

# "The Smart Feedback API"

Tiempo de resolución: 3/4 días.

## Introducción y contexto

¡Hola!

Si estás leyendo esto es porque tu perfil nos resultó interesante y queremos conocerte un poco mejor a través de la forma en que pensás y resolvés problemas con código.

En **Arbusta**, desarrollamos soluciones donde la Inteligencia Artificial y los equipos humanos trabajan en conjunto. Actualmente, el equipo de Operaciones recibe diariamente feedback de usuarios a través de múltiples canales, y necesita una forma eficiente de **identificar y priorizar los casos críticos**.

Tu desafío será construir una solución que permita **analizar automáticamente el sentimiento** de estos mensajes (Positivo / Negativo / Neutral), ayudando a agilizar la toma de decisiones.

## Tu misión

Diseñar, construir y entregar una **aplicación en Python** que exponga un **servicio de análisis de sentimiento** a través de una API.

## El desafío técnico

### 1. Insumos

Tu punto de partida es el archivo [reviews.csv](#) (provisto), que contiene una lista de comentarios recibidos por la empresa.

### 2. Requerimientos funcionales

La solución debe cubrir dos componentes principales:

#### a) Motor de Análisis (NLP Core)

Desarrollar la lógica encargada de clasificar un texto de entrada.

La salida debe incluir, al menos:

- **Sentiment:** Positivo, Negativo o Neutral
- **Score:** un valor que indique la confianza de la predicción

La elección del enfoque o modelo queda a tu criterio.

### b) API REST (Ingeniería de software)

Exponer el motor de análisis a través de una API REST  
(recomendamos **FastAPI** o **Flask**, aunque no es excluyente).

La API debe contar con al menos un endpoint, por ejemplo:

- `POST /analyze`

El endpoint debe:

- Recibir un JSON con un texto de entrada
- Devolver el resultado del análisis de forma estructurada

## Entrega

- **Repositorio:**

El código debe estar alojado en un **repositorio público** de GitHub.

- **Historial de commits:**

Valoramos poder ver el proceso de desarrollo y evolución de la solución  
(*evitá subir un único commit al final*).

- **Documentación:**

El repositorio debe incluir un **README.md** claro que explique:

- Las decisiones técnicas tomadas
- Instrucciones paso a paso para ejecutar el proyecto

- **Envío:**

Compartí el link de tu repositorio respondiendo a este correo.