**Opción 2: Sistema de Detección de Quejas y Soporte Automático para E-commerce**

**Descripción Detallada de la Consigna:**

Construirán un sistema prototipo para una empresa de e-commerce que busca automatizar la gestión de quejas y la respuesta inicial a clientes. El sistema debe ser capaz de identificar quejas, extraer información clave y generar una respuesta de servicio al cliente.

1. **Simulación/Creación de Datos:**
   * Crear un dataset simulado de **10-15 "tickets de soporte"** (textos cortos, de 2-5 oraciones cada uno) que contengan opiniones sobre productos o servicios.
   * Asegúrense de que haya una mezcla de sentimientos (positivos, negativos, neutrales) y que incluyan algunos casos con **modismos o jerga argentina**.
   * **Etiquetar manualmente** cada ticket con su sentimiento (Positivo, Negativo, Neutro) y al menos **3-5 categorías de queja/consulta** (ej. "Problema de Envío", "Producto Defectuoso", "Consulta de Stock", "Elogio", "Consulta de Facturación").
2. **Detección de Sentimiento y Clasificación Zero-Shot:**
   * Aplicar un modelo preentrenado de **análisis de sentimiento** (Hugging Face finiteautomata/beto-sentiment-analysis o Gemini API) para clasificar la polaridad de cada ticket.
   * Utilizar la **clasificación *zero-shot*** de Hugging Face (Recognai/bert-base-spanish-wwm-cased-xnli) o Gemini API para categorizar cada ticket en las categorías predefinidas manualmente (sin necesidad de entrenamiento del clasificador).
   * Comparar los resultados del sentimiento y la clasificación *zero-shot* del modelo con sus etiquetas manuales, comentando las discrepancias.
   * *(Referencia: 004\_Análisis de Sentimiento con Modelos Preentrenados de HuggingFace.ipynb para sentiment-analysis; EJERCICIOS.ipynb/HF\_Transformers.ipynb para zero-shot-classification; GEMINI.ipynb para alternativas con Gemini API).*
3. **Extracción de Información Clave (NER/QA):**
   * Para cada ticket, identificar y extraer **entidades relevantes** (ej. nombres de productos, números de pedido simulados, nombres de clientes, ciudades). Pueden usar un pipeline de NER o intentar con *Question Answering*.
   * Formular una **pregunta automática por ticket** (ej. "¿Cuál es el problema principal?", "¿Qué producto menciona el cliente?") y usar un modelo de *Question Answering* (Hugging Face PlanTL-GOB-ES/roberta-large-bne-sqac o similar) para extraer la respuesta principal.
   * *(Referencia: EJERCICIOS.ipynb/HF\_Transformers.ipynb para NER y QA con transformers pipelines; GEMINI.ipynb para alternativas con Gemini API).*
4. **Generación de Respuesta Personalizada:**
   * Para cada ticket (especialmente los clasificados como negativos o de "problema"), generar una **respuesta de atención al cliente** utilizando un LLM (Hugging Face transformers como datificate/gpt2-small-spanish o Gemini API).
   * La respuesta debe ser **empática**, reconocer el problema y sugerir un siguiente paso (ej. "Derivaremos su caso al área técnica", "Le contactaremos para más detalles"). La respuesta debe tener un límite de longitud (ej. 2-4 líneas) y, si es posible, intentar un tono amable y local.
   * *(Referencia: Introducción a los Modelos de Lenguaje.ipynb, Transformers.ipynb, generacion\_de\_texto.ipynb para LLMs y generación de texto; EJERCICIOS.ipynb/HF\_Transformers.ipynb para text-generation con transformers pipelines; GEMINI.ipynb para text-generation con Gemini API).*
5. **Interfaz Interactiva (Obligatoria):**
   * Desarrollar una interfaz en **Gradio** que permita al usuario introducir un nuevo "ticket de soporte" y ver en tiempo real:
     + El sentimiento predicho del ticket.
     + La categoría predicha (zero-shot).
     + Las entidades o la respuesta clave extraída.
     + La respuesta generada por el LLM.
   * *(Referencia: Gradio.ipynb para la construcción de la interfaz).*

**Posibles Entregables:**

* Un Jupyter Notebook (.ipynb) completo que incluya:
  + La definición del dataset simulado y su etiquetado manual.
  + El código para la aplicación de los pipelines/APIs de PLN en cada etapa.
  + Una discusión de los resultados de cada etapa, incluyendo la comparación con las etiquetas manuales.
  + El código funcional de la interfaz Gradio, con instrucciones claras para ejecutarla.
* Una reflexión final en el notebook sobre la viabilidad de tal sistema en un entorno real, los desafíos éticos (sesgos), y las limitaciones de los modelos en este contexto.

**Criterios de Evaluación Específicos (Complementarios a los Globales):**

* Calidad y representatividad del dataset simulado, incluyendo la incorporación de jerga.
* Precisión de las clasificaciones de sentimiento y zero-shot.
* Efectividad en la extracción de información clave.
* Coherencia, empatía y adecuación contextual/cultural de las respuestas generadas por el LLM.
* Diseño y funcionalidad de la interfaz Gradio.