

Primeros Pasos Programando con Grandes Modelos de Lenguaje en Python

Matías Torres Esteban

InforSanLuis 2025

Departamento de Informática
Universidad Nacional de San Luis

3, 4 y 5 de diciembre de 2025

1. Introducción
2. Contexto
3. Interpretación de Textos con LLMs
4. Consigna

Introducción

Motivación

- Los **Grandes Modelos de Lenguaje** (LLMs) son capaces de aprender patrones complejos de los textos y de resolver en consecuencia varias tareas de PLN.
- Podemos aprovechar estas capacidades para automatizar nuestros procesos.

Objetivos

- Aprender a programar con LLMs en Python y resolver problemas no triviales en Python.
- Entender las virtudes y deficiencias de los LLMs.

Organización

- Material del curso disponible aquí:
<https://github.com/matizzat/InforSanLuis-2025-LLMs>
- **Aprobación:** Entregar al menos una tarea antes del viernes 5, 23:59

- Los modelos de lenguaje son artefactos que predicen la próxima palabra en una secuencia dado un contexto que le precede:

$$P(\text{Papás} | \text{Me comí una milanesa con}) = 0,6$$

- La **decodificación** permite generar palabras a partir de esta distribución de probabilidad.
- Los LLMs son modelos de lenguaje implementados con una red neuronal profunda de millones de parámetros y entrenada con una gran cantidad de textos.
- Muchas tareas de PLN pueden ser modeladas como predicción de palabras.

$$P(w | P: \text{¿Quién escribió el señor de los anillos? R:})$$

- **Riesgo:** Debido a su naturaleza estocástica, los LLMs son modelos de caja negra, no interpretables y sufren de alucinaciones y sesgos.

Tipos de LLMs

- **Encoder Only:**

Transforma secuencias de tokens a vectores contextualizados

Me llamo Matías \mapsto $[0, 1, 2, \dots]$ $[3, 1, 2, \dots]$ $[0, 0, 2, \dots]$

Estos vectores capturan el significado de un texto.

Ejemplos: BERT, RoBERTa

Permiten construir bases de datos semánticas.

- **Decoder Only:**

Generan texto prediciendo el próximo token dada una secuencia de tokens previa.

Utilizados para responder preguntas y generar resúmenes.

Ejemplos: Gemini, GPT4 y LLaMa.

- **Híbridos:**

Ejemplos: T5 y BART.

Ingeniería de Prompts

Hay dos maneras de lograr que un LLM resuelva una tarea:

- **Ajuste Fino:**

Entrenamos al modelo con nuevos datos que se ajusten a nuestro dominio.
Modifica los parámetros.

- **Ingeniería de Prompts:**

Escribimos prompts ingeniosos y construimos sistemas informáticos que invocan al LLM preentrenado.

No modifica los parámetros.

Nosotros utilizaremos **ingeniería de prompts** para interpretar texto y responder preguntas de dominios especializados.

Utilizar LLMs y Python para construir **Grafos de Conocimiento** a partir de textos expositivos.

Ejemplo

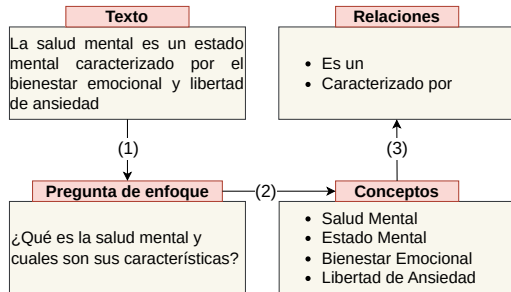
"El agua es un gas y está compuesta de carbono."



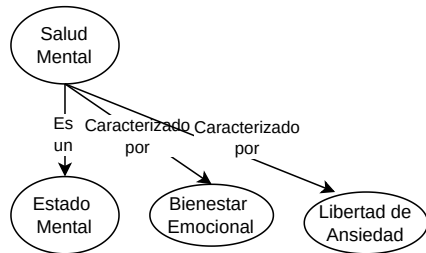
Figura: Resultado del modelo Gemini 2.5 Flash para el ejemplo dado.

El Proceso de Creación

Idea: Instruir al modelo a construir un mapa conceptual en un proceso de cuatro pasos mediante **ingeniería de prompts**.



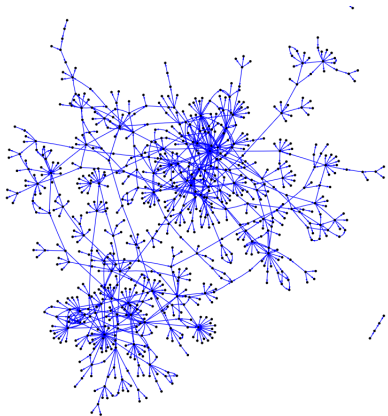
(a) Primeros pasos obtienen la pregunta de enfoque, los conceptos y las relaciones.



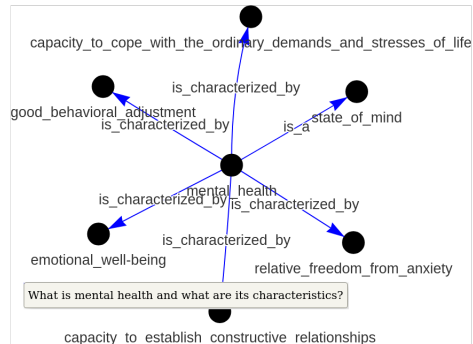
(b) Último paso obtiene las ternas de conocimiento.

Figura: Proceso de interpretación de texto con LLMs

Resultados



(a) Grafo global

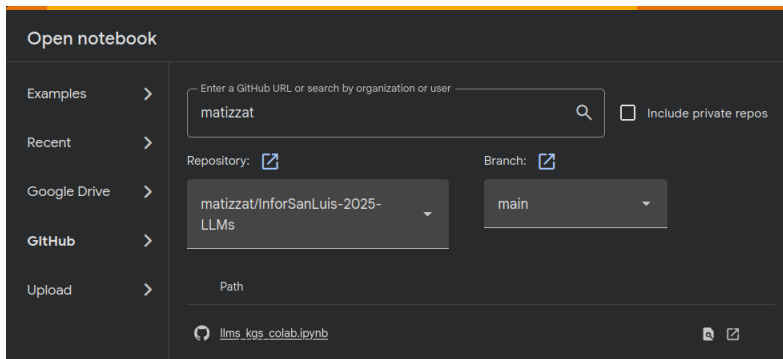


(b) Subgrafo de la entrada "Mental health"

Figura: Resultado del proceso aplicado a 100 entradas de un diccionario de psicología.

Consigna

1. Entrar a Google Colab: <https://colab.google/>
2. Abrir la Notebook de Github con la tarea de hoy:



3. Guardar una **copia en Drive**, leer la notebook y resolver la **tarea** de la última sección.



Muchas Gracias por su Atención