



República del Ecuador

Universidad de Especialidades Espíritu Santo/División Online

Maestría en Inteligencia de Negocios y Ciencia de Datos

## **INTELIGENCIA ARTIFICIAL**

### **Proyecto**

Grupo N° 3

Autores:

Liz Eliana Castillo Zamora

Pablo Mauricio Castro Hinostroza

Erick Sebastián Rivas

Ángel Israel Romero Medina

Tutora:

PhD. Gladys María Villegas Rugel

20 de diciembre del 2025

Guayaquil – Ecuador

## 1. Título y Resumen Ejecutivo

### **Nombre descriptivo del proyecto:**

Plataforma de Análisis de Sentimientos en tiempo real para E-Commerce basada en técnicas de procesamiento del lenguaje natural y modelos Transformers.

### **Resumen**

El proyecto propone el diseño y desarrollo de una plataforma inteligente de análisis de sentimientos en tiempo real dirigida a empresas de comercio electrónico que requieren monitorear grandes volúmenes de reseñas, comentarios y publicaciones alojadas en diversas fuentes digitales.

El problema que se presenta actualmente es que en los marketplaces no se cuenta con un entorno que sea dinámico para que de forma inmediata se pueda identificar problemas de calidad, de envíos en los pedidos y en la variación en la percepción del cliente en la plataforma.

Por lo tanto, la plataforma a desarrollarse integra un sistema de Procesamiento de Lenguaje Natural (PLN) para realizar una clasificación adecuada de los sentimientos de los clientes, ya sean positivo, negativo o neutro, en base a la calidad del producto, precio y logística, que permita identificar niveles de atención urgente según la severidad del problema detectado. Se utilizarán modelos Transformer, versiones ajustadas de BERT, ya que su arquitectura permite un entendimiento contextual más robusto del lenguaje natural en concordancia con el concepto de agentes inteligentes y aprendizaje automático. (García Serrano, 2016)

Complementariamente, se incorpora un módulo generativo basado en modelos tipo GPT para producir respuestas automáticas coherentes y contextualizadas.

## Introducción

Mediante técnicas de Inteligencia Artificial, se realizará el entrenamiento inicial del sistema en el cual se utilizará el dataset público de “Amazon Product Reviews”, este es seleccionado por su estructura limpia, volumen significativo de reseñas y disponibilidad de calificaciones numéricas que permiten derivar etiquetas de sentimiento. Este dataset servirá como base para construir y ajustar los modelos de análisis de sentimientos y detección de aspectos antes de integrarlos con fuentes en tiempo real.

Adicionalmente, y con la ayuda de reseñas históricas y flujos de tiempo real obtenidos mediante web scraping (Selenium y BeautifulSoup), APIs que son públicas de redes sociales y bases de datos preexistentes, garantizando así baja latencia y escalabilidad, se estructura un pipeline distribuido con tecnologías como Apache Kafka y Spark Streaming, en concordancia con los principios de los sistemas inteligentes de soporte a la decisión aplicados en contextos de alto volumen de datos. (Galar Pascual, 2020).

Es importante mencionar que, el proyecto contempla además un dashboard de monitoreo con métricas clave, tendencias de sentimiento, alertas automáticas y análisis de aspectos por producto o categoría. Todo esto con el fin de mejorar la gestión de reputación digital, reducir tiempos de respuesta al cliente, anticipar incidencias operativas y fortalecer la toma de decisiones basada en datos.

## 2. Definición del Problema

**Contexto:** ¿En qué dominio o sector se enmarca el problema?

El proyecto se sitúa en el sector del comercio electrónico, un entorno caracterizado por la alta competitividad y por una gran dependencia de la reputación digital. En este sector de e-commerce, como Amazon las reseñas de clientes son

fundamentales para evaluar la calidad del producto, la eficacia del proceso logístico y la satisfacción del consumidor. Sin embargo, la magnitud de datos generados diariamente por los usuarios excede la capacidad de los equipos humanos para analizarlos manualmente. El reto técnico es construir un modelo de Deep Learning capaz de entender el contexto completo de una reseña en inglés y clasificar con precisión casi humana para permitir una reacción rápida ante incidentes.

