (inteligencia artificial) ha sido un campo de gran crecimiento en la actualidad, esto tiene que ver en gran parte por el enorme potencial que refleja, además de los variados sectores donde está podría llegar a aplicarse, por varias razones es complicado definir inteligencia razón por la cual han existido diferentes autores que tienen su definición de inteligencia artificial, dentro de [15] se define la IA como una serie de algoritmos los cuales son capaces de realizar procesos que simularían los que consideramos necesarios para la inteligencia humana, cumpliendo características presentes en la misma como: la capacidad de aprender la cual requiere que se mantenga información y las reglas para el uso de la misma; dentro del campo de la lógica, que tenga la capacidad de razonar, esto es tener la capacidad de llegar a conclusiones haciendo uso de las reglas impuestas en su aprendizaje; la capacidad de realizar autocorrecciones lo que supone una capacidad de evaluación en las conclusiones que obtiene.

Uno de los pioneros en esta temática fue el matemático inglés Alan Turing el cual aportó con una de las pruebas icónicas en esta disciplina la cual es el test de turing, el cual establece que para demostrar la inteligencia en agentes no biológicos era necesario que este tenga patrones al actuar que se pudieran denominan inteligentes, sobre todo en el intercambio de palabras, en una conversación, siendo el objetivo de la IA, camuflarse y hacer creer al otro, que está teniendo una conversación con una persona, esta visión tuvo sus consecuencias, pues si bien la IA se construyó encima del conocimiento de varias áreas, debido al test de turing, las investigaciones se han centrado más en la lingüística [16].

La IA ya se encuentra integrada en la sociedad, esto es más notorio en la vida diaria, con aplicaciones como Spotify o Google Maps, que hacen uso de esta tecnología para diversos fines, aun así, se considera que la IA está en sus fases iniciales, por lo que su potencial aún no ha sido explotado por completo, en [17] se menciona que esta tecnología permitirá un crecimiento en diversas áreas entre ellas el económico, pues se espera que para 2030 un 70% de empresas tenga integrada alguna solución con esta tecnología, entre los campos de esta se encuentra el aprendizaje automático y el aprendizaje profundo .

4.2.2. Procesamiento de lenguaje natural

Es una disciplina que permite el estudio de una forma de interacción entre las computadoras y los seres humanos mediante el lenguaje natural, para lo cual combina

diferentes campos como la matemática, informática y lingüística [18], con todos los retos que esto conlleva, pues el lenguaje que las personas utilizamos diariamente posee una complejidad considerable, donde hay retos como el vocabulario, el contexto, entre otros.

Según [19] las tareas en las cuales se implementa el PLN se ven en gran medida influenciadas por las industrias, las cuales según su temática pueden hacer un uso específico del mismo, industrias como las finanzas, marketing entre otras, han encontrado una gran oportunidad en distintas aplicaciones del PLN, de entre los usos más conocidos tenemos, el análisis de sentimientos, chatbots, clasificación de textos, etc.

Este tiene el potenciar de analizar, modificar y asimilar, el lenguaje natural lo cual tiene varias implicaciones entre ellas la capacidad de llegar a comprender un contexto, identificar las partes más significativas de este y extraerlo, entender las reglas dentro de este lenguaje logrando la capacidad de identificar cuando no se cumplen las mismas [18], [20] . Contiene dos áreas las cuales son: la Comprensión del Lenguaje Natural (CLN) y la Generación del Lenguaje Natural (GLN).[18][21].

4.2.3. Modelos LLM

Los Large Language Models (LLMs), en español Grandes modelos de lenguaje, como tal no cuentan con una definición establecida a través de un consenso, pero [22] nos dice que los LLM son aquellos modelos que han pasado por un proceso de pre entrenamiento con una cantidad muy elevada de datos, estos sin haber tenido un proceso de ajuste por lo que no están ajustados para llevar a cabo una tarea específica. Estos según las investigaciones avanzas han ido demostrando ser capaces de realizar tareas tanto de la cotidianidad como relacionadas al PLN.

Estos surgieron como resultado de la investigación de los modelos PLM (Pre-trained language models) [23] o Modelos de lenguaje previamente entrenados en español, pues al realizar entrenamientos de estos modelos aumentando su tamaño o el número de datos normalmente resulta en una mejora de la capacidad de los mismos, estos experimentos como por ejemplo llegaron a tal punto que los modelos desarrollaron lo que se denominó “habilidades emergentes” que es la capacidad de realizar tareas complejas, tareas que modelos PLM de tamaño inferior no podrían realizar.

4.2.3.1. Características de un LLM con capacidades relevantes [24].

- El lenguaje natural posee varios desafíos para lograr una correcta interpretación del mismo, entre ellos el contexto tiene la capacidad de cambiar por completo el significado en la comunicación, por lo que un LLM tiene que tener la capacidad de comprender de manera profunda el contexto en el lenguaje natural.

- Una capacidad de generación de texto que no sea mecánico es decir que llegue al punto similar al que no sea distinguible si este texto fue escrito por una persona o por el modelo LLM.
- Un modelo LLM debe tener la capacidad de utilizar la comprensión del contexto dado, para generar respuestas que vayan en concordancia a dicho contexto, sobre todo en escenarios donde se requiere que el conocimiento sea especializado.
- Estos modelos deben ser capaces de seguir las instrucciones dadas, de la manera más precisa que sea posible, esto es esencial para obtener resultados que cumplan con las expectativas deseadas.

4.2.4. Modelo Gemma.

Desde Google DeepMind se realizó el lanzamiento de Gemma, el cual se suma a la competencia en cuento a varias tareas, tanto de razonamiento como otros usos, los cuales se están llevando a cabo por varios modelos los cuales son la propuesta de cada empresa, dentro de este escenario aparece la familia de modelos ligeros Gemma el cual es un LLM, que al ser publicado ha permitido que la comunidad lo pueda ajustar dependiendo de sus necesidades, si hablamos de una familia es porque dependiendo a los recursos físicos, es posible elegir una versión del mismo con mejor rendimiento, pues cuenta con un mayor número de parámetros siendo contando con 7 mil millones de parámetros, u otra versión que pueda ser usada en entornos con bajos recursos, en este caso tenemos la versión de Gemma que cuenta con 2 mil millones de parámetros, el cual se muestra mucho más accesible ante recursos limitados [5]. La **Tabla 1** muestra las diferencias presentes entre las versiones disponibles que presenta la familia de modelos Gemma, en esta se presenta que la versión 2B aun si tiene parámetros más limitados que la versión 7B, es la opción tomada para la ejecución de este TIC, debido a que esta permite su ajuste con menor cantidad de recursos computacionales, permitiendo su ajuste en entornos como Google Colab en su versión no paga, aunque en el proceso de la metodología realizada (véase **Sección 5.2**) se puede apreciar que esto dependerá de la cantidad de datos en su entrenamiento y validación pues si estos tienen una cantidad considerable, puede que aun usando la versión más básica de modelo se requiera el uso de más capacidad computacional que la ofrecida en Google Colab.

Tabla 1. Comparación de parámetros entre Gemma 2B y 7B del modelo [5]

Parámetros	**2B**	**7B**
d_model	2048	3072
Layers	18	28
Feedforward hidden dims	32768	49152
Num heads	8	16
Num KV heads	1	16

Parámetros	2B	7B
Head size	256	256
Vocab size	256128	256128

d_model: tamaño de los vectores de entrada y salida del modelo
Layers: número de capas que componen el modelo
Feedforward hidden dims: dimensión de las capas ocultas de la red neuronal feedforward.
Num heads: número de cabezales de atención que utiliza el modelo.
Num KV heads: número de cabezales de atención utilizados para calcular las llaves (keys) y los valores (values) en el mecanismo de atención.
Head size: tamaño de cada cabezal de atención.
Vocab size: tamaño del vocabulario del modelo.

4.2.4.1. Arquitectura del modelo.

La arquitectura del modelo Gemma se basa en decodificador de Transformers, esta cuenta con una longitud en su contexto de 8192 tokens, además de la arquitectura este modelo cuenta con mejoras en la misma [5]:

4.2.4.1.1. Transformer decoder

La diferencia con el modelo encoder-decoder se centra en que este no posee un encoder, contando únicamente con la parte del decoder encargado de generar texto a través del mecanismo de autoregresión [25], además de contar con la capacidad de detectar y reconocer el texto [26]. Inicialmente, este se compuso de una pila de N=6 capas idénticas, posee dos subcapas en cada capa de decodificador, finalmente inserta una tercera subcapa que realiza operaciones de multi-head, siguiendo la normalización de capas, este emplea conexiones residuales alrededor de las subcapas, también se debe modificar la self-attention en la pila de decodificadores para que atienda a posiciones posteriores, este enmascaramiento garantiza las predicciones para una posición que puede depender solo de las salidas conocidas en posiciones menores que la posición establecida [27]. La información sobre este tipo de modelos es posible encontrarlos en distintas fuentes como la web[2], quien para su representación toma en cuenta información obtenida de [27][5], donde se explica la arquitectura Transformers y su funcionamiento. La **Figura 1** refleja una capa de la arquitectura del Transformer decoder esto es debido a que el decoder está compuesto por un número determinado de capas, las cuales tienen la misma estructura, estas se conectan entre sí para obtener una salida como puede ser texto.

[2] Decoder-Only Transformers

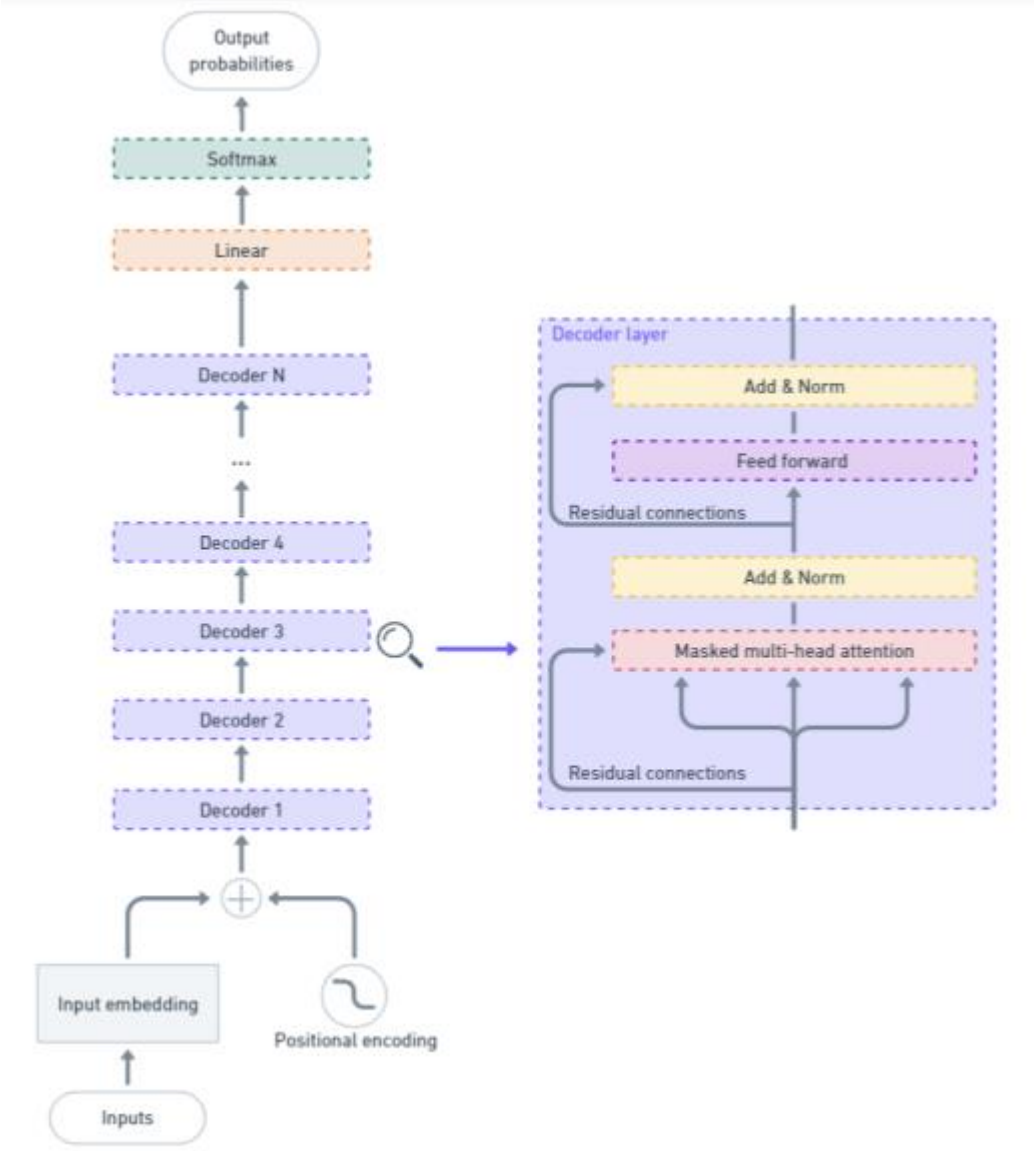


Figura 1. Arquitectura basada en el Transformer decoder [25].

4.2.4.1.2. Multi-Query Attention.

En particular, el modelo 7B utiliza atención de múltiples cabezales, mientras que los puntos de control 2B utilizan atención multiconsulta (con num_kv_heads = 1), basada en ablaciones que demostró que la atención multiconsulta funciona bien a pequeña escala [28].

4.2.4.1.3. Incrustaciones de RoPE.

En vez de usando incrustaciones posicionales absolutas, usamos incrustaciones posicionales rotativas en cada capa; nosotros también compartir incorporaciones en nuestras entradas y salidas para reducir el tamaño del modelo [29]**.**

4.2.4.1.4. Activaciones GeGLU.

La no linealidad estándar de ReLU se reemplaza por la versión aproximada de la función de activación GeGLU [30].

4.2.4.1.5. RMSNorma.

Se normaliza la entrada de cada subcapa transformadora, la capa de atención y la capa feedforward, con RMSNorm para estabilizar el entrenamiento [31].

4.2.5. PEFT

Al realizar el entrenamiento con capacidades industriales de modelos estos pasan a ser denominados modelos preentrenados, el resultado es modelos de gran tamaño por lo cual tendrían muy complejo y en algunos casos inconveniente su ajuste, para tareas específicas, pues los requerimientos computacionales necesarios serían altamente prohibitivos, en casos como estos, no resulta sorpresivo el nacimiento de nuevos enfoques para el ajuste de estos modelos, uno de estos es el método PEFT (ajuste fino eficiente de parámetros) el cual propone una forma de ahorro en capacidad computacional, proponiendo una forma de ajustar parcialmente el modelo, esto lo logra actualizando los pesos de una cantidad reducida de los parámetros del modelo [32].

En modelos LLM especializados en tareas PLN es especialmente importante, debido a que estos cuentan con la característica zero-shot learning, esta explica que cuenta con un nivel elevado de generalización, el cual le permite responder en escenarios nuevos los cuales no formaron parte en su proceso de entrenamiento, pero aun con esto para que estos modelos respondan en escenarios particulares, es necesario realizar un ajuste con datasets que cuente con las actividades especifica que se requiera. Para esto, métodos como PEFT permiten un ajuste de forma específica a los parámetros de los modelos LLM, además este método no se limita solo a capos de PLN, pues se puede explicar su uso en modelos de visión por computadora entre otras tareas [33]. La **Figura 2** muestra algunos ejemplos de tipos de algoritmos PEFT, los cuales forman parte de la taxonomía de los métodos que PEFT engloba, los cuales son usados para el ajuste de modelos LLM.

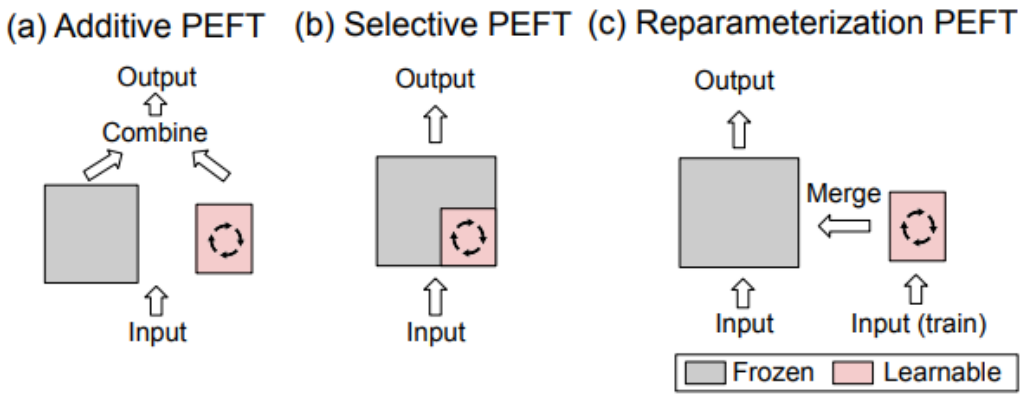


Figura 2. Tipos de algoritmos PEFT [33].

En la actualidad debido al auge de los modelos LLM, las investigaciones de estos han incrementado, la forma en que son ajustados es full fine-tuning (FT) que es un afinamiento

completo de los parámetros. Debido al número tan elevado de parámetros con los que cuentan los LLM, en varios de los casos un full fine-tuning resulta en inviable, esto debido a la capacidad computacional requerida, por lo que, métodos como PEFT han empezado a ser vistos como una alternativa viable, pues añaden una pequeña cantidad de parámetros y los ajustan para la aplicación específica en que serán entrenados, con lo cual logran disminuir el tiempo y recursos computacionales necesarios para el entrenamiento, además de lograr resultados que pueden estar a la altura de los conseguidos con un full fine-tuning o FT [33].

Realizar un ajuste completo del modelo, conlleva varias desventajas entre ellos, el costo necesario tiene un gran aumento, además de requerir una capacidad alta en almacenamiento, pues es necesario guardar una copia de seguridad para el modelo resultante en cada ajuste por cada tarea específica, el enfoque tomado por PEFT permite el ajuste parcial de estos parámetros, logrando disminuir el coste necesario y consiguiendo resultados competitivos a comparación de un full fine-tuning [34].

4.2.5.1. LoRa.

Low-Rank adaptation (LoRA) es un método el cual ofrece una eficiencia de entrenamiento en modelos grandes, aunque el impacto en el rendimiento del modelo es limitado, este junto con Mixture-ofExperts (MoE) son métodos que su enfoque es mejorar el rendimiento del método PEFT, investigaciones para mejorar el rendimiento al usar LoRa continúan realizándose [35]. En los LLM las grandes capacidades que han demostrado han permitido que estos puedan realizar transfer learning en una gran variedad de dominios y para diferentes tareas, debido a que los modelos LLM tienen una gran cantidad de parámetros el ajuste de todos los parámetros requiere capacidades computacionales muy grandes, razón por la cual la comunidad de investigación se ha centrado en métodos PEFT como LoRa [35].

Es funcionamiento de LoRa trata de que esta congela los pesos del modelo una vez entrenado e inyecta matrices de descomposición de rangos entrenable en cada una de las capas de la arquitectura Transformers, lo cual permite que la cantidad de parámetros a entrenar para una tarea específica se reduzca, este es capaz de reducir según el modelo tanto el número de parámetros a entrenar como los requerimientos en memoria de GPU, aun con un número de parámetros limitados se logra un mejor rendimiento [36]. La **Figura 3** presenta el enfoque de lo LoRa para el entrenamiento del modelo, este lo logra entrenar algunas capas densas en una red neuronal indirectamente optimizando las matrices de descomposición de rangos del cambio de las capas densas durante la adaptación mientas se mantienen congelados los pesos previamente entrenados, logrando así que LoRa sea eficiente tanto en requerimientos de almacenamiento como de recursos computacionales, en la figura con LoRa solo se entrenan A y B.

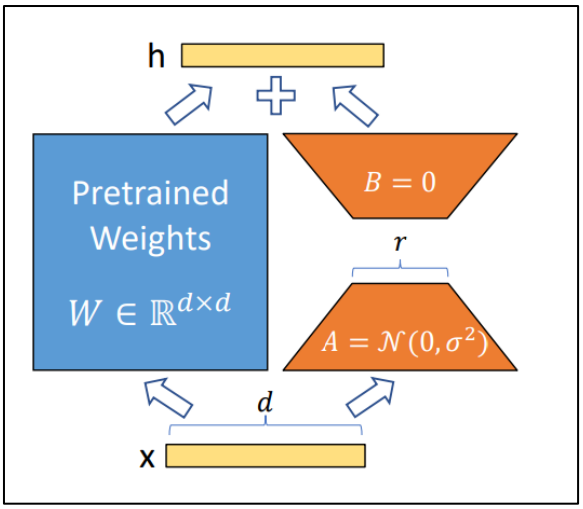


Figura 3. Enfoque de entrenamiento de LoRa [36].

PEFT actualmente es el método más popular para ajustar LLM conjuntamente con LoRa el cual obtiene un gran funcionamiento en escenarios con recursos computacionales medios o bajos, donde LoRa demuestra resultados de alta calidad, hay que tomar en cuenta que es posible optimizar LoRa con granularidades específicas a nivel de módulos o capas logrando una adaptabilidad sin disminuir peligrosamente el rendimiento [37].

4.2.6. Markdown.

Su presentación ocurrió en el año 2004 por parte de su creador John Gruber, este es un lenguaje de marcado considerado ligero, esto es así por su objetivo y características, pues este tiene la capacidad de agregar elementos de formato a texto que no cuenta con un formato definido, la sintaxis de este es sencilla, permitiendo incluso que la información sea legible aun si esta no se renderiza, su objetivo hace énfasis en tener una alta legibilidad al punto de poder presentar un texto con este formato y que no se tenga la percepción que el texto fue marcado con etiquetas o puesto en un formato [38].

Markdown si bien posee una estructura de etiquetado sencilla, ha sido utilizado en varios escenarios y con distintos fines, un ejemplo de esto es almacenar información, que requería contar con una estructura, esto se puede observar en artículos como [39] [40], en ellos se destaca la gran popularidad de este formato, pues en distintos escenarios como la web, este ha sido ampliamente aceptado, pues su facilidad de uso se refleja al momento de su escritura y lectura. El formato Markdown al ser tan flexible, existen herramientas las cuales pueden no llegar a identificar determinadas etiquetas, en Google Colab por ejemplo en las secciones de texto se usa este formato, dando tanto una vista de la escritura con las etiquetas respectivas en la parte izquierda y su respectiva renderización en la parte derecha, como se muestra en la **Figura 4** esta es de autoría propia**.**

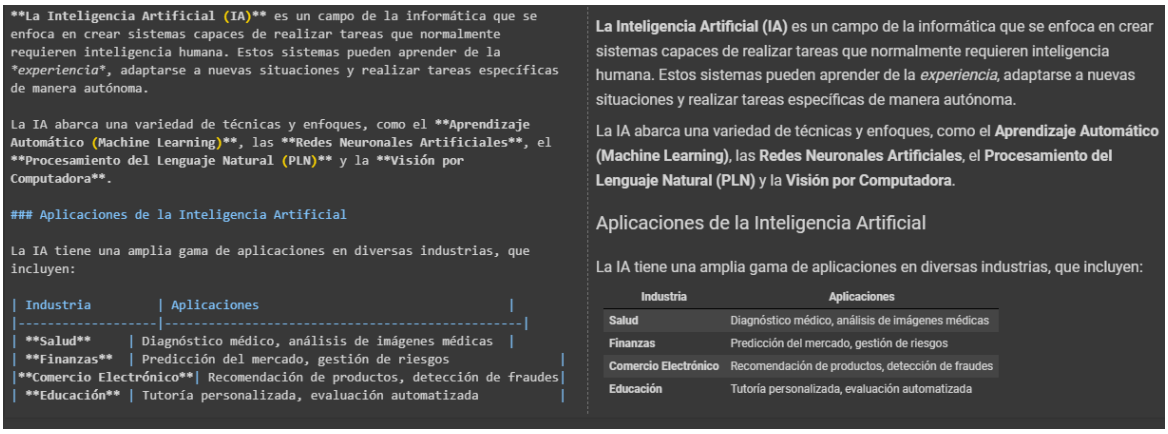


Figura 4. Ejemplo del uso de formato Markdown, con y sin renderización.

4.2.6.1. Tablas.

Las etiquetas que hace posible la creación de una tabla en el formato Markdown, poseen flexibilidad, además existe una variedad tanto de etiquetas como de formas en que estas se organizan en la estructura para la creación de la tabla, es debido a esto que puede existir el caso que determinadas herramientas no reconozcan una sintaxis determinada [41]. Estas tablas se crean con notación en línea, esto es con una sintaxis sencilla y fácil de escribir para estas estructuras de datos [42]. El uso de la sintaxis de Markdown en las presentes estructuras está tan extendido que incluso es usada dentro de herramientas de análisis de datos, para la exportación de tablas de datos en este formato [43].

En la **Figura 5** se muestra la representación de una tabla con su sintaxis correcta para una celda de texto dentro de un cuaderno de Google Colab, además de ser la sintaxis que se utilizó en el desarrollo del proyecto, al almacenar las tablas con la sintaxis Markdown dentro del dataset, esto tomando en cuenta que la alineación en esta sintaxis no es importante como se muestra en la **Figura 6**.

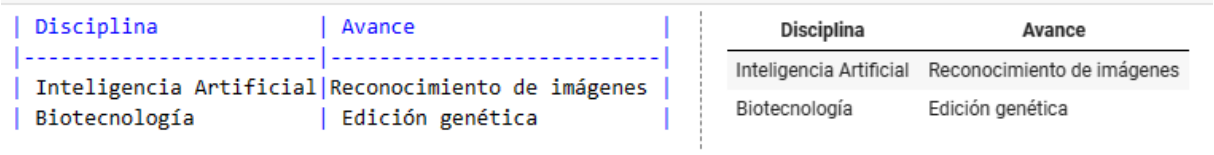


Figura 5. Tabla en Markdown dentro de una celda de texto de Google Colab.

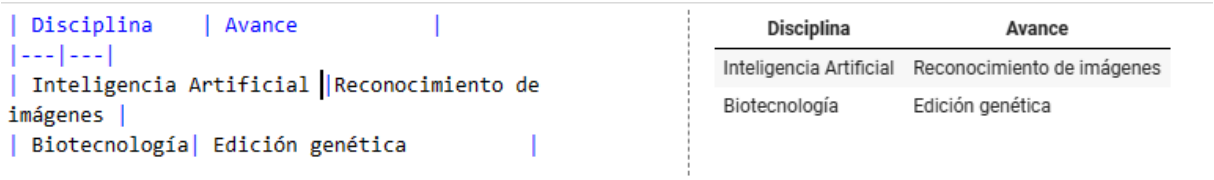


Figura 6. Tabla en Markdown sin una alineación uniforme.

4.2.7. Datos sintéticos.

En el entorno del desarrollo de la IA existen muchas limitaciones, una de las principales es la escasez de información, este problema abarca todos los campos de investigación, es por eso que se han desarrollado procesos para suplir esta necesidad, al aparecer modelos generativos de inteligencia artificial, surgió también la opción de utilizar su capacidad para suplir esta falta de información, esto resulto muy conveniente para diversas aplicaciones industriales como de investigación, además de permitir suplir la cantidad escasa de datos existente, estos permitieron su uso en otros modelos tanto de machine learning como de deep learning, tanto para su creación como ajuste permitiendo mejorar su eficiencia [44].

Los datos sintéticos para el entrenamiento de modelos machine learning y deep learning, permite realizar distintos experimentos disminuyendo la dificultad de encontrar datos, su uso se ha extendido en varios campos de la IA, que van desde el procesamiento de lenguaje natural, hasta la visión por computadora [45], abarcando varias aplicaciones de la IA a su alrededor. La utilidad de los datos sintéticos es alta y su uso en diversos campos va en aumento, esto debido a la dificultad de acceso a los datos necesarios, pues estos deben de ser válidos para la tarea, la cantidad disponible de datos siempre es un reto a superar, ya sea por la inexistencia de los mismos, como con problemas relacionados a la privacidad, seguridad o regulaciones como pueden ser los derechos de autor, estos datos sintéticos nos pueden permitir hacer uso de datos incluso, generándolos con el enfoque específico que requiere la tarea para la cual van a ser utilizados [46].

La **Figura 7** muestra las diferentes aplicaciones de los datos sintéticos, varios campos que su uso conlleva beneficios, tanto en temas de aumento de la cantidad de datos, contribuir en temas de privacidad y seguridad de datos, por lo que se contempla su uso en aplicaciones como: educación (Education), visión (Vision), Cuidado de la salud (Healthcare), voz (Voice), procesamiento de lenguaje natural (Natural Language processing), negocio (Business).

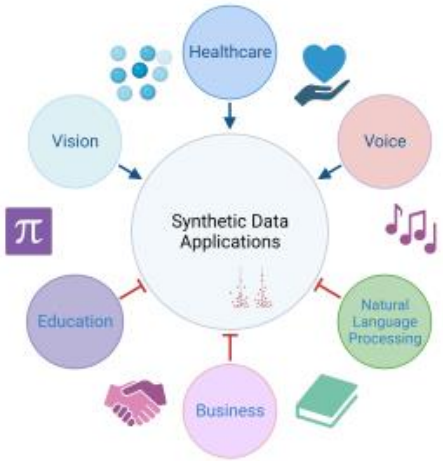


Figura 7. Aplicaciones para los datos sintéticos de [46] .

4.2.8. CRISP-ML(Q).

Esta es una metodología que tiene un enfoque en las aplicaciones de aprendizaje automático, con todos los desafíos que estas presentan, esta es la respuesta de la comunidad compuesta por organizaciones y desarrolladores, los cuales consientes de los riesgos presentes en el desarrollo de este tipo de aplicaciones, han tratado de normalizar un proceso para mitigar los riegos y poder lograr una aplicación de calidad, este cuenta con 6 fases que van desde el entendimiento del negocio y los datos esto es crucial para lograr determinar correctamente los requerimientos que deberá cumplir para satisfacer correctamente la demanda a cubrir ya sea en una organización, en el mercado, etc. Las fases de esta metodología son: Comprensión de datos y negocios, Ingeniería de datos, Ingeniería de modelos de aprendizaje automático, Evaluación del modelo de aprendizaje automático, Implementación del modelo, Monitoreo y Mantenimiento de Modelos [47]. La **Figura 8** muestra las fases de CRISP-ML(Q) en esta se puede apreciar que tiene varias fases que pueden llegar a iterarse, esto para lograr la calidad necesaria, en ella se separa tres grupos importantes los cuales pueden tratar: El negocio y el entendimiento de su información, el desarrollo del modelo y el modelado.

