  
  
**Instrucciones de implementación externa:**

***• Agregar la referencia a google icon font en la sección <head> del sitio web:***

\* Copie y pegue el siguiente código dentro de la sección <head> de su página para incluir los estilos necesarios del widget:

**1 .** <!--Import Google Icon Font-->

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

***• Agregar el contenedor del widget en el <body> del sitio web:***

\* Incluya el siguiente contenedor HTML en la sección <body> de la página, en el lugar donde desea que se muestre el widget:

**2.** <!-- Contenedor Widget -->

<div id="contenedorWidget"></div>

**• Añadir el script JavaScript para la funcionalidad del widget:**

**\*** Incluya el siguiente script al final de la sección <body> del sitio web, antes del cierre de la etiqueta </body>. Esto asegurará que el script necesario para el widget se cargue y funcione correctamente; configurar la constante URL\_BASE con la URL base de la aplicación (APP\_URL):

**3.**  <!-- Import js y css -->

<script>

// Definición de constantes para las URLs

const URL\_BASE = 'APP\_URL';

const URL\_CSS = `${URL\_BASE}/chatWeb.css`;

const URL\_JS = `${URL\_BASE}/chatWeb.js`;

// Función para cargar con timeout

function cargarConTimeout(cargarFn, src, timeout = 1500) {

return new Promise((resolve, reject) => {

let el;

const timer = setTimeout(() => {

if (el && el.parentNode) el.parentNode.removeChild(el);

reject(new Error('Timeout al cargar: ' + src));

}, timeout);

cargarFn(src)

.then(() => {

clearTimeout(timer);

resolve();

})

.catch((err) => {

clearTimeout(timer);

reject(err);

});

// Guardar el elemento para poder removerlo si hay timeout

if (cargarFn === cargarCSS) {

el = document.querySelector(`link[href='${src}']`);

}

if (cargarFn === cargarJS) {

el = document.querySelector(`script[src='${src}']`);

}

});

}

// Funcion para cargar el css

function cargarCSS(src) {

return new Promise((resolve, reject) => {

const link = document.createElement('link');

link.rel = 'stylesheet';

link.href = src;

link.onload = resolve;

link.onerror = reject;

document.head.appendChild(link);

// Devuelve el elemento para poder removerlo si hay timeout

resolve.el = link;

});

}

// Funcion para cargar el js

function cargarJS(src) {

return new Promise((resolve, reject) => {

const script = document.createElement('script');

script.src = src;

script.onload = resolve;

script.onerror = reject;

document.head.appendChild(script);

});

}

// Carga con timeout de 3 segundos

cargarConTimeout(cargarCSS, URL\_CSS, 1500)

.then(() => cargarConTimeout(cargarJS, URL\_JS, 1500))

.then(() => {

if (window.WidgetChat && typeof window.WidgetChat.init === 'function' && document.getElementById('contenedorWidget')) {

window.WidgetChat.init();

console.log('✅ Widget cargado exitosamente');

} else {

console.error('❌ El widget o el contenedor no están disponibles.');

}

})

.catch((error) => {

console.error('❌ Error al cargar recursos del widget, por favor contactenos:', error);

// Configuración de reintentos

const intervaloReintento = 15000; // 15 segundos

const tiempoMaximo = 300000; // 5 minutos

const tiempoInicio = Date.now();

// Funcion para intentar cargar nuevamente los recursos

function intentarCarga() {

const tiempoTranscurrido = Date.now() - tiempoInicio;

if (tiempoTranscurrido >= tiempoMaximo) {

console.error('❌ Tiempo máximo de reintentos alcanzado');

return;

}

console.log('🔄 Reintentando cargar el widget...');

cargarConTimeout(cargarCSS, URL\_CSS, 1500)

.then(() => cargarConTimeout(cargarJS, URL\_JS, 1500))

.then(() => {

if (window.WidgetChat && typeof window.WidgetChat.init === 'function' && document.getElementById('contenedorWidget')) {

window.WidgetChat.init();

console.log('✅ Widget cargado exitosamente');

} else {

throw new Error('El widget o el contenedor no están disponibles');

}

})

.catch((error) => {

console.error('❌ Error en reintento:', error);

setTimeout(intentarCarga, intervaloReintento);

