

```

document.addEventListener('DOMContentLoaded', () => {
  // Referências aos elementos do DOM
  const userInput = document.getElementById('user-input');
  const sendButton = document.getElementById('send-button');
  const messagesDisplay = document.getElementById('messages-display');

  // Adiciona event listeners para enviar mensagem
  sendButton.addEventListener('click', sendMessage);
  userInput.addEventListener('keypress', handleKeyPress);
  userInput.addEventListener('input', adjustTextareaHeight);

  /**
   * Função para enviar uma mensagem do usuário.
   */
  function sendMessage() {
    const messageText = userInput.value.trim();
    if (messageText === "") return; // Não envia mensagens vazias

    // Adiciona a mensagem do usuário na tela
    addMessage(messageText, 'user');

    // Limpa o input e reseta a altura do textarea
    userInput.value = "";
    userInput.style.height = 'auto';

    // Simula uma resposta da IA após um pequeno atraso
    // (Aqui você integraria sua API de IA real)
    setTimeout(() => {
      const iaResponse = "Essa é uma resposta simulada do Mig para: " + messageText +
        "...";
      addMessage(iaResponse, 'ia');
    }, 1000); // Responde após 1 segundo
  }

  /**
   * Função para lidar com o evento de pressionar tecla no input.
   * @param {KeyboardEvent} e - O evento de teclado.
   */
  function handleKeyPress(e) {
    if (e.key === 'Enter' && !e.shiftKey) { // Envia ao apertar Enter, a menos que Shift+Enter
      e.preventDefault(); // Evita nova linha no textarea ao enviar
      sendMessage();
    }
  }

  /**
   * Adiciona uma mensagem ao display do chat.
   * @param {string} text - O texto da mensagem.

```

```

* @param {'user' | 'ia'} type - O tipo da mensagem ('user' ou 'ia').
*/
function addMessage(text, type) {
  const messageDiv = document.createElement('div');
  messageDiv.classList.add('message', `${type}-message`);

  const avatarDiv = document.createElement('div');
  avatarDiv.classList.add('avatar');
  avatarDiv.innerHTML = type === 'user' ? '<i class="fas fa-user"></i>' : '<i class="fas fa-robot"></i>';

  const contentDiv = document.createElement('div');
  contentDiv.classList.add('message-content');

  // Adiciona o nome do remetente para mensagens da IA
  if (type === 'ia') {
    const senderName = document.createElement('div');
    senderName.classList.add('sender-name');
    senderName.textContent = 'Mig'; // Nome da sua IA
    contentDiv.appendChild(senderName);
  }

  // Adiciona o texto da mensagem
  contentDiv.innerHTML += text;

  // Monta a estrutura da mensagem (avatar à esquerda para IA, à direita para usuário)
  if (type === 'user') {
    messageDiv.appendChild(contentDiv);
    messageDiv.appendChild(avatarDiv);
  } else {
    messageDiv.appendChild(avatarDiv);
    messageDiv.appendChild(contentDiv);
  }

  messagesDisplay.appendChild(messageDiv);
  messagesDisplay.scrollTop = messagesDisplay.scrollHeight; // Rola para o final da
  conversa
}

/**
* Ajusta a altura do textarea de acordo com o conteúdo.
*/
function adjustTextareaHeight() {
  userInput.style.height = 'auto'; // Reseta a altura para calcular corretamente
  userInput.style.height = userInput.scrollHeight + 'px'; // Ajusta para o conteúdo
}
});

```