La Inteligencia Artificial, especialmente el Procesamiento del Lenguaje Natural (PLN), ofrece herramientas para automatizar tareas cognitivas como la clasificación de sentimientos y la extracción de información relevante en textos. Según Vaswani (2017) “ Los sistemas inteligentes permiten procesar grandes volúmenes de información y tomar decisiones en entornos complejos, filtrando y representando el conocimiento de manera eficiente para su interpretación”.

En este contexto, surge la necesidad de implementar un sistema automatizado que permita analizar y monitorear en tiempo real las opiniones de los usuarios, utilizando datos provenientes del dataset de Kritanjali (2020) “Amazon Product Reviews”, reconocido por su estructura limpia y su disponibilidad pública.

***Problema específico:*** Descripción clara y concreta del problema a resolver

El marketplace presenta dificultades para monitorear y analizar de forma oportuna las miles de reseñas que los usuarios generan diariamente. Esta limitación conlleva varios problemas:

- Demoras en la identificación de fallas de productos o retrasos de envíos.
- Dificultad para evaluar tendencias de satisfacción del cliente.
- Respuestas tardías ante comentarios negativos que afectan la reputación.
- Ausencia de un mecanismo automatizado que clasifique sentimientos y detecte aspectos críticos.

***El problema específico se define como:***

La necesidad de contar con un sistema automatizado que analice sentimientos y detecte aspectos relevantes en reseñas de productos en tiempo real, utilizando datos estructurados del dataset Amazon Product Reviews como base inicial de entrenamiento.

- ***Justificación:*** ¿Por qué es importante resolver este problema?

Analizar manualmente grandes volúmenes de texto es una actividad ineficiente e insostenible. Los avances en IA permiten automatizar este proceso mediante algoritmos de aprendizaje automático capaces de identificar patrones, clasificar emociones y extraer información relevante del lenguaje natural. Esto es coherente con el planteamiento de Kreimer (2020), quien resalta que la IA busca imitar capacidades humanas como la comprensión del lenguaje, el razonamiento y el aprendizaje para mejorar la toma de decisiones.

Por su parte, Kreimer (2020) destaca que los sistemas inteligentes de soporte a la decisión permiten mejorar la eficiencia y la capacidad de respuesta en contextos de alta carga informativa, integrando modelos predictivos y análisis automatizado de datos para asistir a los equipos operativos.

La solución propuesta es relevante porque:

- Permite detectar tendencias negativas antes de que escalen.
- Mejora la calidad del servicio al cliente mediante respuestas oportunas.
- Optimiza procesos internos relacionados con calidad, logística y postventa.
- Reduce costos asociados a reclamos no atendidos.
- Fortalece la reputación digital del marketplace.

***Stakeholders:*** ¿Quiénes se beneficiarán de la solución?

Los principales actores que interactúan con la solución son:

- **Clientes del marketplace:** beneficiarios directos de mejoras en la calidad del servicio.
- **Equipo de Servicio al Cliente:** quienes gestionan reclamos y comunicaciones automatizadas.
- **Área de Control de Calidad:** encargada de identificar productos con problemas recurrentes.
- **Gerencia y Dirección:** interesadas en métricas estratégicas sobre satisfacción y tendencias.
- **Equipo técnico (TI e IA):** responsable del desarrollo, despliegue y mantenimiento del sistema.
- **Analistas de datos y Marketing:** usuarios del dashboard para la toma de decisiones.

### 3. Objetivos del Proyecto

#### Objetivo general

Desarrollar un sistema de análisis de sentimientos basado en el modelo Transformer xlm-roberta-base que, mediante un pipeline de entrenamiento con datos globales (Inglés) y una capa de inferencia con traducción, clasifique eficazmente reseñas en español para optimizar el monitoreo de calidad.

#### Objetivos específicos

- 1) **Implementar el pipeline de ingesta:** Implementar un pipeline de limpieza y normalización de datos sobre el dataset "Amazon Product Reviews" (21,255 registros), gestionando el desbalance de clases y preparando los textos para la arquitectura Transformer.
- 2) **Entrenar y Validar el Modelo NLP:** Realizar el *fine-tuning* del modelo XLM-RoBERTa Base utilizando exclusivamente datos en inglés, logrando la convergencia del modelo con una función de pérdida (*loss*) validada inferior a 0.40.