});

}

// Iniciar el primer reintento

setTimeout(intentarCarga, intervaloReintento);

});

</script>

**• Consideraciones Adicionales:**

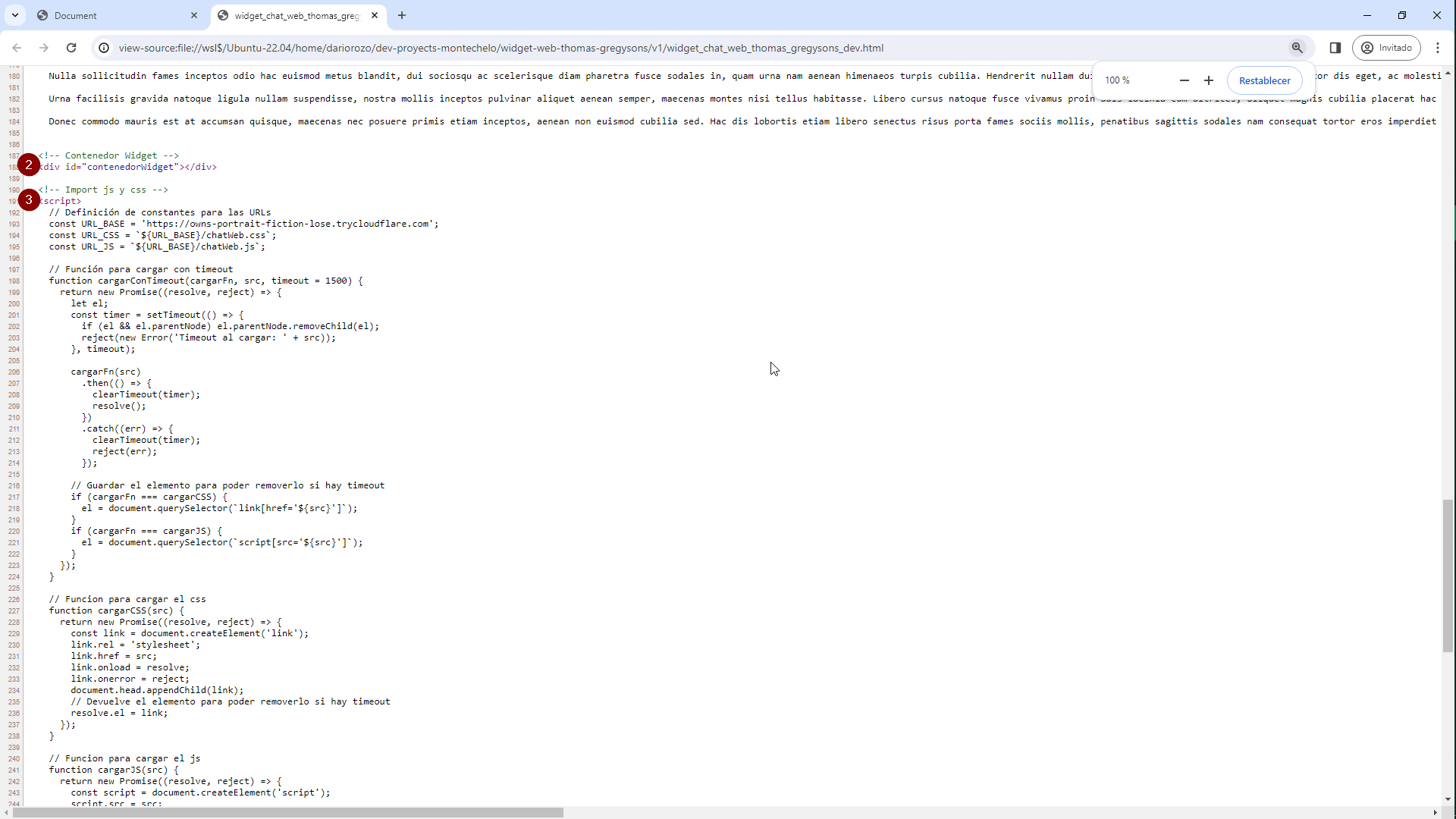
\* Asegúrese de que los archivos referenciados sean accesibles desde la red donde está alojada la página oficial en la cual se desea integrar el widget.

