# Integración de EmailJS en el Formulario de Contacto

Voy a mostrarte cómo integrar EmailJS para que tu formulario de contacto funcione sin necesidad de un backend propio. EmailJS es un servicio que permite enviar correos directamente desde el frontend usando plantillas.

## Paso a paso de la implementación:

```jsx

import React, { useState, useRef } from 'react';

import { PhoneIcon, EnvelopeIcon, MapPinIcon } from '@heroicons/react/24/outline';

import { FaWhatsapp, FaInstagram, FaFacebook, FaTwitter } from 'react-icons/fa';

import emailjs from '@emailjs/browser'; // Importa la librería de EmailJS

const Contact = () => {

const form = useRef(); // Referencia para el formulario

const [isSubmitting, setIsSubmitting] = useState(false);

const [submitMessage, setSubmitMessage] = useState({ type: '', text: '' });

// Función para manejar el envío del formulario

const handleSubmit = (e) => {

e.preventDefault();

setIsSubmitting(true);

setSubmitMessage({ type: '', text: '' });

// Reemplaza con tus credenciales de EmailJS

const serviceId = 'TU\_SERVICE\_ID';

const templateId = 'TU\_TEMPLATE\_ID';

const publicKey = 'TU\_PUBLIC\_KEY';

emailjs.sendForm(serviceId, templateId, form.current, publicKey)

.then(() => {

setSubmitMessage({ type: 'success', text: 'Mensaje enviado con éxito!' });

form.current.reset(); // Limpia el formulario

})

.catch((error) => {

setSubmitMessage({

type: 'error',

text: 'Hubo un error al enviar el mensaje. Por favor, inténtalo de nuevo.'

});

console.error('Error al enviar el correo:', error);

})

.finally(() => {

setIsSubmitting(false);

});

};

return (

<section id="contacto" className="py-16 md:py-20 bg-gradient-to-b from-green-50 to-gray-800 relative overflow-hidden">

<div className="absolute inset-0 overflow-hidden">

<div className="absolute top-0 left-1/3 w-72 h-72 md:w-96 md:h-96 bg-emerald-200 rounded-full opacity-10 animate-pulse"></div>

<div className="absolute bottom-0 right-1/3 w-64 h-64 md:w-80 md:h-80 bg-amber-200 rounded-full opacity-10 animate-pulse"></div>

</div>

<div className="relative z-10 max-w-6xl mx-auto px-4 sm:px-6 lg:px-8">

<div className="text-center mb-12 md:mb-16">

<h2 className="text-3xl md:text-4xl lg:text-5xl font-bold text-gray-800 mb-4 md:mb-6">

CONTACTO

</h2>

<p className="text-lg md:text-xl text-gray-600 max-w-3xl mx-auto">

Estamos aquí para ayudarte a crear el espacio zen perfecto

</p>

</div>

<div className="grid grid-cols-1 lg:grid-cols-2 gap-8 lg:gap-10">

{/\* Columna de información (sin cambios) \*/}

{/\* ... tu código existente para la columna de información ... \*/}

{/\* Columna del formulario con EmailJS \*/}

<div className="flex flex-col h-full">

<div className="bg-white rounded-xl md:rounded-2xl shadow-lg md:shadow-xl p-6 md:p-8 flex-grow">

<h3 className="text-xl md:text-2xl font-bold text-gray-800 mb-4 md:mb-6">

Envíanos un Mensaje

</h3>

{/\* Mensaje de estado \*/}

{submitMessage.text && (

<div className={`mb-6 p-4 rounded-lg ${

submitMessage.type === 'success'

? 'bg-green-100 text-green-700'

: 'bg-red-100 text-red-700'

}`}>

{submitMessage.text}

</div>

)}

<form

ref={form}

onSubmit={handleSubmit}

className="space-y-4 md:space-y-6"

>

<div>

<label htmlFor="name" className="block text-sm md:text-base font-medium text-gray-700 mb-1 md:mb-2">

Nombre

</label>

<input

type="text"

name="user\_name" // IMPORTANTE: Este nombre debe coincidir con tu plantilla

id="name"

className="w-full px-4 py-2 md:px-4 md:py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-emerald-500 focus:border-emerald-500 transition-colors duration-300"

placeholder="Tu nombre"

required

/>

</div>

<div>

<label htmlFor="email" className="block text-sm md:text-base font-medium text-gray-700 mb-1 md:mb-2">

Email

</label>

<input

type="email"

name="user\_email" // IMPORTANTE: Este nombre debe coincidir con tu plantilla

id="email"

className="w-full px-4 py-2 md:px-4 md:py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-emerald-500 focus:border-emerald-500 transition-colors duration-300"

placeholder="tu@email.com"

required

/>

</div>

<div>

<label htmlFor="message" className="block text-sm md:text-base font-medium text-gray-700 mb-1 md:mb-2">

Mensaje

</label>

<textarea

id="message"

name="message" // IMPORTANTE: Este nombre debe coincidir con tu plantilla

rows="4"

className="w-full px-4 py-2 md:px-4 md:py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-emerald-500 focus:border-emerald-500 transition-colors duration-300"

placeholder="Cuéntanos cómo podemos ayudarte..."

required

></textarea>

</div>

<button

type="submit"

disabled={isSubmitting}

className={`w-full ${

isSubmitting ? 'bg-emerald-400' : 'bg-emerald-600 hover:bg-emerald-700'

} text-white px-6 py-3 rounded-lg font-semibold transition-colors duration-300 shadow-md`}

>

{isSubmitting ? 'Enviando...' : 'Enviar Mensaje'}

</button>

</form>

</div>

</div>

</div>

</div>

<div className="mt-16 md:mt-20 pt-6 md:pt-8 border-t border-gray-700">

<div className="text-center text-gray-400 text-sm md:text-base">

<p>&copy; 2024 La Familia del Bonsai. Todos los derechos reservados.</p>

</div>

</div>

</section>

);

};

export default Contact;

