

INTRODUCCIÓN

Tiempo estimado: 10m

Ésta es la primera lección del curso de desarrollo de aplicaciones webs con **Node** mediante **Express**. No es un curso de **Node**, se asume que el estudiante ya tiene esos conocimientos; tampoco se introduce **HTML5** ni **CSS**. No se presenta ningún *framework* cliente como **Angular**, **Backbone.js**, **Ember.js** o **React**, ni sus conocimientos son necesarios para el curso, aunque sí se recomienda que al menos uno de ellos se encuentre en el currículo del estudiante.

Para comenzar, se presenta el *framework* **Express**, la columna vertebral del servidor web. También se lista algunas organizaciones que están usando **Node** y **Express**, mostrando así la aceptación de la plataforma **Node** para el desarrollo de aplicaciones. A continuación, se muestra el concepto de pila web donde se enumera los componentes de software que participan en una aplicación web escrita en **Node**, tanto en el lado servidor como en el cliente. Finalmente, se resume el plan de estudio del curso.

Al finalizar la lección, el estudiante sabrá:

- Qué es **Express**.
- Qué es una pila web.

express

Express es el *framework* de facto para el desarrollo de aplicaciones webs y servicios **REST** mediante **Node** con sus más de seis millones de descargas mensuales. Se utiliza para la implementación del lado *back-end* de la aplicación web. Inicialmente, lo diseñó y desarrolló **TJ Holowaychuk**. Actualmente, su desarrollo está en manos de la **Node.js Foundation**, entre cuyos miembros encontramos compañías como **Google**, **Groupon**, **IBM**, **Intel**, **Joyent**, **Microsoft**, **PayPal**, **Red Hat**, **SAP** y **Yahoo!**

Es de código abierto y se puede utilizar gratuitamente, tanto en aplicaciones personales como comerciales.

Entre las organizaciones que lo usan encontramos **Adobe**, **Amazon**, **CBS**, **Dow Jones**, **eBay**, **Financial Times**, **Flickr**, **Fox**, **Google**, **Groupon**, **HP**, **IBM**, **LinkedIn**, **Microsoft**, **Mozilla**, **MySpace**, **Netflix**, **Samsung**, **Telefónica**, **The New York Times**, **PayPal**, **Pearson**, **Pinterest**, **Rdio**, **Skype**, **Uber**, **Walmart** y **Yahoo!**.

Recordemos que un *framework* es una pieza de software que implementa o proporciona una determinada funcionalidad o infraestructura. Siendo así más fácil el desarrollo y la reutilización de software. Huelga decir que debido a esta reutilización de componentes, el *framework* debe tenerse muy en cuenta durante la fase de diseño del software a desarrollar. Cuando se selecciona un *framework* en el desarrollo de un componente o aplicación, el *framework* es pieza angular en el diseño. Atendiendo al uso de uno u otro, el diseño puede ser distinto.