3) **Implementar un Dashboard de monitoreo:** Desplegar un Dashboard interactivo que visualice métricas clave como; volumen de menciones, tendencia de sentimiento por aspecto y alertas activas, permitiendo la actualización de gráficos en tiempo real.

## Alcance

- **Entrenamiento:** Datos en Inglés (Amazon).
- **Operación:** Entradas en Español (vía traducción).
- **Modelo:** Deep Learning (Transformers).

## Limitaciones

- **Dependencia de Traducción:** La precisión final depende de la calidad de la traducción automática intermedia.
- **Sesgo de Clases:** Dificultad para detectar la clase "Neutra" debido al desbalance natural del dataset de origen.

## 4. Solución Propuesta con IA

**Tipo de problema de IA:** El proyecto aborda principalmente problemas de Procesamiento de Lenguaje Natural (NLP), específicamente:

- Clasificación de Texto (Sentiment Analysis).
- Extracción de Información (Aspect-Based Sentiment Analysis - ABSA).
- Generación de Texto (Natural Language Generation - NLG).

**Enfoque Técnico:** Se utiliza **XLM-RoBERTa**, un modelo basado en Transformers pre-entrenado en 100 idiomas. A diferencia de BERT clásico, XLM-R está optimizado para tareas multilingües, lo que lo hace ideal para nuestra estrategia híbrida de entrenar en un idioma e inferir en otro (con apoyo de traducción).

### ***Arquitectura Preliminar:***

1. **Capa de Entrada:** El usuario ingresa una reseña en **Español** (ej: *"El producto llegó roto"*).
2. **Capa de Adaptación (Traducción):**
  - Uso de **GoogleTranslator** (librería **deep-translator**) para convertir la entrada de ES  $\rightarrow$  EN (*"The product arrived broken"*).
3. **Capa de Procesamiento (Modelo):**
  - Tokenización (sub-word tokenization).
  - Inferencia con **xlm-roberta-base** (pesos ajustados con datos de Amazon).
4. **Capa de Salida:** Clasificación Softmax en 3 categorías (Positivo, Neutro, Negativo) con lógica de umbral para incertidumbre.

### ***Alternativas Consideradas:***

- **LSTM/RNN:** Descartados porque tienen dificultades con secuencias largas y son más lentos de entrenar que los Transformers modernos en paralelo.

## **5. Datos**

**Fuentes de datos:** ¿De dónde se obtendrán los datos?

**Fuente de Datos:** Dataset público **"Amazon Product Reviews"** (versión Kaggle/HuggingFace).

**Descripción de los datos:**

**Tipo:** Texto no estructurado (cuerpo de la reseña) y metadatos estructurados (calificación de estrellas, fecha, categoría del producto).

**Idioma Base:** Inglés.

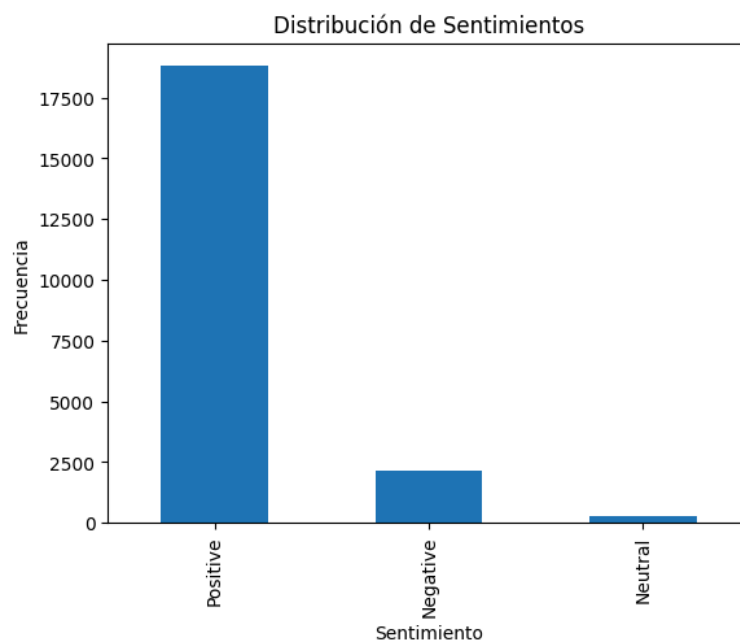
**Entrenamiento:** 21,255 reseñas procesadas tras la limpieza.

