

## Prueba Técnica Científico de datos con énfasis en IA

### Caso de Negocio:

Tuya S.A es una compañía de financiamiento con una propuesta de valor centrada alrededor de la oferta de créditos rotativos y de consumo con cadenas de consumo aliadas, así como de seguros. La compañía desea crear un modelo/solución que pueda responder las preguntas de los clientes sobre los temas previamente mencionados usando herramientas de inteligencia artificial generativa.

Para presentar su propuesta puede tomar como insumo los enlaces de la página web de la compañía que contienen información suficiente y son suministrados a continuación:

- <https://www.tuya.com.co/como-pago-mi-tarjeta-o-credicompras>
- <https://www.tuya.com.co/tarjetas-de-credito>
- <https://www.tuya.com.co/credicompras>
- <https://www.tuya.com.co/otras-soluciones-financieras>
- <https://www.tuya.com.co/nuestra-compania>
- <https://www.tuya.com.co/activacion-tarjeta>

### Respuestas a responder:

1. ¿Cuáles son los nombres de las tarjetas que tiene disponibles Tuya S.A?.
2. ¿Cuáles son los valores la tasa de interés y póliza del producto credicompras?.

### Entregables:

1. Repositorio en github con lo que considere necesario para la estructuración del proyecto.
2. Prueba de ejecución de su propuesta de solución a la necesidad planteada, puede ser un notebook, un reporte con imágenes o el formato que considere conveniente.

### Notas para el desarrollo de la prueba:

Puede hacer uso de las herramientas que considere y tenga disponibles, en caso de requerir el uso de un modelo de lenguaje libre se recomienda usar el modelo Llama 3.2 en su versión de 1B de parámetros para que la necesidad de recursos computacionales sea baja. Puede ser desde el repositorio oficial en HuggingFace o desde [unsloth/Llama-3.2-1B-Instruct](https://huggingface.co/unsloth/Llama-3.2-1B-Instruct) donde no va a realizar pedidos de autenticación adicionales.

Se comparte el código para hacer la conexión al modelo y usarlo bajo el framework de langchain

```
from transformers import AutoTokenizer, AutoModelForCausalLM, pipeline
from langchain.llms import HuggingFacePipeline

model_id = 'unsloth/Llama-3.2-1B-Instruct'

#descarga de componentes base del modelo

tokenizer = AutoTokenizer.from_pretrained(model_id)

model = AutoModelForCausalLM.from_pretrained(model_id)

#creacion del pipeline del modelo

pipe = pipeline(
    "text-generation",
    model=model,
    tokenizer=tokenizer,
    max_length=2000, device = "cuda"
)

#conversión a uso api tipo langchain

local_llm = HuggingFacePipeline(pipeline=pipe)
```

En caso de recurrir a la implementación usando el modelo sugerido tenga en cuenta la carta del modelo para un uso correcto.

[Llama 3.2 | Model Cards and Prompt formats](#)