

Aura: No pudiendo competir mano a mano con otros frameworks que veremos más adelante, la particularidad de Aura es que sus librerías se pueden usar como paquetes independientes en cualquier código base.

Muy usado por aquellos desarrolladores que prefieren ir haciéndose sus propias librerías, sin contaminar demasiado su proyecto con paquetes innecesarios que se han incluido en otros frameworks por su popularidad. Pero no quita que Aura sociabilice con otros frameworks más completos, ya que podremos exportar sus paquetes de librerías para importarlos a otro framework de nuestra elección.

FuelPHP: Este “novato” en el mundo de los frameworks, les está echando un pulso (más igualado de lo que quisieran) a otros gigantes de este campo que veremos a continuación. Es simple, flexible y full-stack (no podremos separar un conjunto de librerías para usar unos recursos y descartar otros).

Se llegó a pensar que este proyecto había fracasado casi antes de empezar, ya que no había mucha gente usando y colaborando en su desarrollo y mejora, pero el proyecto se reactivó hace un par de años con el equipo que lo desarrolla totalmente volcado, implementando novedades debido a la enorme cantidad de gente que se ha dado cuenta del potencial de esta herramienta, y han empezado a colaborar en sus fases de debugging y testeo de las versiones próximas a ver la luz.

El núcleo de FuelPHP tiene todo lo necesario para desarrollar una web de principio a fin, sea cual sea el tamaño y complejidad del proyecto.

Slim: Nos encontramos ante uno de los mejores micro-frameworks (de menor tamaño al contar con menos componentes pre-instalados, pero igual de eficientes en la asistencia al desarrollador). De código abierto, sofisticado e intuitivo, se ha desarrollado desde cero, sin reutilizar “partes” de otros frameworks de este listado.

Aun así, cuenta con URL routing, una de las mejores arquitecturas middleware (gestión de información dentro de un sistema heterogéneo donde convergen diferentes tecnologías, el middleware se encargará de que todo lleve su curso) que podemos encontrar y cuenta con soporte nativo para cookies encriptadas, mensajes flash, plantillas de página y caché además de otras tantas funcionalidades.

Es mayormente utilizado para el desarrollo de API's o aplicaciones simples, pero muy eficaces y flexibles en cuanto a reutilizaciones. Slim permanecerá en el núcleo de nuestro proyecto,

escribiremos el código en torno a él e iremos activando los componentes que lo componen según los necesitemos, así como le podremos añadir aquellos otros que no contenga.

Phalcon: Un framework diseñado para la velocidad. Está escrito en C desde sus cimientos y es con seguridad el framework más rápido existente hoy día. Ofrece la mayoría de los recursos actuales necesarios para desarrollar un proyecto ágil y fresco (routing, controladores, vista de plantillas, ORM, caché, etc...).

No es la primera elección de muchas empresas y desarrolladores, pero seguramente será por moda. Tras muchas pruebas sigue siendo el framework con más margen de rendimiento a la hora de desarrollar nuestro proyecto web sea cual sea su tamaño y complejidad, ya que cuenta con todas las características necesarias para ello, además de un entorno intuitivo y de fácil aprendizaje.

Phalcon, suele gustar a todo aquel que pasa por su interfaz, además de estar muy bien documentado por el equipo de desarrollo, con ejemplos prácticos, vídeos, capturas, etc...

CakePHP: Framework de la “Old School”, que se niega a doblar la rodilla ante los que van llegando, por muy fuerte que vengan pisando. Desarrollado en 2005 cuenta hoy día con licencia del MIT y ha llamado la atención de grandes empresas como Hyundai o BMW para todos sus proyectos web.

Este veterano ha sacado a la luz su versión 3.0 con la que quiere volver a lo más alto del top de los frameworks.

Entre sus novedades cuenta con una gestión de las sesiones mejorada, librerías independientes (como ya vimos con Aura) y componentes modulares para reducir el peso y mejorar la personalización de los proyectos; entre otros tantos recursos de hoy día.

Zend Framework: En lo referente a PHP, no hay un top 10 (incluso me arriesgaría a decir un top 5) donde no se encuentre Zend Framework. Aplicación indiscutible para empresas como BBC, Cisco WebEx o BNP Paribas, nos asegura un entorno robusto, fiable y adaptable a cualquier proyecto, sea pequeño, grande o desmesurado, pero cuenta con una curva de aprendizaje extremadamente difícil.

Sólo para configurarlo se requiere dedicar varios minutos, y a la hora de aprender a usarlo es uno de los más complejos, pero los resultados una vez que nos hemos hecho con él, son indiscutiblemente superiores.

No obstante, para la versión que está por venir (Zend Framework 3), los desarrolladores están dándole todo para aligerar esta carga de la dificultad de iniciación. Prometen una interfaz más amigable y en general un entorno más moderno y acogedor, pero de momento habrá que esperar.