**Estructura:** Texto libre (reviewText) y etiqueta categórica (label: 0=Negativo, 1=Neutro, 2=Positivo).

```
label_map = {  
    'Negative': 0,  
    'Neutral': 1,  
    'Positive': 2  
}  
  
df['label'] = df['sentiment_category'].map(label_map)
```

**Distribución:**

- Positivos: 18.806 (Mayoría dominante).
- Negativos: 2144
- Neutros: 305



**Estrategia de recolección:** ¿Cómo se recopilarán o accederán?

Se desarrollarán scripts en Python utilizando **Selenium** y **BeautifulSoup** para simular navegación y extraer datos.

**Consideraciones éticas:** Privacidad, sesgos, consentimiento, uso responsable

Privacidad: Anonimizar todos los nombres de usuarios (user\_id encriptado) antes del almacenamiento.

Uso: Los datos serán estrictamente para fines académicos. Se respetarán las tasas de limitación (rate limits) de los sitios web para no afectar su servicio.

**Preprocesamiento necesario:** Limpieza, normalización, aumento de datos, etc.

- Eliminación de etiquetas HTML y URLs.
- Manejo de emojis (convertirlos a texto o eliminarlos según impacto).
- Tokenización específica para BERT.
- Normalización de texto (minúsculas, acentos).

## 6. Metodología

La metodología define el conjunto de fases, actividades generales, herramientas tecnológicas y roles requeridos para desarrollar la plataforma de análisis de sentimientos en tiempo real. Este enfoque secuencial garantiza la correcta construcción del pipeline, el entrenamiento de los modelos NLP y la integración del sistema completo, alineándose con los principios de los sistemas inteligentes expuestos por García Serrano (2016) y Galar Pascual (2020) .

### 6.1 Fases del Proyecto

El proyecto se ejecutó en Google Colab siguiendo tres fases documentadas en Notebooks:

### Fase 1: Preparación y Exploración de Datos (Notebook 1)

Se utilizó el dataset de "**Amazon Product Reviews**" en Inglés como base de conocimiento.

- **Limpieza:** Normalización de texto (minúsculas, eliminación de ruido) mediante expresiones regulares.

```
6. Preprocesamiento de texto: Definimos una función de limpieza que nos permite normalizar los textos

import re

def limpiar_texto(texto):
    texto = str(texto).strip().lower()
    texto = re.sub(r'^\w\s', '', texto) # Eliminar puntuación
    return texto

# Aplicar la limpieza al texto
df['reviewText'] = df['reviewText'].apply(limpiar_texto)
```

- **Estratificación:** División 80/20 garantizando la proporción de clases.

```
from sklearn.model_selection import train_test_split

train_df, test_df = train_test_split(
    df, test_size=0.2, random_state=42, stratify=df['label']
)

print(f"Tamaño del conjunto de entrenamiento: {len(train_df)}")
print(f"Tamaño del conjunto de prueba: {len(test_df)}")

... Tamaño del conjunto de entrenamiento: 17004
    Tamaño del conjunto de prueba: 4251
```

### Fase 2: Entrenamiento y Ajuste Fino (Notebook 2)

- **Transfer Learning:** Se ajustaron los pesos de **xlm-roberta-base** para la tarea de clasificación de sentimientos.

- **Configuración:** Entrenamiento durante 1 época con *batch size* de 8, utilizando la GPU T4 de Colab.

- **Persistencia:** Guardado del modelo y tokenizador para su uso

posterior.

```
pruebas = [
    "El producto llegó roto y nadie responde, pésimo servicio.",
    "Está bien, cumple con lo esperado, nada especial.",
    "Excelente calidad, llegó antes de tiempo, totalmente recomendado."
]

for t in pruebas:
    print(f"Texto: {t}")
    print(f"Sentimiento: {predecir_sentimiento(t, umbral_neutro=0.22, mostrar_probs=True)}\n")
```

Texto: El producto llegó roto y nadie responde, pésimo servicio.  
 Probabilidades (neg, neu, pos): [0.54 0.145 0.315]  
 Sentimiento: negativo

Texto: Está bien, cumple con lo esperado, nada especial.  
 Probabilidades (neg, neu, pos): [0.069 0.007 0.924]  
 Sentimiento: positivo

Texto: Excelente calidad, llegó antes de tiempo, totalmente recomendado.  
 Probabilidades (neg, neu, pos): [0.069 0.007 0.924]  
 Sentimiento: positivo

### Fase 3: Evaluación y comparación (Notebook 3)

- **Validación:** Evaluación de métricas sobre el set de prueba.