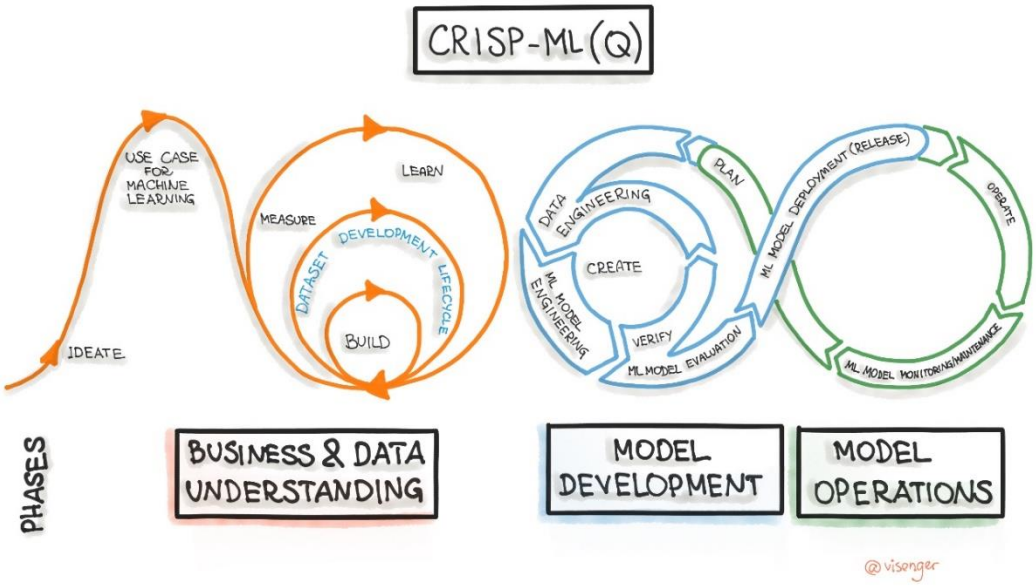


Figura 8. Fases del ciclo de vida la metodología CRISP-ML(Q) [48].

Este en comparación de otras metodologías enfocadas a minería de datos como es la CRISP-DM, cuenta con fases específicas para tareas de aprendizaje automático y deep learning, hay que tomar en cuenta que esta metodología, separa de manera correcta fases que tienen que ver con los datos que van en relación a el negocio u aplicación específica esperada, entre las fases con un enfoque en los datos tenemos a: La comprensión de datos y negocios, Preparación de datos. Con un enfoque en el modelo están las fases de: Modelado, Evaluación, Despliegue, Monitoreo y Mantenimiento [49].

4.2.9. Métricas usadas en la evaluación del modelo.

4.2.9.1. BLEU.

La métrica BLEU fue propuesto por Papineni en 2002 [13] este está contemplado para el uso de medir traducciones automáticas, generación de texto. Esta métrica como tal requiere tener a disposición, mínimo una traducción de referencia, por cada evaluación, esta métrica evalúa los n-gramas y los compara entre el texto obtenido por el modelo con la referencia, se usa dentro del procesamiento del lenguaje natural, dentro de su funcionamiento se encuentran diversos procesos que lleva a cabo, para lograr obtener una medida, entre ellos penalizaciones en cuanto por ejemplo el modelo genere textos muy cortos a comparación de a referencia, para la puntuación se asigna un número del 0 al 1, siendo que 0 nos indica que no existe coincidencia por él contra el número 1 índica que tanto el texto generado, como la referencia son iguales.

Esta trata de calcular la precisión de palabras coincidentes comparando una candidata con una referencia, esto lo logra, calculando la cantidad de palabras coincidentes, para posteriormente dividirlo para el total de palabras en la candidata esto para cada n-grama [50].

$$P_n = \frac{(\Sigma \mathrm{c} \in \{\mathrm{Cantidates}\} \Sigma \mathrm{n} - \mathrm{gram} \in \mathrm{C}\ \mathrm{Count}_{\mathrm{clip}}(\mathrm{n} - \mathrm{gram}))}{(\Sigma \mathrm{c} \in \{\mathrm{Cantidates}\} \Sigma \mathrm{n} - \mathrm{gram} \in \mathrm{C}\ \mathrm{Count}(\mathrm{n} - \mathrm{gram}))}$$

Una vez se obtuvo la precisión, se debe calcular una penalización dada por la métrica, la cual penaliza, las frases candidatas que tengan una dimensión inferior a la o las frases de referencia [50].

$$BP = \begin{cases} 1 & C > R \\ e^{(1-R/C)} & C \leq R \end{cases}$$

En caso de tener varias referencias por cada frase candidata, y estas tiene diferentes longitudes entre ellas, se selecciona la más cercana a la frase candidata, y finalmente se combina la precisión, y se usa un logaritmo promedio con peso uniforme que es igual a 4 como en la ecuación [50]:

$$BLEU\ =\ BP\ \times\ exp(\sum_{N=1}^{N} W(n)\ log\ P(n))$$

Con el avance de los modelos IA la búsqueda de métricas capaces de evaluar de manera automatizada es contante, la salida de textos de modelos IA de procesamiento de lenguaje natural contiene varios retos, y BLEU no es una excepción, pues contiene varias carencias en cuanto a la forma de evaluar que maneja, factores como: el orden de las palabras, la presencia de sinónimos, entre otros factores, pueden concluir que esta métrica, no es una puntuación definitiva para determinar que el modelo, excede las capacidades

esperadas tras realizar su ajuste, con estas observaciones es para preguntarse por qué su uso. El uso de esta métrica nos ayuda a poder determinar de manera automática, en que puntación coinciden las frases candidatas con su respectiva frase o frases de referencia, aun con esto, la necesidad aun con el uso de BLEU u otras métricas, de una revisión manual de las salidas dada por el modelo es necesaria debido a que por la complejidad del lenguaje [51]. En el presente TIC para una mejor representación la puntuación BLEU para su presentación se multiplicará por 100 para dar un número en la escala a 100 para conseguir una mejor presentación.

4.2.9.2. Evaluación humana.

Tanto para tareas que involucren la generación de texto por modelos de aprendizaje automático, como texto generado por humanos requiere incluso de manera obligatoria pasar por una evaluación humana, es una forma de determinar la calidad de la misma, aun sabiendo la importancia de la evaluación humana esta cuenta con determinadas limitaciones que es importante tomar en cuenta [52]:

- Su producción es compleja debido al consumo de recursos como el tiempo que requiere.
- La calidad de esta se considera inestable debido a varios factores como: los aspectos que se consideró para la evaluación, los encargados de realizar la evaluación, entre otros.

Métricas de evaluación automática como BLEU, no son capaces de detectar todos los aspectos de una salida de texto de un modelo LLM, debido a que esta tarea es compleja y aun con evaluación humana su evaluación no está exenta de errores [52]. Cualquier tarea realizada por modelos que requiera generación de texto, en su evaluación no se puede limitar solo a métricas automáticas haciendo que se tenga que llevar a cabo la evaluación humana, aun contando con una calidad inestable, su relevancia resalta al conocer que investigadores de diversas áreas justifican la elección de un modelo en base a la evaluación humana, en donde los tópicos a evaluar en muchas ocasiones se puntúan en una escala Likert [53].

Si bien la evaluación humana es aquella que los investigadores consideran con mayor validez, que su calidad sea inestable es una limitante importante para el desarrollo de modelos con mayor capacidad, pues al inicio con modelos pequeños se evaluaba la fluidez, que en comparativa tomaba una cantidad de tiempo reducida, en comparación con las evaluaciones realizadas a modelos de última generación que requieren evaluaciones enfocadas más al contenido en si del texto generado, como por ejemplo mantener el contexto en las oraciones que aun creciendo el texto generado este se mantenga coherente lo cual requiere una lectura más exhaustiva por parte del evaluador, hay que tener como consideración que la comparación efectiva entre modelos usando este método es compleja debido a factores como:

requerir que sean los mismos evaluadores en las pruebas realizadas en distintos modelos, que la prueba realizada al modelo con el cual se quiere comparar sea lo más semejante posible a las realizadas a los modelos con los cuales se requiere comparar, estos factores hacen que la comparación con entre modelos los cuales han llevado un proceso de desarrollo diferente entre ellos puedan ser comparados efectivamente mediante evaluación humana [54].

La evaluación humana la realizó el autor del presente TIC, tomando en cuenta que las evaluaciones humanas tratadas anteriormente eran de forma libre ocupando expertos en la tarea para la cual el modelo fue entrenados, en comparativa en este TIC se creó un dataset y este contiene freses referencias de las salidas esperadas de los modelos por lo cual se cuenta con texto que sería la referencia de la descripción de la tabla de datos, lo cual limita la evaluación humana a comparativas de entre el modelo ajustado y las referencias esperadas, esto utilizando la escala de Likert.

4.2.9.3. Escala de Likert.

En la actualidad la escala Likert es usada en diversas áreas, convirtiéndola en una con el uso más extendido, es una escala sumativa, su ejecución es sencilla, trata de realizar dar una puntuación cuantitativa a un número determinado de ítems propuesta, normalmente y la más utilizada es la escala del 1 al 5, en esta un encuestado es el encargado de otorgar la puntuación asignada a los ítems propuestos, logrando finalmente una puntuación resultado de obtener la media de la puntuación dada a los diferentes ítems, esta fue desarrollada por un investigador social en 1930, en su base trata de registrar la posición que tomó un sujeto frente a unos aspectos determinados de un objeto, para la obtención de estos resultados, se puede explorar distintas fuentes como: trabajos anteriores, el mismo autor u expertos [55][56].

Determinar que las salidas de los LLMs tienen una alta calidad, luego de un proceso de fine tuning resulta en una tarea compleja, por lo que la evaluación humana es indispensable para lograr un LLM que genere respuestas de alta calidad y aceptación para un uso específico, este tipo de evaluaciones puede abarcar varios tópicos, y en cuestión de herramientas que permitan obtener una valoración aceptable resalta el uso de la escala Likert, esta resulta en un estándar para sistemas que tengan un enfoque en realizar diálogos, aunque entre sus limitaciones esta que para asegurar una calidad en diversas áreas se requieren un número elevado de tópicos a evaluar lo cual puede ser prohibitivo debido a los costos que esto puede requerir [57].

En el presente TIC su uso es debido a que en trabajos que han tenido una problemática “table to text” haciendo uso de LLMs como [11] se ha experimentado con el uso de esta escala, permitiendo evaluar las salidas del modelo en base a una escala sencilla, además que la misma permite presentar un promedio de la puntuación alcanzada lo cual nos permitirá realizar

comparaciones con evaluaciones automáticas como las que se realiza utilizando la métrica BLEU.

4.2.9.4. Pruebas A/B

Las pruebas A/B están presentes en varias aplicaciones dentro de la informática, pues su enfoque principal es encontrar resultados cuantitativos de un producto enfocado a un usuario final [58]. Este tipo de pruebas trata de la compasión de entre dos productos con diversos enfoques, como ejemplo en el caso de las páginas web es dos ventanas con diferentes estilos con un mismo propósito, cuál de las dos logra una mejor experiencia de usuario [59]. Estas pruebas son una forma de medir la satisfacción de un nuevo cambio, por lo tanto, permite determinar si para resultados más efectivos se deben aplicar los nuevos cambios propuestos o por el contrario permanecer con la configuración que se está aplicando.

El uso de este tipo de pruebas es muy extendido y se han aplicado dentro del ámbito de los LLM, en donde se divide las diferentes soluciones y se designa un determinado tráfico a estas y se determina la mejor aquella que logra mejores resultados, en estos casos el uso de dataset para estas pruebas está extendido, en cuanto a evaluar modelos obteniendo aquellos que obtengan salidas que más se adecuen a las necesidades del problema abordado [60].

Este tipo de prueba es flexible en cuenta como formular la comparativa, lo cual es interesante aplicar, al momento de realizar la evaluación y determinar un modelo logra una calidad adecuada, al realizar la evaluación ya se ha explorado la opción de realizar comparativas entre las métricas automáticas como BLEU y la evaluación humana en los LLM [61], por lo que existe la posibilidad de realizar este tipo de prueba poniendo entre las opciones a comprar la evaluación automática en BLEU y la evaluación humana. Teniendo en cuenta lo anterior en el presente TIC se utilizó esta prueba para validar los resultados obtenidos por la métrica de BLEU y la evaluación humana, realizando una comparación entre la puntuación dada en electos tomados del conjunto de test con la evaluación automática con la métrica BLEU en comparación con la puntuación obtenida al realizar una evaluación humana, cuyos ítems fueron calificados en escala de Likert, esta prueba arrojara que evaluación es mejor además de permitir establecer comparaciones entre los resultados obtenidos en cada una.

4.2.10. Herramientas de desarrollo.

4.2.10.1. Python.

Es un lenguaje de programación interpretado, que ha tomado gran popularidad a través de los años, con una sintaxis sencilla. La curva de aprendizaje de este es corta en lo referente a un uso básico del mismo, pues este cuenta con recursos para su uso desde la consola, escritorio, web y más recientemente popular en la ciencia de datos, inteligencia artificial. Si algo caracteriza a este lenguaje es su extensa cantidad de librerías, además de una documentación extensa. En el contexto del proyecto, este lenguaje es de alta utilidad

debido a las herramientas que este presenta, como lo son PyTorch, Tensorflow, además de contar con librerías como Transformres, estas serán de gran utilidad en la tarea de ajustar el modelo Gemma, pues con este lenguaje es posible procesar la creación, verificación de los datos a utilizar para su entrenamiento. Realizar dicho entrenamiento en entornos como Google Colab que permite el uso de este lenguaje. Este lenguaje demuestra su utilidad en sus diversas funcionalidades [62].

El lenguaje Python, con una de las sintaxis más sencillas, tanto que código creado con este lenguaje puede llegar a parecer pseudocódigo, su tipado dinámico ofrece una gran flexibilidad, aunque esto tenga un costo en su velocidad de interpretación y no se recomiende su uso para tareas de bajo nivel, su gran versatilidad lo ha posicionado en el mercado, dentro de diversas industrias. La gran disponibilidad de librerías para diversas tareas ha hecho que este lenguaje esté presente en varias aplicaciones desde el desarrollo web, hasta el desarrollo de inteligencia artificial [63].

En [64] se habla de Python como una herramienta óptima para aplicaciones en diversos campos desde las matemáticas básicas, hasta temas más complejos como campos de la IA: machine learning y el deep learning. Dentro de estas áreas de IA, Python ofrece no solo librerías, herramientas como: Tensorfow, PyTorch entre otras. Para el desarrollo de IA si no también cuenta con una comunidad formada que comparte sus conocimientos, ayuda en el continuo desarrollo de la misma, ejemplo de esto es la plataforma Hugging Face Hub[3].

4.2.10.2. PyTorch.

Es una biblioteca de aprendizaje automático, esta fue desarrollada por Facebook, es de código abierto y muestra una gran flexibilidad y facilidad, tanto al crear modelos de aprendizaje automático como para ajustar los mismos, cuenta con una amplia gama de módulos y herramientas integradas para facilitar el desarrollo de modelos de aprendizaje profundo, incluyendo capas predefinidas, funciones de activación, funciones de pérdida, optimizadores y más [65].

PyTorch es un framework ampliamente utilizado, este forma parte de los denominados, frameworks de segunda generación que en comparación con los de primera generación como TensorFlow, cuenta con eager-mode or imperative execution model (modo ansioso o modelo de ejecución imperativo), donde las operaciones se ejecutan de inmediato como semántica[66]. La razón de su uso para este TIC va más allá de su popularidad elevada al igual que TensorFlow, si no que cuenta con características que hacen que resalte. Este tiene una flexibilidad y simplicidad elevada, debido a esto en varios entornos investigativos y de aprendizaje su uso está más que recomendado, pues se considera que aprender a usar este framework resulta más accesible que otras opciones de desarrollo de inteligencia artificial [65].

[3] Hugging Face Hub

Este cuenta con características que lo hacen Pythonic (su código es: legible, consistente, simple y expresivo), una amplia documentación oficial, proporciona computación acelerada usando unidades de procesamiento gráfico (GPU), su amplia adopción reflejada en sus altos números de citas en concretos internacionales[65], además permite el manejo y monitoreo de la memoria utilizada de la GPU[67], todo esto hacen que sea una opción idónea.

4.2.10.3. Hugging Face.

Sitio en donde se encuentran alojados varios recursos referentes a la inteligencia artificial, que van desde modelos de visión por computadora, de procesamiento de lenguaje natural etc., en este es posible encontrar datasets tanto compartidos por la comunidad como propios en caso de que se desee subirlos a la plataforma, esta es un recurso que cuenta con una documentación detallada además de albergar modelos que han sido liberados tanto para su consumo como para su modificación, esto dependerá de la licencia con la cual hayan sido publicados, dentro de esta plataforma se encuentra el modelo Gemma, entre muchos otros. Además, que en la misma se detallan características de los modelos como: función, arquitectura, idiomas y dominios admitidos, licencias y otros aspectos relevantes [68].

Hugging Face en su sitio web[4] muestra que la creo una empresa estadounidense, esta cuenta con la librería Transformers, la cual es de código abierto, el sitio cuenta con su documentación oficial, por otro lado, hay que tomar en cuenta que esta plataforma ayuda con la guía de personas nuevas en el manejo de modelos, permitiéndoles hacer uso de estos, aunque aún falta trabajo en una categorización más especializada de los modelos preentrenados dentro de la plataforma [69]. Dentro de esta plataforma existe el apartado Hugging Face Hub documentation, aquí se dice que la plataforma hay alojados, más de 350000 modelos, 75.000 conjuntos de datos y 150.000 aplicaciones de demostración (Spaces), todas de código abierto y disponibles públicamente, poniendo a disposición de la comunidad, todos estos recursos, con el fin que esta colaboren en el desarrollo [70].

Se ha convertido en un centro popular para todos aquellos interesados en modelos de inteligencia artificial, y datasets, disponibles para realizar entrenamientos y ajustes de modelos, en este existen modelos de todas las arquitecturas: CNN, Transformers [44]. Enfoques como: Visión por computadora, Generación de texto, Multimodales y entre ellos los LLM con el enfoque en el procesamiento de lenguaje natural. Los modelos que encontramos en esta plataforma no solo permiten consumirlos, sino que también permiten su ajuste, pues varios se encuentran con su documentación [71]. Aun con esto se presentan problemas dentro de esta plataforma, entre los que se puede mencionar esta: falta de exposición de los modelos, falta de una documentación correctamente estructurada en un número representativo de estos, falta de información de los datasets usados en el desarrollo de los modelos, modelos

[4] Hugging Face

usados incumpliendo las normativas de licencia de propiedad intelectual, entre otros problemas presentes [71]. Pero aun con todos es tos problemas sigue siendo la plataforma que nuclea, modelos y documentación de esto, razón por la cual cuando se realiza un proyecto que implique el uso de modelos de IA, es requisito visitar este sitio, y el desarrollo de este TIC no es la excepción.

4.3. Trabajos Relacionados

En el contexto de búsqueda de trabajos que estén relacionados sobre la generación de descripciones de tablas de datos, o también conocido en algunos trabajos como: “table to text”. Se han descubierto algunos artículos que tratan esta temática, no como se plantea en este proyecto. La **Tabla 2** presenta los trabajos relacionados obtenidos al investigar sobre el tema del presente TIC, junto con un código el cual se utilizó para referenciar estos trabajos, en este van desde enfoque similar del tema hasta el artículo del modelo en uso.

Tabla 2. Trabajos relacionados

Código	Título	Descripción
TR01	Table-to-text Generation by Structure-aware Seq2seq Learning	En el presente artículo lo que se presenta para resolver la problemática es el uso de una arquitectura seq2seq con reconocimiento de estructura que consta de un codificador de activación de campo y un generador de descripciones con doble atención, para lo cual el dataset utilizado es WIKIBIO el cual contiene 700.000 biografías y los cuadros de información correspondientes de Wikipedia, hay que tener en cuenta que el dataset utilizado denota un sesgo pues se enfoca en bibliografías lo cual reduce el vocabulario necesario además estas tablas son pequeñas por lo cual las dimensiones de la misma denotan las limitaciones de los modelos con arquitecturas seq2seq [8].
TR02	Variational Template Machine for Data-to-Text Generation	Este artículo explora el uso de la máquina de plantillas variacionales (VTM), el cual es un método que permite generar descripciones de texto a partir de tablas de datos, pone el foco varias de las limitantes presentes en modelos que tratan de llevar a cabo esta temática la cual es que los enfoques existentes que utilizan modelos de codificador-decodificador neuronal a menudo adolecen de una

Código	Título	Descripción
		falta de diversidad, lo cual se ve presente en que varios de los modelos construidos utilizan los mismos datasets lo cual puede limitar el resultado obtenido, con lo cual afirma que un conjunto abierto de plantillas es crucial para enriquecer las estructuras de frases y comprender las diferentes generaciones. Aprender este tipo de plantillas no es muy accesible, pues se necesita de un gran conjunto, que no está disponible. En el artículo se afirma que es posible lograr una calidad comparable en generación con Table2seq, además este promueve la diversidad por un amplio margen [9].
TR03	HTLM: HyperText Pre-Training and Prompting of Language Models	El siguiente artículo muestra un enfoque que no está directamente relacionado con temática tablas a texto, pero nos da determinadas conclusiones, el código del mismo no se encuentra liberado por lo que solo es posible analizar los hallazgos registros en el artículo, lo interesante es que entre sus hallazgos habla de la posibilidad de lograr que modelos como Bert aprendan estructuras como HTML lo cual puede ser un enfoque interesante en la temática de generación de tablas de datos a texto [57].
TR04	Investigating Table-to-Text Generation Capabilities of LLMs in RealWorld Information Seeking Scenarios	Este artículo denota una investigación interesante para la problemática, pues estudia la adopción de LLM en aplicaciones del mundo real para la búsqueda de información de tablas, estableciendo en un principio que aún no se ha explorado lo suficiente esta aplicación de los modelos en esta problemática. En este artículo, se investiga las capacidades de conversión de tabla a texto de diferentes LLM utilizando cuatro conjuntos de datos dentro de dos escenarios de búsqueda de información del mundo real. Estos incluyen LogicNLG, LoTNLG los cuales comenta que fueron recientemente construidos, FeTaQA y F2WTQ recientemente construidos para la

Código	Título	Descripción
		generación basada en consultas. Este artículo logra dar un poco de conocimiento de la aplicación de búsqueda de datos dentro de estructuras como las tablas y dar resultados en lenguaje natural lo cual concluye con que esta es una aplicación con gran potencial, además de nombrar a BLEU como una métrica de evaluación popular para este escenario [10].
TR05	Few-Shot Table-to-Text Generation with Prototype Memory	En el presente artículo se nos da a conocer que el desafío de pasar de una tabla a texto, no es nuevo y han existido varios enfoques para llegar a una buena solución, se hace notar que el desafió presente es la diferencia de formato que existe entre la tabla y el texto, al cual el modelo deba llegar, se menciona métricas como el BLEU, también que para que el resultado de los modelos tenga un buen nivel, este requiere de una cantidad de datos elevada, y según el dominio de los datos utilizados se puede obtener resultados distintos, en el documento [11].
TR06	REASTAP: Injecting Table Reasoning Skills During Pre-training via Synthetic Reasoning Examples	El artículo pone en evidencia la dificultad existente en los modelos tanto generales como especializados en datos tabulares, pues estos tienen dificultades al requerir realizar tareas de razonamiento dentro de tablas de datos, estas pueden ser desde responder preguntas, realizar operaciones matemáticas entre las celdas, la generación de texto basándose en una tabla de datos. Los modelos preentrenados que han abordado este tipo de estructura como son las tablas de datos han logrado avances, pues uno de los puntos iniciales es que dichos modelos deben aprender la estructura de la tabla, con REASTAP se presenta un método pensado para modelos LM logrando facilitar la comprensión de las estructuras de las tablas, con lo cual se podría realizar tareas que requieran

Código	Título	Descripción
		razonamiento, en el artículo también se muestra que para evaluar el rendimiento en cuanto a la tarea de generación de texto, se usa BLEU además de usar datos sintéticos para su pre entrenamiento, los cuales son datos generados de manera artificial demostrando así su potencial en la creación de datasets para distintas problemáticas [12].
TR07	Robust (Controlled) Table-to-Text Generation with Structure-Aware Equivariance Learninge	Este artículo presenta algunos de los retos, al tratar de generar texto a partir de tablas de datos, esto es la gran variedad de estas, de entre los desafíos que pueden aparecer es que una vez se realizara un modelo este no sea capaz de generar el texto a partir de la tabla debido a que la estructura de la misma ha cambiado, esto puede ocurrir aún incluso si la información sigue siendo la misma, en este se experimentó con un modelo T5 el cual demuestran que no es capaz de generar texto si la tabla tiene una complejidad determinada en su estructura. Da a conocer como en experimentos anteriores para poder enviar tablas en datasets usados para entrenamiento es necesario que se encuentre en una línea de texto, esto con ayuda de un formato definido y además de tokens especiales, todo esto para que la estructura de la tabla y por tanto su contenido lleguen a ser identificados correctamente, en las métricas de evaluación del texto generado a partir de la tabla de datos se menciona el uso de BLEU la cual la cataloga como popular [13].
TR08	Gemma: Open Models Based on Gemini Research and Technology	En el artículo se detalla que Gemma como tal es una familia de modelos livianos de última generación, pues ha sacado la versión de 2 mil millones y 7 mil millones de parámetros, lo cual causa un rendimiento diferente en cada uno de estos, estos modelos han sido el desarrollo más moderno en cuanto al desarrollo de los LLM de manera responsable, este

Código	Título	Descripción
		cuenta con resultados en cuanto a razonamiento que es superior a otros LLM como Llama, entre otros, si comparamos con otros modelos este cuenta con 8192 tokens, además que al ser un modelo preentrenado destinado a que se puede especializar según el ajuste realizado, puede producir resultados con entrenamientos de menor duración, en comparación con otros trabajos relacionados, que para lograr resultados requieren de días enteros de entrenamiento [5].

TR: Trabajos relacionados

Como conclusión de los trabajos relacionados, la arquitectura Transformers ha demostrado ser competente en trabajos de NLP, pero como tal los experimentos existentes han demostrado tener varias limitaciones en cuenta a las descripciones de las tablas realizadas, pues debido a sus datos no se realizan de forma correcta tampoco toman en cuenta toda la información de la tabla, razón por la cual se ha optado por una tecnología recientemente liberada que es el modelo Gemma[5] el cual en cuestiones de razonamientos en su sitio oficial supera a los experimentos de los trabajos relacionados, pues cuenta varias ventajas como lo es su entrenamiento que al ser para propósitos generales tienen un extenso conocimiento.

5. Metodología

Dentro de esta sección se presentan detalles necesarios para el desarrollo del TIC tanto de la metodología implementada como las herramientas necesarias. Para cumplir el cometido antes mencionado en la sección 5.1 se detalla el área de estudio ocupada en el desarrollo del proyecto; la sección 5.2 detalla el procedimiento necesario que se requirió para lograr cumplir con los objetivos planteados; finalmente dentro de la sección 5.3 se muestra información de los recursos necesarios en el desarrollo del proyecto.

5.1. Área de estudio

Para el desarrollo de este TIC, establecido dentro del periodo académico marzo del 2024 a agosto del mismo año, se hizo uso de las instalaciones dadas por la universidad, en específico el espacio físico de la “Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables” en la sección de la carrera de ingeniería en Computación, la cual se encuentra ubicada en la Av. Reinaldo Espinosa entre la Av. Pio Jaramillo Alvarado y Eduardo Kingman en las coordenadas -4.030718884063791, -79.19969956629753, la dirección se muestra en la **Figura 9.**

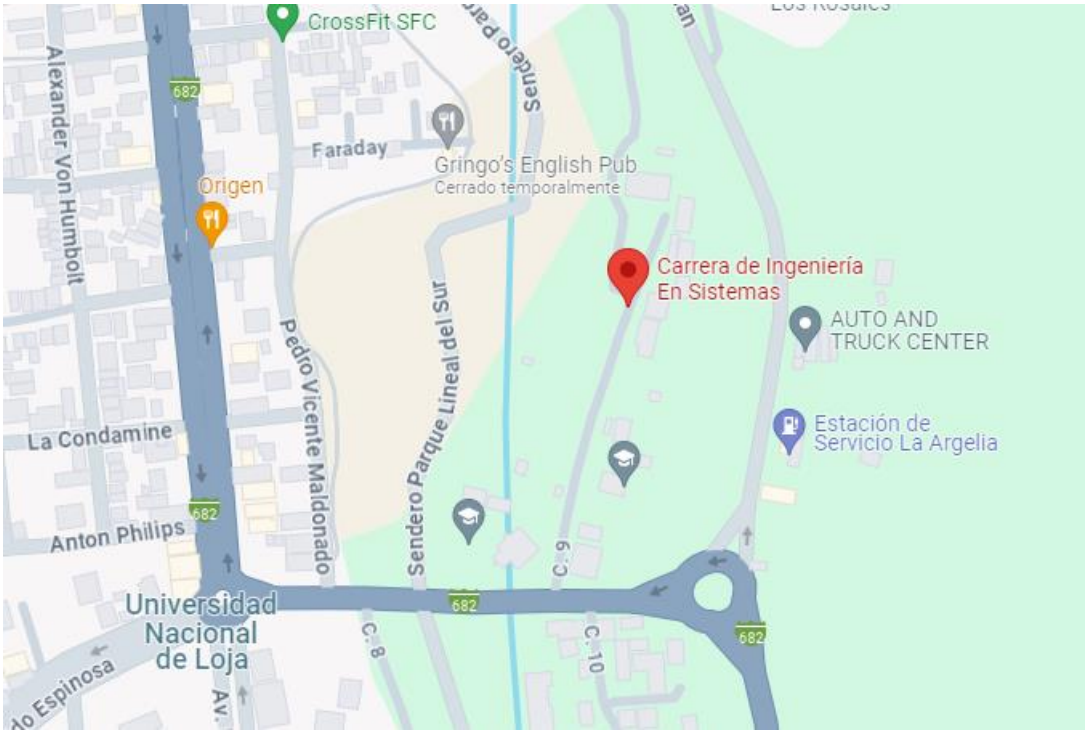


Figura 9. Carrera de Ingeniería en Computación de la Universidad Nacional de Loja.

5.2. Procedimiento

Para alcanzar la meta planteada en el objetivo general del TIC se hizo uso de una metodología inspirada en CRISPML(Q), tomando sus fases y tareas con modificaciones para adaptarlo a los requerimientos de los objetivos, se llegó a implementar fases como:

Comprensión de datos, Ingeniería de datos, Ingeniería de modelos de aprendizaje automático y evaluación del modelo.