Yii: El framework Yii data de 2008, entrando en la categoría de veteranos como Phalcon o CakePHP que vimos en la entrega anterior. Nunca ha estado en las listas de los frameworks más usados, hasta hace un par de años.

Con su versión Yii 2.0 está ganando peso a la hora de elegir framework para el desarrollo de aplicaciones o entornos web a nivel empresarial, dado que cuenta con un conjunto de características de lo más completo, que abarcará tanto proyectos pequeños como de gran envergadura.

Una de sus bazas para ganar tantos puntos y salir elegido en gran cantidad de ocasiones, es una herramienta llamada Gii de generación de código de clase, que facilita enormemente el desarrollo.

Haciendo un poco de spoiler de lo que vendrá a continuación, voy a decir que Yii en la mayoría de las ocasiones en las que se debate su uso o implementación en una empresa o proyecto, compite ni más ni menos que con Symfony, sobre todo si se trata proyectos donde el multiusuario es el pilar fundamental (por aquello del almacenamiento en caché de grandes cantidades de datos), como por ejemplo los CMS o CRM.

La comunidad de Yii tiene mucho peso en su éxito, dado de que es una de las más activas, tanto en soporte y resolución de incidencias como en aportación y desarrollo. Parece que no quieren que Yii se quede donde está y alcance un nuevo y mayor nivel de implantación y uso.

Codeigniter: Ha sido referente durante muchísimo tiempo en el mundo de la programación con PHP, pero parece que no había sabido adaptarse a la competencia que iba surgiendo y las novedades que éstas incorporaban al desarrollo de proyectos web.

Creado en 2006 y distribuido por el MIT bajo licencia de uso libre era la elección lógica para la mayoría de desarrolladores PHP por su ligereza y rapidez de acceso a librerías. Siempre que se recomendaba un framework para iniciarse, Codeigniter aparecía en la conversación como opción de más peso.

Aunque los fieles al proyecto Codeigniter insistan en mantener el framework vivo, su comunidad no parece hacer que la aplicación alcance el nivel de mejoras que incorpora la competencia, y el que su creador EllisLab haya abandonado el proyecto, no ayuda en su resurgimiento.

Pero a pesar de no ofrecer todas las novedades que nos brindan sus rivales, Codeigniter es muy fácil de aprender a usar y continúa siendo una de las mejores elecciones para iniciarse con PHP, ya que la comunidad de soporte es una de las más experimentadas que podemos encontrar.

Symfony2: En el segundo puesto de nuestra tabla de los Frameworks más recomendados y requeridos por las empresas, encontramos Symfony. Nos encontramos ante un ecosistema ideal para enormes y complejos proyectos, que por supuesto, también nos será de gran ayuda en nuestros particulares menesteres con PHP.

Cuenta con un amplio conjunto de componentes de gran reutilización, así como con uno de los aspectos más importantes para que una aplicación salga adelante, una comunidad muy activa que no cesa en aportar ideas o código al desarrollo de novedades o mejoras de cara a futuras actualizaciones.

Su versatilidad se basa como decía, en un conjunto de componentes independientes y reutilizables en su interior. Tal es su eficacia que el framework que encabeza este top 10, usa multitud de elementos de Symfony para mantenerse. Ejemplo de empresas que han visto en Symfony el aliado ideal, Drupal, Piwik, OROCRM... además de que algunos de los componentes de este framework se han usado para mover uno de las mayores plataformas de plantillas de Internet, Twig por supuesto cuenta con licencia de libre uso del MIT. Si te interesa aprender a desarrollar aplicaciones web usando este framework no puedes perderte nuestro curso de Symfony2 con más de 28 horas en vídeo.

Laravel: Y por fin alcanzamos la cima de este top para conocer al rey de los Frameworks de PHP, Laravel. Estamos ante la elección más implantada en las empresas de desarrollo, por lo tanto será requisito para multitud de ofertas de trabajo que encontremos.

Pero no hay que preocuparse, ya que su éxito radica en que ha sido diseñado para ser extremadamente fácil de aprender y usar, además de para favorecer la rapidez en el desarrollo de aplicaciones web. También ayuda el que cuente con su propio motor de plantillas, al que llaman Blade, que nos ofrece el poder escribir nuestro código directamente sobre sus plantillas, ahorrando posibles líneas innecesarias y aligerando por consiguiente la producción y el desempeño de la misma.

Otro gran punto a destacar es el HomeStead, que nos ahorrará la instalación de un servidor web, PHP o cualquier otro software de servidor en nuestra máquina local. Aunque también tiene una pega, y es que es el único punto de Laravel cuya instalación puede resultar “incómoda”, aunque la documentación de su web es de lo más esclarecedora.

Y como el equipo de Laravel no quiere dejar escapar nada, ha escuchado las múltiples peticiones de su comunidad, y ha lanzado Lumen, un micro-framework de PHP con el que aquellos habituados al entorno de trabajo de Laravel, podrán desarrollar proyectos de menor envergadura con una velocidad pasmosa y con la seguridad y robustez del entorno Laravel.