**Recomendaciones finales**

A lo largo de esta sección de Node.js, vimos cómo funciona éste internamente, como implementar un servidor con *Express*, etcétera. Aun así, lo que vimos en esta guía es sólo la implementación básica de un servidor con Node.js. Así que recomendamos que se profundice en lo siguiente:

Documentación de **Node.js**:

<https://nodejs.org/dist/latest-v9.x/docs/api/>

Documentación de **Express**:

<https://expressjs.com/>

***Async***, una librería que nos permite ejecutar código asíncrono con si fuera código síncrono. Realmente útil para escenarios más complejos que no tratamos en la guía, pero que comúnmente se presentan en el desarrollo de aplicaciones en Node.js:

<https://caolan.github.io/async/>

***Promises***, una forma de manejar código asíncrono que viene integrada en JavaScript desde la especificación ES2015 (también llamada *ES6*).

<https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Promise>

***Loopback***, un framework de Node.js, basado en *Express* que facilita muchísimo la contrucción de *APIs* en Node.js. Se recomienda tener un conocimiento medio de cómo funciona *Node.js* y *Express* ya que la curva de aprendizaje de este *framework* es muy alta. Sin embargo, con un buen conocimiento de *Node.js, Express* y *Loobpack*, podemos ahorrar tiempo drásticamente al implementar *Loopback*. Así que recomendamos que se profundice en él.

<https://loopback.io/>

Por último, recomendamos encarecidamente que aprendas a utilizar ***Git****,* uno de los controladores de versiones más famoso de los últimos tiempos. *Git* facilita enormemente el desarrollo de aplicaciones de cualquier lenguaje de programación y también muchísimo cuando se trabaja en equipo a la hora de compartir el código de un determinado proyecto.

<https://git-scm.com/>

Hay distintas plataformas que soportan *Git*, como *GitHub, BitBucket, GitLab,* así que recomendamos que les eches un vistazo.

<https://github.com/>

<https://bitbucket.org/product>

<https://about.gitlab.com/>