# 3. Fuentes de información

En los sub-apartados de este apartado se deben indicar documentos de interés para aprender sobre el tipo de tecnología en general, y sobre cada una de las tecnologías elegidas.

## 3.1 Fuentes sobre Web Server Framework

### **3.1.1 Wikipedia**

Un framework para aplicaciones web es un framework diseñado para apoyar el desarrollo de [sitios web](https://es.wikipedia.org/wiki/Sitio_web) dinámicos, [aplicaciones web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web) y [servicios web](https://es.wikipedia.org/wiki/Servicio_web). Este tipo de frameworks intenta aliviar el exceso de carga asociado con actividades comunes usadas en desarrollos web. Muchos framework proporcionan [bibliotecas](https://es.wikipedia.org/wiki/Biblioteca_%28inform%C3%A1tica%29) para acceder a bases de datos, estructuras para plantillas y gestión de sesiones, y con frecuencia facilitan la [reutilización de código](https://es.wikipedia.org/wiki/Reutilizaci%C3%B3n_de_c%C3%B3digo).

Para proporcionar una página web dinámica que refleja las entradas del usuario , se introdujo la norma ( CGI ) Common Gateway Interface para conectar aplicaciones externas con los servidores web. CGI podría afectar negativamente a la carga del servidor , sin embargo ,cada solicitud inicia un proceso separado .

Muchos marcos siguen el patrón de arquitectura MVC para separar el modelo de datos con reglas de negocio de la interfaz de usuario

### **3.1.2 Universidad de Sevilla /Javier J. Gutiérrez.**

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. Para comprender como trabajan los frameworks Web existentes es imprescindible conocer el patrón MVC.

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.

### **3.1.3** [**http://kokiing.blogspot.com.es**](http://kokiing.blogspot.com.es)

El propósito de un Framework es ayudar y facilitar el proceso de desarrollo de aplicaciones. Debe permitir desarrollar la aplicación rápido y fácilmente y debe resultar en una aplicación superior finalizada.

Frameworks, en resumen, proveen una poderosa caja de herramientas. Las herramientas en esta caja ayudan en muchas áreas diferentes del desarrollo de aplicaciones. Proveen patrones de diseño esenciales y estructura al proyecto de desarrollo de la aplicación, y también provee el esqueleto y contenedor para los componentes que se crean para operar la aplicación.

Los Frameworks son importantes en todas las fases de desarrollo, desde el diseño hasta el desarrollo y quizá más en el mantenimiento continúo.

## 3.2 Fuentes sobre Django

### **3.2.1 Wikipedia**

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–controlador. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt.

En junio de 2008 fue anunciado que la recién formada Django Software Foundation se haría cargo de Django en el futuro.

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés *Don't Repeat Yourself*). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

Los orígenes de Django en la administración de páginas de noticias son evidentes en su diseño, ya que proporciona una serie de características que facilitan el desarrollo rápido de páginas orientadas a contenidos. Por ejemplo, en lugar de requerir que los desarrolladores escriban controladores y vistas para las áreas de administración de la página, Django proporciona una aplicación incorporada para administrar los contenidos, que puede incluirse como parte de cualquier página hecha con Django y que puede administrar varias páginas hechas con Django a partir de una misma instalación; la aplicación administrativa permite la creación, actualización y eliminación de objetos de contenido, llevando un registro de todas las acciones realizadas sobre cada uno, y proporciona una interfaz para administrar los usuarios y los grupos de usuarios (incluyendo una asignación detallada de permisos).

### **3.2.2** [**http://tutorial.djangogirls.org/es/django/**](http://tutorial.djangogirls.org/es/django/)

Django es un framework para aplicaciones web gratuito y open source, escrito en Python. Es un WEB framework - un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente.

Verás, cuando estás construyendo un sitio web, frecuentemente necesitas un conjunto de componentes similares: una manera de manejar la autenticación de usuarios (registrarse, iniciar sesión, cerrar sesión), un panel de administración para tu sitio web, formularios, una forma de subir archivos, etc.

Por suerte para ti, hace tiempo varias personas notaron que los desarrolladores web enfrentan problemas similares cuando construyen un sitio nuevo, por eso juntaron cabezas y crearon frameworks (Django es uno de ellos) que te ofrecen componentes listos para usarse.

Los frameworks existen para ahorrarte tener que reinventar la rueda y ayudarte a aliviar la carga cuando construyes un sitio.

Cuando llega una petición a un servidor web es pasado a Django que intenta averiguar lo que realmente es solicitado. Toma primero una dirección de página web y trata de averiguar qué hacer. Esta parte es realizada por urlresolver de Django (tenga en cuenta que la dirección de un sitio web es llamada URL - Uniform Resource Locator - así que el nombre *urlresolver* tiene sentido). Este no es muy inteligente - toma una lista de patrones y trata de igualar la URL. Django comprueba los patrones de arriba hacia abajo y si algo se coincide entonces Django le pasa la solicitud a la función asociada (que se llama *vista*).

### **3.2.3** [**http://computerhoy.com**](http://computerhoy.com)

### **Django está de moda. Aunque sus orígenes se remontan a un diario online en 2003, por alguna razón ha sido en los últimos años cuando ha adquirido relevancia.**

Si has intentando conocer un poco más acerca de Django, te habrás topado con la densa definición de la entradilla. En pocas palabras, resume muchas de las propiedades de este proyecto, así que veamos de qué se trata.

Proporciona una serie de herramientas para facilitar la creación de páginas, siguiendo los principios DRY (Don’t Repeat Yourself; No Te Repitas) para evitar duplicidad en las líneas de código e invertir el menor esfuerzo posible. Por ejemplo, levantar un panel de administración básico sólo requiere un par de líneas de Python.