5.2.1. Objetivo 1: Ajustar el modelo Gemma para que sea capaz de describir tablas de datos, las cuales estarán en un formato Markdown, un tamaño determinado con un máximo de 3*3, sin tomar en cuenta signos poco usados como los presentes en fórmulas matemáticas, mediante una metodología basada en CRISP-ML(Q).

Fase 1: Comprensión de datos:

Creación del dataset

Al no contar con un dataset que cumpla con las características necesarias para el desarrollo del presente TIC, se optó por la opción de crear uno desde cero. Al requerir tablas de datos, es necesario establecer el tipo de contenido, es decir, las temáticas de información que albergará, para esto, tomando en cuenta las temáticas que han abordado trabajos relacionados como TR01, TR02, TR04, TR05, TR06 y TR07, además de una exploración por los artículos de los datasets utilizados en trabajos relacionados como [14][70], se determinó los tópicos abordados. Se tomó este enfoque debido a observaciones en los artículos de datasets mencionados anteriormente, pues identificaron una disminución del rendimiento de los modelos, conforme aumente la diversidad de temáticas en la información contenida en las tablas de datos, teniendo en cuenta estas consideraciones, se determinaron las siguientes temáticas:

- Ciencia y Tecnología
- Datos bibliográficos, Personas Relevantes
- Datos Demográficos Globales
- Educación y Cultura en el Mundo
- Medio Ambiente, Animales, Plantas y Sostenibilidad

Además de las temáticas, se establecieron los casos necesarios en función del objetivo, en este caso, los tipos de tablas de datos por sus dimensiones, con un límite de dimensiones de tres filas por tres columnas, esto sin contar la fila de las características de la tabla. En la **Tabla 3**, se presentan los tipos de tablas de datos para la creación del dataset, los cuales fueron el resultado de combinar las temáticas con las dimensiones de las tablas de datos requeridas por el alcance del presente objetivo.

Tabla 3. Tipos de tablas de datos.

Nro	Temática	Tipos de tablas por dimensiones
1	Ciencia y Tecnología	1 columna con 1 fila 1 columna con 2 filas 1 columna con 3 filas

Nro	Temática	Tipos de tablas por dimensiones
		2 columnas con 1 fila
		2 columnas con 2 filas
		2 columnas con 3 filas
		3 columnas con 1 fila
		3 columnas con 2 filas
		3 columnas con 3 filas
2	Datos bibliográficos, Personas Relevantes	1 columna con 1 fila
		1 columna con 2 filas
		1 columna con 3 filas
		2 columnas con 1 fila
		2 columnas con 2 filas
		2 columnas con 3 filas
		3 columnas con 1 fila
		3 columnas con 2 filas
		3 columnas con 3 filas
3	Datos Demográficos Globales	1 columna con 1 fila
		1 columna con 2 filas
		1 columna con 3 filas
		2 columnas con 1 fila
		2 columnas con 2 filas
		2 columnas con 3 filas
		3 columnas con 1 fila
		3 columnas con 2 filas
		3 columnas con 3 filas
4	Educación y Cultura en el Mundo	1 columna con 1 fila
		1 columna con 2 filas
		1 columna con 3 filas
		2 columnas con 1 fila
		2 columnas con 2 filas
		2 columnas con 3 filas
		3 columnas con 1 fila
		3 columnas con 2 filas
		3 columnas con 3 filas

Nro	Temática	Tipos de tablas por dimensiones
5	Medio Ambiente, Animales, Plantas y Sostenibilidad	1 columna con 1 fila 1 columna con 2 filas 1 columna con 3 filas 2 columnas con 1 fila 2 columnas con 2 filas 2 columnas con 3 filas 3 columnas con 1 fila 3 columnas con 2 filas 3 columnas con 3 filas

En la **Figura 10** se muestra la configuración de carpeta creada de manera temporal, para lograr un orden al momento de la creación de los datos, estos archivos json se repitieron en cada carpeta, representando las combinaciones de las tablas de datos por cada temática.

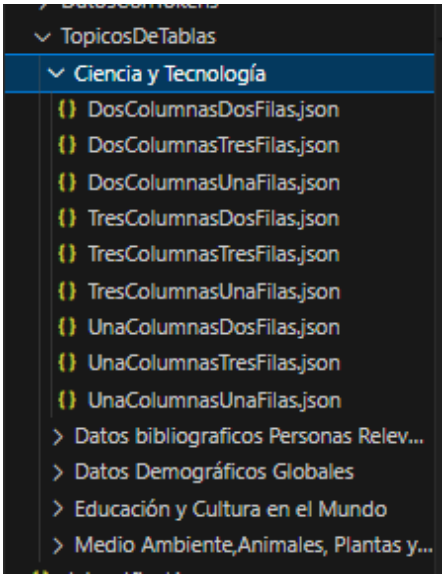


Figura 10. Configuración de carpetas temporal para la creación de los datos.

En la **Figura 11** se muestra la estructura de objetos usada dentro de los archivos json creados (véase **Figura 10**), en esta se denotan tres características: un “id” como identificador del objeto, “tabla” donde se almacenó la tabla de datos en formato Markdown (véase la **Sección 4.2.6.1**) y “texto” en la cual se guardó la descripción de la tabla de datos esperada.

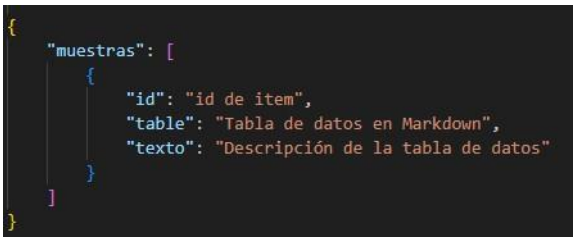


Figura 11. Estructura de los objetos dentro de los archivos json.

Para la generación de datos sintéticos (véase la **Sección 4.2.7**), se realizó una revisión en la web, de tablas de datos existentes. La **Tabla 4** menciona los buscadores utilizados para la verificación de tablas de datos existentes usando las temáticas establecidas (véase **Tabla 3**). Para la obtención de estos datos sintéticos en la **Tabla 5** se detallan las herramientas de IA que se hizo uso, donde se especifican sus nombres, dirección web y el uso que tuvo cada una de ellas.

Tabla 4. Buscadores usados en la búsqueda de tablas de datos.

Nombre	Dirección web
Google	https://www.google.com.ec/
Duckduckgo	https://duckduckgo.com/

Tabla 5. Herramientas IA utilizadas para la creación de datos.

Nombre	Dirección web	Uso
Gemini	https://gemini.google.com/app?hl=es	Su interfaz web se usó para la expansión de las temáticas de las tablas de datos, y su API para la creación de datos sintéticos mediante un prompt.
Perplexity	https://www.perplexity.ai/search/	Se uso para la diversificación del contenido de las tablas de datos, donde se realizó búsquedas de subtemas relacionados a las temáticas establecidas.

Los datos sintéticos fueron creados con ayuda de la IA Gemini (véase **Tabla 5**). La **Figura 12** muestra una porción del código del Script en el cual se utilizó una llave para consumir el modelo Gemini, logrando automatizar la generación de datos, los scripts utilizados se encuentran en un repositorio de GitHub[5]. En la **Figura 13** se muestra el código del Script

[5] Repositorio Github de Scripts

usado para de manera automática, crear el objeto completo en los json (véase **Figura 11**). Esto se lo realizó manejando las diferentes rutas de las carpetas, los json separados por las dimensiones de las tablas de datos (véase **Figura 10**) y un prompt determinado.

```python
import os
import json
import uuid



import pathlib
import textwrap

import google.generativeai as genai

# Used to securely store your API key

from IPython.display import display
from IPython.display import Markdown


def to_markdown(text):
  text = text.replace('•', '  *')
  #return Markdown(textwrap.indent(text, '> ', predicate=lambda _: True))
  return text

genai.configure(api_key="AIzaSyCou9G86XNV-X_L1-_BTeoGi7F4cq55AoM")

model = genai.GenerativeModel('gemini-pro')

```

Figura 12. Script para crear datos sintéticos con Gemini.

```python
iteraciones_totales = 1
iteraciones_por_minuto = 40
tiempo_por_iteracion = 60 / iteraciones_por_minuto  # Calcula el tiempo en segundos entre iteraciones
file_path = "./TopicosDeTablas/"+direccion+"/DosColumnasTresFilas.json"
for i in range(iteraciones_totales):
    # Hacer algo en este momento
    print(f"Iteración {i}")
    response = model.generate_content(prompt)
    agregar_o_crear_json(limpiarcadenaDeasteriscos(limpiar_cadena(normalize_text(str(to_markdown(response.text))))), file_path)
    # Hacer algo cada 40 iteraciones
    #if i % 40 == 0:
    #    time.sleep(1)
```

Figura 13. Script para la creación de datos en los json.

La **Tabla 6** muestra los diferentes prompts usados para la generación de datos, estos variaron según el tipo de tabla requerida, la temática de la misma y alguna variación en la descripción

Tabla 6. Tabla de prompts usados para la generación de datos sintéticos con Gemini.

Prompts	**Variables**		
	dimensiones	**topicoTabla**	**ejemplo**
genera una tabla en formato markdown escrita de manera seguida en una línea, la tabla debe tener ”+dimensiones+”, el	Una columna con una fila Una columna con dos filas	Tópico usado por cada dimensión	El ejemplo del formato establecido

Prompts	Variables		
	dimensiones	**topicoTabla**	**ejemplo**
encabezado de la columna tendrá el tema y las filas deberán expresar información en relevancia con la temática de la columna, " + topicoTabla + ", luego una oración que describa el contenido de la tabla fila por fila, usa solo la información puesta en la tabla no expandas el tema , la información se presente detalladamente pero solo en base a la información de la tabla no agregues información que no esté en la tabla, no pongas la tabla contiene, separa la tabla de la descripción con ?, como el ejemplo: "+ ejemplo	Una columna con tres filas	establecida de las 5 temáticas (ver Tabla1)	en cada tipo de tabla, dado en el código.
	Dos columnas con una fila		
	Dos columnas con dos filas		
	Dos columnas con tres filas		
	Tres columnas con una fila		
	Tres columnas con dos filas		
	Tres columnas con tres filas		

Nota: El prompt usado es el mismo para cada tipo de tabla (véase Tabla 1), las Variables son variables de código, usadas dentro delo prompt, para obtener datos en cada caso, las variables que van cambiando son, dimensiones que almacena las dimensiones de la tabla, topicoTabla esta variable almaceno la temática correspondiente, y ejemplo almacena un ejemplo dado de la salida a nivel de código.

La **Figura 14** muestra el código del Script implementado para el procesamiento del texto de salida obtenido del modelo Gemini usando prompts determinados (véase **Tabla 5**), este tuvo la tarea de filtrar el texto, tanto para quitar saltos de línea innecesarios, darle al texto encoding de utf-8 además de crear el objeto, este lo reconocerá debido a caracteres propios del formato Markdown. Para diferenciar la tabla de la descripción, se hizo uso de un carácter determinado, para posteriormente almacenarlo en el json correspondiente, el script para esta tarea disponible en GitHub[6], el nombre del script es generarDatos.py.

[6] Repositorio Github de Scripts

```python
def agregar_o_crear_json(input_text, file_path):
    try:
        # Intenta dividir la cadena solo en la primera aparición del carácter '?'
        if '?' not in input_text:
            raise ValueError("El carácter '?' no encontrado en el texto de entrada.")

        table, text = input_text.split('?', 1)
        # Elimina espacios en blanco alrededor de 'table' y 'text'
        table = table.strip()
        text = text.strip()

        # Verificar si el archivo JSON ya existe
        if os.path.exists(file_path):
            # Si el archivo ya existe, cargar el JSON y añadir el nuevo objeto
            with open(file_path, 'r', encoding='utf-8') as file:
                data = json.load(file)
                data["muestras"].append({
                    "id": str(uuid.uuid4()),
                    "table": table,
                    "texto": text
                })
            # Escribir los datos actualizados de vuelta al archivo JSON
            with open(file_path, 'w', encoding='utf-8') as file:
                json.dump(data, file, indent=4, ensure_ascii=False)
        else:
            # Si el archivo no existe, crear un nuevo archivo JSON con el objeto
            data = {
                "muestras": [
                    {
                        "id": str(uuid.uuid4()),
                        "table": table,
                        "texto": text
                    }
                ]
            }
            with open(file_path, 'w', encoding='utf-8') as file:
                json.dump(data, file, indent=4, ensure_ascii=False)

    except ValueError as e:
        # Si ocurre un error, imprimir un mensaje de error con información detallada
        print(f"Error al procesar '{file_path}': {e}")
        return
```

Figura 14. Código necesario para el procesamiento del texto de salida del modelo.

Una vez se construyeron todos estos elementos, se consiguió, tanto los archivos y rutas para la creación de datos sintéticos, como el código para generarlos uno por uno con ayuda de las herramientas IA (véase **Sección 6.1.1**)

Fase 2: Ingeniería de datos

Selección de datos

Al utilizar herramientas de IA para generar los datos sintéticos (véase **Sección 4.2.7**), estos no consiguen un existo en el 100% de los casos. Al realizar la generación dato por dato, Gemini obtuvo datos de toda calidad, y en un alto porcentaje estos no servían para el propósito de este TIC, aun así, al generarlos se obtuvieron tablas de datos en formato Markdown de buena calidad por lo que no se descartó su uso. La **Figura 15** muestra como en varias ocasiones el resultado del uso de determinados prompts demasiado genéricos resultaba en tablas de datos que redundaban información, resultando en archivos json con objetos que contenían información de forma repetitiva, la **Figura 16** muestra otro problema presente, que aunque se colocaron filtros, al recibir el texto de salida del modelo aún se almacenan objetos

que no cumplen con la calidad requerida, razón por la cual fue necesario establecer criterios de inclusión y exclusión (véase **Sección 6.1.2**).

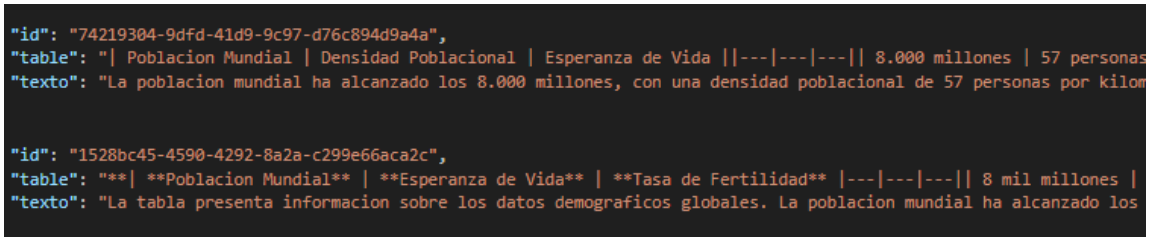


Figura 15. Datos con información y estructura con poca variedad.

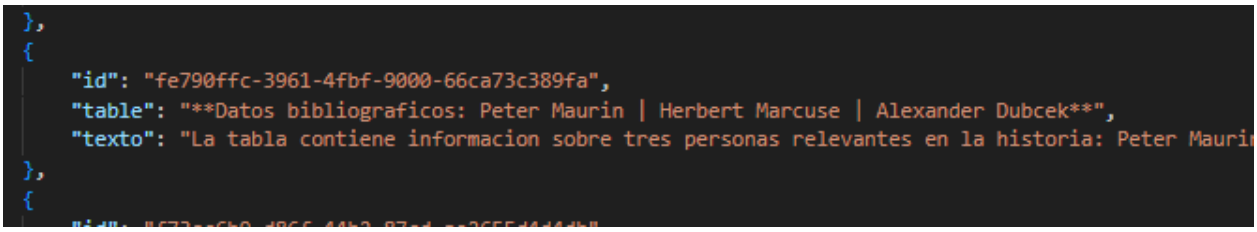


Figura 16. Datos con estructura incorrecta.

Ingeniería de características.

Se realizó una evaluación humana de los datos sintéticos, posteriormente se realizaron rectificaciones de errores de manera manual tanto en la tabla de datos, como dentro de las descripciones generadas, eliminación de datos repetitivos, y balanceando la cantidad de datos por caso, además se realizó correcciones manuales de datos que pasaron por la selección bajo criterios de inclusión y exclusión (véase **Tabla 7**) de manera parcial y era posible adecuarlos para que los cumplieran en su totalidad (véase **Sección 6.1.3**).

Estandarización de datos.

Para la estandarización de los datos se visitó la web del modelo en Hugging Face[7], donde se determina que, la entrada del modelo se debe hacer en una cadena de texto con una pregunta o sugerencia, debido a lo cual revisando dentro de kaggle[8] en ajustes realizados al modelo Gemma se pudo confirmar una plantilla de prompt. La **Figura 17** muestra la plantilla de prompt usada en los ajustes del modelo Gemma, la cual se utilizó para normalizar los datos para el dataset, en donde como “Instruction” se enviara la tabla de datos en formato Markdown y en “Response” se enviara la descripción de la tabla de datos.

```
prompt = f"Instruction:\n{instruction}\n\nResponse:\n{response}"
```

Figura 17. Plantilla de prompt usada en el modelo Gemma.

[7] Gemma en Hugging Face
[8] Plantilla de prompt para normalizar datos.

Debido a una observación dada en un artículo del dataset ToTTo[14], donde se especifica que el rendimiento del ajuste del modelo para la descripción de tablas decaía en su rendimiento conforme se aumentaran los tópicos tratados dentro de las tablas de datos. Se realizó un experimento para verificarlo, se optó por crear dos versiones del dataset (véase **Sección 6.1.4**). La **Figura 18** presenta el script necesario para normalizar los datos a la plantilla de prompt establecida para el modelo GEMMA, el script se encuentra en un repositorio de Github[9], para la normalización primeramente se obtuvo archivos json con la cantidad de datos totales según la versión del dataset, donde se dividió en los conjuntos para train 80%, validation 10% y test 10%, posteriormente estos json pasaron a un formato jsonl muy utilizado en dataset dentro de Huggingface[10], logrando con esto los conjunto de las dos versiones del dataset (véase **Sección 6.1.4**).

```python
import json
import jsonlines as jl

def procesar_dataset(input_json, output_jsonl):
    # Cargar el archivo JSON de entrada
    with open(input_json, 'r') as file:
        dataset = json.load(file)

    # Filtrar y transformar los datos
    filtered_dataset = []
    for _, data in enumerate(dataset["muestras"]):
        if "contexto" not in data:
            text = f"Instruction:\n{data['table']}\n\nResponse:\n{data['texto']}"
            filtered_dataset.append({"text": text})

    # Escribir el resultado en un archivo JSONL
    with jl.open(output_jsonl, 'w') as writer:
        writer.write_all(filtered_dataset)

```

Figura 18. Script necesario para la colocación de tokens especiales del modelo.

Fase 3: Ingeniería de modelos de aprendizaje automático

Selección del modelo preentrenado Gemma.

Para el uso de cualquiera de las versiones del modelo Gemma, fue necesario conocer que, si bien este es un modelo libre, cuenta con una licencia propia establecida por Google, que libera el modelo para su ajuste tanto para fines lucrativos como de investigación, pero establece algunas cláusulas que limitan esta libertad de manejo del modelo. La **Figura 19** muestra dentro de la web de Hugging Face[11] como esta cuenta con una licencia propia, en esta se definen varias condiciones de uso del modelo Gemma (véase **Anexo 2**), para poder descargar el modelo para su ajuste, se tuvo que aceptar los términos y condiciones

[9] Repositorio de Github con el Script utilizado.
[10] Formato de dataset
[11] Modelo Gemma 2B en Hugging Face.

establecidos, pues esta licencia se une a la cuenta de Hugging Face, dando solo acceso a llaves generadas desde cuentas que hayan aceptado estos términos.

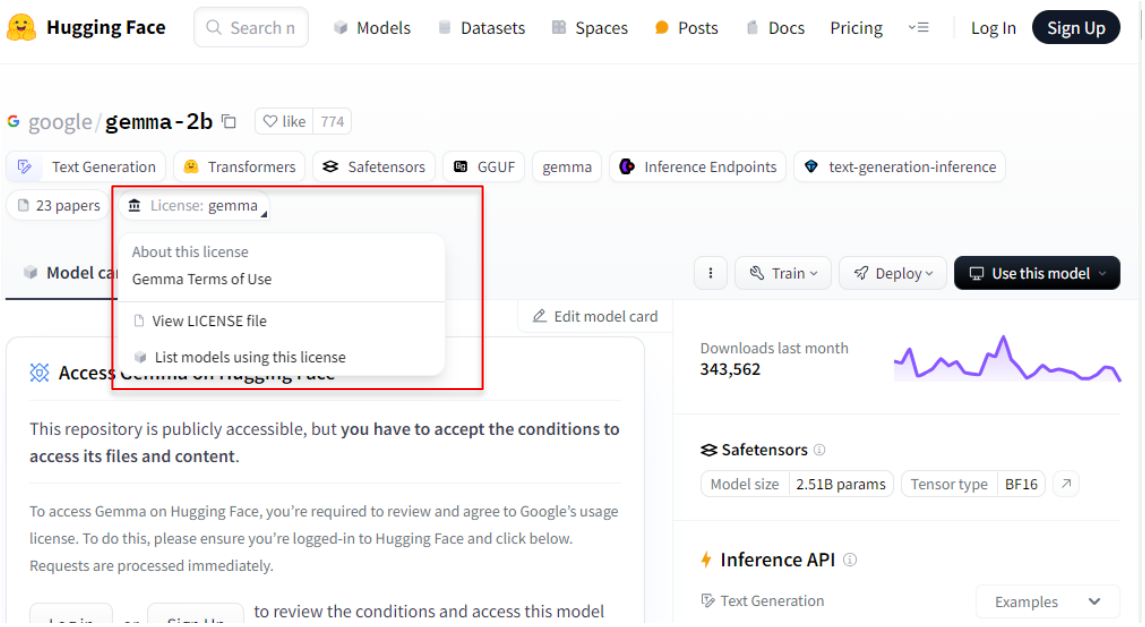


Figura 19. Licencia del modelo Gemma 2B.

La familia de modelos Gemma (véase **Sección 4.2.4**) cuenta con versiones que se diferencian entre ellos en características como el número de parámetros, para realizar la comparativa se tuvo en consideración las características de cada uno (véase **Tabla 1**) donde se presentaron las versiones 2B y 7B, llegando a elegir el modelo que mejor cumplía las necesidades del presente TIC (véase **Sección 6.1.5**).

Comprensión del modelo.

Para lograr una comprensión del modelo fue necesario conocer el funcionamiento de las capas que este contiene además de la arquitectura con la que cuenta, se puede observar la arquitectura de este modelo que se muestra al imprimirla en Google Colab (véase en la **Figura 33**) mostrando además las capas ocultas que contiene, pero para más comprensión de este fue necesario una revisión de tanto de su artículo como de artículos que explicaban el funcionamiento de las capas que contiene el modelo (véase la **Sección 4.2.4**), al investigar se pudo comprobar que la arquitectura de LlaMA2 comparte muchas semejanzas con la arquitectura de Gemma 2B, al revisar la web donde se graficaba la arquitectura de este modelo[12], en otros sitios se obtuvo información gráfica del funcionamiento de capas como la MLP[13] presente en la arquitectura de Gemma, además de conocer como es el funcionamiento de un modelo only decoder transformer[14], se visualizaron las semejanzas lo cual sirvió para

[12] Arquitectura Llama2.
[13] Capa MLP.
[14] Modelo only decoder transformer.

conocer el funcionamiento de capas comunes, con todo esto se logró realizar una representación gráfica del modelo (véase **Figura 34**) en esta se presenta cada una de las capas con las que cuenta el modelo, cabe recalar que las capas al imprimir el modelo le añaden como prefijo a las capas Gemma, estos prefijos no se colocaron en la arquitectura debido a que puede dificultar su identificación.

Aplicar el transfer learning

Para aplicar el transfer learning se eligió una técnica que permita el ajuste en un entorno de capacidades de hardware limitado, la técnica elegida fue PEFT (véase **Sección 4.2.5**). La **Figura 20**, proporciona una visualización de las librerías que se requirió instalar, haciendo especial atención, en la librería Transformers la cual deriva del sitio Hugging Face, que permitió acceso a la documentación necesaria, otras librerías como las necesarias para la ejecución en entornos de hardware limitado, además de importar la librería para el método de transfer learning aplicado PEFT.

```
# Para cualquier actividad básica de HF como cargar modelos y tokenizadores para ejecutar inferencia
# La actualización es imprescindible para el modelo Gemma más nuevo.
!pip install --upgrade datasets
!pip install --upgrade transformers

# Por hacer cosas eficientes - PEFT
!pip install --upgrade peft
!pip install --upgrade trl
!pip install bitsandbytes
#!pip install accelerate

# Para registrar y visualizar el progreso del entrenamiento
!pip install tensorboard
```

Figura 20. Librerías necesarias para realizar el ajuste del modelo.

La **Figura 21** presenta las librerías importadas, en la cual se denota la importancia del paquete “Transformers” el cual nos brindó acceso a herramientas indispensables en el proceso de ajuste del modelo, destacando el método de PEFT implementado, el cual es LoRa (**véase Sección 4.2.5.1).**

```
[ ]  from transformers import AutoTokenizer, AutoModelForCausalLM
     import torch
```

```
[ ]  from transformers import (
         AutoModelForCausalLM,
         AutoTokenizer,
         BitsAndBytesConfig,
         HfArgumentParser,
         TrainingArguments,
         logging,
     )
     from peft import LoraConfig, PeftModel
     from trl import SFTTrainer
```

```
[ ]  from datasets import load_dataset
```

Figura 21. Importación de dependencias.

En la **Figura 22** se establece la forma de cargar los datasets, ya procesados, por lo que están normalizados a la plantilla de input propuesta, que el modelo necesita para su funcionamiento, este al utilizar el método load_dataset cargó las características de los datos en un campo denominado texto, según cada caso se cargó los archivos de la primera o segunda versión del dataset.

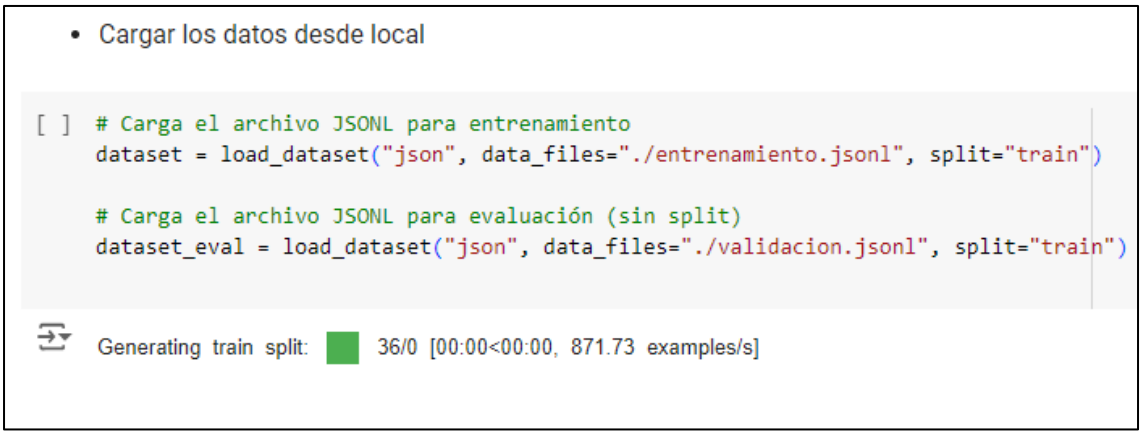


Figura 22. Cargar datasets de entrenamiento y validación.

Para reducir el uso de memoria debido a la gran cantidad de parámetros con la que cuenta el modelo Gemma 2B, se siguió la documentación de ajuste PEFT en LLM de Hugging Face para disminuir la memoria requerida, para esto se utilizó Quantization[15]. Aplicando

[15] Documentación de Quantization en Hugging Face.

Quantization se logró representar datos con menos bits, la opción de Quantization elegida es la precisión de 4 bits con la biblioteca bitsandbytes, según la documentación en Hugging Face se recomienda QloRA el cual es un método que cuantifica un modelo a 4 bits y luego lo entrena con LoRA, lo cual permitió el ajuste en el entorno de una GPU de colab, los valores usados en la configuración (véase **Tabla 10**) fueron los recomendados en su documentación. En la **Figura 23** se demuestra, el proceso realizado para limitar la precisión del modelo, a 4bits la función de cada uno de los valores es:

- load_in_4bit: Para cuantificar el modelo a 4 bits cuando lo cargue.
- bnb_4bit_quant_type: Para usar un tipo de datos especial de 4 bits
- bnb_4bit_use_double_quant: Para usar un esquema de cuantificación anidado para cuantificar los pesos ya cuantificados
- bnb_4bit_compute_dtype: Para usar bfloat16 para un cálculo más rápido

```
# Load QLoRA configuration
# Cargar configuración QLoRA
compute_dtype = getattr(torch, bnb_4bit_compute_dtype)

bnb_config = BitsAndBytesConfig(
    load_in_4bit=use_4bit, # Activa la carga de precisión de 4 bits / Activates 4-bit precision loading
    bnb_4bit_quant_type=bnb_4bit_quant_type, # nf4
    bnb_4bit_compute_dtype=compute_dtype, # float16
    bnb_4bit_use_double_quant=use_nested_quant, # False
)
```

Figura 23. Configuración establecida para el método PEFT denominado QLoRA.

La **Figura 24** muestra el código necesario para cargar el modelo base (Gemma 2B), en Google Colab, fue necesario además un token de escritura, esto debido a que se necesitó la descarga del modelo, este proceso de descarga en cada oportunidad, fue la mejor opción, pues si se decidía tener el modelo descargado, sería dentro del drive el cual debido a su capacidad no nos permitió almacenar el modelo, pues en su versión más ligera tiene un peso de 10 GB, siendo esta la versión denominada 2B, la cual se utilizó para realizar el proceso de ajuste del modelo.

```
# Cargar el modelo Base
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    token=hf_token,
    quantization_config=bnb_config,
    device_map=device_map
)
model.config.use_cache = False
model.config.pretraining_tp = 1

tokenizer = AutoTokenizer.from_pretrained(model_name,
                                          token=hf_token,
                                          trust_remote_code=True)
tokenizer.pad_token = tokenizer.eos_token
# Fix weird overflow issue with fp16 training
# Solucionar un extraño problema de desbordamiento con el entrenamiento fp16
tokenizer.padding_side = "right"
```

Figura 24. Proceso para cargar el modelo pre-entrenado Gemma 2B.

