

**PROFESSIONAL WEBMASTER**

Módulo: 2

Node JS



Introducción (2).png

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

[¿Qué son Express y Node? 4](#_Toc34725751)

[Node 4](#_Toc34725752)

[Express 4](#_Toc34725753)

[¿Cómo es el código para Express? 5](#_Toc34725754)

[Creando manejadores de rutas 6](#_Toc34725755)

[Renderización de data (vistas) 6](#_Toc34725756)

[Estructura de Archivos 7](#_Toc34725757)

# 

# ¿Qué son Express y Node?

[Node](https://nodejs.org/) (o más correctamente: Node.js) es un entorno que trabaja en tiempo de ejecución, de código abierto, multi-plataforma, que permite a los desarrolladores crear toda clase de herramientas de lado servidor y aplicaciones en **JavaScript.** La ejecución en tiempo real está pensada para usarse fuera del contexto de un explorador web (es decir, ejecutarse directamente en una computadora o sistema operativo de servidor).

Desde una perspectiva de desarrollo de servidor web, Node tiene un gran número de ventajas:

* Ha sido diseñado para optimizar el rendimiento y la escalabilidad en aplicaciones web.
* El código está escrito en "simple JavaScript".
* JavaScript es un lenguaje de programación relativamente nuevo y se beneficia de los avances en diseño de lenguajes cuando se compara con otros lenguajes de servidor web tradicionales (ej, Python, PHP, etc.)
* El gestor de paquetes de Node (NPM del inglés: Node Packet Manager) proporciona acceso a cientos o miles de paquetes reutilizables. Tiene además la mejor en su clase resolución de dependencias y puede usarse para automatizar la mayor parte de la cadena de herramientas de compilación.
* Es portable, con versiones que funcionan en Microsoft Windows, OS X, Linux, Solaris, FreeBSD, OpenBSD, WebOS, y NonStop OS. Además, está bien soportado por muchos de los proveedores de hospedaje web, que proporcionan infraestructura específica y documentación para hospedaje de sitios Node.
* Tiene un ecosistema y comunidad de desarrolladores de terceros muy activa, con cantidad de gente deseosa de ayudar.

# Express

Es el framework web más popular de Node, y es la librería subyacente para un gran número de otros [frameworks web de Node](https://expressjs.com/en/resources/frameworks.html) populares.   
  
Proporciona mecanismos para:

* Escritura de manejadores de peticiones con **diferentes verbos HTTP** en diferentes caminos URL (rutas).
* Integración con motores de **renderización de "vistas"** para generar respuestas mediante la introducción de **datos en plantillas**.
* Añadir procesamiento de peticiones **"middleware"** adicional en cualquier punto dentro de la tubería de manejo de la petición.

A pesar de que **Express** es en sí mismo bastante minimalista, los desarrolladores han creado **librerías** para trabajar con:

Reiniciar automáticamente el servidor: **nodemon**  
Envio de mail: **nodemailer**

Sesiones: **express-session**

Base de datos: **mysql**

ETC…

## ¿Cómo es el código para Express?

En sitios web o aplicaciones web dinámicas, que accedan a bases de datos, el servidor espera a recibir peticiones HTTP del navegador (o cliente).

Cuando se recibe una petición, la aplicación determina cuál es la acción adecuada correspondiente, de acuerdo a la estructura de la URL y a la información (opcional) indicada en la petición con los métodos **POST o GET.**

Dependiendo de la acción a realizar, puede que se necesite leer o escribir en la base de datos, o realizar otras acciones necesarias para atender la petición correctamente.

Express posee métodos para especificar que función ha de ser llamada dependiendo del verbo HTTP usado en la petición (**GET, POST, SET, etc.)** y la estructura de la URL **("ruta").** También tiene los métodos para especificar que plantilla **("view")** o gestor de visualización utilizar, donde están guardadas las plantillas de HTML que han de usarse y como  generar la visualización adecuada para cada caso.