#### 5. Reporte de clasificación

```
target_names = ["negativo", "neutro", "positivo"]

print("=== Classification Report ===")
print(classification_report(y_true, y_pred, target_names=target_names, digits=4))

acc = accuracy_score(y_true, y_pred)
print("Accuracy:", round(acc, 4))
```

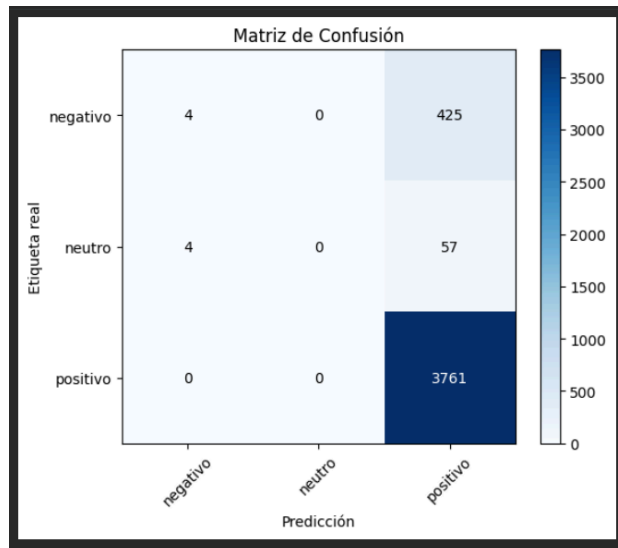
```
=== Classification Report ===
              precision    recall  f1-score   support

negativo      0.8438      0.0629      0.1171        429
neutro         0.0000      0.0000      0.0000         61
positivo      0.8912      0.9997      0.9424       3761

accuracy      0.8908
macro avg     0.5783      0.3542      0.3532        4251
weighted avg  0.8736      0.8908      0.8456        4251
```

Accuracy: 0.8908

- **Matriz de confusión:**



## 6.2 Sobre el Dataset Utilizado

El dataset **Amazon Product Reviews** constituye la base inicial del entrenamiento por las siguientes razones:

- Contiene millones de reseñas organizadas por producto.
- Incluye calificaciones numéricas que permiten derivar etiquetas de sentimiento.
- Tiene variedad de aspectos comentados como calidad, precio y logística.
- Su disponibilidad pública facilita la replicabilidad.

La naturaleza textual del dataset permite aplicar técnicas modernas de NLP y conformar un modelo generalizable antes de integrarlo con datos en tiempo real provenientes de otras fuentes.

## 6.3 Herramientas y Tecnologías

Para la implementación del prototipo funcional se utilizó el siguiente stack tecnológico:

- **Lenguaje y Entorno:**

- Python 3.10+: Lenguaje base para todo el procesamiento.
- Google Colab: Entorno de ejecución en la nube con aceleración por hardware (GPU T4) para el entrenamiento del Transformer.
- **Librerías de Procesamiento y Modelado:**
  - Hugging Face Transformers: Acceso al modelo xlm-roberta-base y la API Trainer.
  - PyTorch: Backend de Deep Learning para el cálculo de tensores y gradientes.
  - Pandas & NumPy: Manipulación de estructuras de datos y cálculos matriciales.
  - Scikit-Learn: Utilizado estrictamente para la división estratificada de datos (train\_test\_split) y cálculo de métricas.
- **Visualización:**
  - Matplotlib / Seaborn: Generación de gráficos de distribución y matrices de confusión.

### 6.3 Métricas de Evaluación

Debido al desbalance de clases (predominancia de reseñas positivas), se seleccionaron métricas que penalizan los sesgos:

- Accuracy (Exactitud): Métrica global para verificar el cumplimiento del objetivo (>85%).
- Weighted F1-Score: Métrica crítica que combina Precisión y Recall ponderando por el soporte de cada clase. Es fundamental asegurar que el buen desempeño no sea solo producto de acertar con la clase mayoritaria.
- Matriz de Confusión: Herramienta visual para diagnosticar errores específicos (ej. cuántos negativos reales fueron confundidos como positivos).