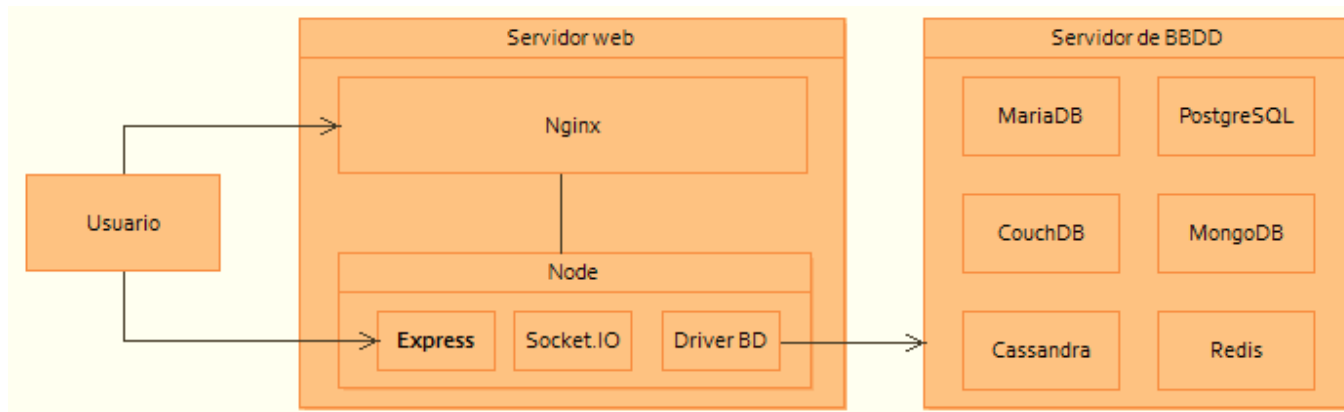
pila web

El objeto principal de este curso es mostrar una pila para el desarrollo de aplicaciones webs en la plataforma **Node**. Una *pila web* (*web stack*) es la combinación de componentes de software usados para el desarrollo y la producción de aplicaciones webs. Una *pila web de JavaScript* (*full-JavaScript web stack*) es aquella que utiliza como único lenguaje de programación **JavaScript**.

A la hora de atender una pila web, se suele distinguir entre la parte servidora y la cliente.

parte servidora de la pila web

Por un lado, hay que distinguir el lado servidor. En **Node**, se puede distinguir básicamente lo siguiente:



En empresas pequeñas y medianas, donde los recursos financieros son escasos en muchas ocasiones, e incluso en algunas empresas grandes, se suele utilizar software de código abierto y gratuito como, por ejemplo, [Linux](#), [Node](#), [CouchDB](#), [Cassandra](#), [MariaDB](#), [PostgreSQL](#) y [Redis](#). Algunas empresas prefieren no arriesgar y usar software de fabricantes reconocidos mundialmente como, por ejemplo, [Microsoft](#) u [Oracle](#), lo que aumenta los costes, pero se sienten más seguros. Es importante tener en cuenta que hay mucho software de código abierto y/o gratuito que suele implementarlo un grupo de programadores profesionales de la comunidad. Y en muchos de los desarrollos suele haber personal de organizaciones privadas que participan y ayudan en su desarrollo y después proporcionan soporte. Por ejemplo, detrás de [Apache Cassandra](#) se encuentra [DataStax](#), una empresa con capital de millones de dólares y oficinas en EE.UU., Europa y Asia; detrás de [PostgreSQL](#) se encuentran [EnterpriseDB](#), [VMware](#) y [Red Hat](#), también con millones de dólares de capital y oficinas en EE.UU., Europa y Asia. Tanto [DataStax](#) como [EnterpriseDB](#), [VMware](#) y [Red Hat](#) dan soporte 24x7 a aquellas empresas que lo necesitan; al igual que [Microsoft](#) y [Oracle](#) lo hacen con sus productos.

Por lo general, en entornos productivos se utiliza [Linux](#) como sistema operativo como, por ejemplo, [Red Hat Enterprise](#), [Ubuntu](#), [CentOS](#) o [Debian](#). Estas máquinas pueden ser máquinas físicas o bien virtuales bajo [VMware](#) o [KVM](#). Actualmente, algunas empresas han decidido virtualizar todo o la mayor parte de su entorno productivo, siendo frecuente que las máquinas servidoras no se encuentren en máquinas físicas dedicadas sino virtuales. En entornos de desarrollo, también es muy común el uso de virtualización porque reduce drásticamente los costes y su administración.

Cuando se usa [Node](#), se suele utilizar como servidor web el *framework* [Express](#). En muchas ocasiones, se combina con un servidor web como [Apache](#) o [Nginx](#). Además, se suele utilizar [Socket.IO](#) para proporcionar comunicación bidireccional entre el cliente web y la aplicación servidora en tiempo real.

Cuando se usa una pila web [JavaScript](#) completa con [Node](#), a nivel de sistema de gestión de bases de datos se suele utilizar bases de datos [SQL](#), principalmente, [MariaDB](#), [MySQL](#), [PostgreSQL](#), [SQL Server](#) y [SQLite](#) o bien [NoSQL](#) como [Cassandra](#), [CouchDB](#), [MongoDB](#) o [Redis](#). Para el acceso a la base de datos desde la aplicación web se utiliza *drivers* de [Node](#); existe *drivers* para la mayoría del top 10 de bases de datos.

parte cliente de la pila web

En el lado cliente, también hay pila como, por ejemplo, la que se ilustra a continuación, que no es completa ni mucho menos:



Se puede utilizar *frameworks* como [Angular](#) de [Google](#) o [React](#) de [Facebook](#). Además, se puede utilizar el

protocolo **WebSocket**, mediante **Socket.IO**, para la comunicación entre el cliente y el servidor, tal como vimos en el lado servidor. Además no puede faltar ni **HTML5**, ni **CSS**, ni como no **JavaScript**.

información del curso

Este curso es la primera parte de dos dedicados exclusivamente al *framework* **Express**.