Una vez se realizó la investigación de métodos eficientes de ajuste en este caso PEFT, se determinó una amplia gama de los mismos esto dependiendo del uso que se le iba a dar, pues en artículos como [33] se demuestra que, según la utilidad, se recomienda el uso de diferentes métodos, de entre ellos en la **Figura 25** se muestra uno de los que más resaltan para esta tarea es el que se utiliza para tareas de tipo chatbot, el cual es conveniente debido, al proceso que el modelo realiza, determinando el formato del input al modelo, el mismo que se divide en dos, los del usuario y los que contienen la respuesta dada del modelo. La configuración dada para los hiperparámetros de LoRa se estableció con los mismos utilizados de ejemplo de la documentación de LoRa en Hugging Face[16] (véase **Tabla 11**). Según la documentación LoRa en PEFT[17] los hiperparámetros utilizados en el TIC tienen las siguientes funciones:

- r: el rango de las matrices de actualización, expresado en int. Un rango más bajo da como resultado, matrices de actualización más pequeñas con menos parámetros entrenables.
- target_modules: Los módulos (por ejemplo, bloques de atención) para aplicar las matrices de actualización de LoRA.
- lora_alpha: factor de escala de LoRA.
- lora_dropout (flotante): la probabilidad de abandono de las capas de Lora.

[16] Documentación LoRa en Hugging Face.
[17] Documentación LoRa con PEFT en Hugging Face.

- Bias: específica si los parámetros de sesgo deben entrenarse. Puede ser 'none', 'all' o 'lora_only'.

```
# Load LoRA configuration
# Cargar configuración LoRA
peft_config = LoraConfig(
    lora_alpha=lora_alpha,
    lora_dropout=lora_dropout,
    r=lora_r,
    bias="none",
    task_type="CAUSAL_LM",
    target_modules=["q_proj", "k_proj", "v_proj", "o_proj","gate_proj", "up_proj"]
)
```

Figura 25. Se carga la configuración de QLorRa dentro de PEFT.

Según los experimentos realizados, se usó el constructor SFTConfig para cargarlo en el entrenador, este fue cambiando, conforme la configuración lo necesitase, pues valores como las épocas, iba cambiando para comprobar como varió la perdida al momento de realizar el entrenamiento. Para determinar los hiperparámetros se consultaron diversas fuentes: En cuanto a bach size, epoch y steps los trabajos relacionados no pueden ser tomados en cuenta debido a que estos se llevaron a cabo usando clusters de GPUs por lo que no se tomó en cuenta estos valores, para determinarlos se realizaron experimentos buscando los valores que obtengan mejores resultados. En learning rate TR03 propone 5e-5, el TR06 le da un valor de 1e-4 y en el TR07 se propuso un valor de 2e-4, estos son los valores de learning rate que se utilizaron para los experimentos realizados, tomando en cuenta experimentos realizados con LlaMA2 otro LLM han demostrado que un learning rate de 2e-4 tiene un aprendizaje óptimo con el rango LoRA [72] por lo que es el learning rate con el que se empezó los experimentos. Para el optimizador se tomó en consideración TR06 debido a que utilizó LLMs recomendando Adamw y Adafactor, debido a que se limitó la capacidad de memoria en datos al modelo se obtuvo opciones de estos optimizadores existentes para la configuración aplicada[18]. Para lr_scheduler_type al consultar un foro[19] y documentación de Hugging Face[20] se recomienda el uso de “constant” para que la tasa de aprendizaje se mantenga constante, para los hiperparámetros faltantes se tomó en cuenta los valores por defecto en la documentación de Hugging Face[21]. En la **Figura 26** se presenta el constructor en el cual se cargó la configuración más importante, esto es valores como: la cantidad de épocas, la métrica

[18] Documentación 8-bit optimizers Hugging Face.
[19] Foro para determinar lr_scheduler_type
[20] Documentación lr_scheduler_type en Hugging Face.
[21] Documentación de la configuración de SFT en Hugging Face.

por la cual se monitorea entre otros hiperparámetros, se tomó en cuenta también que por ejemplo en el TR02 la métrica que se monitorea y se usa para determinar el mejor modelo es el Loss, en este TIC se planteó la posibilidad de monitorear la métrica BLEU y guardar el mejor modelo según la misma con las limitaciones que esto presento, la imagen muestra la configuración básica, pero según los experimentos esta fue cambiando según las necesidades de cada caso.

```
training_arguments = SFTConfig(
    output_dir=output_dir,
    num_train_epochs=num_train_epochs,
    per_device_train_batch_size=per_device_train_batch_size,
    load_best_model_at_end = True,
    metric_for_best_model = "loss",
    gradient_accumulation_steps=gradient_accumulation_steps,
    optim=optim,
    save_steps=save_steps,
    logging_steps=logging_steps,
    learning_rate=learning_rate,
    weight_decay=weight_decay,
    fp16=fp16,
    bf16=bf16,
    max_grad_norm=max_grad_norm,
    max_steps=max_steps,
    warmup_ratio=warmup_ratio,
    group_by_length=group_by_length,
    lr_scheduler_type=lr_scheduler_type,
    report_to="tensorboard",
)
```

Figura 26. Configuración determinada para el constructor SFTConfig.

La **Figura 27** muestra el constructor que se requirió declarar, para poder poner en marcha el método PEFT, este consumió la configuración de SFTConfig (véase **Tabla 12**), cuyo código se reflejó en la **Figura 26** este permitió el uso de los conjuntos de datos, en el apartado “text” en la cual se almacenaron las diferentes líneas que representaron a los datos, ya normalizados al formato de template del modelo, en este se estableció la cantidad máxima de tokens que el tokenizador podrá aceptar, además se consumió la configuración QloRa (véase **Tabla 10**) y la configuración de hiperparámetros LoRa (véase **Tabla 11**) establecida en la **Figura 25.**

```
# Set supervised fine-tuning parameters
# Establecer parámetros de ajuste fino supervisados
trainer = SFTTrainer(
    model=model,
    train_dataset=dataset,
    eval_dataset= dataset_eval,  # Aquí se agrega el conjunto de datos de evaluación
    peft_config=peft_config,
    dataset_text_field="text",
    # formatting_func=format_prompts_fn,
    max_seq_length=max_seq_length,
    tokenizer=tokenizer,
    args=training_arguments,
    packing=packing,
    #predict_with_generate = True,
    callbacks = [early_stop],
    compute_metrics = compute_metrics
)
```

Figura 27. Configuración de parámetros para el ajuste PEFT.

Teniendo en consideración el código necesario para llevar a cabo el ajuste del modelo, se determinaron los valores para cada Hiperparámetro en su correspondiente configuración, los valores establecidos se asignaron tomando en cuenta factores como la limitada capacidad computacional, el tiempo de entrenamiento, la memoria (véase **Sección 6.1.6**).

Documentar el modelo

Para la documentación del modelo con el ajuste llevado a cabo, se creó un cuaderno en Google Colab por cada uno de los experimentos realizados, esto en la primera versión de los experimentos solo se dio seguimiento al Loss obtenido, en ellos se puede observar cómo se carga el modelo, con la llave dada en Hugging Face, el proceso de cargar los datos y los resultados obtenidos como se puede ver en la **Figura 28,** esta carpeta se encuentra dentro del drive la cuenta institucional[22].

Figura 28. Experimentos realizados

Debido a que el método utilizado para el ajuste del modelo PEFT con LoRa, la clase que permite este ajuste SFTTrainer[23] es un enfoque en investigación no cuenta con la función implementada de manera nativa para el seguimiento de la métrica BLEU, pero mediante la revisión de foros, en un foro de Github[24] se encontró una manera para poder realizar este

[22] Carpeta de los primeros experimentos realizados.
[23] Documentación de SFTTrainer en Hugging Face.
[24] Foro sobre la forma para hacer seguimiento de la métrica establecida BLEU.

seguimiento con limitaciones. Una vez obtenido una manera de realizar el seguimiento de la métrica BLEU durante el entrenamiento, se volvió a realizar una serie de experimentos los cuales se encuentran en drive[25], a cada experimento se le asignó el nombre “experimento” acompañado del número que lo identifica, esto tanto en las carpetas como en los cuadernos de Google Colab, una vez realizados se registró los resultados obtenidos, siendo posible en estos casos registrar tanto el Loss como la puntuación BLEU alcanzada (véase **Tabla 14**).

Con ayuda de la librería tensorboard fue posible presentar mediante gráficas la evaluación tanto del Loss como de la puntuación BLEU durante el entrenamiento, del segundo grupo de experimentos realizados (véase **Tabla 14**), lo cual permitió establecer los hiperparámetros que mejor resultados estaban obteniendo o concluir que los hiperparámetros usados no realizaban un ajuste de manera eficiente (véase **Sección 6.1.8**).

5.2.2. Objetivo 2: Evaluar mediante la métrica BLEU los distintos modelos obtenidos al usar hiperparámetros diferentes, obteniendo aquel que mejor logre describir tablas de datos en formato Markdown.

Fase 4: Evaluación del modelo.

Evaluar el modelo usando BLEU.

Para evaluar los diferentes experimentos realizados, primeramente, se cargó el mejor modelo guardado en la fase de entrenamiento, se hizo uso del conjunto de test de la versión del dataset con el que fueron entrenados, se procedió a extraer los datos del conjunto de test correspondiente en formato JSON, debido a que este tiene separada de manera más detallada la tabla de datos, con la referencia correspondiente. Estos se almacenaron en listas, posteriormente, por cada experimento se cargó el modelo Gemma ajustado, en el cual se envió cada tabla de datos en formato Markdown de la lista de tablas antes obtenida como input al modelo, para la obtención de la descripción generada por el modelo fue necesario procesar la salida del modelo (véase **Figura 36**). Seguidamente, estas salidas fuerovin almacenadas en una lista, que finalmente, con la lista de referencias, se realizó el cálculo de la puntuación BLEU correspondiente con ayuda del paquete “evaluate” de Python.

Se calculó la puntuación BLEU de los primeros experimentos realizados (véase **Tabla 15**) en estos no se monitoreó la evolución de la métrica BLEU, por lo que solo se registró el Loss, medido con el conjunto de “Validation” de la versión del dataset correspondiente, cada vez que se realizaba una validación por steps al modelo en su fase de entrenamiento, cabe recalcar que la validación en estos experimentos se realizó cada 100 steps.

Utilizando la misma forma de cargar el conjunto de test según la versión del dataset que le corresponda al modelo, se realizó la evaluación de BLEU en el segundo grupo de experimentos realizados, para valorar el rendimiento del modelo se tomó en cuenta la

[25] Carpeta con experimento dando seguimiento a BLEU.

puntuación BLEU obtenida con el conjunto de test de la versión del dataset correspondiente, se registró la puntuación BLEU por experimento (véase **Tabla 16**).

Documentar la fase de evaluación.

Se detalló conclusiones obtenidas tanto de los primeros experimentos realizados (véase **Tabla 15**), además se realizó una explicación de la necesidad de realizar el segundo grupo de experimentos (véase **Tabla 16**). Se detalló los hallazgos encontrados con los experimentos en los cuales se varió los hiperparámetros, se evaluó BLEU con test del dataset correspondiente como corpus, se obtuvo la puntuación BLEU de cada dato de test, además de la puntuación BLEU para 2,3 y 4 n-gramas (véase **Sección 6.2.2**).

Se procedió a realizar una prueba para verificar que tanto puede generalizar en cuanto a temáticas no presentes en el dataset, para lo cual se tomó el mejor modelo obtenido, con la primera versión del dataset y tomando el conjunto de test de la segunda versión de manera randómica con una semilla se eligieron una cantidad de datos igual a la contenida en el conjunto de test con el cual fue entrenado, con esto se obtuvo una puntuación que verifico si el modelo entrenado con la primera versión del dataset era capaz de generar descripciones de tablas con información sobre temas de los cuales no se lo entreno (véase **Sección 6.2.2**).

Finalmente, para validar que modelo obtenido con una puntuación BLEU alta genera descripciones de calidad, se realizó una prueba A/B, teniendo en cuenta que el mejor modelo se obtuvo entrenándolo con la segunda versión del dataset y su conjunto de test cuenta con 180 datos, se procedió a tomar de forma random con una semilla de 12345 el 10 % de los datos del conjunto de test. La prueba A/B consistió en comparar la puntuación BLEU asignada a los datos escogidos de manera random con una puntuación que se asignó en la evaluación humana, donde se midió con ayuda de una escala Likert del 1 al 5 la cual es usada en TR02 y TR5 al realizar evaluaciones humanas, los siguientes tópicos se determinaron con base en evaluaciones humanas realizadas en TR02, TR04 y TR05, siendo el 1 desacuerdo y el 5 de acuerdo, en el caso de la presencia de alucinaciones el 1 representa un alto grado de presencia de alucinaciones y un 5 que no existen alucinaciones en la salida del modelo:

- Integridad de la información.
- Coherencia y fluidez.
- Precisión del significado.
- Presencia de alucinaciones

Elegir y empaquetar el modelo.

Por cada experimento que se realizó se guardó el mejor modelo obtenido al realizar el ajuste y una vez finalizados todos los experimentos se eligió el modelo ajustado Gemma final (véase **Sección 6.2.3**).

5.3. Recursos

5.3.1. Recursos Científicos.

- **Método Analítico:** Con ayuda de este método, se logró separar el proceso a realizar, con el fin de cumplir con los dos objetivos específicos planteados en este TIC, además de adaptar las fases de la metodología CRISP-ML(Q) para ello.
- **Investigación Bibliográfica:** El proyecto realizado en este TIC, cuenta con una variedad de conceptos específicos, en su mayoría relacionado con la IA, para lograr su definición, así como para explicar el fundamento de su uso, al lograr realizar el objetivo general. La investigación de la bibliografía se realizó de manera exhaustiva, revisando fuentes de alto impacto y fiabilidad, tales como: artículos científicos, revistas científicas, tesis entre otras.
- **Método Experimental:** Este método aportó al proceso realizado en las diferentes fases de la metodología CRISP-ML(Q), este aún con la modificación en las fases de la metodología, aporto a la idea de realizar experimentos en cada fase que se realizó, desde la creación del dataset, el ajuste del modelo y la posterior evaluación del mismo (verse **Sección 6**).
- **Entrevista:** La entrevista se realizó con el fin de obtener información de los retos que presentaba el cumplimiento del objetivo del TIC, esta se realizó mediante Zoom, y fue grabada, subida a la plataforma de YouTube, para posteriormente extraer la información obtenida. La entrevista (véase **Anexo 1**), contiene información dada por un experto en temas de deep learning y procesamiento de lenguaje natural, en la misma se da información, tanto del tema del dataset, el proceso de ajuste y la métrica de evaluación.

5.3.2. Recursos técnicos.

- **GPU NVIDIA Tesla T4:** Es la GPU usada para el ajuste del modelo Gemma, la misma es la disponible en el entorno de Google Colab, esta cuenta con 16 GB de VRAM y capacidades de aceleración de IA optimizadas para cargas de trabajo de inferencia.
- **Google Colab:** El entorno dado por Google, para usar recursos computacionales, dentro de un cuaderno, la GPU dada en el entorno T4, fue usada para el ajuste del modelo.
- **Gemini:** Modelo multimodal de Google, el cual fue usado para la creación de datos sintéticos, los cuales permitieron la creación del dataset para finalizar la primera fase del primer objetivo del TIC.
- **Gemma:** Es una familia de modelos open source de Google los cuales entran en la clasificación LLM, de entre los cuales se eligió la versión 2B, para cumplir con

el objetivo del TIC, esto debido a que es el más ligero de las versiones, por lo que permite su ajuste en entornos de bajos recursos computacionales, en este caso fue ajustado en el entorno de Google Colab.

5.3.3. Participantes.

Para conseguir completar el objetivo del presente TIC, se contó con los siguientes participantes:

- Patricio Paredes, autor del presente TIC responsable de completar el objetivo general planteado.
- Ing. Oscar Cumbicus, como director del TIC, quien llevo a cabo correcciones y supervisiones de las tareas realizadas para el objetivo general.

6. Resultados

Los resultados obtenidos del Proyecto de Integración Curricular se presentan en la presente sección:

6.1. Objetivo 1: Ajustar el modelo Gemma para que sea capaz de describir tablas de datos, las cuales estarán en un formato Markdown, un tamaño determinado con un máximo de 3*3, sin tomar en cuenta signos poco usados como los presentes en fórmulas matemáticas, mediante una metodología basada en CRISP-ML(Q).

El primer objetivo se logró llevar a cabo con ayuda de las fases: Comprensión de datos, Ingeniería de datos e Ingeniería de modelos, los cuales fueron una adaptación de las fases propuestas de la metodología CRISP-ML(Q) a continuación se detallan las tareas realizadas en cada fase:

Fase 1: Comprensión de datos.

6.1.1. Creación del dataset

Para la creación del dataset en primera instancia se establecieron los casos tanto por temática establecida, como los casos existentes dada las combinaciones posibles de las tablas según el límite fijado en el objetivo de 3 filas por 3 columnas esto sin tener en cuenta los encabezados de las filas. La **Figura 29** muestra una ventana de Visual Code en la cual en la parte izquierda se muestran las carpetas una por cada temática, y los diferentes archivos json albergando cada una combinación posible de las dimensiones de las tablas de datos (véase **Tabla 3)** y el archivo abierto es una de las combinaciones o casos de tabla el cual contiene objetos con la estructura establecida (véase **Figura 11**) para cada dato, en estos archivos se fueron cargando los datos sintéticos generados con la herramienta AI Gemini, la carpeta con la información mostrada se encuentra almacenada en drive[26].

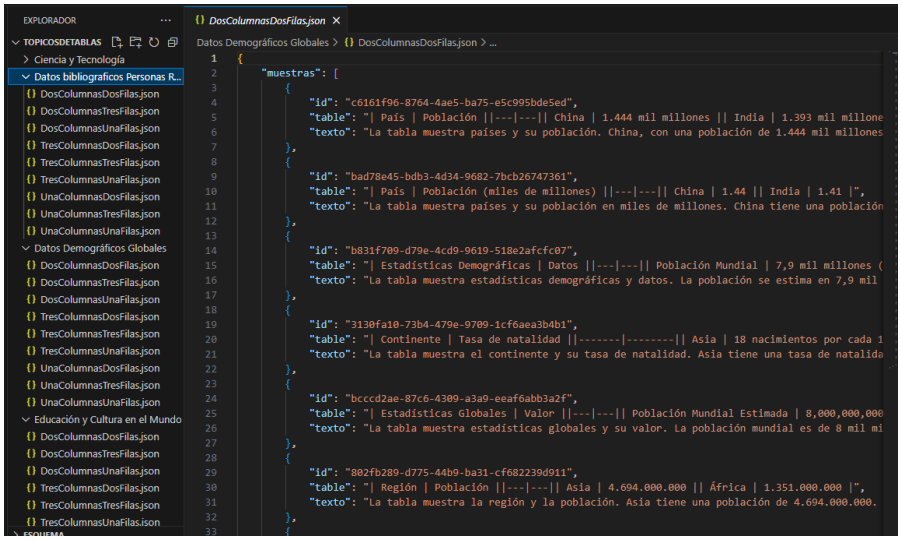


Figura 29. Datos sintéticos generados sobre los tópicos del dataset.

[26] Distribución de carpetas por casos para la generación de datos sintéticos.

Fase 2: Ingeniería de datos.

6.1.2. Selección de datos.

Al hacer uso de la herramienta Gemini para la generación de datos sintéticos, esta solo generaba datos con un porcentaje de éxito del 20%, reflejando a su vez la incapacidad de los LLM genéricos para llevar a cabo tareas concretas, se logró garantizar datos válidos estableciendo criterios de inclusión y exclusión. La **Tabla 7** detalla los criterios de inclusión y exclusión usados al momento de selección de datos, estos se evaluarán en cada objeto (véase **Figura 10**) de los archivos json que contienen los datos correspondientes, para esto se debe evaluar tanto el elemento "tabla" que corresponde a la tabla en donde una de las más grandes observaciones fue que cumpla el formato Markdown de tablas de datos (véase la **Sección 4.2.6.1**), cabe destacar que al tener un porcentaje de éxito tan bajo al generar los datos sintéticos, se aceptó datos que cumplieran los criterios de inclusión y exclusión de manera parcial.

Tabla 7. Criterios de inclusión y exclusión de datos.

Inclusión	Exclusión
Cumple con la sintaxis en la tabla del formato Markdown.	Existen caracteres que no correspondan al formato Markdown establecido.
La tabla tiene las dimensiones correctas.	La tabla posee dimensiones que no corresponden.
La estructura de la tabla es correcta, sin tener combinaciones en filas ni columnas	La tabla posee caracteres de la sintaxis de las tablas Markdown que no corresponden
Las dimensiones de la tabla coinciden con los establecidos en la ruta	La tabla contiene caracteres poco usados, o extraños
La descripción corresponde a la tabla	No hay descripción
La descripción toma en cuenta el encabezado de la tabla	Descripción demasiado genérica.
La descripción explica el contendió de la tabla fila por fila	La descripción de la tabla explica la misma por sus columnas
La descripción no abarca todo el contenido de la tabla de datos.	La descripción contiene información que no corresponde a la tabla

6.1.3. Ingeniería de características.

Una vez culmino la evaluación de cada dato según los criterios de inclusión y exclusión (véase **Tabla 7**) se procedió a evaluar la calidad de dichos datos con una revisión manual donde se realizó diversas actividades, se eliminó datos redundantes o que no pertenecieran a la temática, en los casos de datos que cumplieron con los criterios de inclusión y exclusión de manera parcial, se los evaluó uno por uno modificándolos de manera manual para que cumplieran con la calidad establecida por los criterios de inclusión y exclusión. La **Figura 30** muestra datos cuando se realizó una verificación manual, al evaluar la calidad de este, si no cumplía con una mínima calidad (criterios de inclusión y exclusión), requirió de una

modificación habiendo casos donde la descripción fue realizada completamente manual, el proceso de evaluación y modificación manual en caso de requerirlo se realizó en cada uno de los datos.

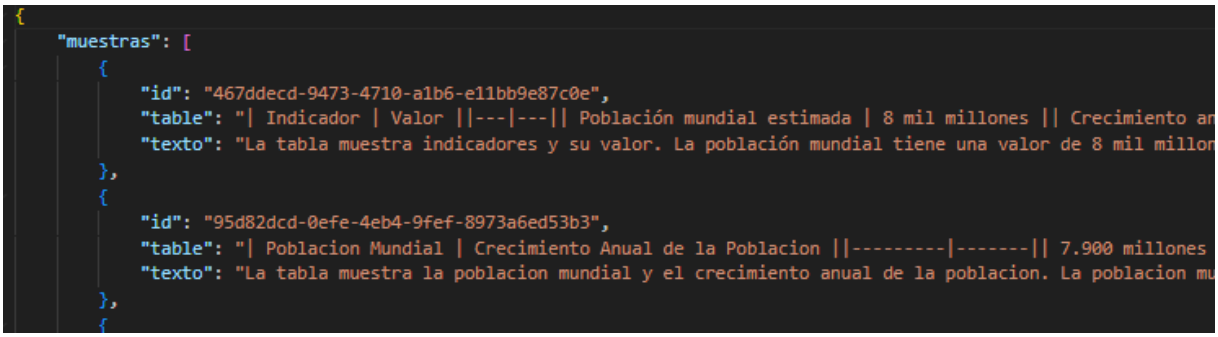


Figura 30. Datos con revisión manual.

Una vez se realizó una revisión manual, se contabilizó para dejar balanceados todos los casos de tablas de datos, se eliminó y creo nuevos datos según el caso para obtener el mismo número en cada uno de los json por cada temática. La **Tabla 8** muestra la cantidad de datos sintéticos por cada tipo, estos ya han pasado por factores de inclusión y exclusión, además de la corrección manual por lo que se consideran datos válidos listos para pasar a la fase de normalización, estos se encuentran en una carpeta del drive institucional[27].

Tabla 8. Cantidad de datos finales.

Nro	Temática	Tipos de tablas por dimensiones	Cantidad de datos
1	Ciencia y Tecnología	Una columna con una fila	40
		Una columna con dos filas	40
		Una columna con tres filas	40
		Dos columnas con una fila	40
		Dos columnas con dos filas	40
		Dos columnas con tres filas	40
		Tres columnas con una fila	40
		Tres columnas con dos filas	40
		Tres columnas con tres filas	40
2	Datos bibliográficos, Personas Relevantes	Una columna con una fila	40
		Una columna con dos filas	40
		Una columna con tres filas	40
		Dos columnas con una fila	40

[27] Datos obtenidos

Nro	Temática	Tipos de tablas por dimensiones	Cantidad de datos
		Dos columnas con dos filas	40
		Dos columnas con tres filas	40
		Tres columnas con una fila	40
		Tres columnas con dos filas	40
		Tres columnas con tres filas	40
3	Datos Demográficos Globales	Una columna con una fila	40
		Una columna con dos filas	40
		Una columna con tres filas	40
		Dos columnas con una fila	40
		Dos columnas con dos filas	40
		Dos columnas con tres filas	40
		Tres columnas con una fila	40
		Tres columnas con dos filas	40
		Tres columnas con tres filas	40
4	Educación y Cultura en el Mundo	Una columna con una fila	40
		Una columna con dos filas	40
		Una columna con tres filas	40
		Dos columnas con una fila	40
		Dos columnas con dos filas	40
		Dos columnas con tres filas	40
		Tres columnas con una fila	40
		Tres columnas con dos filas	40
		Tres columnas con tres filas	40
5	Medio Ambiente, Animales, Plantas y Sostenibilidad	Una columna con una fila	40
		Una columna con dos filas	40
		Una columna con tres filas	40
		Dos columnas con una fila	40
		Dos columnas con dos filas	40
		Dos columnas con tres filas	40
		Tres columnas con una fila	40
		Tres columnas con dos filas	40
		Tres columnas con tres filas	40

6.1.4. Estandarización de datos.

La distribución de los datos sintéticos es simétrica, esto es que el número de datos por tópico es el mismo, esto es debido a que en artículos como [73] pone en consideración, el problema que este tipo de experimentos presentó, explicando que si no se presenta temáticas similares, el modelo puede mostrar falencias en temáticas de las que no cuente datos, esto sumado a la variedad presente en las tablas de datos, hace que la tarea sea más exigente, para la comprobación de esto se generó dos versiones del dataset, en el primero aislando una temática, se aisló solo la temática “Ciencia y Tecnología” al aislarlo la cantidad de datos se limita a 360 manteniendo balanceadas los tipos de tablas establecidos (véase **Tabla 3**) y la segunda versión contiene los datos creados en la versión de 360 datos además de completarlos con las temáticas faltantes, contando que por cada temática se crearon 360 datos, con las cinco temáticas se logró un total de 1800 datos manteniendo la distribución de datos que se refleja en la **Figura 31,** de igual manera manteniendo todos tipos de tablas (véase **Tabla 3**) balanceadas.

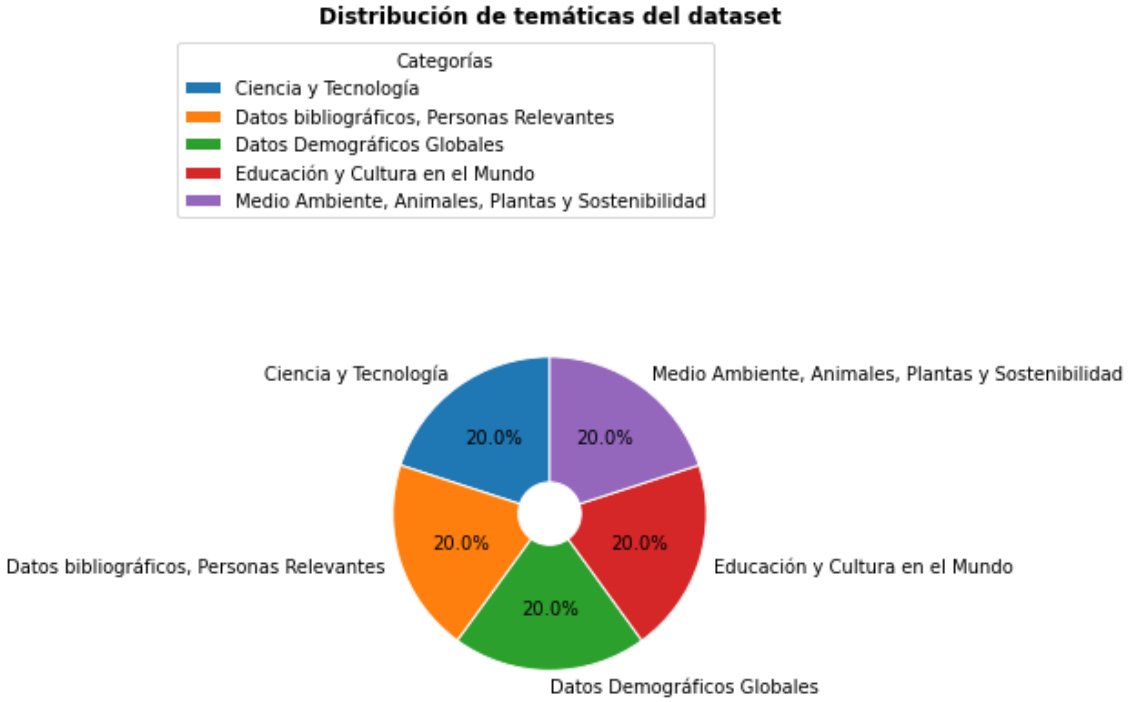


Figura 31. Distribución de tópicos usados para la construcción del dataset.

Se logró las dos versiones de dataset combinando los json que contenían los datos válidos de manera random con una semilla de 123, toma tanto la tabla de datos en Markdown, como su descripción uniéndolos en el formato de un prompt con el cual el modelo Gemma puede realizar su entrenamiento, logrando así la normalización de los datos. La **Tabla 9** muestra la configuración de las dos versiones del dataset logradas, la primera con una sola temática y, por tanto, con un número limitado de datos 360 y la segunda versión al tener en cuenta todas las temáticas cuenta con 1800 datos, estos una vez superada la fase de

ingeniería de datos y dividiendo los conjuntos de datos en: 80% para entrenamiento, 10% para validación y 10% para prueba, los datasets se encuentran dentro de una carpeta en el drive institucional[28].

Tabla 9. Configuración de los dataset creados.

	Primera versión	Segunda versión
Split	**Cantidad de datos**	**Cantidad de datos**
Train	288	1440
Validation	36	180
Test	36	180
Total	360	1800

Fase 3: Ingeniería de modelos.