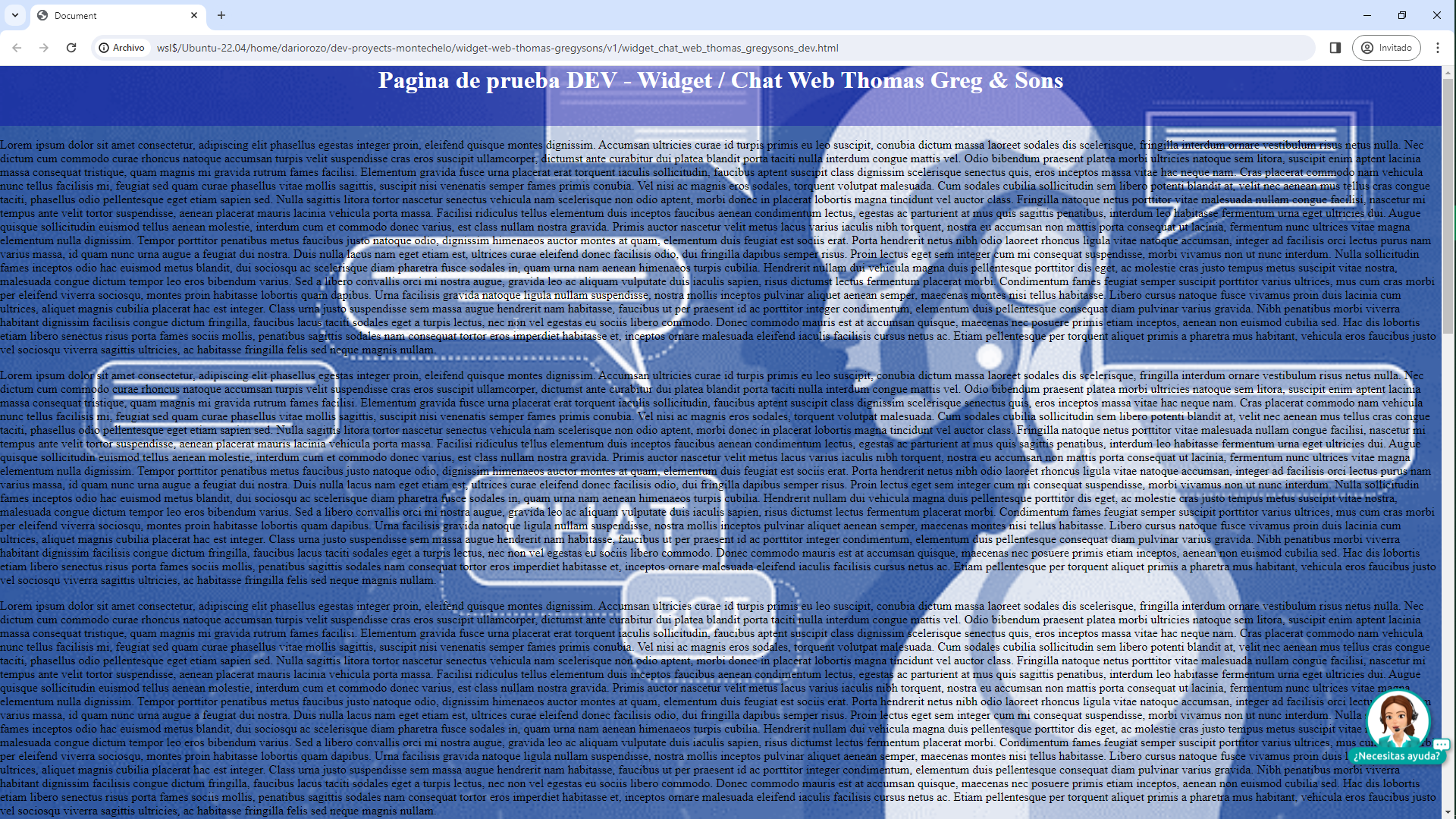
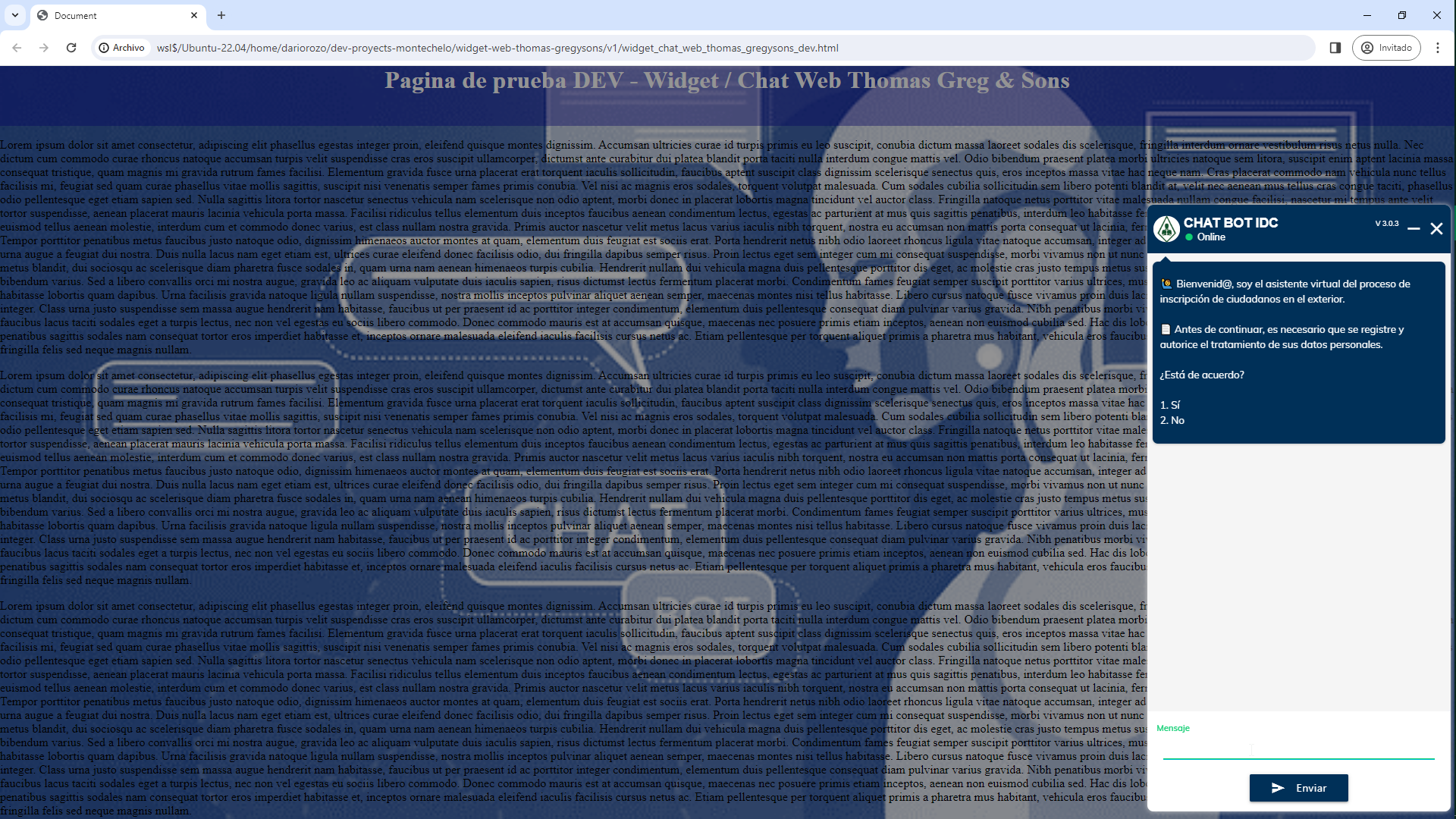
\* Verifique que no existan conflictos con otros estilos o scripts en el sitio web que puedan afectar el diseño o el funcionamiento del widget.

\* Se recomienda realizar pruebas en un ambiente de desarrollo antes de implementar los cambios en el sitio de producción.

\* Por favor, no duden en contactarnos si necesitan mayor información o soporte técnico durante el proceso de integración.

**Ejemplo visual de implementación**

**Ejemplo de visualización UI  
  
**