

# Mood Tracker - Explicación del código para generar el gráfico

*Fecha de generación: 16/04/2025 12:53*

Este documento explica paso a paso el código utilizado para generar un gráfico con Chart.js a partir de datos JSON recibidos por fetch().

1. `.then(response => response.text()):`

Lee la respuesta del servidor como texto (por si no es JSON bien formateado). Esto evita errores inesperados al hacer `JSON.parse()` directamente.

2. `.then(text => { ... }):`

Una vez recibido el texto, se intenta convertirlo a JSON utilizando `JSON.parse(text)`.

3. `try { const data = JSON.parse(text); }:`

Si el texto es un JSON válido, se convierte en un objeto JavaScript llamado `data`. Este contiene un array de objetos como:

```
[{"estado_animo": "feliz", "total": 3}, {"estado_animo": "triste", "total": 2}]
```

4. `const contenedor = document.getElementById("grafico-container");`

Se selecciona el div del HTML donde irá el gráfico.

5. `const labels = data.map(item => item.estado_animo);`

`const valores = data.map(item => item.total);`

Se crean dos arrays: uno con los nombres de las emociones (labels) y otro con los totales (valores).

6. `if (window.graficoInstancia) { window.graficoInstancia.destroy(); }`

Antes de dibujar un gráfico nuevo, se destruye el anterior para evitar duplicados.

7. `const ctx = document.getElementById("grafico-emociones").getContext("2d");`

Se obtiene el contexto 2D del canvas donde se dibujará el gráfico.

8. `new Chart(ctx, {...})`:

Se crea el gráfico con Chart.js, tipo "bar", usando los labels y valores generados.

9. `catch (e)`:

Si `JSON.parse()` falla, se captura el error y se imprime en consola junto con el texto que no se pudo parsear.

Este flujo asegura que:

- Siempre se limpien errores de respuesta mal formateada.
- El gráfico se dibuje correctamente con los datos que devuelve el backend.
- La experiencia del usuario sea fluida y sin errores visibles.