6.1.5. Selección del modelo Gemma.

Se escogió de entre la familia de modelos Gemma una versión de tamaño reducida, la versión con el menor número de parámetros, tal y como se muestra en la **Figura 32** el modelo elegido de la familia Gemma es la versión 2B, se pudo observar que al tener más capacidad el modelo 7B requiere más recursos computacionales, además de razones como que es la única versión que puede ejecutarse en Google Colab.

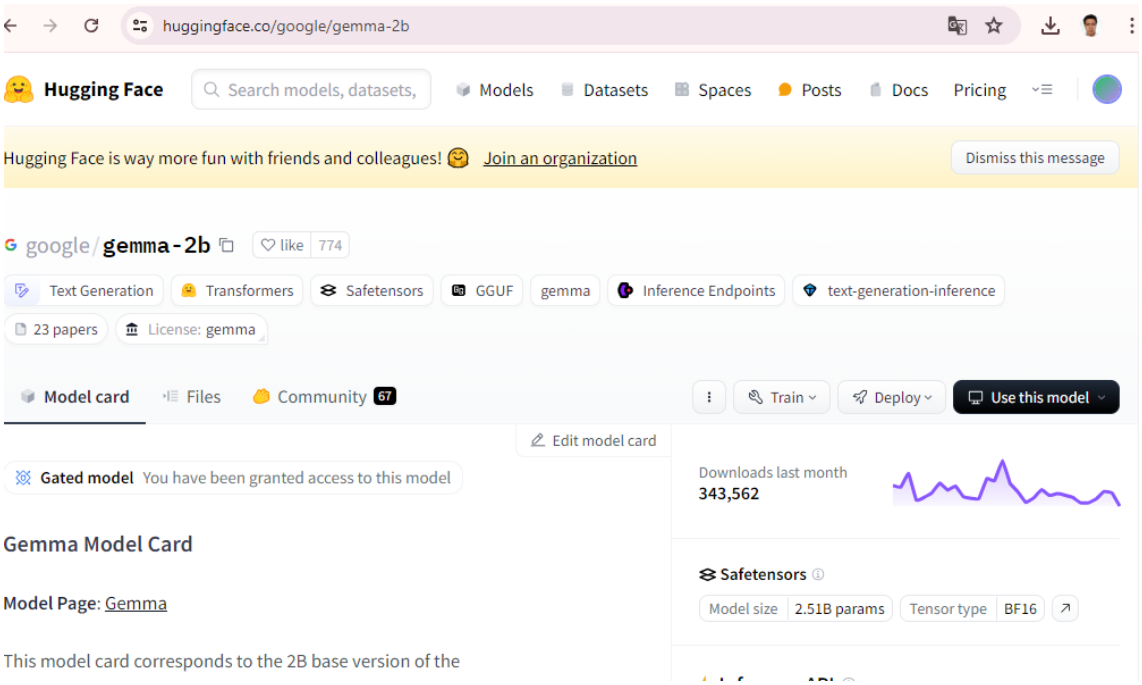


Figura 32. Tarjeta de información del modelo Gemma 2b de Hugging Face.

[28] Datasets logrados.

Se logó consumir el modelo mediante una cuenta de Hugging Face, pues para se requiere una llave de acceso la cual se consiguió siguiendo las indicaciones de la plataforma[29], esto debido a que la licencia de Gemma se asigna a la cuenta de Hugging Face. Al cargar el modelo base en este caso Gemma 2B tanto el modelo como su tokenizador, es necesario enviar en sus parámetros la llave de acceso antes generada. La **Figura 33** muestra la arquitectura del modelo base cargado (Gemma 2B), en esta fue posible observar características ya detalladas anteriormente (véase **Tabla 1**) como las 18 capas decoder, funciones de activación entre otras, la figura es el resultado de imprimir el modelo dentro del cuaderno en Google Colab.

```
GemmaForCausalLM(
  (model): GemmaModel(
    (embed_tokens): Embedding(256000, 2048, padding_idx=0)
    (layers): ModuleList(
      (0-17): 18 x GemmaDecoderLayer(
        (self_attn): GemmaSdpaAttention(
          (q_proj): Linear4bit(in_features=2048, out_features=2048, bias=False)
          (k_proj): Linear4bit(in_features=2048, out_features=256, bias=False)
          (v_proj): Linear4bit(in_features=2048, out_features=256, bias=False)
          (o_proj): Linear4bit(in_features=2048, out_features=2048, bias=False)
          (rotary_emb): GemmaRotaryEmbedding()
        )
        (mlp): GemmaMLP(
          (gate_proj): Linear4bit(in_features=2048, out_features=16384, bias=False)
          (up_proj): Linear4bit(in_features=2048, out_features=16384, bias=False)
          (down_proj): Linear4bit(in_features=16384, out_features=2048, bias=False)
          (act_fn): PytorchGELUTanh()
        )
        (input_layernorm): GemmaRMSNorm()
        (post_attention_layernorm): GemmaRMSNorm()
      )
    )
    (norm): GemmaRMSNorm()
  )
  (lm_head): Linear(in_features=2048, out_features=256000, bias=False)
)
```

Figura 33. Arquitectura impresa en Google Colab del modelo Gemma versión 2B.

6.1.6. Compresión del modelo

El modelo Gemma es un modelo LLM por lo que cuenta con las características de estos (véase la **Sección 4.2.3.1**) de allí que cuenta con una arquitectura de solo decoder de transformer, además cuenta con capas que le permiten un nivel elevado de abstracción (véase la **Sección 4.2.4**), debido al conocimiento que nos da su documentación fue necesario entender que este permite un número de parámetros para u ajuste con el método PEFT(véase la **Sección 4.2.5**). La **Figura 34** muestra la arquitectura con la que cuenta el modelo Gemma 2B hay que tomar en cuenta que este modelo es de base only decoder tansformer,

[29] Guía llave de acceso Hugging Face.

añadiéndole capas, para la representación de su arquitectura se tuvo en cuenta los elementos con los que cuenta el modelo al imprimirlo (véase **Figura 33**).

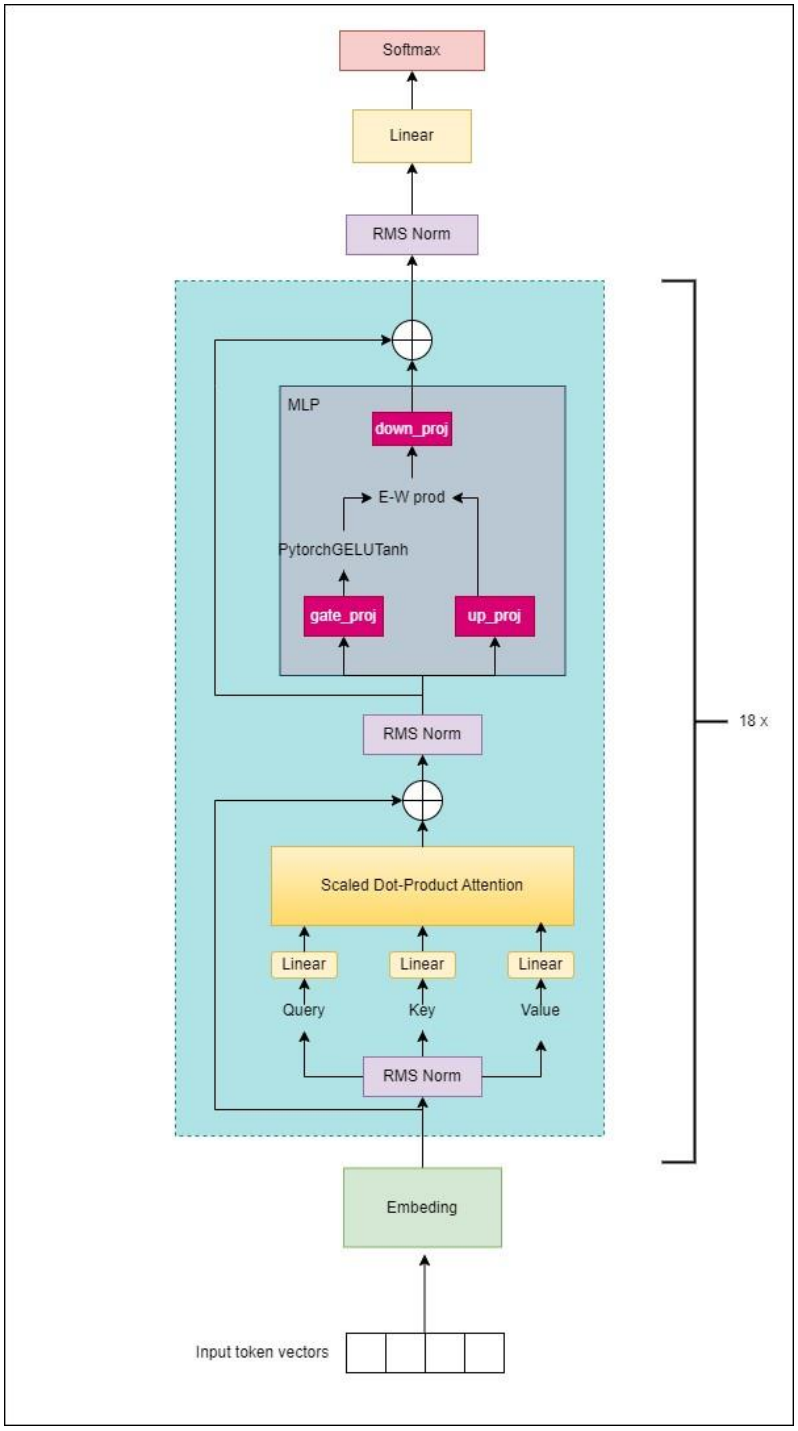


Figura 34. Arquitectura del modelo Gemma 2B.

6.1.7. Aplicar Transfer learning.

El ajuste de un modelo como Gemma 2B en un entorno de una sola GPU dada por Google Colab requiero que se optimice el uso de memoria cuanto sea posible por lo cual se

utilizó quantizing de 4-bit precisión con la biblioteca bitsandbytes. La **Tabla 10** muestra los valores asignados tomados de su documentación[30], con el cual se logró utilizar LoRa con pocos recursos computacionales optimizando el uso de memoria.

Tabla 10. Configuración de QLoRa

Parámetro	Valor asignado
load_in_4bit	True
bnb_4bit_quant_type	nf4
bnb_4bit_use_double_quant	False
bnb_4bit_compute_dtype	float16

Para el ajuste del modelo se tuvo en cuenta varios valores, en sus variables las cuales puedan maximizar los resultados obtenidos, en la **Tabla 11** se empezó con la configuración del método PEFT utilizado, al configurar LoRA, los hiperparámetros se asignaron haciendo enfoque en las capacidades de cómputo limitada que ofrece Google Colab razón por la cual se establecieron valores de menor consumo de memoria posible.

Tabla 11. Configuración de los hiperparámetros LoRA.

Hiperparámetro	Valor asignado
lora_r	4
lora_alpha	16
lora_dropout	0.1
bias	none
task_type	CAUSAL_LM
target_modules	"q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", “up_proj"

La **Tabla 12** muestra los parámetros que se utilizó para el entrenamiento, en estos se denotan varios que se mantendrán, esto debido a que ya son los más eficientes, además que en casos como el batch size depende de los recursos computacionales por los que se inició con un valor de 4, pero se experimentó también con 8 y 16, esto se aplicó en los distintos cuadernos de Google Colab donde se realizaron los experimentos[31] esta primera versión de los experimentos se realizó sin monitorear la métrica BLEU en el proceso de entrenamiento, por lo que, una vez se encontró una forma de darle seguimiento se realizaron nuevos

[30] Documentación Quantizing en Hugging Face.
[31] Carpetas de los cuadernos de Google colab con los experimentos realizados.

experimentos, dentro de la tabla aquellos hiperparámetros con más de una opción es donde se realizó experimentos utilizando cada opción determinando las opciones con mejores resultados a nivel de la puntuación de la métrica que se dio seguimiento.

Tabla 12. Configuración de los parámetros para el ajuste del modelo.

Parámetro	**Valor asignado**
batch_size	4
epoch	1 o 10 o 20
gradient_accumulation_steps	1
learning_rate	2e-4
save_strategy	steps
metric_for_best_model	“loss” o ”eval/bleu”
save_total_limit	2
load_best_model_at_end	True
evaluation_strategy	steps
logging_strategy	steps
do_eval	True
eval_steps	10 o 100
greater_is_better	False o True
gradient_accumulation_steps	1
optim	paged_adamw_32bit o Adafactor
save_steps	10 o 100
logging_steps	10 o 100
learning_rate	2e-4 o 1e-4 o 5e-6
weight_decay	0.001
fp16, bf16	False
max_grad_norm	0.3
max_steps	-1
warmup_ratio	0.03
lr_scheduler_type	"constant"
report_to	tensorboard

La **Figura 35** muestra el código, que permitió arrancar con el entrenamiento, en él se pudieron visualizar varios indicadores importantes que dieron lugar, a la verificación realizada en cada experimento que se realizó, pues este presenta una monitorización por steps

realizados, se marcó el tiempo consumido, además de un marcador de las épocas realizadas, y finalmente como se guardó el módulo generado por PEFT con los parámetros modificados.

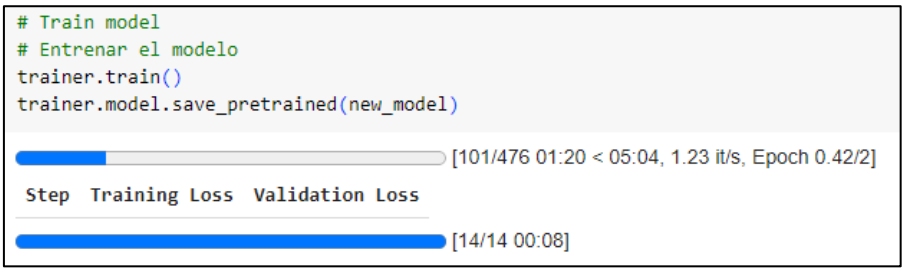


Figura 35. Ejecutar el entrenamiento del modelo Gemma 2B.

6.1.8. Documentar el modelo.

La **Tabla 13** representa los experimentos iniciales que se llevó a cabo con distintos valores en este caso en la variable épocas, si se observa esta tabla se puede verificar que no se ha llegado a una perdida baja, esto lo que reflejo es que existe oportunidad de mejora en el modelo, el impedimento más grande presentado son los requerimientos computacionales, pues el tiempo estimado por época es de 15 min, por lo que se requirió alargar el tiempo para realizar un mayor número de experimentos, estos experimentos realizados se guardaron en diferentes cuadernos de Google Colab en una carpeta del drive institucional[32], al realizaros solo se evaluó la perdida, ya que la evaluación por métrica se trató en el segundo objetivo.

Tabla 13. Experimentos iniciales realizados.

	Loss	
Cantidad de Épocas	**Primera versión**	**Segunda versión**
1	2.281	1.720
10	0.1854	1,643
20	0.09	--------

La **Tabla 14** muestra los resultados obtenidos de los experimentos realizados, cuando se tuvo a disposición una forma de realizar el seguimiento de la métrica BLEU durante el entrenamiento, en la columna Dataset si contiene un valor 1 se trata de la primera versión del dataset, y si es 2 es la segunda versión del dataset (véase **Tabla 9**), cabe recalcar que estos experimentos cuentan con un Loss mayor a los primeros experimentos de la (véase **Tabla 13)** debido a que al momento de programar la función compute_metrics()[33] para su uso es necesario EarlyStoppingCallback[34] en el cual fue posible determinar un parámetro de nombre

[32] Carpeta de experimentos realizados.
[33] https://huggingface.co/docs/trl/sft_trainer
[34] https://huggingface.co/docs/transformers/main_classes/callback

“early_stopping_patience” el cual permite determinar una paciencia, si por ejemplo se puso una paciencia de 20, luego de encontrar la puntuación más alta para guardar el mejor modelo, realizo de nuevo la validación y si en 20 validaciones el puntaje del mejor modelo no mejoró el entrenamiento se detiene, esto es muy útil en entrenamientos de mucho tiempo puesto que llegan a un punto que no existe una mejora en la métrica y tiene sentido seguir con el entrenamiento, estos experimentos se encuentran almacenados dentro de drive[35], una nota importante es que los experimentos del 1 al 6 se consideró para guardar el mejor modelo el Loss, y los experimentos del 7 al 12 se consideró la puntuación BLEU para guardar el mejor modelo.

Tabla 14. Experimentos realizados haciendo un seguimiento de BLEU.

Experimento	Épocas	Loss	BLEU	Dataset	Patience/ Paciencia
1	1	2,00	59,80	1	20
2	10	1,879	62,86	1	40
3	20	2,156	61,14	1	40
4	1	1,439	68,34	2	20
5	10	1,38	68,96	2	40
6	20	1,401	69,62	2	2
7	1	2,00	59,61	1	20
8	10	1,904	61,45	1	10
9	20	2,296	61,37	1	40
10	1	1,443	68,54	2	20
11	10	1.4859	69,812	2	20
12	20	1.4463	69,28	2	20
13	10	1.4364	69,311	2	20
14	10	1.5292	69,190	2	20
15	10	1.5729	66,395	2	20
16	10	1.5053	69,552	2	20

Nota: Patience se refiere a un parámetro de EarlyStoppingCallback el cual determina cuantos en este caso pasos (steps) esperar sin que la puntuación de la métrica mejore, si esta no lo hace pasado el límite establecido el entrenamiento termina.

A partir del experimento 12 se hizo uso de la configuración de hiperparámetros del experimento 11 junto con variaciones en determinados hiperparámetros con el fin de conseguir un modelo con mejor rendimiento. En el experimento 13 se modificó el batch size a 8, en otro se modificó a 16 pero este valor no se pudo ejecutar por limitaciones de memoria

[35] Carpeta de experimentos monitoreando BLEU

del entorno de Google Colab[36]. En el experimento 14 se modificó el learning rate a 1e-4 y en el experimento 15 se modificó el learning rate a 5e-6. Finalmente, el experimento 16 se modificó el optimizador a Adafactor.

Para una mejor representación libreras como tensorboard nos permitieron visualizar y dar seguimiento al desarrollo del entrenamiento realizado para el ahorro de espacio se descartó los primeros experimentos donde solo se monitorio Loss debido a que al evaluar BLEU con su conjunto de test los resultados fueron muy inferiores a los obtenidos a los experimentos donde se monitoreó tanto el Loss como el BLEU durante el entrenamiento (véase **Anexo 3**).

6.2. Objetivo 2: Evaluar mediante la métrica BLEU los distintos modelos obtenidos al usar hiperparámetros diferentes, obteniendo aquel que mejor logre describir tablas de datos en formato Markdown.

Fase 4: Evaluación del modelo

6.2.1. Evaluar el modelo usando BLEU.

Para evaluar la métrica BLEU en cada experimento realizado fue necesario por cada uno de ellos cargar el modelo ajustado, esta evaluación se realizó con la parte de test, del dataset utilizado para entrenar el modelo, según la versión que corresponda (véase **Tabla 9**), haciendo uso de su versión en json lo cual facilito cargarlas en listas, las tablas fueron enviadas una por una al modelo. La **Figura 36** muestra el proceso que es necesario realizar en cada salida del modelo, para obtener la descripción generada con la tabla en Markdown como entrada, es en el código en la imagen es una función que toma como parámetros el texto generado, la lista fue guardando las descripciones y una lista donde guardo las salidas originales del modelo sin procesar, posteriormente en el interior de esta se digitaron líneas que permitieron tomar el texto luego de “\n\nResponse:\n”, en caso de que no cumpla con este formato se guardó la cadena completa, cabe destacar que este tipo de salidas obtuvieron una puntuación BLEU de 0,0. En cuanto se obtuvo las listas: de referencias cargadas del json y de respuestas generadas por el modelo ajustado. Se concluye calculando la puntuación BLEU con ayuda del paquete “evaluate”.

[36] Cuaderno del experimento fallido con batch size de 16.

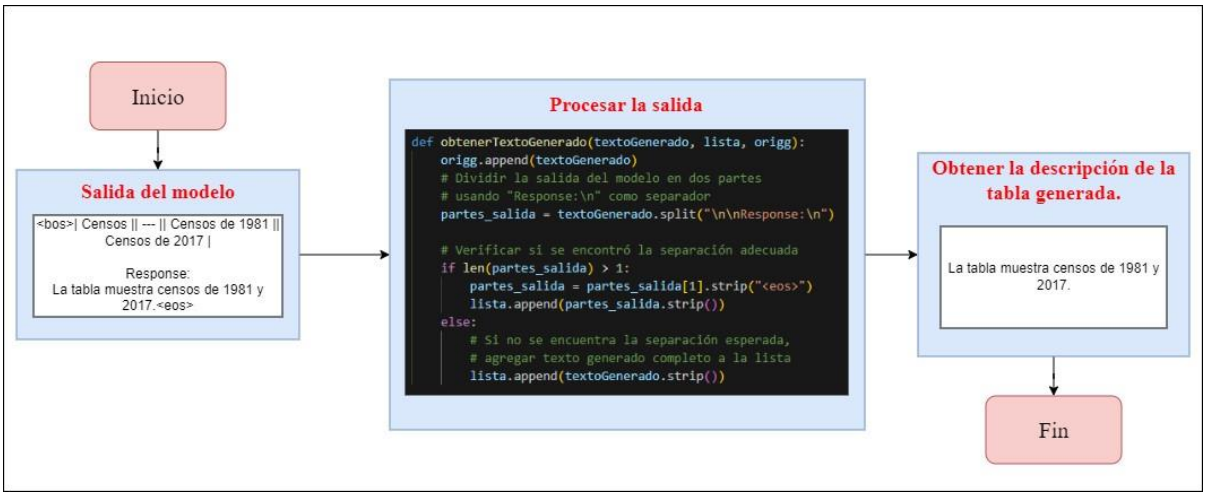


Figura 36. Proceso llevado a cabo para extraer la descripción de la tabla generada.

La **Tabla 15** muestra los resultados en los primeros experimentos realizados donde solo se dio seguimiento a Loss, aquí se obtuvo la evaluación BLEU, con el conjunto de test correspondiente según la versión del dataset utilizada para su entrenamiento y validación. Siendo que para los items del 1 al 3 al usar la primera versión del dataset este tiene 360 datos en contando “train”, “validation” y “test”; por otra parte, los items del 4 y 5 al usar la segunda versión del dataset se utilizó el que cuenta con 1800 datos en total, esto se detalla en la configuración de los datasets creados (véase **Tabla 9**).

Tabla 15. BLEU en evaluación con el conjunto de test.

Item	Cantidad de Épocas	Dataset	BLEU en test
1	1	Primera versión	22,87
2	10	Primera versión	35,54
3	20	Primera versión	41,30
4	1	Segunda versión	26,41
5	10	Segunda versión	53,09

La **Tabla 16** muestra los resultados de los experimentos en los que se midió BLEU durante el entrenamiento, calculando BLEU con el conjunto de test que corresponda, cabe recalcar que para evaluar se realizó el procesamiento de la salida del modelo (véase **Figura 36**), en este se observó que el modelo entrenado con la primera versión del dataset, tuvo un puntaje mejor que los modelos entrenados con la segunda versión del dataset comprobando

que una diversidad de datos ayuda a tener mejores resultados, los resultados de la puntuación BLEU se registraron en el cuaderno de cada experimento[37].

Tabla 16. BLEU evaluando con el conjunto test.

Experimento	Épocas	Dataset	Métrica para guardar el mejor modelo en entrenamiento	BLEU en test
1	1	1	Loss	45,636
2	10	1	Loss	49,807
3	20	1	Loss	50,277
4	1	2	Loss	70,939
5	10	2	Loss	69,626
6	20	2	Loss	71,716
7	1	1	BLEU	51,245
8	10	1	BLEU	50,639
9	20	1	BLEU	55,267
10	1	2	BLEU	63,810
11	10	2	BLEU	73,015
12	20	2	BLEU	72,776
13	10	2	BLEU	72,692
14	10	2	BLEU	72,758
15	10	2	BLEU	61,774
16	10	2	BLEU	74,009

6.2.2. Documentar la fase de evaluación.

En el caso de los primeros experimentos (véase **Tabla 15**) estos al realizarse sin el monitoreo de BLEU no fue posible comprobar cómo evoluciona en la métrica, por lo que para evaluar sus resultados, se tomó en cuenta la puntuación BLEU obtenida con el conjunto de test, a lo cual debido a que en estos experimentos no se tomó cada qué número de steps se debería realizar la evaluación, razón por la cual el puntaje BLEU obtenido fue muy bajo, razón por la cual se los descartó, no sin antes establecer que el mejor modelo de estos es el entrenado con la segunda versión del dataset por lo que se determina que mientras más datos y temáticas se tenga en el dataset mejor será el rendimiento del modelo obtenido.

Dentro de las pruebas del segundo grupo (véase **Tabla 16**), el puntaje a tomar en cuenta fue el que se obtuvo al momento de realizar la medición de BLEU con el conjunto de

[37] Carpeta con los cuadernos del segundo grupo de experimentos.

test, en cuanto a las configuraciones de LoRa y QloRa estas se mantienen estables en todos los experimentos, cuando se evaluó los experimentos del 1 al 6 donde se guardó el mejor modelo según el Loss, y los experimentos del 7 al 12 donde se guardó el mejor modelo según la mejor puntuación BLEU, se determinó que guardar según BLEU obtenía mejores resultados, los experimentos fueron planteados de esta forma para ir modificando hiperparámetros con el fin de obtener una configuración óptima, al concluir el experimento 12, se determinó que la mejor opción para registrar métricas, evaluar y guardar el modelo fue cada 100 steps, dado que fue la configuración del experimento 11, el mejor hasta ese momento, con el experimento 13 se trató de subir el batch size, pero en cuanto a la puntuación BLEU de test tuvo una disminución en su puntaje por lo que se descartó volviendo a un batch size de 4. Con el experimento 14 se modificó el learning rate 1e-4 que una vez evaluado en BLEU con el conjunto test no presento mejora por lo que fue descartado. Con el experimento 15 se volvió a intentar modificar el learning rate a un valor de 5e-6, dando una vez más resultados negativos por lo que el learning rate de 2e-4 se mantiene como el mejor. Finalmente, el experimento 16 modifica el optimizador hasta ahora se utilizó paged_adamw_32bit una versión de Adamw, pero con este experimento si hubo mejora donde se obtuvo una puntuación BLEU de 74,009 lo cual lo convirtió en el mejor modelo obtenido.

En cada uno de los experimentos se evaluó no solo BLEU en 1 n-grama sino también en 2,3 y 4 n-gramas, estos se encuentran en el cuaderno de cada experimento realizado, además se generó tanto un informe en el cual se escribió la puntuación BLEU un n-grama, de cada dato del conjunto test de la versión del dataset correspondiente. La **Figura 37**, muestra los archivos que contiene la carpeta de cada uno de los experimentos, en el “informe.txt” se encuentra el puntaje BLEU obtenido junto con la puntuación BLEU de cada dato del conjunto test, el archivo “respuestas.txt” contiene la descripción de la tabla generada por el modelo, cada salto de línea es una descripción generada, el archivo “originales.txt” contiene la salida del modelo completa con la plantilla del prompt, al aislar solo la descripción de esta se genera “respuestas.txt”. El archivo “references.txt” contiene las referencias de los datos de test cargadas desde el dataset.

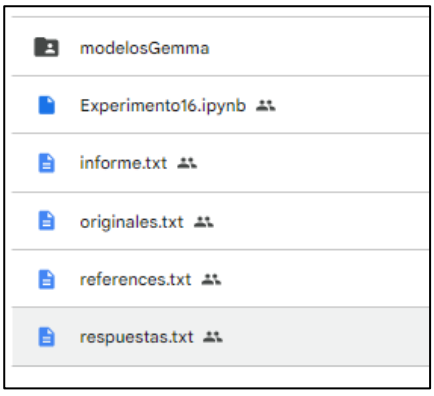


Figura 37. Archivos generados para el informe de BLEU.

En la **Tabla 17** se muestra como ejemplo las cinco primeras salidas del mejor modelo obtenido en el experimento 16, con la puntuación BLEU de cada una de las salidas, estos datos fueron tomados del archivo “respuestas.txt” para los ouput del modelo y el “informe.txt” donde están registrados las puntuaciones BLEU obtenidas de forma individual, la puntuación BLEU 2 n-gramas es de 65,470, la de 3 n-gramas es 58,954 y la de 4 n-gramas es 53,722, esta información se obtuvo del cuaderno del experimento 16[38].

Tabla 17. Muestra de las 5 primeras salidas con su puntaje BLEU del mejor modelo.

No	Origen	Ouput del modelo	Puntuación BLEU
1	**Input**	| Personaje Histórico ||---|| Simón Bolívar, libertador de varias naciones sudamericanas |	80,85
	Ouput del modelo	La tabla contiene información sobre un personaje histórico, Simón Bolívar, quien fue libertador de varias naciones sudamericanas.	
2	**Input**	| País | Población (millones) | Esperanza de vida (años) ||---|---|---|| India | 1.380 | 69,4 || Estados Unidos | 332,4 | 78,8 |	78,33
	Ouput del modelo	La tabla muestra información del país, su población en millones y su esperanza de vida en años. El país de India tiene una población de 1.380 millones y su esperanza de vida es de 69,4 años. El país de Estados Unidos tiene una población de 332,4 millones y su esperanza de vida es de 78,8 años.	
3	**Input**	| Aves || --- || Loros || Pingüinos |	43,94
	Ouput del modelo	La tabla muestra aves, nombra los loros y el pingüino.	
4	**Input**	| Ventajas | Desventajas ||---|---|| Flexibilidad y conveniencia | Falta de interacción social || Acceso a recursos globales | Potenciales problemas técnicos || Costos reducidos | Dificultad para establecer conexiones personales |	75,36
	Ouput del modelo	La tabla muestra ventajas y desventajas. Como ventaja es la flexibilidad y la conveniencia, como desventaja tiene la falta de interacción social. Como	

[38] Ruta del cuaderno del experimento 16.

No	Origen	Ouput del modelo	Puntuación BLEU
		ventaja es el acceso a recursos globales, como desventaja pueden surgir problemas técnicos. Como ventaja es el costo reducido, como desventaja se menciona la dificultad para establecer conexiones personales.	
5	**Input**	| Planta | Causas de extinción ||---|---|| Cicuta Virosa | Está experimentando extinción debido a la destrucción del hábitat |	45,16
	Ouput del modelo	La tabla muestra información sobre una planta y sus causas de extinción. La planta es la cicuta virosa, que está experimentando extinción debido a la destrucción del hábitat.	

