

MARKETING RED

MARKETING RED: Es Marketing en Internet.

El servicio especializado y técnicamente adecuado para implementar su negocio en Internet con éxito.

PHP Y SU USO EN PROYECTOS COMPLEJOS

Se ha escrito y debatido mucho acerca de PHP, tanto de sus bondades como de sus limitaciones. Este artículo no pretende mostrar al lenguaje PHP como la gran panacea ni como EL lenguaje de programación, pero sí mostrar cómo es que se puede utilizar para elaborar proyectos de relativa complejidad.

Modularidad

PHP nació como un lenguaje procedimental, orientado a funciones (lo que se conoce como programación estructurada). Sabemos también que desde la versión 5 de PHP se incluye el soporte para la programación orientada a objetos (POO), que si bien no es aún perfecta brinda al desarrollador herramientas suficientes para desarrollar código complejo aprovechando las ventajas principales de la POO como la herencia, el polimorfismo o la implementación de interfaces. Basándonos en este paradigma de programación, en el desarrollo de ADMINMKRED hemos aprovechado las bondades de la POO para, sobre la base de unas pocas clases principales, tener librerías de código estandarizadas e incluso módulos completos ya listos para incorporar a cualquier proyecto web que desarrollemos. Si necesitamos sacar un módulo nuevo no hay que escribir el código desde cero, basta con "heredar" desde una de estas clases base y "extenderla" para que tenga nuevas funcionalidades, con el consiguiente ahorro en tiempo y esfuerzo de programación.

Organización, orden, claridad

Se dice mucho de que PHP lleva al programador a hacer un código desordenado. Ello puede ser cierto, e incluso he visto programas en PHP con tal grado de desorden que seguirles la pista para buscar un error en ellos es toda una faena. Pero creo que código desordenado o ilegible se puede hacer en virtualmente cualquier lenguaje de programación. Si yo construyo una casa, está en mí hacerlo de manera ordenada, en base unos planos ya hechos de antemano y siguiendo una serie de pasos y pautas, o podría hacerlo "a lo que salga". Del mismo modo, si sigo pautas de codificación, nombres estándar para variables, un formato de código adecuado y comentarios donde deben ir, el código resultante se verá mucho más ordenado y será más legible, tanto al momento de examinarlo como al depurarlo y modificarlo si hiciera falta. La correcta abstracción de un programa en clases y objetos, una correcta separación entre las interfaces de usuario y la lógica de la aplicación, son temas que si se aplican correctamente permiten manejar proyectos grandes de forma ordenada, pero son temas que dependen más del desarrollador que de la herramienta.

Una parte importante del desarrollo de sistemas y que más de uno pasa por alto es la documentación interna (dentro del mismo código fuente). Algunos argumentan que esta actividad toma demasiado tiempo y no es tan importante, otros confían tanto en su memoria que no ven necesario documentar, pues el programador que hizo el código necesariamente debe recordar cómo funciona el programa. Lo cierto es que la documentación no sólo es una gran ayuda para comprender un código fuente, tanto si lo escribió otro programador como si lo escribió uno mismo, sino también permite echar una mirada rápida al funcionamiento general de ese código sin tener que seguir la pista a cada instrucción o función del mismo. Ahora bien, PHP permite la inclusión de comentarios en el código fuente pero no incluye de forma "nativa" una herramienta de documentación; pero eso no significa que no exista una. Está disponible una herramienta llamada phpDocumentor que permite generar documentación de código fuente para sistemas PHP de gran complejidad extrayendo los comentarios del propio código fuente del proyecto (los cuales deben seguir una sintaxis especial similar al JavaDoc que es propio del lenguaje Java). Esta herramienta la utilizamos para documentar las librerías de ADMINMKRED y nos permite saber en pocos minutos para qué sirve determinado método de clase, qué parámetros necesita, qué salida genera, en qué archivo encontrarlo, e incluso ejemplos de cómo debe utilizarse en un proyecto.

Ahora bien, muchos dirán "bueno pero el código de mi programa sólo lo veo yo, el visitante de mi web no lo ve así que qué importa si está 'bonito', se pierde tiempo, además lo que importa es que funcione", o peor aún, hay desarrolladores que intencionalmente hacen un código desordenado, como buscando que nadie más lo entienda y recurrir obligadamente a la persona que hizo el programa si se quiere modificar algo. Puede que eso sea cierto si soy un desarrollador único trabajando en proyectos relativamente pequeños; pero en el desarrollo de grandes sistemas, se trabaje solo o en equipo, el orden y claridad es importante, y por varias razones.

Por ejemplo, imaginemos que nos toca hacer un proyecto web con un módulo único (en el sentido de funcionalidades) que no tenemos implementado aún. Bien, cogemos nuestras librerías base y escribimos el código necesario y lo implantamos en el web. Pasa un tiempo y necesitamos modificar dicha librería para implantarla en otro web. En Marketing Red manejamos una cantidad importante de proyectos, cada uno con su propio conjunto de módulos que le dan funcionalidad y que tiene su propia lógica interna, así que no siempre recordamos al instante la lógica interna de todos y cada uno de los módulos que desarrollamos, y si se trata de un módulo que no tocamos hace un tiempo a veces se necesita "refrescar la memoria" y examinar la librería. Es evidente que esta tarea es más fácil si la librería está ordenada y bien documentada.

Otro ejemplo: Incorporamos un nuevo programador a Marketing Red y hay que enseñarle el funcionamiento del código para que comience a desarrollar su primer proyecto web. Para esto el nuevo programador deberá entender cómo es que funciona el código PHP con el que va a trabajar y la metodología de desarrollo para producir tal código. De nuevo, el desarrollador la tendrá más fácil si le damos como muestra un código lo suficientemente legible no sólo para que lo pueda entender rápidamente sino para que se adapte a un estilo de programación único (que nosotros ya hemos definido previamente) y que permitirá tener al final un código uniformizado que se integrará sin problemas al resto de librerías y, por ende, a cualquier proyecto web actual o futuro que manejemos.

Productividad

Por último está el tema de la productividad. PHP de por sí no viene con un entorno de desarrollo (IDE) propio. Algunos salvan este punto diciendo que no es tan necesario, que todo está en la destreza del programador. Los más radicales proclaman que un simple editor de textos (como el Bloc de notas o vi) son suficientes. En mi opinión, si se va a desarrollar un proyecto grande que involucre todo un análisis previo de concepción y desarrollo, un esquema modular y organizado en clases y librerías, y se necesite seguir la pista de ejecución de nuestro proyecto, ver cómo se procesan los datos, trazar errores, detectar instrucciones redundantes y localizar rápidamente tal o cual librería o método de clase, es claro que un Bloc de notas no nos va a servir de mucho.

Lo bueno es que para PHP hay variedad de herramientas de administración de proyectos de software, desde editores de código sencillos hasta IDE completos como Zend Studio, Eclipse o Netbeans, todos multiplataforma. Estos dos últimos están disponibles gratuitamente para su descarga.

Conclusión

Como conclusión, PHP se ha convertido de un tiempo a esta parte en una poderosa herramienta para desarrollar proyectos de envergadura. Evidentemente no es perfecto (¿qué herramienta lo es?) pero posee características que lo hacen idóneo para el desarrollo de proyectos Web. Pero tener una buena herramienta de programación no garantiza un producto final de calidad: La capacidad del programador para organizarse en función a pautas determinadas, unido con una metodología de desarrollo claramente definida, son también determinantes para cualquier proyecto de desarrollo.

Autor:

**Danilo Quispe Lucana. dquispe@marketingred.com
Jefe de Desarrollo MARKETING RED**