

# Introducción a Express

**DigitalHouse** >  
Coding School



**Express** es un **framework** que facilita y **agiliza** el desarrollo de aplicaciones web en **Node.js**.



# ¿Qué es un **framework**?

Un framework es un **entorno** de **trabajo** que trae resueltas una serie de tareas, automatizando así el desarrollo de cualquier aplicación.

El **framework** de **Node.js** se utiliza principalmente por la productividad que brinda, así como por su escalabilidad y velocidad.

**Express** es uno de los más populares y estables del entorno de Node.js y es muy utilizado tanto para aplicaciones web como móviles.

The logo for Express.js, featuring the word "Express" in a thin, outlined, sans-serif font.

# ¿Cómo usar **Express**?

Lo primero que hay que hacer es instalar la librería en un proyecto de Node.js ya inicializado. Es decir, haber ejecutado previamente `npm init` y tener creado el archivo `package.json`.

Con el siguiente comando estaremos **guardando**, en la propiedad `dependencies` del archivo `package.json`, una **referencia** a la **librería** que estamos instalando.

```
_> npm install express
```

Luego, quien quiera clonar el proyecto, podrá instalar todas las dependencias necesarias para que funcione con el comando `npm install`.

# ¿Cómo usar Express?

Una vez instalado Express, tendremos que requerir el módulo en nuestro *entry-point*, que normalmente es un archivo llamado: `app.js`.

```
{ }  const express = require('express');
```

Lo que devuelve la librería es una función que encapsula toda la funcionalidad de Express. Para poder empezar a usarla, hace falta ejecutar esa función.

Lo próximo, entonces, será crear una variable nueva y almacenar en ella la ejecución de `express` para tener todos los métodos de la librería disponibles.

```
{ }  const app = express();
```



Cada vez que necesitemos trabajar con Express, hace falta **instalarlo** dentro del **proyecto** que estemos desarrollando.



DigitalHouse>  
Coding School