### 6.4 Plan de Validación

La validación del modelo siguió una estrategia rigurosa para evitar el sobreajuste (*overfitting*):

1. División Estratificada (Stratified Split): No se utilizó una división aleatoria simple. Se empleó `stratify=y` para garantizar que la proporción de positivos, negativos y neutros fuera idéntica en el set de entrenamiento (80%) y en el de prueba (20%).
2. Evaluación en Datos No Vistos: Las métricas reportadas se calcularon exclusivamente sobre el set de prueba (4,251 registros) que el modelo nunca vio durante el entrenamiento.
3. Análisis de Errores (Error Analysis): Se realizó una revisión manual de los casos fallidos en el Notebook 3 para entender por qué el modelo confundía ciertas reseñas, llevando a la implementación del "umbral de confianza" para mejorar la clase neutra.

## **6.6 Roles Necesarios**

### ***Equipo completo (proyecto real, escala grande con 10 roles)***

1. Data Engineer (Kafka, Spark, pipelines)
2. Machine Learning Engineer
3. NLP Specialist
4. Data Scientist
5. Backend Developer
6. Frontend/Dashboard Developer
7. Scraping Engineer
8. Cloud/DevOps Engineer
9. Product Owner
10. QA Tester / Validación

### ***Roles para el Piloto (Equipo de 4 personas)***

- **Persona 1:** Responsable de la colección y preparación de datos, incluyendo la obtención del dataset Amazon Product Reviews, limpieza del texto, normalización y construcción del corpus para entrenamiento.
- **Persona 2:** Encargada del desarrollo de los modelos NLP, realizando el fine-tuning de BERT/BETO para clasificación de sentimientos, el modelo de análisis de aspectos (ABSA) y el clasificador de urgencia.
- **Persona 3:** Responsable de la integración del pipeline, configurando Apache Kafka para la ingesta en tiempo real, Spark Streaming para el procesamiento distribuido e integración de los modelos dentro del flujo.
- **Persona 4:** Encargada del dashboard y validación, desarrollando la interfaz de monitoreo, gestionando las métricas de desempeño, verificando la coherencia de los resultados y realizando las pruebas del prototipo final.

Este desglose refleja la complejidad del sistema y permite escalar la solución según los recursos disponibles.

## 7. Viabilidad y Recursos

**Resultados Cuantitativos (Validación Final):** El modelo alcanzó los siguientes indicadores de desempeño en el set de prueba (4,251 muestras en inglés): Accuracy 89.08% (Meta superada) y F1-Score (Weighted): 0.8456.

### **Análisis de Errores (Matriz de Confusión):**

Fortaleza: El modelo distingue con excelente precisión entre reseñas positivas (F1: 0.94) y negativas.

Debilidad Detectada: La clase "Neutra" tiene un desempeño bajo.

Causa: El dataset original tiene menos del 2% de muestras neutras.

Mitigación: Se implementó en el código de inferencia una regla de "Baja Confianza": si la diferencia de probabilidad entre las clases principales es menor a un umbral (0.22), se clasifica como "Neutro".

El proyecto es viable y exitoso para detectar polaridad (satisfacción o insatisfacción) en comentarios en inglés, logrando resultados de alta precisión con recursos computacionales estándar.

### ***Recursos Técnicos para Prototipo***

Dado que se desarrolló un prototipo funcional (MVP), los recursos se optimizaron para validar el concepto sin grandes inversiones iniciales:

**Cómputo:** Se utilizó la infraestructura de **Google Colab (GPU T4 Gratuita)** para el entrenamiento, lo que permitió ajustar el modelo Transformer sin costos de hardware dedicado.

**Modelo Eficiente:** Se optó por **XLM-RoBERTa Base**, que ofrece un equilibrio ideal entre rendimiento (SOTA) y eficiencia computacional.

**Software Open-Source:** Todo el stack se basa en herramientas gratuitas: Python, Hugging Face Transformers, PyTorch y deep-translator.