```

## Cómo funciona EmailJS paso a paso:

1. \*\*Crear una cuenta en EmailJS\*\*:

- Ve a [https://www.emailjs.com/](https://www.emailjs.com/) y regístrate

- El plan gratuito permite 200 correos/mes

2. \*\*Configurar un servicio de correo\*\*:

- En el dashboard, ve a "Email Services" y agrega un servicio (Gmail, Outlook, etc.)

- Sigue las instrucciones para conectar tu cuenta de correo

3. \*\*Crear una plantilla de correo\*\*:

- Ve a "Email Templates" y crea una nueva plantilla

- Usa variables como `{{user\_name}}`, `{{user\_email}}` y `{{message}}`

- Ejemplo de plantilla:

```

Asunto: Nuevo mensaje de contacto de {{user\_name}}

Nombre: {{user\_name}}

Email: {{user\_email}}

Mensaje:

{{message}}

```

4. \*\*Obtener tus credenciales\*\*:

- En "Integration" encontrarás:

- \*\*Service ID\*\*: ID del servicio de correo que creaste

- \*\*Template ID\*\*: ID de la plantilla que creaste

- \*\*Public Key\*\*: Tu clave pública de EmailJS

5. \*\*Instalar la librería en tu proyecto\*\*:

```bash

npm install @emailjs/browser

# o

yarn add @emailjs/browser

```

6. \*\*Configurar el formulario en React\*\*:

- Usa `useRef` para acceder al formulario

- Asegúrate que los `name` de los inputs coincidan con los de tu plantilla

- Implementa el manejo de estado para mostrar mensajes de éxito/error

## Ventajas de usar EmailJS:

1. \*\*No necesitas backend\*\*: Todo se maneja desde el frontend

2. \*\*Fácil integración\*\*: Solo necesitas unas pocas líneas de código

3. \*\*Seguro\*\*: No expones credenciales sensibles de tu servidor de correo

4. \*\*Gratis para empezar\*\*: 200 correos/mes en el plan gratuito

## Consejos adicionales:

1. \*\*Protege tu cuenta\*\*:

- Usa una cuenta específica para los correos del formulario

- No uses tu correo personal principal

2. \*\*Personaliza la experiencia\*\*:

- Añade un spinner durante el envío

- Implementa validación adicional de campos

- Puedes añadir más campos si los necesitas

3. \*\*Pruebas\*\*:

- Prueba el formulario con diferentes escenarios

- Verifica que los correos lleguen a tu bandeja de entrada

¿Necesitas ayuda para configurar tu cuenta de EmailJS o crear las plantillas?