

**PROFESSIONAL WEBMASTER**

Módulo: 2

Node JS



Introducción (2).png

|  |  |
| --- | --- |
|  |  |

[**Envio de datos 4**](#_Toc34766564)

[**Envio email desde Node con Nodemailer 5**](#_Toc34766565)

# 

# Envio de datos

Existen dos formas de recuperar datos enviados desde un formulario http, **GET** y **POST.**

* **GET:** Cuando se envían datos al servidor mediante GET estos serán **visibles** en la url.
* **POST:** Cuando se envían datos al servidor mediante POST estos estarán **ocultos.** *(login – Funcionamiento de un formulario, etc)*

**En el código hay elementos claves a completar:**

* El atributo "action", que indica "a qué página" le está pasando las variables, siendo la página que nos va a mostrar cuando pulsemos en el botón "Enviar", tal como si fuese un link hacia esa página.
* El atributo **"method"**, que especifica uno de los dos posibles "métodos" o formas de pasar las variables: **a la vista** de todos en la URL del navegador (method="**GET**) o si las vamos a pasar ocultas (method="**POST**").
* Algún **campo** que permita al usuario el ingreso de datos (**input**). Lo fundamental de cada campo será su **nombre** (atributo "name"), ya que ése será el nombre de la variable que enviará a la página de destino.
* Un **botón** para enviar los datos.

# Envio email desde Node con Nodemailer

**Nodemailer** es un **módulo** que nos permitirá añadir estas funcionalidades de envio de email desde **Node.js**.

**Dotenv** es un módulo de dependencia que carga variables de entorno de un archivo .env.

Para enviar un email, primero debemos definir un **transporter**.  
  
SMTP es el transporte principal en Nodemailer para entregar mensajes.   
En el caso que solo necesita cambiar las opciones de configuración para reemplazar un proveedor por otro y ya está listo.

Lo haremos de la siguiente manera, sustituyendo los datos por los de nuestra cuenta:

var transporter = nodemailer.createTransport({  
 host: "smtp.example.com",

port: 587,

auth: {

user: "username",

pass: "password"

}  
 }  
});

Ahora que ya tenemos el transporter, definiremos el propio email:

var obj = {  
 to: 'destinatario@gmail.com',  
 subject: 'Asunto',  
 text: 'Contenido del email'  
};

y ya sólo nos queda enviar el email!

transporter.sendMail(obj)

**