También se adscribe al diseño MVC (Modelo-Vista-Controlador), por lo que las diferentes partes del sitio están claramente separadas. Por ejemplo, el código de acceso a los datos es completamente independiente al que gobierna el aspecto externo de la página.

Al tratarse de Python, Django permite que el desarrollador escriba código ágilmente. El resultado son menos líneas de código y, en consecuencia, menos probabilidades de que haya bugs. De ahí que digan que “fomenta el desarrollo rápido y el diseño limpio y pragmático”.

Django no pretende sustituir a WordPress como CMS. Sólo está ahí para ayudarte a levantar sitios web desde cero, sea del tipo que sea. En vista de la versatilidad forzosa a la que está siendo sometido WordPress últimamente, quizá quieras optar por Django en busca de la flexibilidad necesaria para cualquier tipo de proyecto.

Aprenderás Python de una forma práctica y limpia, pues la implementación de Django obliga a respetar muchos principios de la buena programación.

## 3.3 Fuentes sobre Node.js

### **3.3.1 Wikipedia**

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web.[2](https://es.wikipedia.org/wiki/Node.js#cite_note-2) Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla.

Node.js es similar en su propósito a Twisted o Tornado de Python, Perl Object Environment de Perl, libevent o libev de [C](https://es.wikipedia.org/wiki/C), EventMachine de Ruby, vibe.d de Dy JEE de Java existe Apache MINA, Netty, Akka, Vert.x, Grizzly o Xsocket. Al contrario que la mayoría del código JavaScript, no se ejecuta en un navegador, sino en el servidor. Node.js implementa algunas especificaciones de CommonJS.[5](https://es.wikipedia.org/wiki/Node.js#cite_note-5) Node.js incluye un entorno REPL para depuración interactiva.

Node.js funciona con un modelo de evaluación de un único hilo de ejecución, usando entradas y salidas asíncronas las cuales pueden ejecutarse concurrentemente en un número de hasta cientos de miles sin incurrir en costos asociados al cambio de contexto.[6](https://es.wikipedia.org/wiki/Node.js#cite_note-6) Este diseño de compartir un único hilo de ejecución entre todas las solicitudes atiende a necesidades de aplicaciones altamente concurrentes, en el que toda operación que realice entradas y salidas debe tener una función callback. Un inconveniente de este enfoque de único hilo de ejecución es que Node.js requiere de módulos adicionales como *cluster*para escalar la aplicación con el número de núcleos de procesamiento de la máquina en la que se ejecuta.

### **3.3.2** [**http://www.netconsulting.es/blog/nodejs/**](http://www.netconsulting.es/blog/nodejs/)

Node.js es una librería y entorno de ejecución de E/S dirigida por eventos y por lo tanto asíncrona que se ejecuta sobre el intérprete de JavaScript creado por Google V8. Lo cierto es que está muy de moda aunque no es algo nuevo puesto que existen librerías como Twisted que hacen exactamente lo mismo pero si es cierto que es la primera basada en JavaScript y que tiene un gran rendimiento.

Node.js es un entorno Javascript del lado del servidor, basado en eventos. Node ejecuta javascript utilizando el motor V8, desarrollado por Google para uso de su navegador Chrome. Aprovechando el motor V8 permite a Node proporciona un entorno de ejecución del lado del servidor que compila y ejecuta javascript a velocidades increíbles. El aumento de velocidad es importante debido a que V8 compila Javascript en código de máquina nativo, en lugar de interpretarlo o ejecutarlo como bytecode. Node es de código abierto, y se ejecuta en Mac OS X, Windows y Linux.

Aunque Javascript tradicionalmente ha sido relegado a realizar tareas menores en el navegador, es actualmente un lenguaje de programación totalmente, tan capaz como cualquier otro lenguaje tradicional como C++, Ruby o Java. Además Javascript tiene la ventaja de poseer un excelente modelo de eventos, ideal para la programación asíncrona. Javascript también es un lenguaje omnipresente, conocido por millones de desarrolladores. Esto reduce la curva de aprendizaje de Node,js, ya que la mayoría de los desarrolladores no tendrán que aprender un nuevo lenguaje para empezar a construir aplicaciones usando Node.js.

### **3.3.3** [**http://www.genbetadev.com/frameworks/como-funciona-node-js**](http://www.genbetadev.com/frameworks/como-funciona-node-js)

Todos hemos oído hablar de Node.js como tecnología que permite trabajar con Javascript del lado del servidor. Poco a poco Node se está ganando un hueco en la comunidad. Sin embargo hay muchas dudas sobre su funcionamiento. Vamos a intentar en este artículo acercarnos un poco más a esta tecnología tan interesante.

Node.js está basado en el motor V8 de Javascript de Google. Este motor está diseñado para correr en un navegador y ejecutar el código de Javascript de una forma extremadamente rápida.

La tecnología que está detrás de Node.js permite ejecutar este motor en el lado del servidor abriendo un nuevo abanico de posibilidades en cuanto al mundo de desarrollo se refiere.

Para ello el entorno de Node.js ha desarrollado un conjunto amplio de librerías que no tienen nada que envidiar a otras plataformas. Por otro lado se han eliminado algunas funcionalidades que en el entorno de servidor no tenían sentido como por ejemplo el uso de Document Object Model.

Para poder trabajar de una forma óptima Node.js delega todo el trabajo en un pool de threads. Este pool de threads está construido con la librería libuv. Esta librería dispone de su propio entorno multithread asíncrono. Node.js envía el trabajo que hay que realizar al pool.