Implementar un chat Bot que use un servicio gratuito de IA para dar respuestas como psicólogo es posible utilizando HTML, CSS y JavaScript. Aquí tienes una guía paso a paso para lograrlo, basada en las mejores prácticas y ejemplos disponibles[1](https://www.toolify.ai/es/ai-news-es/tutorial-crea-un-chatbot-de-estilo-de-mensajera-con-javascript-1086617)[5](https://jvcodes.com/responsive-ai-chatbot-using-html-css-and-javascript/)2.

**Paso 1: Estructura HTML del Chatbot**

Crea un archivo index.html con la estructura básica del chatbot:

xml

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title>Chatbot Psicólogo</title>

<link rel="stylesheet" href="chatbot.css">

</head>

<body>

<div class="chat-container">

<div id="chat-box"></div>

<div class="input-area">

<input type="text" id="user-input" placeholder="Escribe tu pregunta...">

<button id="send-btn">Enviar</button>

</div>

</div>

<script src="chatbot.js"></script>

</body>

</html>

**Paso 2: Estilizar el Chatbot (CSS)**

Crea un archivo chatbot.css:

css

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.chat-container {

width: 350px;

background: #fff;

border-radius: 10px;

box-shadow: 0 0 10px rgba(0,0,0,0.1);

overflow: hidden;

}

#chat-box {

height: 400px;

overflow-y: auto;

padding: 10px;

background: #f9f9f9;

}

.input-area {

display: flex;

padding: 10px;

background: #fff;

border-top: 1px solid #ddd;

}

#user-input {

flex: 1;

padding: 10px;

border: 1px solid #ddd;

border-radius: 5px;

}

#send-btn {

margin-left: 10px;

padding: 10px 15px;

background: #4CAF50;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

**Paso 3: Implementar la lógica del Chatbot (JavaScript)**

Para un chatbot básico, puedes usar respuestas predefinidas o, si quieres usar un servicio gratuito de IA, puedes llamar a una API pública como la de ChatGPT o alguna alternativa gratuita.  
**Nota:** Muchas APIs de IA gratuitas tienen límites y requieren clave de API, pero puedes usar algunas opciones de prueba.

Este ejemplo muestra una integración básica con un servicio de IA genérico (sustituye la URL y el prompt según el servicio que elijas):

js

*// Archivo chatbot.js*

*// Configuración básica para simular una API de IA (puedes reemplazar por una real)*

**async** **function** getAIResponse(userMessage) {

*// Aquí puedes poner la lógica para llamar a una API real de IA*

*// Ejemplo con fetch (sustituye la URL y los datos según la API que uses)*

*/\**

*const response = await fetch('https://api.ejemplo.com/chat', {*

*method: 'POST',*

*headers: { 'Content-Type': 'application/json' },*

*body: JSON.stringify({ prompt: userMessage })*

*});*

*const data = await response.json();*

*return data.reply;*

*\*/*

*// Simulación local mientras pruebas (puedes borrar esto cuando integres la API)*

**return** "...(respuesta simulada de IA) " +

"Como psicólogo, te recomiendo reflexionar sobre tus emociones y buscar apoyo profesional si lo necesitas.";

}

**function** addMessageToChat(message, isUser) {

**const** chatBox = document.getElementById('chat-box');

**const** messageDiv = document.createElement('div');

messageDiv.className = isUser ? 'user-message' : 'bot-message';

messageDiv.textContent = message;

chatBox.appendChild(messageDiv);

chatBox.scrollTop = chatBox.scrollTop + messageDiv.offsetHeight;

}

document.getElementById('send-btn').addEventListener('click', **async** () => {

**const** userInput = document.getElementById('user-input');

**const** userMessage = userInput.value.trim();

**if** (!userMessage) **return**;

addMessageToChat(userMessage, true);

userInput.value = '';

*// Pide a la IA (simulada o real) que responda como psicólogo*

**const** aiResponse = **await** getAIResponse(userMessage);

addMessageToChat(aiResponse, false);

});

*// Opcional: Permitir enviar con Enter*

document.getElementById('user-input').addEventListener('keypress', (e) => {

**if** (e.key === 'Enter') {

document.getElementById('send-btn').click();

}

});

**Paso 4: Personalizar el Prompt para Psicología**

Cuando integres una API real de IA (como ChatGPT, Gemini, etc.), asegúrate de enviar un prompt específico para que la IA responda como psicólogo. Por ejemplo:

js

**const** promptPsicologo = "Eres un psicólogo profesional. Responde a las preguntas del usuario con empatía y consejos útiles.";

Y luego lo envías junto con el mensaje del usuario:

js

**const** response = **await** fetch('https://api.ejemplo.com/chat', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({ prompt: promptPsicologo + userMessage })

});

**Paso 5: Despliega y prueba**

* **Guarda los archivos en una carpeta.**
* **Abre index.html en tu navegador.**
* **Para usar una API de IA real, regístrate en un servicio gratuito (por ejemplo, OpenAI, Gemini, etc.) y reemplaza la función getAIResponse con la llamada real a la API.**

**Resumen**

* **HTML:** Define la estructura del chat.
* **CSS:** Estiliza el chatbot para que sea atractivo y funcione en móviles.
* **JavaScript:** Maneja la interacción del usuario y la integración con una API de IA (usando el prompt de psicólogo).
* **IA:** Usa un servicio gratuito (o de prueba) para generar respuestas como psicólogo.