### ***Recursos Humanos - Equipo de 4 Personas***

Con un equipo compacto de 4 personas, asignaremos roles multifuncionales. El especialista en NLP/Data Scientist liderará el fine-tuning de DistilBERT para español con el dataset de Amazon, desarrollará el modelo de análisis de aspectos (precio, calidad, envío) y adaptará los algoritmos a las particularidades del

e-commerce. Esta persona también se encargará de la preparación y limpieza del dataset.

El desarrollador Full-Stack implementará tanto el backend con FastAPI como un dashboard básico de visualización usando Streamlit o Dash, que permita mostrar los análisis de sentimiento en tiempo real. También integrará el sistema de procesamiento de streams y las conexiones a bases de datos.

El ingeniero DevOps configura el entorno de desarrollo, los pipelines de entrenamiento de modelos, y desplegará el prototipo en una plataforma cloud económica. Implementará monitoreo básico y asegurará que el sistema sea reproducible.

El coordinador técnico asumirá roles de project management junto con responsabilidades de arquitectura del sistema, pruebas de integración y comunicación con posibles usuarios pilotos. Esta persona también investigará alternativas para enriquecer el dataset.

### ***Presupuesto Estimado para Prototipo***

Para el desarrollo del prototipo en 3-4 meses, el presupuesto se estima en \$25,000-\$35,000. Los salarios del equipo de 4 personas representarán aproximadamente \$18,000-\$25,000. Los costos de infraestructura cloud se mantendrán bajo \$500 mensuales utilizando servicios básicos y aprovechando créditos gratuitos de proveedores cloud. Las herramientas de desarrollo y APIs tendrán un costo máximo de \$300 mensuales, priorizando siempre opciones open-source. Se destinarán \$2,000-\$3,000 específicamente para abordar el desafío del dataset, incluyendo posibles servicios de etiquetado de datos complementarios o herramientas de aumento de datos.

### ***Riesgos Identificados y Plan de Mitigación***

El riesgo más crítico identificado es la limitación del dataset de Amazon para capturar matices, particularmente variantes regionales y jerga específica de e-commerce en diferentes países. Para mitigar esto, implementaremos una estrategia de tres fases: primero, aumentaremos el dataset existente con técnicas de data augmentation específicas para español; segundo, complementaremos con un pequeño conjunto de datos etiquetados manualmente (500-1,000 muestras) que incluya expresiones regionales; tercero, desarrollaremos un módulo de retroalimentación en el prototipo para que los usuarios puedan corregir clasificaciones erróneas, creando así un dataset de mejora continua.

El riesgo de rendimiento en tiempo real con recursos limitados se mitigará optimizando el modelo desde el inicio: usaremos cuantización para reducir el tamaño del modelo, implementaremos caching agresivo de consultas similares, y priorizaremos la precisión sobre la cobertura en esta fase inicial. Para el riesgo de integración futura, diseñamos una arquitectura modular que permite cambiar componentes fácilmente, como el conector de datos o el modelo de análisis, sin reescribir todo el sistema.

Finalmente, estableceremos métricas de éxito claras para el prototipo (precisión mínima del 85% en clasificación básica, latencia inferior a 2 segundos) que nos permitan validar si el concepto justifica una inversión mayor en la siguiente fase, donde sí consideraríamos recursos más robustos como GPU.

**Conclusión:** La estrategia híbrida es altamente viable, permite desplegar una solución robusta para el mercado hispanohablante de manera inmediata, aprovechando la calidad de los datos globales y la potencia de los Transformers modernos.

## 8. Referencias

García Serrano, A. (2016). *Inteligencia artificial: Fundamentos, práctica y aplicaciones* (2.<sup>a</sup> ed.). RC Libros.

Galar Pascual, D. (2020). *Artificial Intelligence Tools: Decision Support Systems in Condition Monitoring and Diagnosis*. Elsevier.

Vaswani, A., et al. (2017). "Attention Is All You Need." *Advances in Neural Information Processing Systems*.

Kritanjali, J. (2020). Amazon reviews. kaggle.  
<https://www.kaggle.com/datasets/kritanjali/jain/amazon-reviews>

Kreimer, P. (2020). "Real-time Sentiment Analysis with Apache Kafka and Spark Streaming." *O'Reilly Media Technical Blog*.