**EJEMPLO**

var express = require('express');

var app = express();

**app.get('/', function(req, res) {**

**res.send('Hola Mundo!');**

**});**

Las primeras dos líneas incluyen (mediante la orden **require()**) el **módulo** **de Express** y crean una [aplicación de Express](https://expressjs.com/en/4x/api.html#app). Este elemento se denomina comúnmente **app**, y posee métodos para el enrutamiento de las peticiones HTTP, configuración del 'middleware', y visualización de las vistas de HTML, uso del motores de 'templates', y gestión de las [configuraciones de las aplicaciones](https://expressjs.com/en/4x/api.html#app.settings.table)que controlan la aplicación.

Las líneas que siguen en el código (las tres líneas que comienzan con app.get) muestran **una definición de ruta** que se llamará cuando se reciba una petición  HTTP GET con una dirección ('/') relativa al directorio raíz.

La función 'callback' coge una petición y una respuesta como argumentos, y ejecuta un [send()](https://expressjs.com/en/4x/api.html" \l "res.send) en la respuesta, para **enviar** la cadena  de caracteres: "Hola Mundo!".

## Creando manejadores de rutas

Las rutas le permiten igualar patrones particulares de caracteres en la URL, y extraer algunos valores de ella y pasarlos como parámetros al manejador de rutas.

Usualmente es útil agrupar **manejadores de rutas** para una parte del sitio juntos y accederlos usando un prefijo de ruta en común.   
  
En Express esto se logra usando el objeto **[express.Router](http://expressjs.com/en/guide/routing.html" \l "express-router).**   
  
**EJEMPLO: nosotros.js**

var express = require('express');

var router = express.Router();

// Home page route

router.get('/', function(req, res) {

res.send('index');

});

module.exports = router;

## Renderización de data (vistas)

El Motor de plantilla (referido como "motor de vistas" por Express) le permite definir la estructura de documento de salida en una plantilla, usando marcadores de posición para datos que serán llenados cuando una página es generada.   
Las plantillas son utilizadas generalmente para crear HTML, pero también pueden crear otros tipos de documentos.

Express tiene soporte para  numerosos motores de plantillas.

Nosotros vamos a utilizar: **Handlebars**,   
  
Handlebars sirve para generar HTML a partir de objetos con datos en formato JSON.

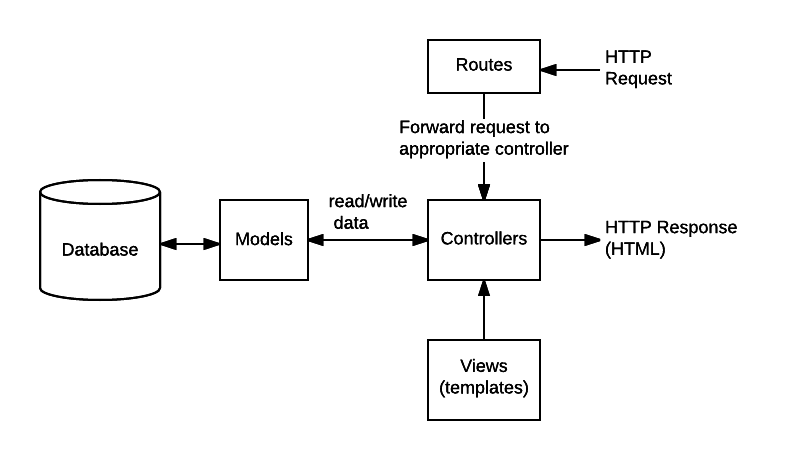
Handlebars es un popular sistema de plantillas en Javascript que te permite crear y formatear código HTML de una manera muy sencilla.

Los sistemas de plantillas son muy importantes, porque nos permiten separar el código por responsabilidades y crear la salida de las aplicaciones de una manera sencilla, manteniendo el código HTML separado del código Javascript.

.

## Estructura de Archivos

**MVC de express**



**