



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.');

```

```

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

```
1 <!-- Import de HTML -->
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <!-- Import Google Icon Font -->
9   <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
10
11 <style>
12   html,
13   body {
14     height: 100%;
15     /* Asegura que html y body ocupen el 100% de la altura */
16     margin: 0;
17     /* Elimina márgenes por defecto */
18   }
19
20   body {
21     background-image: url('https://oams-portrait-fiction-lose.trycloudflare.com/images/imagen-corporativa/fondo.png');
22     /* Línea a la imagen de fondo */
23     background-repeat: no-repeat;
24     /* No repetir la imagen */
25     background-size: cover;
26     /* Ajustar la imagen para cubrir todo el fondo */
27     background-position: center;
28     /* Centrar la imagen */
29     background-attachment: fixed;
30     /* Hacer que la imagen de fondo quede fija al hacer scroll */
31   }
32 </style>
33 </head>
34 <body>
35   <div class="titlePagina" style="background: #000070; height: 8%;>
36     <h1 style="text-align: center; margin: 0px; color: rgb(255, 255, 255);"><strong>Pagina de prueba DEV - Widget / Chat Web Thomas Greg & Sons</strong></h1>
37   </div>
38
39   <!-- Lorem ipsum dolor sit amet consectetur, adipiscing elit phasellus egestas integer proin, eleifend quisque montes dignissim. Accumam ultricies curae id turpis primis eu leo suscipit, conubia dictum massa laoreet sodales dis scelerisque, fringilla interdum
40   Elementum gravida fusce urna placerat erat torquent laculis sollicitudin, faucibus aptent suscipit class dignissim scelerisque senectus quis, eros inceptos massa vitae hac neque nam. Cras placerat commodo nam vehicula nunc tellus facilisis mi, feugiat sed q
41   Nulla sagittis litora tortor nascetur senectus vehicula nam scelerisque non odio aptent, morbi donec in placerat lobortis magna tincidunt vel auctor class. Fringilla natoque netus porttitor vitae malesuada nullam congue facilisi, nascetur mi tempus ante vel
42   Augue quisque sollicitudin euismod tellus aenean molestie, interdum cum et commodo donec varius, est class nullam nostra gravida. Primis auctor nascetur velit metus lacus varius laculis nibh torquent, nostra eu accumam non mattis porta consequat ut lacina
43   Porta hendrerit netus nibh odio laoreet rhoncus ligula vitae natoque accumam, integer ad facilisis orci lectus purus nam varius massa, id quam nunc urna augue a feugiat dui nostra. Duis nulla lacus nam eget etiam est, ultrices curae eleifend donec facilisi
44   Nulla sollicitudin fames inceptos odio hac euismod metus blandit, dui sociosq ac scelerisque diam pharetra fusce sodales in, quam urna nam aenean himenaeos turpis cubilia. Hendrerit nullam dui vehicula magna dui pellentesque porttitor dis eget, ac molesti
45   Urna facilisis gravida natoque ligula nullam suspendisse, nostra mollis inceptos pulvinar aliquet aenean semper, maecenas montes nisi tellus habitasse. Libero cursus natoque fusce vivamus proin dui lacinia cum ultrices, aliquet magnis cubilia placerat hac
46   Donec commodo mauris est at accumam quisque, maecenas nec posuere primis etiam inceptos, aenean non euismod cubilia sed. Hac dis lobortis etiam libero senectus risus porta fames sociis mollis, penatibus sagittis sodales nam consequat tortor eros imperdiet
47   Cras ipsum dolor sit amet consectetur, adipiscing elit phasellus egestas integer proin, eleifend quisque montes dignissim. Accumam ultricies curae id turpis primis eu leo suscipit, conubia dictum massa laoreet sodales dis scelerisque, fringilla interdum
48   Elementum gravida fusce urna placerat erat torquent laculis sollicitudin, faucibus aptent suscipit class dignissim scelerisque senectus quis, eros inceptos massa vitae hac neque nam. Cras placerat commodo nam vehicula nunc tellus facilisis mi, feugiat sed q
49   Nulla sagittis litora tortor nascetur senectus vehicula nam scelerisque non odio aptent, morbi donec in placerat lobortis magna tincidunt vel auctor class. Fringilla natoque netus porttitor vitae malesuada nullam congue facilisi, nascetur mi tempus ante vel
50   Augue quisque sollicitudin euismod tellus aenean molestie, interdum cum et commodo donec varius, est class nullam nostra gravida. Primis auctor nascetur velit metus lacus varius laculis nibh torquent, nostra eu accumam non mattis porta consequat ut lacina -->
51
52 <!-- Contenedor Widget -->
53 <div id="contenedorWidget"></div>
54
55 <!-- Import JS y CSS -->
56 <script>
57   // Definición de constantes para las URLs
58   const URL_BASE = 'https://oams-portrait-fiction-lose.trycloudflare.com';
59   const URL_CSS = `${URL_BASE}/chatWeb.css`;
60   const URL_JS = `${URL_BASE}/chatWeb.js`;
61
62   // Función para cargar con timeout
63   function cargarConTimeout(cargaFn, src, timeout = 1500) {
64     return new Promise((resolve, reject) => {
65       let el;
66       const timer = setTimeout(() => {
67         if (el && el.parentNode) el.parentNode.removeChild(el);
68         reject(new Error('Timeout al cargar: ' + src));
69       }, timeout);
70
71       cargaFn(src)
72         .then(() => {
73           clearTimeout(timer);
74           resolve();
75         })
76         .catch((err) => {
77           clearTimeout(timer);
78           reject(err);
79         });
80     });
81
82     // Guardar el elemento para poder removerlo si hay timeout
83     if (cargaFn === cargarCSS) {
84       el = document.querySelector('link[href="' + src + '"]');
85     }
86     if (cargaFn === cargarJS) {
87       el = document.querySelector('script[src="' + src + '"]');
88     }
89   }
90
91   // Función para cargar el CSS
92   function cargarCSS(src) {
93     return new Promise((resolve, reject) => {
94       const link = document.createElement('link');
95       link.rel = 'stylesheet';
96       link.href = src;
97       link.onload = resolve;
98       link.onerror = reject;
99       document.head.appendChild(link);
100      // Remover el elemento para poder removerlo si hay timeout
101      resolve.el = link;
102    });
103  }
104
105  // Función para cargar el JS
106  function cargarJS(src) {
107    return new Promise((resolve, reject) => {
108      const script = document.createElement('script');
109      script.src = src;
110    });
111  }
112
113  // Cargar CSS y JS
114  cargarConTimeout(cargarCSS, URL_CSS);
115  cargarConTimeout(cargarJS, URL_JS);
116
117  // Inicializar el widget
118  const widget = new ChatWidget();
119  widget.init();
120
121  // Cerrar el script
122  </script>
```


Ejemplo de visualización UI

