

# Clase 16: Organización del proyecto y envío de mail

Sitio: [Centro de E-Learning - UTN.BA](#)  
Curso: Curso de Backend Developer - Turno Noche  
Libro: Clase 16: Organización del proyecto y envío de mail

Imprimido por: Nelson Brian Avila Solano  
Día: Tuesday, 30 de December de 2025, 19:53

## Descripción

### Objetivos

- Aprender a enviar correos electrónicos utilizando Nodemailer y Ethereal.
- Comprender el funcionamiento de Nodemailer y cómo configurar el transporte.
- Utilizar Handlebars como motor de plantillas para generar correos electrónicos personalizados.
- Separar la lógica del formulario del archivo principal app.js para mantener un código limpio y organizado.

# Tabla de contenidos

**1. Introducción**

**2. Envío de mails**

**3. Práctica**

# 1. Introducción

El envío de correos electrónicos desde una aplicación web es una funcionalidad esencial para muchas plataformas, ya sea para confirmaciones de registro, notificaciones o recuperación de contraseñas. En Node.js, Nodemailer es una de las bibliotecas más utilizadas para esta tarea, ya que permite configurar un servidor SMTP de manera sencilla y enviar correos electrónicos de forma eficiente. Para el desarrollo y prueba de estos envíos sin afectar a usuarios reales, se puede utilizar Ethereal, un servicio que proporciona cuentas temporales para simular el envío de correos sin que realmente lleguen a los destinatarios.

Además, para mejorar la organización del código, es recomendable separar la lógica del formulario en un archivo específico en lugar de manejarla directamente en app.js. También, se pueden generar correos electrónicos personalizados mediante motores de plantillas como Handlebars, permitiendo una mejor presentación del contenido. Para entornos de producción, es posible configurar Nodemailer con servidores de correo reales, como Gmail, asegurando una integración funcional y segura dentro de la aplicación.

## 2. Envío de mails

### Envío de mails

#### Nodemailer

Nodemailer es una biblioteca de Node.js que permite enviar correos electrónicos fácilmente.

##### ▪ instalar nodemailer:

<https://www.npmjs.com/package/nodemailer>

```
const nodemailer = require('nodemailer');
```

#### Ethereal

<https://ethereal.email/>

Ethereal es un servicio SMTP falso, principalmente dirigido a usuarios de Nodemailer (pero no limitado a). Es un servicio de correo electrónico anti-transaccional completamente gratuito donde los mensajes nunca se entregan.

En su lugar, puede generar una cuenta de correo electrónico personalizada directamente desde Nodemailer, enviar un correo electrónico utilizando esa cuenta como lo haría con cualquier otro proveedor de SMTP y, finalmente, obtener una vista previa del mensaje enviado aquí, ya que en realidad no se entregan correos electrónicos.



#### POST

```
app.post("/send-email", (req, res) =>{  const
transporter = nodemailer.createTransport({
host:"smtp...",    port:"",    secure:false,
auth: {    user: 'xxxxxxxx@ethereal.email',
pass: 'xxxxxxx'
}
```

## Opciones

```
const mailOptions={  from:
"Remitente",
to:"xxxxxxxx@hotmail.com",    subject:
"Enviado desde nodemailer",
text:"¡Hola mundo!"
};    transporter.sendMail(mailOptions, (error,
info)=>{    if(error){
res.status(500).send(error.message);
    }else{
console.log("Email enviado")
res.status(200).jsonp(reqbody);
    }
  });});
```

## Separación de la lógica del formulario

Separar la lógica del formulario del archivo principal app.js nos permitirá mantener un código limpio y organizado. Para ello, deberás crear un archivo formulario.js en la carpeta de rutas, y mover la lógica del formulario allí.

## Pasar a un correo tipo gmail

Para pasar a un correo de tipo Gmail en lugar de utilizar Ethereal, los siguientes pasos pueden ser útiles:

1. Acceder a la cuenta de Gmail que se desea utilizar para enviar los correos.
2. Habilitar el acceso de aplicaciones menos seguras en la cuenta de Gmail siguiendo estos pasos:
  - Ir a "Mi Cuenta" en la página de Google.
  - Seleccionar "Seguridad".
  - En la sección "Acceso de aplicaciones menos seguras", habilitar la opción para permitir el acceso.

3. Modificar la configuración del transporte en el archivo formulario.js para utilizar el servidor SMTP de Gmail y las credenciales correspondientes, por ejemplo:

```
// configurar el transportador SMTP const
transporter = nodemailer.createTransport({
  service: 'gmail',  auth: {    user:
'tu_correo@gmail.com',    pass:
'tu_contraseña'    }
});
```

4. En el formulario formulario.hbs, cambiar la dirección de correo electrónico del remitente a la dirección de correo de Gmail:

```
<input type="email" id="from" name="from" placeholder="Correo electrónico"
required>
```

5. Actualizar cualquier referencia a la dirección de correo electrónico de Ethereum en el archivo formulario.js con la dirección de correo electrónico de Gmail correspondiente.

## 3. Práctica

### ¡A practicar!

Desarrollar una aplicación web utilizando Node.js y Express que permita a los usuarios enviar correos electrónicos utilizando Nodemailer y Ethereal.

#### Requerimientos:

- utilizar Handlebars o EJS como motor de plantillas para generar correos electrónicos personalizados.
- Permitir a los usuarios llenar un formulario con la información necesaria para enviar un correo electrónico, incluyendo campos para el remitente, destinatario, asunto y mensaje.
- Enviar el correo electrónico utilizando Nodemailer y Ethereal.
- Mostrar una página de confirmación después de que se haya enviado el correo electrónico.
- Separar la lógica del formulario del archivo principal app.js para mantener un código limpio y organizado.
- Tener un estilo coherente y profesional utilizando CSS.