Tiene como objetivo presentar los fundamentos del desarrollo de aplicaciones webs servidoras con **Express**. Dejándose para el siguiente, los aspectos avanzados como, por ejemplo, el desarrollo de APIs **REST**, la negociación de contenido, la compresión de contenido, el desarrollo de aplicaciones compuestas, la autenticación, etc.

Al finalizarlo, el estudiante sabrá:

- Qué es **Express**.
- Cómo desarrollar aplicaciones **Express**.
- Cómo usar **nodemon** para monitorizar los cambios en los archivos y reiniciar automáticamente la aplicación web.
- Cómo usar **Handlebars** como motor de plantillas.
- Cómo procesar los datos remitidos por los usuarios mediante formularios en el lado servidor.
- Cómo usar generadores para facilitar y mejorar el desarrollo.

Este curso se recomienda a aquellas personas que deseen mejorar su conocimiento de **Node** para extraer todo su jugo mediante el desarrollo de aplicaciones webs en el lado servidor.

No es un curso de **Node** ni **HTML**. Asume que el estudiante ya sabe programar en ambos lenguajes.

PLAN DE ESTUDIO

El curso tiene una duración aproximada de 8 horas. Se divide en 16 lecciones, cada una de ellas con una parte de teoría y generalmente una de práctica. El enfoque a seguir es muy sencillo: ir lección a lección; primero hay que leer la teoría y, después, realizar la práctica. Se recomienda encarecidamente que el estudiante realice cada lección, tanto teoría como práctica, en el mismo día, con el menor número de interrupciones a lo largo de su estudio.

A continuación, se enumera las distintas lecciones y el tiempo estimado para su estudio:

Lección	Teoría	Práctica	Descripción
1 Introducción	10min	-	Esta lección.
2 Aplicaciones Express	10min	10min	Descripción de una aplicación de Express .
3 Pila de middleware	10min	25min	Descripción de la pila de <i>middleware</i> de las aplicaciones Express .
4 Contenido estático	15min	15min	Cómo servir contenido estático en una aplicación Express .
5 Objeto petición	15min	10min	Descripción del objeto request de las aplicaciones Express .
6 Registro de eventos	15min	15min	Cómo llevar a cabo el registro de mensajes de una aplicación Express .
7 Objeto respuesta	25min	15min	Descripción del objeto response de las aplicaciones Express .
8 nodemon	5min	15min	Uso de nodemon para monitorizar cambios del proyecto y reiniciar la aplicación Express automáticamente.
9 Rutas	15min	15min	Cómo definir y procesar las URLs específicas de la aplicación Express .
10 Introducción a las plantillas	15min	-	Introducción al concepto de motor de plantillas.
11 Handlebars	25min	25min	Descripción detallada del motor de plantillas Handlebars .

12	Redireccionamiento HTTP	5min	5min	Cómo redireccionar recursos a otras URLs.
13	Cookies	20min	15min	Cómo usar <i>cookies</i> para almacenar datos.
14	Estados de sesión	10min	15min	Cómo usar el estado de sesión para almacenar datos particulares de las sesiones.
15	Formularios HTML	10min	10min	Cómo procesar los datos remitidos por los usuarios mediante formularios HTML.
16	Generadores	10min	25min	Cómo usar el generador oficial de Express y el de Justo para facilitar la creación de aplicaciones Express.

INFORMACIÓN DE PUBLICACIÓN

Título Fundamentos de Express (Volumen 1)

Autor Raúl G. González - raulgonzalez@nodemy.com

Primera edición Agosto de 2016

Versión actual 1.0.0

Versión de Express 4.14

Contacto hola@nodemy.com