Además de esta prueba se realizó una prueba al mejor modelo entrenado con el primer dataset, esto para comprobar que tan bien podría generar descripciones de tablas de temáticas con las que no fue entrenado, para lo cual utilizando un script[39] se seleccionó del conjunto test de la segunda versión del dataset 36 datos de manera aleatoria con una semilla de 12345 y se evaluó BLEU con estos datos obteniendo una puntuación de 49,204 de BLEU, por lo que se puede concluir que para lograr descripciones de calidad, las tablas ingresadas deben ser de temáticas contempladas en su entrenamiento, esta prueba, los resultados de esta prueba se encuentra en drive[40].

Finalmente, se planteó una prueba A/B comparando la puntuación obtenida en BLEU y una comparación humana dando como resultado que si la descripción obtenida tiene una puntuación BLEU de más de 50% cuenta además con una puntuación en la evaluación humana alta lo cual demuestra que esta métrica, se puede considerar para esta tarea, algunas limitaciones de la misma son que no puede medir la cantidad de alucinaciones, además que el modelo evaluado mientras menos información contenga la tabla de datos, mejor puntuación BLEU y evaluación humana obtendrá, logrando así realizar la prueba con éxito (véase **Tabla 18**). La **Tabla 18** cuenta con la prueba A/B que se realizó para obtener la comparativa, entre la puntuación BLEU asignada a cada salida del modelo en comparación con la evaluación humana, la cual evaluó 4 tópicos con ayuda de la escala Likert, al obtener la calificación media, de estos cuatro tópicos se logró comparar con BLEU, en donde existe una relación entre una

[39] Script para obtener datos aleatorios
[40] Resultados al evaluar múltiples temáticas.

buena evaluación humana en una salida que haya obtenido una puntuación BLEU superior a 50, por lo que en salidas con una puntuación baja corresponde con una evaluación humana baja, demostrando que la métrica es útil en el presente TIC, aunque la evaluación humana siempre será recomendable.

Tabla 18. Prueba A/B realizada con la comparativa entre la métrica BLEU y la evaluación humana realizada

Item	Input	Partes implicadas en la comparación		Puntuación BLEU	Evaluación Humana		
		Origen	Contenido		Parámetros a evaluar		Valoración Promedio
					Parámetro	Puntaje	
107	| Provincia | Programa ||---|---|| Azuay | Lengua y Literatura || Guayas | Ciencias Naturales |	Ouput del modelo	La tabla muestra las provincias y sus programas. Azuay tiene el programa de Lengua y Literatura. Guayas tiene el programa de Ciencias Naturales.	51,405	Integridad de la información	5	4,75
					Coherencia y fluidez	4	
		Referencia	La tabla muestra las provincias y su programa. Azuay tiene en su programa lengua y literatura. Guayas tiene en su programa ciencias naturales.		Precisión del significado	5	
					Presencia de alucinaciones	5	
3	| Aves || --- || Loros || Pingüinos |	Ouput del modelo	La tabla muestra aves, nombra el loros y el pingüino.	43,949	Integridad de la información	3	3.5
					Coherencia y fluidez	3	
		Referencia	La tabla muestra aves, se nombra a los loros y los pingüinos.		Precisión del significado	3	
					Presencia de alucinaciones	5	
77	| País | Cantidad de especies de aves ||---|---|| India | 1.350 || Colombia | 1.900 || Perú | 1.814 |	Ouput del modelo	La tabla muestra países y la cantidad de especies de aves. India tiene 1.350 especies de aves. Colombia tiene 1.900 especies de aves. Perú tiene 1.814 especies de aves.	69,696	Integridad de la información	5	5
					Coherencia y fluidez	5	
		Referencia	La tabla muestra países y la cantidad de especies de aves. India tiene 1.350.		Precisión del significado	5	

Item	Input	Partes implicadas en la comparación		Puntuación BLEU	Evaluación Humana		
		Origen	Contenido		Parámetros a evaluar		Valoración Promedio
					Parámetro	Puntaje	
			Colombia tiene 1.900. Perú tiene 1.814.		Presencia de alucinaciones	5	
95	| Elemento de la Tabla Periódica | Propiedades ||---|---|| Helio | Gas inerte, no inflamable, no tóxico || Carbono | Sólido, no metálico, puede formar enlaces covalentes || Hierro | Metal de transición, magnético, conductor de electricidad |	Ouput del modelo	La tabla presenta información sobre los elementos de la Tabla Periódica: su nombre, su clasífica y sus propiedades. El helio es un gas inerte, no inflamable y no tóxico. El carbono es un no metal, sólido y no metálico y puede formar enlaces covalentes. El hierro es un metal de transición, magnético y conductor de electricidad.	57,132	Integridad de la información	4	4,5
					Coherencia y fluidez	5	
		Referencia	Esta tabla enumera tres elementos de la Tabla Periódica y sus propiedades más destacadas. La primera fila describe el helio como un gas inerte, no inflamable, no tóxico. La segunda fila destaca el carbono como un sólido no metálico capaz de formar enlaces covalentes. Finalmente, la tercera fila presenta el hierro como un metal de transición con propiedades magnéticas y conductoras de electricidad .		Precisión del significado	4	
					Presencia de alucinaciones	5	
50	| Personaje histórico | Fecha de fallecimiento ||---|---|| Martin Luther King Jr. | 4 de abril de 1968 || Mahatma Gandhi | 30 de enero de 1948 |	Ouput del modelo	La tabla muestra personajes históricos y la fecha en que fallecieron. Martin Luther King Jr. falleció el 4 de abril de 1968. Mahatma Gandhi falleció el 30 de enero de 1948.	86,540	Integridad de la información	5	5
					Coherencia y fluidez	5	
		Referencia	La tabla muestra personajes históricos y la fecha en que fallecieron. Martin		Precisión del significado	5	

Item	Input	Partes implicadas en la comparación		Puntuación BLEU	Evaluación Humana		
		Origen	Contenido		Parámetros a evaluar		Valoración Promedio
					Parámetro	Puntaje	
			Luther King Jr. falleció el 4 de abril de 1968. Mahatma Gandhi falleció el 30 de enero de 1948.		Presencia de alucinaciones	5	
70	| Marca | Modelo | Año ||---|---|---|| BMW | M3 | 2023 || Tesla | Model S | 2022 || Toyota | Camry | 2021 |	Ouput del modelo	La tabla muestra información de marcas, modelos y años. La marca BMW tiene el modelo M3 y en el año 2023. La marca Tesla tiene el modelo Model S y en el año 2022. La marca Toyota tiene el modelo Camry y en el año 2021.	39,215	Integridad de la información	5	4,75
					Coherencia y fluidez	5	
		Referencia	La tabla contiene información sobre tres modelos de automóviles fabricados por diferentes marcas: un BMW M3 fabricado en 2023, un Tesla Model S fabricado en 2022 y un Toyota Camry fabricado en 2021.		Precisión del significado	4	
					Presencia de alucinaciones	5	
145	| País | Población (millones) | Superficie (kilómetros cuadrados) ||---|---|---|| Lituania | 2,79 | 65.300 || Luxemburgo | 0,64 | 2.586 || Malta | 0,54 | 316 |	Ouput del modelo	La tabla muestra información de países, su población en millones y su superficie en kilómetros cuadrados. El país de Lituania tiene una población de 2,79 millones y una superficie de 65.300 kilómetros cuadrados. El país de Luxemburgo tiene una población de 0,64 millones y una superficie de 2.586 kilómetros cuadrados. El país de Malta tiene una población de 0,54 millones y una superficie de 316 kilómetros cuadrados.	80,555	Integridad de la información	5	5
					Coherencia y fluidez	5	

Item	Input	Partes implicadas en la comparación		Puntuación BLEU	Evaluación Humana		
		Origen	Contenido		Parámetros a evaluar		Valoración Promedio
					Parámetro	Puntaje	
		Referencia	La tabla muestra información de países, su población en millones y su superficie en kilómetros cuadrados. Lituania tiene una población de 2,79 millones y una superficie de 65.300 kilómetros cuadrados. Luxemburgo tiene una población de 0,64 millones y una superficie de 2.586 kilómetros cuadrados. Malta tiene una población de 0,54 millones y una superficie de 316 kilómetros cuadrados.		Precisión del significado	5	
					Presencia de alucinaciones	5	
112	| País | Anfibios | Plantas vasculares ||---|---|---|| Uruguay | 96 | 2.512 || Venezuela | 324 | 11.800 |	Ouput del modelo	La tabla muestra el número de especies de anfibios y plantas vasculares. Uruguay tiene 96 especies de anfibios y 2.512 especies de plantas vasculares. Venezuela tiene 324 especies de anfibios y 11.800 especies de plantas vasculares.	90,221	Integridad de la información	5	5
					Coherencia y fluidez	5	
		Referencia	La tabla muestra el número de especies de anfibios y plantas vasculares. Uruguay tiene 96 especies de anfibios y 2.512 especies de plantas vasculares. Venezuela tiene 324 especies de anfibios y 11.800 especies de plantas vasculares.		Precisión del significado	5	
					Presencia de alucinaciones	5	
42	| Experimento | Año ||---|---|| Experimento de Michelson-Morley | 1887 || Experimento de Stern-Gerlach | 1922 |	Ouput del modelo	La tabla muestra experimentos y los años. El experimento de Michelson-Morley se realizó en el año 1887. El experimento de Stern-Gerlach se realizó en el año 1922.	45,384	Integridad de la información	5	4,75
					Coherencia y fluidez	5	

Item	Input	Partes implicadas en la comparación		Puntuación BLEU	Evaluación Humana		
		Origen	Contenido		Parámetros a evaluar		Valoración Promedio
					Parámetro	Puntaje	
		Referencia	La primera fila de la tabla muestra el Experimento de Michelson-Morley, que se realizó en el año 1887. La segunda fila de la tabla muestra el Experimento de Stern-Gerlach, que se realizó en el año 1922.		Precisión del significado	4	
					Presencia de alucinaciones	5	
96	| Cantidad de turistas ||---|| Perú en 2021, que fueron 2,4 millones |	Ouput del modelo	La tabla muestra el número de turistas que visitaron Perú en 2021, que fueron 2,4 millones.	79,945	Integridad de la información	5	5
					Coherencia y fluidez	5	
		Referencia	La tabla muestra el número de turistas que visitaron Perú en 2021, que fueron 2,4 millones.		Precisión del significado	5	
					Presencia de alucinaciones	5	
32	| Interacción entre plantas y animales ||:---|| Polen que las abejas transfieren de flor a flor || Plantas que ofrecen néctar a los polinizadores || Árboles que proporcionan hábitat a las aves |	Ouput del modelo	La tabla muestra interacciones entre plantas y animales. El polen que las abejas transfieren de flor a flor. Plantas que ofrecen néctar a los polinizadores. Árboles que proporcionan hábitat a las aves.	69,003	Integridad de la información	5	5
					Coherencia y fluidez	5	
					Precisión del significado	5	
		Referencia	La tabla describe interacciones entre plantas y animales como el Polen que las abejas transfieren de flor a flor, plantas que ofrecen néctar a los		Presencia de alucinaciones	5	

Item	Input	Partes implicadas en la comparación		Puntuación BLEU	Evaluación Humana		
		Origen	Contenido		Parámetros a evaluar		Valoración Promedio
					Parámetro	Puntaje	
			polinizadores y árboles que proporcionan hábitat a las aves.				
111	| Alumnos || --- || Juan José Rodríguez || María Fernanda Pérez |	Ouput del modelo	La tabla muestra alumnos como Juan José Rodríguez y María Fernanda Pérez.	44,895	Integridad de la información	4	4,5
					Coherencia y fluidez	5	
		Referencia	La tabla contiene los nombres completos de los alumnos Juan José Rodríguez y María Fernanda Pérez.		Precisión del significado	4	
					Presencia de alucinaciones	5	
67	| Misión espacial | Agencia espacial ||---|---|| Apolo 11 | NASA || Shenzhou 5 | CNSA |	Ouput del modelo	La tabla muestra misión espacial y sus respectivas agencias espaciales. La misión espacial Apolo 11 fue realizada por la NASA. La misión espacial Shenzhou 5 fue realizada por la CNSA.	50,555	Integridad de la información	5	4,75
					Coherencia y fluidez	5	
		Referencia	La tabla muestra dos misiones espaciales significativas y sus agencias espaciales correspondientes. La primera fila indica la misión Apolo 11 llevada a cabo por la NASA. La segunda fila muestra la misión Shenzhou 5, realizada por la CNSA.		Precisión del significado	4	
					Presencia de alucinaciones	5	
144	| Región Geográfica | Especies Endémicas | Amenazas ||---|---|---|| Arrecife de Coral de Mesoamérica | Coral Elkhorn, Coral Staghorn |	Ouput del modelo	La tabla muestra información de una región geográfica, las especies endémicas y las amenazas. La región geográfica es Mesoamérica, las especies endémicas son el coral de mesoamérica, el coral elkhorn y el	48,979	Integridad de la información	4	4,5
					Coherencia y fluidez	5	

Item	Input	Partes implicadas en la comparación		Puntuación BLEU	Evaluación Humana		Valoración Promedio
		Origen	Contenido		Parámetros a evaluar		
					Parámetro	Puntaje	
	Sobrepesca, Cambio Climático |		coral staghorn, las amenazas son la sobrepesca y el cambio climático.				
		Referencia	La tabla resume información sobre una región geográfica del arrecife de Coral de Mesoamérica, resaltando las especies endémicas únicas, como el Coral Elkhorn y el Coral Staghorn, así como las amenazas que enfrentan, como la sobrepesca y el cambio climático		Precisión del significado	4	
					Presencia de alucinaciones	4	
161	| Nombre completo || ---------- || Ana María González Rodríguez |	Ouput del modelo	La tabla muestra el nombre completo de una persona, es Ana María González Rodríguez.	62,5	Integridad de la información	5	5
					Coherencia y fluidez	5	
		Referencia	La tabla presenta el nombre completo de Ana María González Rodríguez.		Precisión del significado	5	
					Presencia de alucinaciones	5	
45	| Planta | Causas de extinción ||---|---|| Santalum fernandezianum | Posible deforestación, sobrepastoreo, intrusión de especies exóticas |	Ouput del modelo	La tabla muestra información sobre una planta y las causas de su extinción. El Santalum fernandezianum tiene como causas de extinción la posible deforestación, el sobrepastoreo y la intrusión de especies exóticas.	65,714	Integridad de la información	5	5
					Coherencia y fluidez	5	
		Referencia	La tabla muestra la planta Santalum fernandezianum y su causa de extinción, que incluyen una posible deforestación, el sobrepastoreo y la intrusión de especies exóticas.		Precisión del significado	5	
					Presencia de alucinaciones	5	

Item	Input	Partes implicadas en la comparación		Puntuación BLEU	Evaluación Humana		
		Origen	Contenido		Parámetros a evaluar		Valoración Promedio
					Parámetro	Puntaje	
157	| Nombres | Apellidos | Nacionalidad ||-----|------|-------|| Camila | Perez | Argentina || Carlos | Ramirez | Chilena |	Ouput del modelo	La tabla muestra información de nombres, apellidos y nacionalidad. El nombre es Camila, el apellido Perez y la nacionalidad argentina. El nombre es Carlos, el apellido Ramirez y la nacionalidad chilena.	79,750	Integridad de la información	4	4,5
					Coherencia y fluidez	5	
		Referencia	La tabla contiene información de nombres, apellidos y nacionalidades. El nombre es Camila, el apellido Pérez y la nacionalidad argentina. El nombre es Carlos, el apellido Ramírez y la nacionalidad chilena.		Precisión del significado	4	
					Presencia de alucinaciones	5	
142	| Nombre de la Especie | Causas de la Amenaza | Medidas de Conservación ||---|---|---|| Águila calva | Pérdida de hábitat, contaminación, uso de pesticidas | Protección y restauración del hábitat, prohibición del uso de DDT || Lechuza moteada	Ouput del modelo	La tabla muestra información de nombres de especies, las causas de la amenaza y las medidas de conservación. El nombre de la especie es el águila calva, las causas de la amenaza son la pérdida	03,728	Integridad de la información	1	2,75
					Coherencia y fluidez	1	

Item	Input	Partes implicadas en la comparación		Puntuación BLEU	Evaluación Humana		
		Origen	Contenido		Parámetros a evaluar		Valoración Promedio
					Parámetro	Puntaje	
	del norte | Tala de bosques, fragmentación del hábitat | Protección y gestión de los bosques, reintroducción de lechuzas || Guacamayo jacinto | Tráfico ilegal de vida silvestre, pérdida de hábitat | Creación de zonas protegidas, programas de cría en cautiverio |	Referencia	La tabla proporciona información sobre nombre de especies, sus causas de amenaza y las medidas de conservación implementadas para protegerlas. La primera fila describe el Águila calva, las causas de su declive (pérdida de hábitat, contaminación, uso de pesticidas) y las medidas de conservación (protección y restauración del hábitat, prohibición del DDT). La segunda fila se centra en la Lechuza moteada del norte, las amenazas que enfrenta (tala de bosques, fragmentación del hábitat) y las medidas de conservación (protección y gestión de bosques, reintroducción de lechuzas). La última fila aborda el Guacamayo jacinto, las causas de su amenaza (tráfico ilegal de vida silvestre, pérdida de hábitat) y las medidas de conservación (creación de zonas protegidas, programas de cría en cautiverio).		Precisión del significado	1	
					Presencia de alucinaciones	5	
	Puntuación promedio			**74,009**			**4,625**

6.2.3. Elegir y empaquetar el modelo.

En cada iteración del experimento se guardó el mejor modelo obtenido por lo que solo se requirió elegirlo de entre las comparativas anteriores, al momento de guardarlo de manera empaquetada se guardan los archivos que se muestra en la **Figura 38,** estos archivos se encuentran dentro de cada carpeta de los experimentos realizados en una carpeta de nombre "gemma-descripcion-tablas-markdown", además de esta carpeta también se guardaron los dos mejores checkpoint durante el entrenamiento.

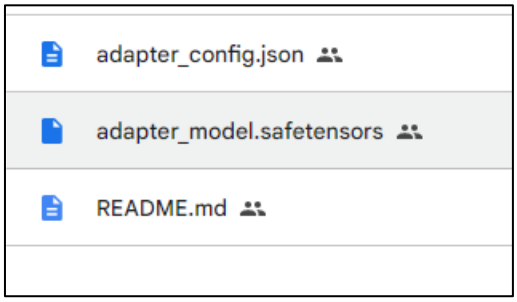


Figura 38. Archivos que el modelo guarda

Finalmente, para consumirlo se debe cargar el modelo base y los archivos de la imagen **Figura 38**, como se muestra en la **Figura 39** se carga el modelo base Gemma 2B y se carga los archivos de ajuste colocando la ruta donde se encuentran en la variable "new_model" de la imagen y listo. Con esto es posible consumir el modelo.

```
base_model = AutoModelForCausalLM.from_pretrained(
    model_name,
    low_cpu_mem_usage=True,
    return_dict=True,
    torch_dtype=torch.float16,
    device_map=device_map,
)
model = PeftModel.from_pretrained(base_model, new_model)
model = model.merge_and_unload()

# Reload tokenizer to save it
# Recarga el tokenizador para guardarlo
tokenizer = AutoTokenizer.from_pretrained(model_name, trust_remote_code=True)
tokenizer.pad_token = tokenizer.eos_token
tokenizer.padding_side = "right"
```

```
Loading checkpoint shards: 100% 2/2 [00:15<00:00,  6.58s/it]
```

Figura 39. Forma de cargar el modelo ajustado.

Al constatar los resultados obtenidos el experimento 16 (véase **Tabla 16**) muestra la mejor puntuación BLEU obtenida, con 74,009 en el conjunto de test, además con la evaluación humana realizada durante la prueba A/B (véase **Tabla 18**), se considera al modelo obtenido en este experimento como el resultado del presente TIC, está empaquetado junto a los demás experimentos en drive [41].

[41] Modelo resultado del presente TIC.

7. Discusión

7.1 Objetivo 1: Ajustar el modelo Gemma para que sea capaz de describir tablas de datos, las cuales estarán en un formato Markdown, un tamaño determinado con un máximo de 3*3, sin tomar en cuenta signos poco usados como los presentes en fórmulas matemáticas, mediante una metodología basada en CRISP-ML(Q).

Para poder realizar el ajuste del modelo Gemma un paso crucial fue la creación del dataset, por lo que primeramente se determinaron las temáticas que se abordarían, en los datasets utilizados en los trabajos relacionados se pone en foco que muestras más diverso sea el contenido el rendimiento del modelo bajará [14], por lo que la mayoría de estos solo abordan temas determinados, para el dataset creado se determinaron 5 temáticas (véase **Sección 5.2.1**), al contrario que datasets como Wikibio utilizado en TR01 el cual aborda información extraída de Wikipedia, con la contraparte que estos contaban con una cantidad de datos muy superior a las dos versiones de un conjunto (véase **Tabla 9**), creadas durante este TIC, no fue posible hacer uso de datasets de los tabajos relacionados debido a que estos estuvieron en el idioma inglés, no contaban con la estructura de la tabla en Markdown, y no realizaban la tarea requerida, pues en comparación en el dataset del TR04 solo se disponía de descripciones parciales de la tabla de datos, por lo que se ha aportado, con las versiones del dataset, con las tablas en una estructura sencilla como es Markdown y sus descripciones.

Al realizar el proceso de creación del dataset, se obtuvieron dos versiones del mismo (véase Tabla 9), las cuales se requirieron inicialmente para realizar un ajuste de prueba con una cantidad de datos pequeña para verificar, si el modelo Gemma obtenía buenos resultados en español pues en [5] se detalla que para su entrenamiento inicialmente se ocupó datos en su mayoría en inglés. Al realizar los experimentos de ajustes iniciales con la primera versión del dataset (véase Tabla 9), se pudo comprobar que el modelo con un entrenamiento de 5 incluso 1 época fue capaz de generar descripciones, aunque de baja calidad, esto se comprueba con la puntuación BLEU de los experimentos con una baja cantidad de épocas (véase Tabla 14). El entorno de entrenamiento fue él una GPU NVIDIA Tesla T4 otorgado en Google Colab, la cual para el método ocupado PEFT el cual ajusta un número limitado de parámetros del LLM fue suficiente para realizar con éxito el ajuste, en comparación con trabajos relacionados como TR03, TR06 en los cuales para ajustar todos los parámetros de los modelos, requirieron usar clústeres de GPUs. En cuanto al uso de LLMs para fines como el planteado en este TIC el TR04 señala que para esta tarea los LLM han sido poco explorados, además que para que un modelo pueda llevarla a cabo se necesita un nivel alto en razonamiento siendo esta otra razón para utilizar el modelo Gemma el cual en el TR08 muestra que este modelo destaca en esta capacidad.

Al realizar el entrenamiento otro aspecto a tener en cuenta es que en los LLM regularmente se monitorea durante el entrenamiento el Loss, por lo que es la misma métrica que se toma en cuenta para guarda el mejor modelo, por ejemplo, en TR02 se aclara que la métrica tomada en cuenta para determinar el mejor modelo es el Loss, por lo que en los experimentos llevados a cabo se tomó esta medida en consideración además de monitorear el BLEU, con esta hubo complicaciones debido a que la clase utilizada para realizar el ajuste "SFTTrainer" no cuente con un parámetro específico que se permita la generación de salidas de manera ordenada para poder realizar la evaluación y seguimiento de la métrica BLEU en cada step del entrenamiento con ayuda del método compute_metric(), por lo que se tuvo que adaptar y monitoria BLEU por cadenas del tamaño 100, cabe recalcar que los modelos con mejor rendimiento fueron aquellos que se guardó monitoreando BLEU.

7.2 Objetivo 2: Evaluar mediante la métrica BLEU los distintos modelos obtenidos al usar hiperparámetros diferentes, obteniendo aquel que mejor logre describir tablas de datos en formato Markdown.

Al realizar la evaluación se pudo observar que el número de épocas en los resultados de los experimentos variaron, pero una mayor cantidad de época no siempre significó una mejor puntuación BLEU en el conjunto de test, esto se apreció en los experimentos que activaban la paciencia, incluso con una paciencia de 20 o 40, que quería decir que si en 20 o 40 evaluaciones la puntuación de la métrica seccionada no mejoraba, el entrenamiento se terminaba, en este sentido se tuvo los mejores resultados en las primeras épocas, y en cuento a la evaluación estas se realizaban cada cierto número de steps, cantidad que influyo en el rendimiento del modelo, pues aquellos que se evaluaban cada 10 steps superaban el rendimiento obtenido de aquellos que se evaluaban cada 100 steps, las tareas de procesamiento de lenguaje natural tienen una dificultad añadida, pues las métricas no son lo suficientemente precisas para determinar completamente su utilidad, por lo que la evaluación humana es primordial. En el presente TIC para responder la pregunta de investigación, se logró una puntuación de 74,009 con la métrica BLEU, y buenos resultados con la evaluación humana, en donde se logró un promedio de 4,625 sobre 5 en escala Linkert en los cuatro aspectos evaluados: integridad de la información, coherencia y fluidez, precisión del significado y presencia de alucinaciones. Mediante la prueba A/B se demostró que un alto puntaje BLEU significa una calidad considerable aún en la evaluación humana, existe una relación en cuanto a puntuaciones BLEU mayores a 50 con una nota en la evaluación humanan elevada, hay que tomar en cuenta que cuando la salida del modelo no daba la información completa esperada, BLEU sancionaba fuertemente su puntaje. Además, se obtienen mejores resultados mientras menor sea la información contenida en la tabla de datos, esto puede llegar a corregirse aumentando el número de datos.

8. Conclusiones

- Al ajustar el modelo Gemma fue posible obtener una puntuación BLEU de 74,009, la misma que se consiguió al evaluar el conjunto de test, el cual cuenta con el 10% de los datos, de la segunda versión del dataset construido el cual posee 1800 datos en total, por lo cual se lo considera como el mejor modelo obtenido, para la tarea de descripción de tablas de datos en un formato Markdown, con un tamaño determinado máximo de 3 filas por 3 columnas, este modelo logra generar descripciones capaces de explicar el contenido de estas.
- Utilizando una metodología basada en CRISP-ML(Q) fue posible realizar el ajuste del modelo Gemma, las fases de comprensión de datos, la ingeniería de datos e ingeniería de modelos, fueron fundamentales en el proceso, pues con ellas se logró la creación de dos versiones de un dataset, variando entre ellos el número de temáticas del contenido de las tablas de datos abordadas, dichos datos ya se encuentra validados y normalizados para el modelo Gemma, el cual fue ajustado con el método PEFT y LoRa que permitieron realizar diversos experimentos del ajuste en un entorno de limitados recursos computacionales como los es Google Colab.
- Finalmente para evaluar los diferentes modelos obtenidos de los experimentos realizados en el objetivo específico 1 variando hiperparámetros y configuración de la clase usada para el ajuste del modelo, se utilizó el conjunto de test con el cual se calculó la puntuación BLEU, obtenida en cada modelo, donde se observó que el entrenamiento realizado con un mayor número de temáticas en las tablas de datos, obtuvo una puntuación BLEU más alta, y la variación de hiperparámetros obtuvo leves mejoras en la puntuación final obtenida. Además, con un test A/B se pudo constar que una alta puntuación BLEU puede ser evaluada como una buena descripción en una evaluación humana, esto se comprobó al evaluar el 10% de los datos de test tomados de manera randómica con una semilla en el test A/B comparando la puntuación BLEU de 74,009 con una puntuación de 4,625 en la escala Likert.

9. Recomendaciones

- Para mejorar la puntuación BLEU de las descripciones de tablas de datos en Markdown, es necesario expandir el número de datos del dataset, además que estos datos sean de temáticas diversas, pues esto ayudará al momento de entrenar, evitar el sobre ajuste, permitiendo obtener una puntuación BLEU más alta, para la creación de datos se recomienda el uso de herramientas de IA como Géminis, llamados también datos sintéticos.
- Para entornos con capacidad computacional limitada se recomienda el método PEFT y LoRa. En el modelo Gemma se recomienda empezar a experimentar con los métodos antes mencionados, pues al tener un número muy elevado de parámetros no es posible un ajuste completo en este tipo de entornos.
- Al entrenar el modelo se recomienda limitar la cantidad de checkpoint que pueda guardar el modelo, para evitar que se llene nuestro almacenamiento, además es recomendable declarar que la validación se realice con un número pequeño de pasos, pues con esto se puede obtener un modelo con mejores valores en la puntuación BLEU y Loss.
- Se recomienda utilizar este TIC como punto de partida en la investigación de las capacidades de procesamiento de tablas de datos en formato Markdown en modelos Gemma ajustados, pues si bien los resultados fueron positivos se pone en consideración que: al evaluar con temáticas nuevas, o tablas con una estructura diferente su rendimiento y puntuación BLEU en sus salidas puede disminuir.

Limitaciones

- La cantidad de datos sintéticos creados, se ve limitada por el tiempo del periodo académico, pues los datos creados con herramientas, aunque son de ayuda requieren de una revisión humana para verificar que cumplan con las necesidades a resolver en este TIC.
- Los valores de los hiperparámetros fueron escogidos en gran medida tomando en cuentas las limitaciones de cálculo computacional presente en el entorno donde se realizó el ajuste del modelo que en este caso fue Google Colab.

Trabajos futuros.

- Incrementar el número de datos en el dataset junto con las temáticas tratadas en las tablas de datos, evaluar utilizando otras métricas, explorar otros hiperparámetros al implementar LoRa.

10. Bibliografía

[1] J. D. Castellanos Bonilla, “Evaluación de modelos de lenguaje de gran tamaño (LLM) en ciberseguridad,” Jun. 14, 2024, *Universidad de los Andes*. Accessed: Jul. 13, 2024. [Online]. Available: https://hdl.handle.net/1992/74392

[2] “Seguimiento a la ejecución física del Plan Estratégico Institucional 2019-2023 Período: Julio -Diciembre 2022”.

[3] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, “Revisiting Deep Learning Models for Tabular Data,” *Adv Neural Inf Process Syst*, vol. 34, pp. 18932–18943, Dec. 2021, Accessed: Jul. 13, 2024. [Online]. Available: https://github.com/yandex-research/rtdl.

[4] M. Mostafavi Ghahfarokhi, H. Jahantigh, A. Asadi, S. Kianiangolafshani, A. Khademian, and A. Heydarnoori, “A Roadmap for Enriching Jupyter Notebooks Documentation with Kaggle Data,” pp. 271–272, Apr. 2024, doi: 10.1145/3644815.3644984.

[5] Gemma Team *et al.*, “Gemma: Open Models Based on Gemini Research and Technology,” Mar. 2024.

