

Lab Cloud – Parte 3: Crea un Frontend HTML que use tu API RESTful de AWS

Objetivo: Construir una página web básica que permita enviar reseñas a la API que creamos con API Gateway + Lambda + DynamoDB

Nivel: Principiante a intermedio

Tabla de contenidos

- Escenario práctico
- ¿Qué vamos a construir?
- Estructura de archivos
- Código HTML + JS explicado paso a paso
- Prueba y validación
- Retos comunes
- Ideas para mejorar





Escenario práctico

CloudFizz ahora quiere que sus usuarios puedan enviar reseñas desde una página web. Ya tienen su API funcional, solo necesitan una interfaz para conectarse con ella. Vamos a crear un formulario HTML que use fetch para enviar datos a la API sin necesidad de frameworks.

🚆 ¿Qué vamos a construir?

Una página simple que incluye:

- Un formulario con campos usuario y comentario
- Un botón para enviar la reseña
- Un mensaje de éxito o error al enviar
- Todo en una sola página HTML estática

Estructura de archivos

/frontend

- index.html

Reemplaza con tu URL

Antes de continuar, anota tu endpoint de API Gateway, algo así:

https://abc123.execute-api.us-east-1.amazonaws.com/dev/resenas



Lo usaremos en el código como valor de API_URL.

Código HTML y JavaScript

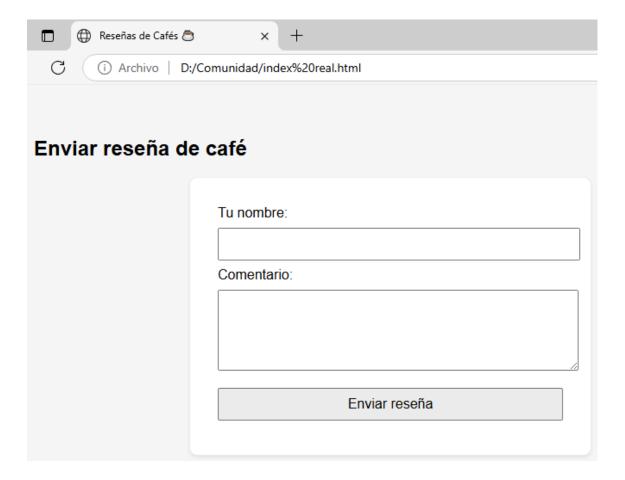
Guarda este archivo como index.html en una carpeta:

```
<!DOCTYPE html>
<html lang="es">
<head>
 <meta charset="UTF-8">
 <title>Reseñas de Cafés <a></title></a>
  body { font-family: sans-serif; padding: 2rem; background: #f8f8f8; }
  form { background: white; padding: 2rem; border-radius: 8px; max-width: 400px; margin: auto; box-shadow: 0
2px 6px rgba(0,0,0,0.1); }
  input, textarea, button { width: 100%; margin: 0.5rem 0; padding: 0.5rem; font-size: 1rem; }
  .success { color: green; }
  .error { color: red; }
 </style>
</head>
<body>
 <h2>Enviar reseña de café</h2>
 <form id="resenaForm">
  <label for="usuario">Tu nombre:</label>
  <input type="text" id="usuario" required />
  <label for="comentario">Comentario:</label>
  <textarea id="comentario" rows="4" required></textarea>
  <button type="submit">Enviar reseña</button>
 </form>
 <script>
  const API_URL = "https://<tu-api-id>.execute-api.<region>.amazonaws.com/stage/resource"; // ← Reemplaza
  document.getElementById("resenaForm").addEventListener("submit", async function(e) {
   e.preventDefault();
```



```
const usuario = document.getElementById("usuario").value;
   const comentario = document.getElementById("comentario").value;
   const mensajeEl = document.getElementById("mensaje");
    const response = await fetch(API_URL, {
      method: "POST",
     headers: { "Content-Type": "application/json" },
     body: JSON.stringify({ usuario, comentario })
    const data = await response.json();
    if (response.ok) {
      mensajeEl.textContent = data.mensaje;
      mensajeEl.className = "success";
      document.getElementById("resenaForm").reset();
      mensajeEl.textContent = data.error || "Error al enviar la reseña.";
      mensajeEl.className = "error";
    }
   } catch (error) {
    mensajeEl.textContent = "Error de conexión con la API.";
    mensajeEl.className = "error";
  });
 </script>
</body>
</html>
```





- 1. Guarda el archivo como index.html
- 2. Ábrelo en tu navegador (doble clic o arrastrar al navegador)
- 3. Llena el formulario y haz clic en Enviar reseña
- 4. Verifica el mensaje y revisa en **DynamoDB** si la reseña se guardó



a Retos comunes

Problema	Solución
CORS Error	Asegúrate de habilitar CORS en el método POST en API Gateway
El formulario no hace nada	Verifica si fetch está lanzando errores en la consola del navegador
La reseña no aparece en DynamoDB	Verifica el endpoint y revisa los logs en Lambda

🚀 Ideas para mejorar

- Mostrar todas las reseñas (GET + render en HTML)
- Validar campos más estrictamente
- Agregar un campo de calificación (1–5 estrellas)
- Estilizar con Bootstrap o Tailwind
- Subir el frontend a S3 como sitio estático público