[6] “Estadísticas de Discapacidad – Consejo Nacional para la Igualdad de Discapacidades.” Accessed: Jun. 18, 2024. [Online]. Available: https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/

[7] “CUADRO BASE ESTUDIANTES CON DISCAPACIDAD ABRIL OCTUBRE 2022-signed.pdf - Google Drive.” Accessed: Jun. 26, 2024. [Online]. Available: https://drive.google.com/file/d/1_E9TXaDJ3hygAmMOlqfTOJhAzJhXlOcJ/view

[8] T. Liu, K. Wang, L. Sha, B. Chang, and Z. Sui, “Table-to-Text Generation by Structure-Aware Seq2seq Learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018, doi: 10.1609/aaai.v32i1.11925.

[9] R. Ye, W. Shi, H. Zhou, Z. Wei, and L. Li, “VARIATIONAL TEMPLATE MACHINE FOR DATA-TO-TEXT GENERATION”.

[10] Y. Zhao, H. Zhang, S. Si, L. Nan, X. Tang, and A. Cohan, “Investigating Table-to-Text Generation Capabilities of LLMs in Real-World Information Seeking Scenarios,” pp. 160–175.

[11] Y. Su, Z. Meng, S. Baker, and N. Collier, “Few-Shot Table-to-Text Generation with Prototype Memory,” pp. 910–917, Accessed: May 01, 2024. [Online]. Available: https://lucene.apache.org/core/

[12] Y. Zhao, L. Nan, Z. Qi, R. Zhang, and D. Radev, “REASTAP: Injecting Table Reasoning Skills During Pre-training via Synthetic Reasoning Examples,” pp. 9006–9018, Accessed: May 01, 2024. [Online]. Available: https://github.com/attardi/

[13] F. Wang, Z. Xu, P. Szekely, and M. Chen, “Robust (Controlled) Table-to-Text Generation with Structure-Aware Equivariance Learning,” *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pp. 5037–5048, May 2022, doi: 10.18653/v1/2022.naacl-main.371.

[14] A. P. Parikh *et al.*, “ToTTo: A Controlled Table-To-Text Generation Dataset,” *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1173–1186, Apr. 2020, doi: 10.18653/v1/2020.emnlp-main.89.

[15] J. F. Avila-Tomás, M. A. Mayer-Pujadas, and V. J. Quesada-Varela, “La inteligencia artificial y sus aplicaciones en medicina I: introducción antecedentes a la IA y robótica,” *Aten Primaria*, vol. 52, no. 10, pp. 778–784, Dec. 2020, doi: 10.1016/J.APRIM.2020.04.013.

[16] J. Cesar *et al.*, “Inteligencia artificial,” 2014, Accessed: May 01, 2024. [Online]. Available: https://rephip.unr.edu.ar/bitstream/handle/2133/17686/1520250496_Inteligencia-Artificial-CC-BY-SA-3.0-86.pdf?se

[17] U. Y. Sociedad *et al.*, “Inteligencia artificial y propiedad intelectual,” *Universidad y Sociedad*, vol. 13, no. S3, pp. 362–368, Dec. 2021, Accessed: May 06, 2024. [Online]. Available: https://rus.ucf.edu.cu/index.php/rus/article/view/2490

[18] Y. Kang, Z. Cai, C. W. Tan, Q. Huang, and H. Liu, “Natural language processing (NLP) in management research: A literature review,” *Journal of Management Analytics*, vol. 7, no. 2, pp. 139–172, Apr. 2020, doi: 10.1080/23270012.2020.1756939.

[19] H. T. Kesgin and F. Amasyali, “Advancing NLP models with strategic text augmentation: A comprehensive study of augmentation methods and curriculum strategies ☆,” *Natural Language Processing Journal*, vol. 7, p. 100071, 2024, doi: 10.1016/j.nlp.2024.100071.

[20] C. Maciejewski *et al.*, “AssistMED project: Transforming cardiology cohort characterisation from electronic health records through natural language processing – Algorithm design, preliminary results, and field prospects,” *Int J Med Inform*, vol. 185, p. 105380, May 2024, doi: 10.1016/J.IJMEDINF.2024.105380.

[21] U. Kamath, J. Liu, and J. Whitaker, *Deep learning for NLP and speech recognition*. Springer International Publishing, 2019. doi: 10.1007/978-3-030-14596-5/COVER.

[22] J. Yang *et al.*, “Article 160. 2024. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond,” *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 6, p. 32, 2024, doi: 10.1145/3649506.

[23] A. Alajrami and N. Aletras, “How does the pre-training objective affect what large language models learn about linguistic properties?,” vol. 2, pp. 131–147, Accessed: May 01, 2024. [Online]. Available: https://github.com/huggingface/datas

[24] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, “A Survey on Large Language Model (LLM) Security and Privacy: The Good, The Bad, and The Ugly,” *High-Confidence Computing*, vol. 4, no. 2, p. 100211, Jun. 2024, doi: 10.1016/j.hcc.2024.100211.

[25] A. Natallia, T. Tutor, J. Antonio, and P. Ortiz, “Estudio de la arquitectura neuronal transformer y comparación entre el mecanismo de atención multicabezal frente al multiconsulta,” Jul. 2024, Accessed: Jul. 31, 2024. [Online]. Available: http://rua.ua.es/dspace/handle/10045/145624

[26] A. Vaswani *et al.*, “Attention Is All You Need,” 2023.

[27] N. S. Google, “Fast Transformer Decoding: One Write-Head is All You Need,” Nov. 2019, Accessed: May 01, 2024. [Online]. Available: https://arxiv.org/abs/1911.02150v1

[28] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, “RoFormer: Enhanced transformer with Rotary Position Embedding,” *Neurocomputing*, vol. 568, p. 127063, Feb. 2024, doi: 10.1016/J.NEUCOM.2023.127063.

[29] N. S. Google, “GLU Variants Improve Transformer,” Feb. 2020, Accessed: May 01, 2024. [Online]. Available: https://arxiv.org/abs/2002.05202v1

[30] B. Zhang and R. Sennrich, “Root Mean Square Layer Normalization,” *Adv Neural Inf Process Syst*, vol. 32, 2019, Accessed: May 01, 2024. [Online]. Available: https://github.com/bzhangGo/rmsnorm.

[31] Y. Xia *et al.*, “Assessing parameter efficient methods for pre-trained language model in annotating scRNA-seq data,” *Methods*, vol. 228, pp. 12–21, Aug. 2024, doi: 10.1016/j.ymeth.2024.05.007.

[32] Z. Han, C. Gao, J. Liu, J. Zhang, and S. Q. Zhang, “Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey,” Mar. 2024, Accessed: Jun. 18, 2024. [Online]. Available: https://arxiv.org/abs/2403.14608v5

[33] H. Zhou, X. Wan, I. Vulić, and A. Korhonen, “ AutoPEFT : Automatic Configuration Search for Parameter-Efficient Fine-Tuning ,” *Trans Assoc Comput Linguist*, vol. 12, pp. 525–542, May 2024, doi: 10.1162/TACL_A_00662/120914/AUTOPEFT-AUTOMATIC-CONFIGURATION-SEARCH-FOR.

[34] C. Gao *et al.*, “Higher Layers Need More LoRA Experts”.

[35] E. Hu *et al.*, “LORA: LOW-RANK ADAPTATION OF LARGE LAN-GUAGE MODELS”, Accessed: Jul. 19, 2024. [Online]. Available: https://github.com/microsoft/LoRA.

[36] G. Pu, A. Jain, J. Yin, and R. Kaplan, “EMPIRICAL ANALYSIS OF THE STRENGTHS AND WEAKNESSES OF PEFT TECHNIQUES FOR LLMS”.

[37] “The Markdown Guide”.

[38] N. Frey, K. Panovich, and I. Essa, “Automatic Instructional Video Creation from a Markdown-Formatted Tutorial,” p. 14, doi: 10.1145/3472749.3474778.

[39] Y. Li, “I♡ LA: Compilable Markdown for Linear Algebra,” 2021, doi: 10.1145/3478513.3480506.

[40] T. Mailund, “Introducing Markdown and Pandoc,” *Introducing Markdown and Pandoc*, 2019, doi: 10.1007/978-1-4842-5149-2.

[41] B. Baumer and D. Udwin, “R Markdown,” *Wiley Interdiscip Rev Comput Stat*, vol. 7, no. 3, pp. 167–177, May 2015, doi: 10.1002/WICS.1348.

[42] V. Arel-Bundock, “modelsummary: Data and Model Summaries in R,” *J Stat Softw*, vol. 103, no. 1, pp. 1–23, Jul. 2022, doi: 10.18637/JSS.V103.I01.

[43] M. Báez, S. Barredo, L. Stinco, and H. Merlino, “ENTORNO DE GENERACIÓN DE DATOS SINTÉTICOS”.

[44] I. Merino Bermejo, “Estrategias de visión por computador para la estimación de pose en el contexto de aplicaciones robóticas industriales: avances en el uso de modelos tanto clásicos como de Deep Learning en imágenes 2D,” Oct. 2023, Accessed: Jun. 22, 2024. [Online]. Available: http://addi.ehu.es/handle/10810/63969

[45] Y. Lu *et al.*, “Machine Learning for Synthetic Data Generation: A Review”.

[46] S. Studer *et al.*, “Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology,” *Machine Learning and Knowledge Extraction 2021, Vol. 3, Pages 392-413*, vol. 3, no. 2, pp. 392–413, Apr. 2021, doi: 10.3390/MAKE3020020.

[47] “CRISP-ML(Q).” Accessed: Jun. 19, 2024. [Online]. Available: https://ml-ops.org/content/crisp-ml

[48] S. Studer *et al.*, “Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology,” *Machine Learning and Knowledge Extraction 2021, Vol. 3, Pages 392-413*, vol. 3, no. 2, pp. 392–413, Apr. 2021, doi: 10.3390/MAKE3020020.

[49] A. Al-Rukban and A. K. J. Saudagar, “Evaluation of english to Arabic machine translation systems using BLEU and GTM,” *ACM International Conference Proceeding Series*, pp. 228–232, Dec. 2017, doi: 10.1145/3175536.3175570.

[50] P. Homola, V. Kuboň, K. Kuboň, and P. Pecina, “A Simple Automatic MT Evaluation Metric,” pp. 33–36, 2009, doi: 10.5555/1626431.1626436.

[51] C. H. Chiang and H. Y. Lee, “Can Large Language Models Be an Alternative to Human Evaluations?,” *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 15607–15631, May 2023, doi: 10.18653/v1/2023.acl-long.870.

[52] M. Karpinska, N. Akoury, and M. Iyyer, “The Perils of Using Mechanical Turk to Evaluate Open-Ended Text Generation,” *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 1265–1285, Sep. 2021, doi: 10.18653/v1/2021.emnlp-main.97.

[53] E. Clark, T. August, S. Serrano, N. Haduong, S. Gururangan, and N. A. Smith, “All That’s ‘Human’ Is Not Gold: Evaluating Human Evaluation of Generated Text,” *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 7282–7296, 2021, doi: 10.18653/V1/2021.ACL-LONG.565.

[54] M. Guil Bozal, “ESCALA MIXTA LIKERT-THURSTONE”.

[55] S. Margarita, M. Luna, L. María, M. Hinojosa, J. Armando, and P. Moreno, “Manual Práctico Para El Diseño De La Escala Likert,” *Xihmai*, vol. 2, no. 4, Nov. 2007, doi: 10.37646/XIHMAI.V2I4.101.

[56] B. Abeysinghe and R. Circi, “The Challenges of Evaluating LLM Applications: An Analysis of Automated, Human, and LLM-Based Approaches,” Jun. 2024, Accessed: Jul. 17, 2024. [Online]. Available: https://arxiv.org/abs/2406.03339v2

[57] R. Kohavi and R. Longbotham, “Online Controlled Experiments and A/B Tests,” *Encyclopedia of Machine Learning and Data Science*, pp. 1–13, 2023, doi: 10.1007/978-1-4899-7502-7_891-2.

[58] R. Johari, P. Koomen, L. Pekelis, and D. Walsh, “Peeking at A/B Tests: Why it matters, and what to do about it,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. Part F129685, pp. 1517–1525, Aug. 2017, doi: 10.1145/3097983.3097992/SUPPL_FILE/WALSH_PEEKING_TESTS.MP4.

[59] Z. Ye, H. Yoganarasimhan, and Y. Zheng, “LOLA: LLM-Assisted Online Learning Algorithm for Content Experiments,” Jun. 2024, Accessed: Jul. 18, 2024. [Online]. Available: https://arxiv.org/abs/2406.02611v1

[60] G. Dogru, “LARGE LANGUAGE MODELS ‘AD REFER-ENDUM’: HOW GOOD ARE THEY AT MA-CHINE TRANSLATION IN THE LEGAL DO-MAIN? VICENT BRIVA-IGLESIAS”.

[61] E. la información con Python, “Python para todos Explorando la información con Python 3 Charles R. Severance”.

[62] R. G. Duque, “Python PARA TODOS”, Accessed: Jun. 25, 2024. [Online]. Available: http://mundogeek.net/tutorial-python/

[63] D. A. L. Altamirano *et al.*, “Python una escalera para el desarrollo de la inteligencia artificial en el proceso enseñanza y aprendizaje de las matemáticas,” *Dominio de las Ciencias*, vol. 9, no. 4, pp. 363–374, Jun. 2023, doi: 10.23857/DC.V9I4.3594.

[64] E. Stevens, “Deep leanring with PyTorch,” *Manning*, pp. 87–104, 2021.

[65] N. P. Lopes, “Torchy: A Tracing JIT Compiler for PyTorch,” *CC 2023 - Proceedings of the 32nd ACM SIGPLAN International Conference on Compiler Construction*, pp. 98–109, Feb. 2023, doi: 10.1145/3578360.3580266.

[66] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, “SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models.” Accessed: Jun. 25, 2024. [Online]. Available: https://github.com/mit-han-lab/smoothquant

[67] Y. Shen *et al.*, “HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face,” *Adv Neural Inf Process Syst*, vol. 36, pp. 38154–38180, Dec. 2023, Accessed: May 01, 2024. [Online]. Available: https://github.com/microsoft/JARVIS

[68] C. Di Sipio, R. Rubei, J. Di Rocco, D. Di Ruscio, and P. T. Nguyen, “Automated categorization of pre-trained models in software engineering: A case study with a Hugging Face dataset,” *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, pp. 351–356, Jun. 2024, doi: 10.1145/3661167.3661215.

[69] “Hugging Face Hub documentation.” Accessed: Jun. 26, 2024. [Online]. Available: https://huggingface.co/docs/hub/index

[70] F. Pepe, V. Nardone, A. Mastropaolo, G. Bavota, G. Canfora, and M. Di Penta, “How do Hugging Face Models Document Datasets, Bias, and Licenses? An Empirical Study,” *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension*, pp. 370–381, Apr. 2024, doi: 10.1145/3643916.3644412.

[71] S. Wang *et al.*, “PROLORA: Partial Rotation Empowers More Parameter-Efficient LoRA”.

[72] F. Wang, Z. Xu, P. Szekely, and M. Chen, “Structure-Aware Equivariance Learning,” pp. 5037–5048, Accessed: May 01, 2024. [Online]. Available: https://github.com/

11. Anexos

Anexo 1. Entrevista para entender la problemática tratada en el TIC.

Entrevista

Fecha: 29 / 01/ 2024

Nombre: Patricio Paredes

Título: Entrevista al responsable del proyecto de vinculación con la Sociedad "Inclusión lectora de los estudiantes con discapacidad visual de la Universidad Nacional de Loja mediante innovación tecnológica" además especialista en deep learning, redes Transformers, Procesamiento de Lenguaje Natural.

Datos personales del entrevistador:

Nombre: Patricio Oswaldo Paredes Chamba
Cargo: Estudiante de la carrera de Computación en la Universidad Nacional de Loja

Email: patricio.paredes@unl.edu.ec

Celular: 0986665365

Información del entrevistado:

Nombre: Ing. Oscar Cumbicus
Cargo: responsable del proyecto de vinculación con la Sociedad "Inclusión lectora de los estudiantes con discapacidad visual de la Universidad Nacional de Loja mediante innovación tecnológica", Docente de la carrera de Computación / Experto en: PLN y la aplicación de esta en modelos transformers, Deep learning.

Objetivos

- Obtener información valiosa sobre las necesidades presentes del proyecto de vinculación con la sociedad Inclusión lectora.
- Determinar los desafíos técnicos presentes en la problemática de generación de descripciones de tablas de datos.

Presentación

Buenas tardes, mi nombre es Patricio Paredes y me dirijo al responsable del proyecto de vinculación de la carrera de Computación, además experto en campos como: PLN, deep learning. Primeramente, me gustaría agradecer al ingeniero Oscar Cumbicus por tomarse el tiempo para esta entrevista. El propósito de la misma es explorar una problemática específica dentro del proyecto de vinculación con la sociedad "Inclusión lectora de los estudiantes con discapacidad visual de la Universidad Nacional de Loja mediante innovación tecnológica" la cual es la generación de descripciones en base a tablas de datos.

Mi objetivo es que dada la problemática antes mencionada como mi trabajo de titulación pueda contribuir al proyecto, proponiendo una solución a esta problemática.

- **¿Podría proporcionar un resumen general del proyecto de vinculación con la sociedad de la carrera de computación de la Universidad Nacional de Loja de nombre: "Inclusión lectora de los estudiantes con discapacidad visual de la Universidad Nacional de Loja mediante innovación tecnológica"?**
    - El proyecto de vinculación justamente está encaminado a desarrollar un sistema tecnológico que les permita a los estudiantes con discapacidad visual poder, desarrollar de mejor manera la habilidad lectora, este sistema pretende ser una ayuda para todos los estudiantes para que cuando ellos reciban un trabajo por parte de los profesores o quieran hacer alguna consulta, ellos puedan usar este sistema y puedan generar audio de los pdfs o de los documentos ya sean Word, de los profesores o que descarguen de internet para poder realizar sus tareas, con esto pretendemos que se pueda, dar una mayor inclusión lectora de estos estudiantes, dentro de la educación en la universidad nacional de Loja.
- **¿Cómo surgió la problemática de generar texto entendible a partir de tablas de datos?**
    - El problema principal es que algunos lectores de pantalla que ya utilizan las personas con discapacidad visual realizan las lecturas de los textos sin problemas sin problema mientras sean un texto, pero cuando se encuentran con tablas, existe una dificultad, ya que la lectura de las tablas por lo general ese sistema lo hace, en forma de columnas es decir, en la primera columna, si tiene 10 filas , lee la primera columna y las 10 filas, después viene la segunda columna y las 10 filas, pero en realidad la tabla debería ser leída de izquierda a derecha, es decir, fila por fila, primero el encabezado de cada una de las columnas luego los datos que existen en cada una de esas filas, entonces , la problemática principal surge en que en la actualidad las herramientas que ellos

utilizan no realizan una lectura adecuada de las tablas, si no llevan un sentido ordenado y eso les dificulta a las personas con discapacidad visual, comprender el contenido de las tablas, cuando se revisa esta lectura ellos no entienden cómo está estructurada la tabla y que datos tienen cada una de las filas, porque la lectura en realidad se hace por columnas no por filas como debería ser.

- **¿El desafío específico que se enfrentaron al tratar de generar texto de una tabla de datos es nuevo para usted o ya se ha presentado con anterioridad?**
    - Dentro del procesamiento del lenguaje natural que es una rama de la inteligencia artificial, se han tratado de hacer esfuerzos para poder realizar este tipo de lectura no solamente dentro del texto si no través de la web, existen ya algunos intentos de hacer esto, pero el reto es bastante grande ya que se tiene que tomar de base que no todas las tablas son del mismo tipo, no todas van a tener 3 columnas, 10 filas si no que hay diferentes tipos de tablas entonces el reto se vuelve grande pero con la utilización de técnicas de procesamiento de lenguaje natural y la creación de modelos de generación de lenguaje como se ha venido viendo últimamente partiendo desde Bert hasta llegar por ejemplo a GPT o a Bart, ha sido grande así que creemos, estamos convencidos de que podemos o que se podría crear un modelo que permita justamente traducir o entender tablas y luego llevarlas a texto.
- **¿Qué modelos o tecnologías se han utilizado hasta ahora para abordar la problemática identificada?**
    - En la actualidad existen algunos modelos como comentaba en la pregunta anterior de generación de lenguaje natural como por ejemplo GPT que es un modelo multimodal que permite realizar un sin número de tareas, conozco de un modelo de datos denominado Bloom que fue creado por la universidad del País Vasco en España también es un modelo multimodal de generación de lenguaje que también permite realizar de alguna forma la descripción de estas tablas pero lo que se debería hacer en realizar es adaptar uno de estos modelos para los casos específicos que nosotros necesitamos, ya que los modelos creados en la actualidad son generales, es decir no son específicos para un caso lo que se debería hacer es el ajuste de uno de estos modelos para poder crear un modelo para el proyecto de vinculación.
- **¿Los enfoques utilizados para abordar la problemática involucran la extracción de la información de tablas en documentos y la posterior generación de texto basada en dichas tablas mediante un único modelo integral, o son considerados como problemas separados con distintos modelos o procesos?**

- Si se podría abordar como problemáticas distintas, primero un modelo que obtenga la tabla del texto, es decir que primero se haga la extracción de la tabla de texto y luego otro modelo que traduzca esa tabla a texto entonces serían a la par dos modelos diferentes, se podría trabajar con cualquiera de los dos modelos pero la intención es que se puedan crear modelos para cada una de las tareas y luego unificarlos en un modelo multimodal, por eso, la importancia de que se puedan hacer proyectos de integración curricular para cada uno de estos modelos en su defecto que varios alumnos trabajen sobre las problemáticas en un solo proyecto de integración curricular puedan hacer estas dos tareas, se debería hacer un modelo para extraer la tabla y luego para poder hacer la compresión de la tabla y generar el texto.

- **¿Cuáles considera que son los requisitos fundamentales que el modelo de generación de texto en base a tablas de datos debe cumplir para ser una herramienta efectiva y beneficiosa en el proyecto de vinculación con la sociedad?**
    - Bueno las consideraciones es que me permita ingresar una tabla es decir que yodado una tabla voy a traducirla a texto, esa es la primera, ahora hay que tomar en cuenta que la traducción no va a ser exactamente o al 100 por ciento correcta , yo creo que si podemos escribir net un 80 a un 90 por ciento de la tabla correctamente estaríamos llegando al objetivo que es la descripción, pro que hemos probado ya modelo de lenguaje multimodales que igual tienen dificultades, por el mismo hecho de la estructura de las tablas, entonces que nosotros podamos traducir en un 80 un 90 por ciento se considera un éxito, ya que en la actualidad no se hace nada, entonces hacer la descripción de un 80, 90 por ciento a no tener nada es un gran avance, entonces ese es uno de los parámetros importantes que debería cubrir y otros parámetros es que debería hacerse este modelo con tecnologías actuales como Transformers, con modelos de generación de lenguaje, ese también sería un aspecto importante que se debería tomar en cuenta para tener el éxito deseado dentro del proyecto.
- **Si se llegara a crear un modelo para estos requerimientos ¿Cuál considera que serían las métricas con las cuales evaluar su efectividad?**
    - Hay muchas métricas que nosotros podríamos utilizar dentro de la evaluación como es una traducción de una tabla a un texto podríamos utilizar la métrica BLEU, que es una métrica justamente para medir la traducción pero más comúnmente podríamos utilizar la matriz de confusión, es decir dadas 10 tablas es decir cuán bien ha descrito estas tablas, es decir cuantas describió bien y

cuantas describió mal, con la matriz de confusión es decir con el porcentaje global de aciertos del modelo podríamos evaluarlo sin ningún problema, si queremos utilizar métricas más específicas deberemos utilizar BLEU, que también es una métrica para generación de texto , pero podríamos hacer sin ningún problema con el porcentaje global de acierto del modelo , dado un número de entradas cuántas de esas entradas las describió bien y cuántas de esas entradas las describió mal.

- **Para la creación de este modelo uno de los problemas más frecuentes son los datos utilizados para el entrenamiento de dichos modelos. ¿Qué desafíos ve presente en el tema de datasets y que número de datos sería representativo para un modelo como este?**
    - Bueno en la actualidad uno de los mayores problemas del machine learning en cualquier parte es justamente los datos, entonces existen datos o dataset que se podrían utilizar como COC, MNTS que tienen imágenes de forma general, en este caso sería conveniente poder generar un dataset propio con tablas que nosotros podamos extraer de textos, para así tener una validación más acercada a la realidad nuestra , nosotros podríamos presentar o crear un dataset, y luego con ese dataset entrenarlo, la cantidad de datos que tengamos será acorde a como el modelo vaya realizando las predicciones de manera correcta, los modelos de procesamiento de lenguaje natural, el tener más datos no significa tener un mejor resultado, porque todo depende de los datos, si las tablas son de buena calidad podríamos tener más resultados, pero si termos 10000 tabas de mala calidad talvez los resultados sean buenos, entonces lo importante no es el número de dos si no que tan buenos son si la traducción que tengamos de esas tablas son buenas nosotros seguramente tendremos resultados buenos estamos hablando cerca de unos 1000 datos para poder crear un dataset como el que necesitaríamos, pero también se podría ir midiendo de acuerdo a como va funcionando el modelo .
- **En el ámbito de la sostenibilidad ¿Usted considera un punto válido para la sostenibilidad el uso de modelos pre entrenados para partiendo de estos realizar un ajuste poder obtener nuevos modelos permitiendo así un menor número de recursos ya sea electricidad entre otros?**
    - Efectivamente el punto de la sostenibilidad es importante ya que podemos aprovechar todo el conocimiento de modelo grandes como Bart como el propio GPT que son modelos que han sido entrenados con millones desde parámetros, cerca de 500 millones de parámetros, podemos utilizar ya ese

conocimiento lo que se hace en la actualidad entonces dentro de sostenibilidad partir de uno de esos modelos pre entrenados sería muy bueno ya que ganamos mucho entrenamiento tiempo, ahorramos mucho tiempo y podemos aprovechar esas características que talvez nosotros en el ambiente que nos desarrollamos el proyecto de vinculación no lo vamos a tener, entonces me parece muy importante que dentro de la sostenibilidad podamos partir de ese tipo de modelos pre entrenados para que luego nosotros también generemos un modelo pre entrenado que luego otra persona lo pueda utilizar entonces es importante el tema de la sostenibilidad si vamos a utilizar recursos humanos, recursos técnicos y recursos logísticos

Estimado Ingeniero Oscar Cumbicus,

Quiero expresar mi sincero agradecimiento por brindarme la oportunidad de conversar con usted durante la entrevista realizada hoy. Su disposición a compartir información valiosa sobre el proyecto de vinculación "Inclusión lectora de los estudiantes con discapacidad visual de la Universidad Nacional de Loja mediante innovación tecnológica", y los desafíos técnicos que enfrenta la problemática específica de generación de descripciones en base a tablas de datos ha sido de gran utilidad para mi comprensión del proyecto.

Aprecio enormemente su tiempo y la claridad con la que abordó las preguntas planteadas. La información proporcionada será de gran ayuda para orientar mi trabajo de titulación y contribuir de manera efectiva a los objetivos del proyecto.

Atentamente,

Patricio Paredes

Ing. Oscar Miguel Cumbicus Pineda

Entrevistado

Nota: Esta entrevista se encuentra firmada dentro del PIC aprobado, para validarla en el presente documento se comparte el link de YouTube donde se encuentra grabada esta entrevista:

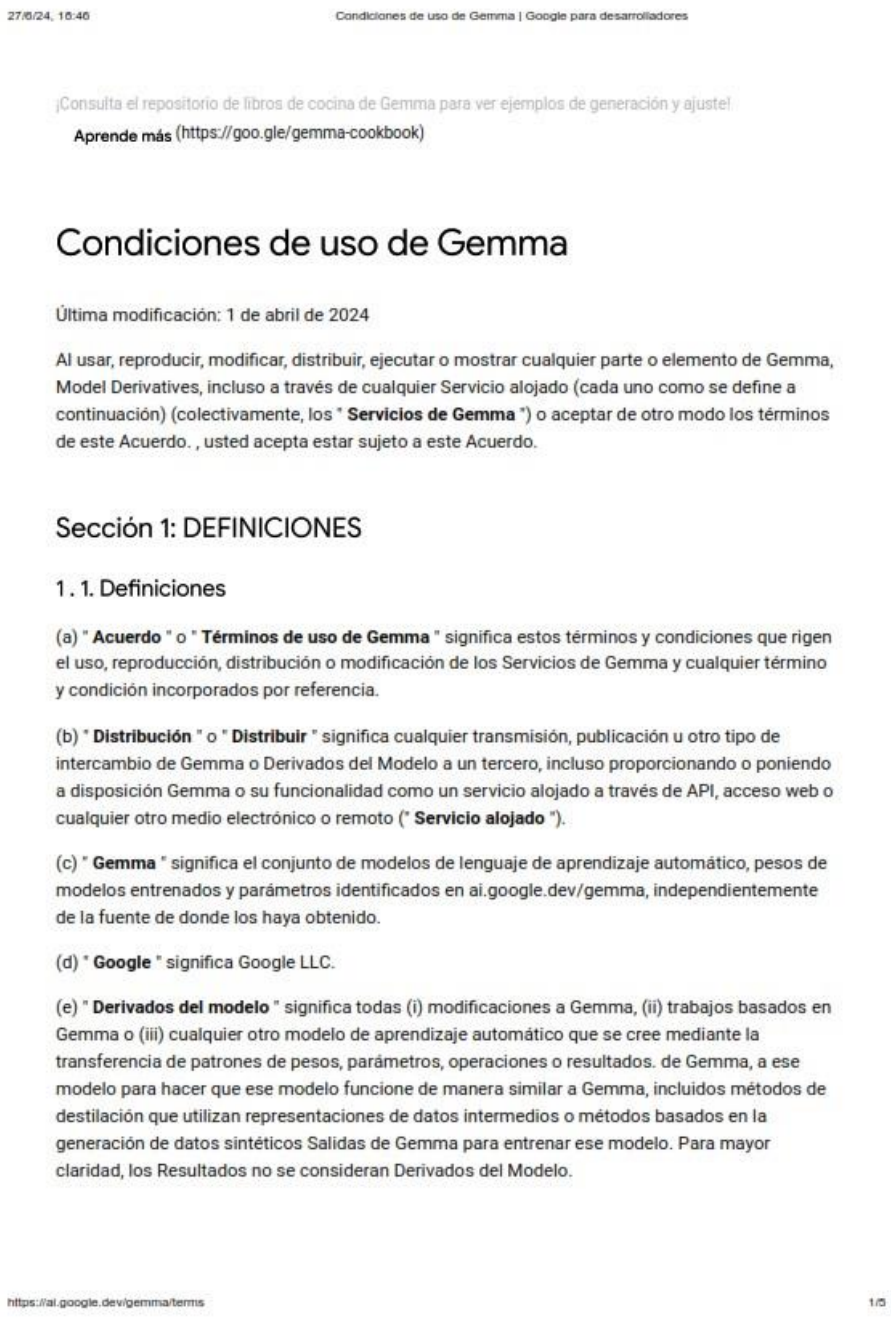
https://www.youtube.com/live/kxzgJ57Fk8o?si=hWtzvvrCU1gR7xSJ

Anexo 2. Condiciones de uso de Gemma

Las condiciones de uso de Gemma se encuentran detalladas dentro de la página web de Google en la dirección:

- https://ai.google.dev/gemma/terms

Donde se puede visualizar varios puntos a tener en cuenta en el manejo de la licencia del modelo Gemma en este caso la versión 2B como se puede observar en la **Figura 40.**



27/6/24, 16:46 Condiciones de uso de Gemma | Google para desarrolladores

¡Consulta el repositorio de libros de cocina de Gemma para ver ejemplos de generación y ajuste!

Aprende más (https://goo.gle/gemma-cookbook)

Condiciones de uso de Gemma

Última modificación: 1 de abril de 2024

Al usar, reproducir, modificar, distribuir, ejecutar o mostrar cualquier parte o elemento de Gemma, Model Derivatives, incluso a través de cualquier Servicio alojado (cada uno como se define a continuación) (colectivamente, los " **Servicios de Gemma** ") o aceptar de otro modo los términos de este Acuerdo. , usted acepta estar sujeto a este Acuerdo.

Sección 1: DEFINICIONES

1 . 1. Definiciones

(a) " **Acuerdo** " o " **Términos de uso de Gemma** " significa estos términos y condiciones que rigen el uso, reproducción, distribución o modificación de los Servicios de Gemma y cualquier término y condición incorporados por referencia.

(b) " **Distribución** " o " **Distribuir** " significa cualquier transmisión, publicación u otro tipo de intercambio de Gemma o Derivados del Modelo a un tercero, incluso proporcionando o poniendo a disposición Gemma o su funcionalidad como un servicio alojado a través de API, acceso web o cualquier otro medio electrónico o remoto (" **Servicio alojado** ").

(c) " **Gemma** " significa el conjunto de modelos de lenguaje de aprendizaje automático, pesos de modelos entrenados y parámetros identificados en ai.google.dev/gemma, independientemente de la fuente de donde los haya obtenido.

(d) " **Google** " significa Google LLC.

(e) " **Derivados del modelo** " significa todas (i) modificaciones a Gemma, (ii) trabajos basados en Gemma o (iii) cualquier otro modelo de aprendizaje automático que se cree mediante la transferencia de patrones de pesos, parámetros, operaciones o resultados. de Gemma, a ese modelo para hacer que ese modelo funcione de manera similar a Gemma, incluidos métodos de destilación que utilizan representaciones de datos intermedios o métodos basados en la generación de datos sintéticos Salidas de Gemma para entrenar ese modelo. Para mayor claridad, los Resultados no se consideran Derivados del Modelo.

https://ai.google.dev/gemma/terms 1/5

Figura 40. Términos y condiciones del modelo Gemma.

Anexo 3. Seguimiento de métricas BLEU y Loss en los experimentos.

En la **Figura 41** se puede apreciar que tanto la métrica BLEU aumenta y el Loss decrece de manera sostenida a través de los distintos steps.

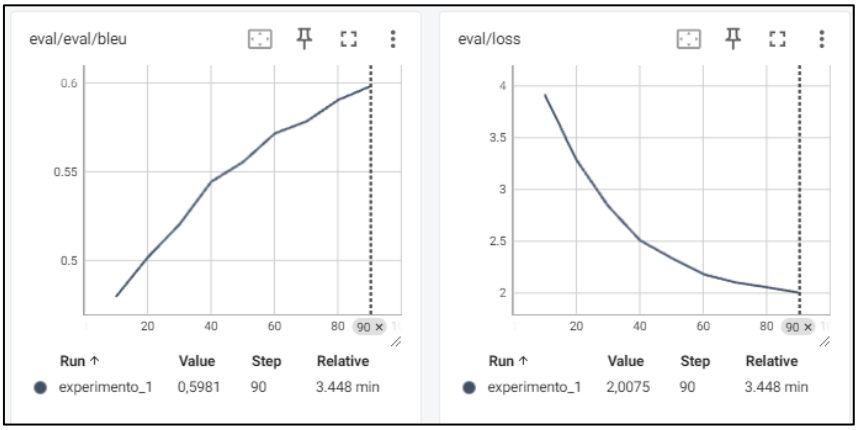


Figura 41. Seguimiento del entrenamiento del experimento 1.

La **Figura 42** permite visualizar como el entrenamiento solo produjo buenos resultados en los primeros steps, para luego decaer completamente tanto el Loss, como en BLEU, en escenarios como estos configurar la paciencia en un numero de steps permitió evitar seguir con el entrenamiento si no se percibió mejoras luego de x steps.

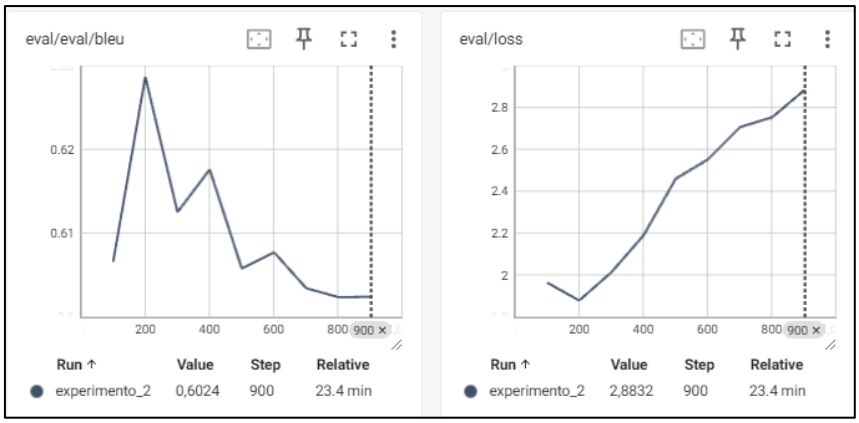


Figura 42. Seguimiento del entrenamiento del experimento 2.

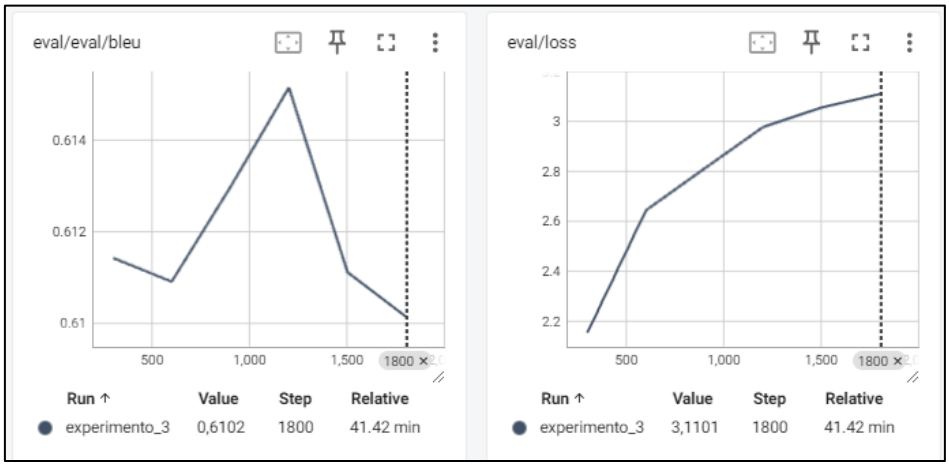


Figura 43. Seguimiento del entrenamiento del experimento 3.

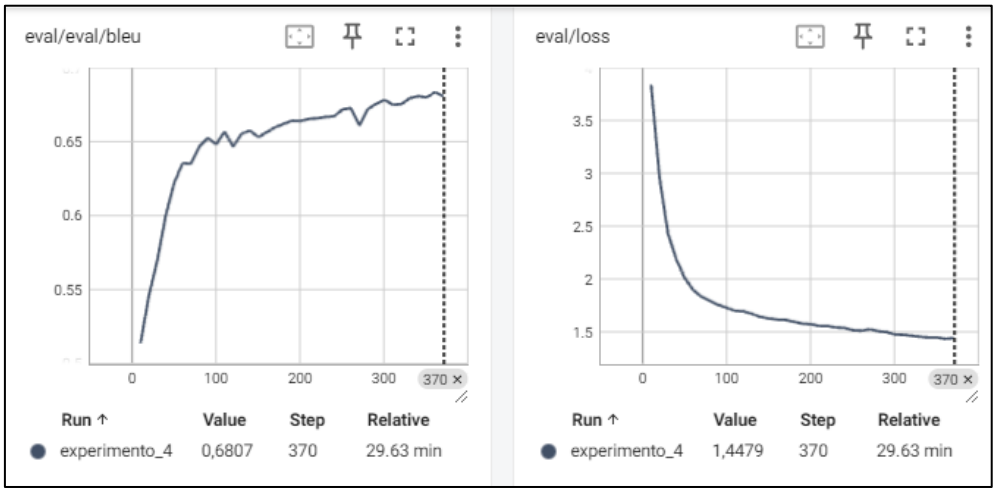


Figura 44. Seguimiento del entrenamiento del experimento 4.

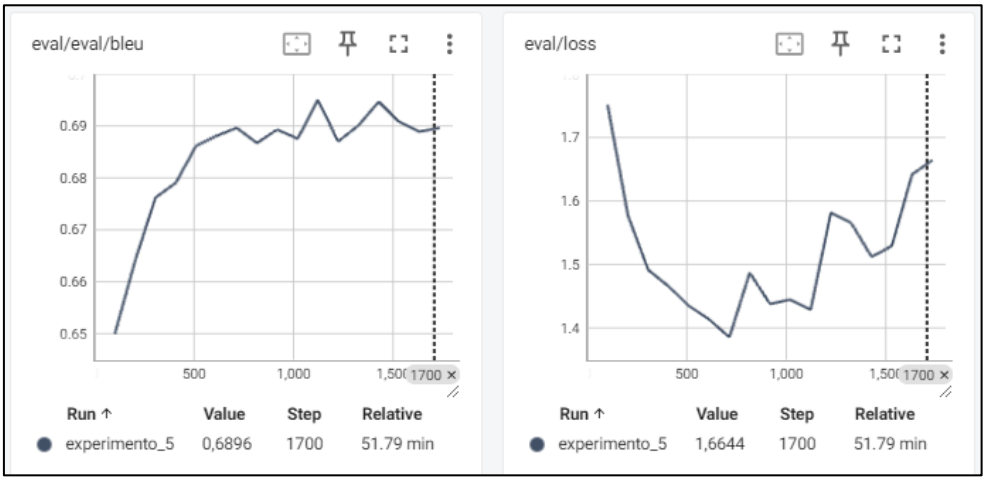


Figura 45. Seguimiento del entrenamiento del experimento 5.

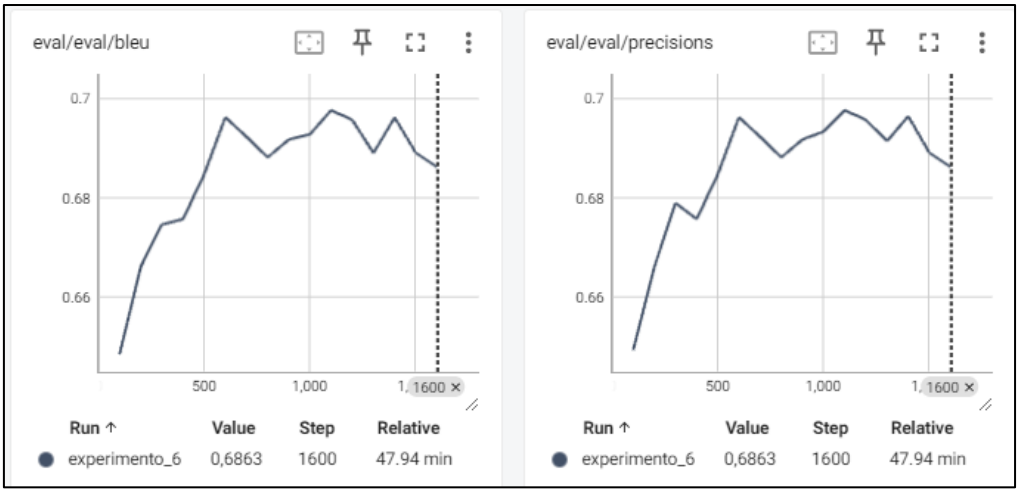


Figura 46. Seguimiento del entrenamiento del experimento 6.

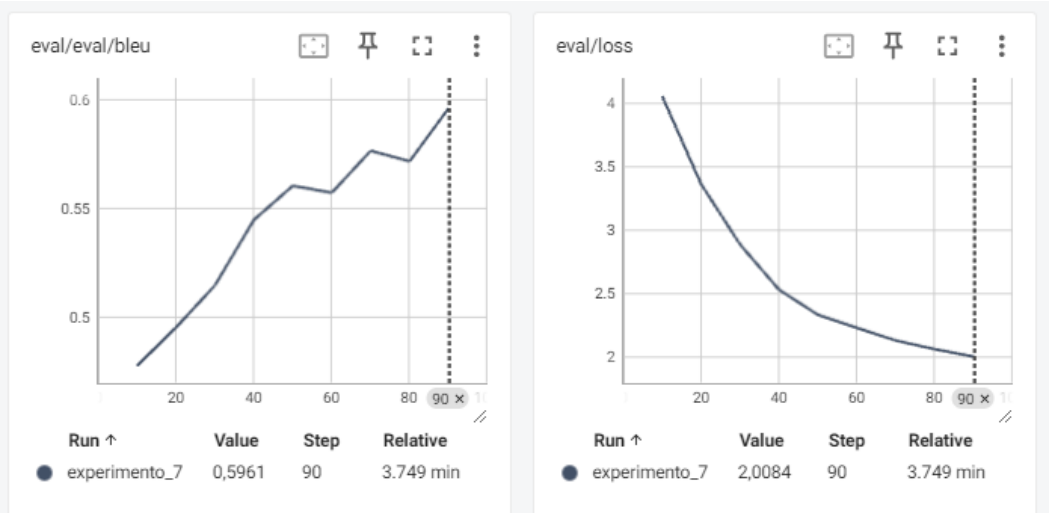


Figura 47. Seguimiento del entrenamiento del experimento 7.

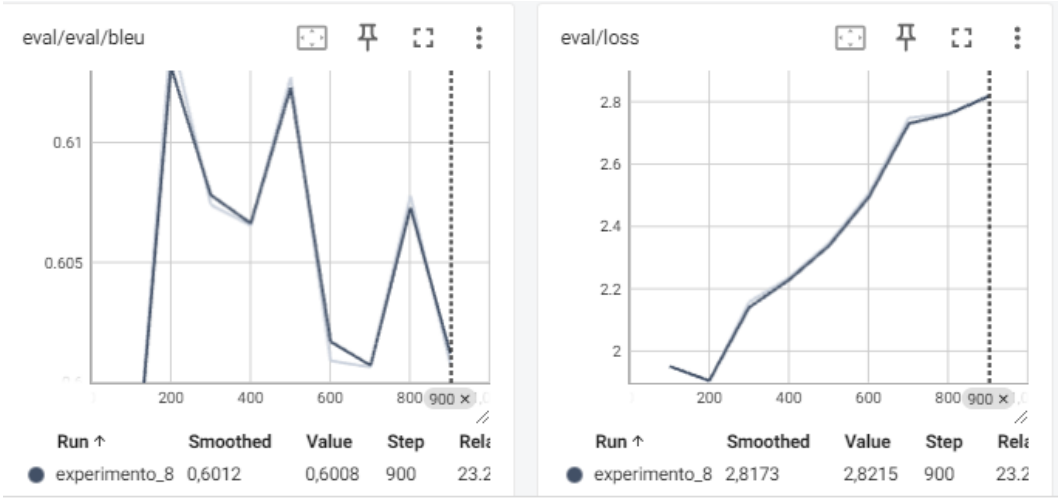


Figura 48. Seguimiento del entrenamiento del experimento 8.

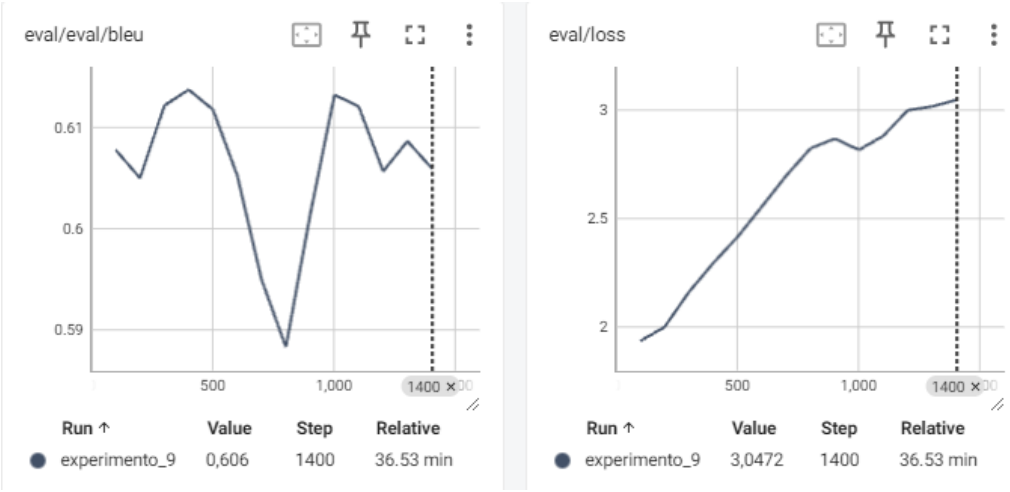


Figura 49. Seguimiento del entrenamiento del experimento 9.

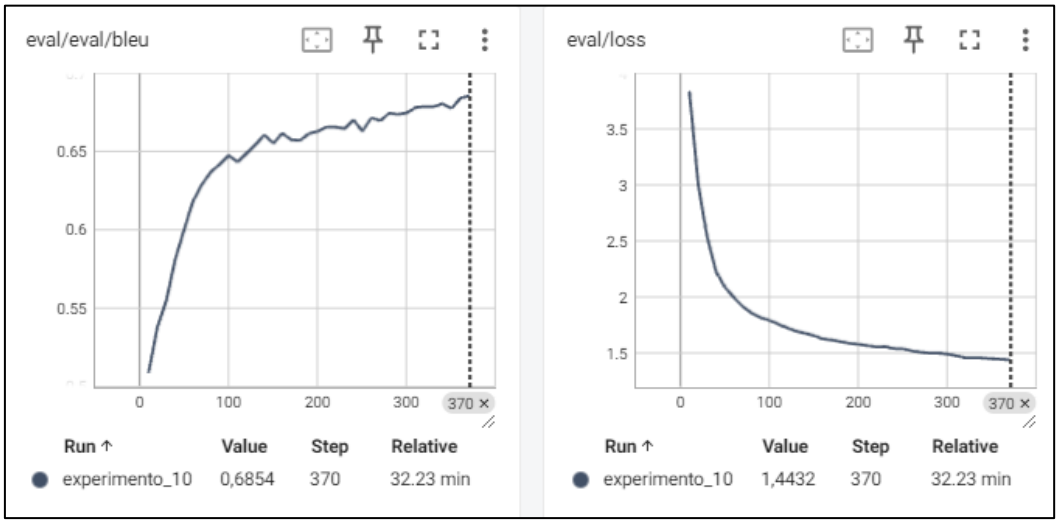


Figura 50. Seguimiento del entrenamiento del experimento 10.

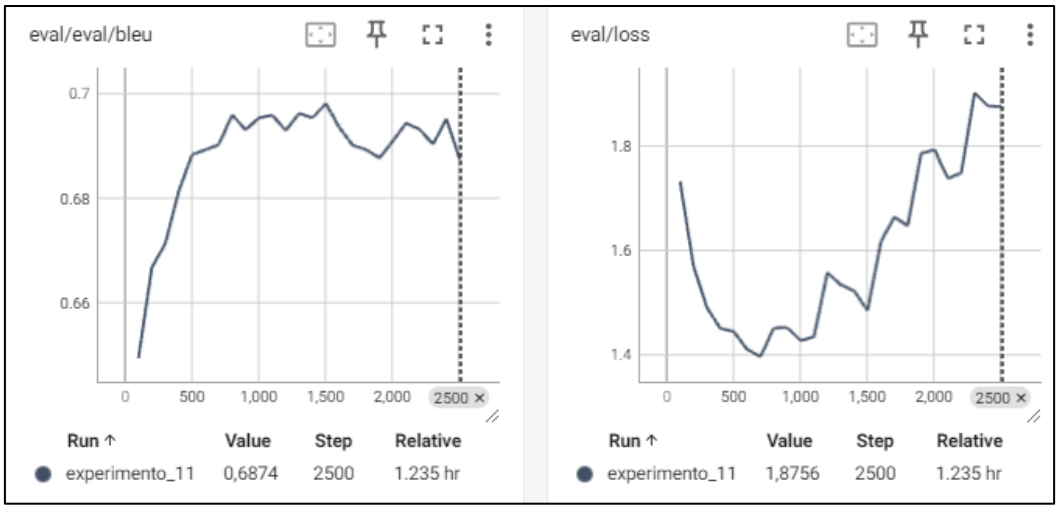


Figura 51. Seguimiento del entrenamiento del experimento 11.

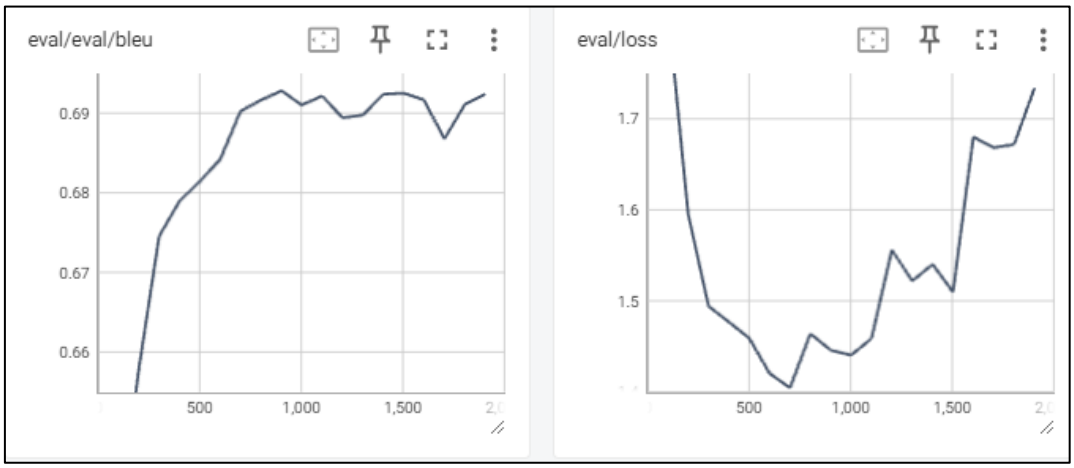


Figura 52. Seguimiento del entrenamiento del experimento 12.

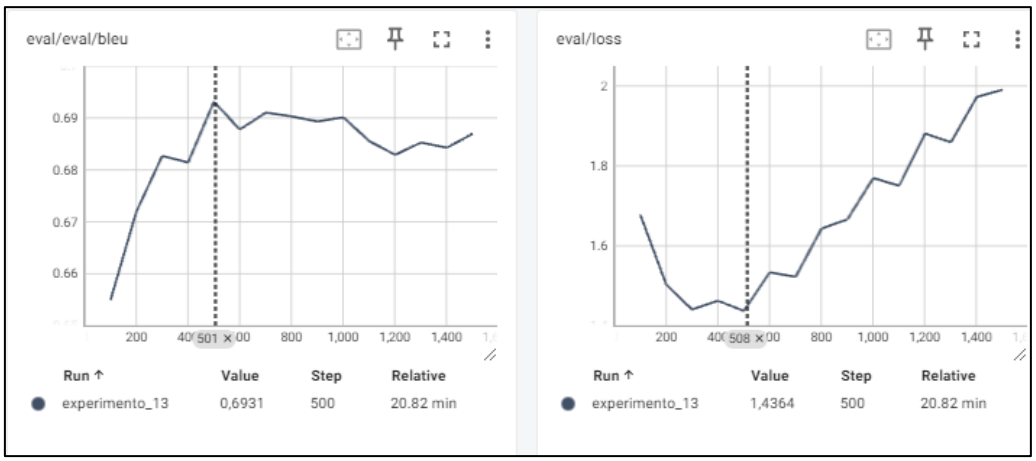


Figura 53. Seguimiento del entrenamiento del experimento 13.

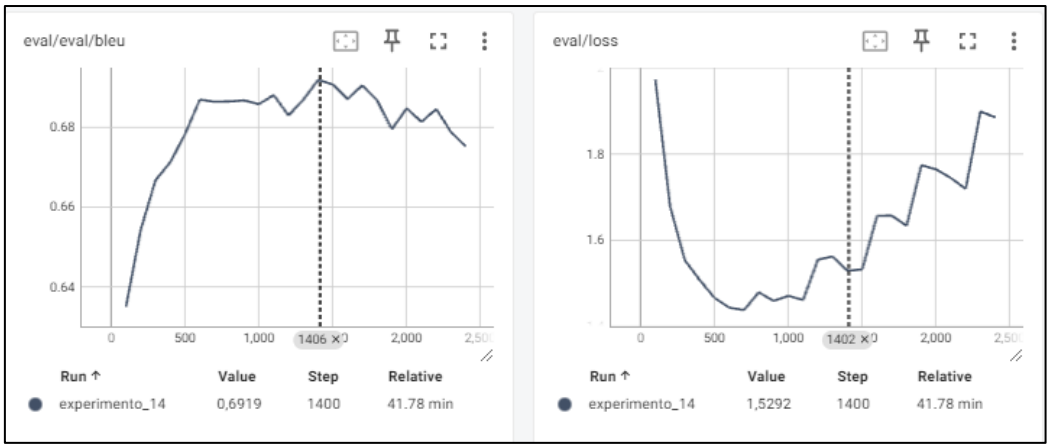


Figura 54. Seguimiento del entrenamiento del experimento 14.

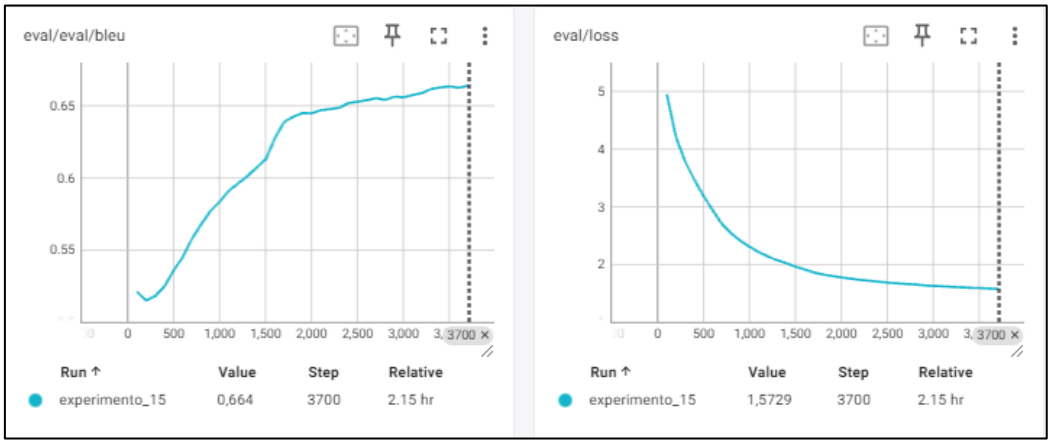


Figura 55. Seguimiento del entrenamiento del experimento 15.

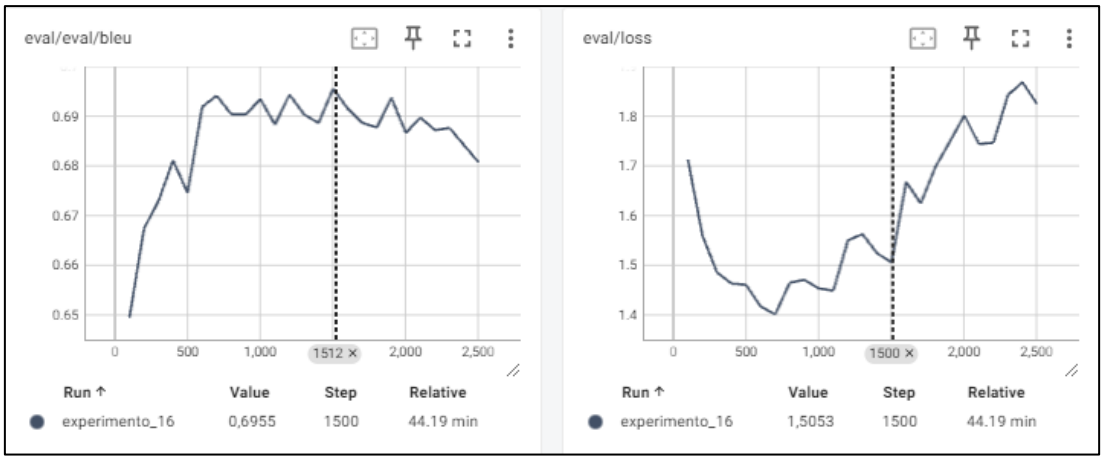


Figura 56. Seguimiento del entrenamiento del experimento 16.

Anexo 4. Certificado de traducción del resumen al idioma inglés

Loja, 02 de agosto de 2024

Lic. Pedro Geovanny Calva Jiménez
LICENCIADO EN PEDAGOGÍA DEL IDIOMA INGLÉS

CERTIFICO:

Que el resumen del Trabajo de Integración Curricular cuyo título es: **Ajuste del modelo Gemma para la descripción de tablas de datos en formato Markdown**, del aspirante **Patricio Oswaldo Paredes Chamba**, con cédula de identidad Nro. **1150011805**, de la Carrera de Computación de la Universidad Nacional de Loja, ha sido traducido al inglés y cumple con las características propias del idioma extranjero.

Lo certifico en honor a la verdad y autorizo hacer uso del presente en lo que a sus intereses convenga.

_____
Lic. Pedro Geovanny Calva Jiménez
1150428496
Nro. Reg. Senecyt: 1031-2022-2421774
LICENCIADO EN PEDAGOGÍA DEL IDIOMA INGLÈS