

Consultoría Bootstrap

Gonzalo Garde | DSM Practica 1 | 14/02/2018

# Frameworks considerados:

Bootstrap: es uno de los frameworks más empleados, basándose en el uso de Grid para el diseño de la página por columnas. Se encuentra bien documentado, hincapié en el diseño adaptativo (especialmente relevante si nuestros productos están orientados a ser consumido en diferentes tipos de dispositivos: pc, Tablet, móviles…). Al ser uno de los más usados, en Internet tenemos a nuestra disposición muchos ejemplos y elementos creados por la comunidad, así como un fuerte respaldo ante posibles problemas. Al tener tanto componentes que se apoyan tanto en CSS como Javascript, nos permite darle más dinamismo a nuestra página, a costa de un mayor peso de las librerías. Como es el sistema en el que veníamos trabajando, parte de la curva de aprendizaje para la nueva versión ya la tenemos hecha. Tiene como desventaja que puede llegar a emborronar el código HTML en algunos puntos, ya que hay que adaptarse a su forma de integrar componentes. En el caso de frameworks “completos” orientados a proyectos grandes es el que mejor prestaciones suele presentar.

PureCSS: solo usa CSS. Debido a esto, el peso de la librería es muy pequeña. Esta principalmente pensado a dispositivos móviles, pero el hecho de no usar Javascript hace que en proyectos donde se busque un mayor dinamismo se encuentre limitado. Punto en contra es que su blog oficial no refleja actividad en los últimos 2 años, por lo que no cuenta con el soporte que tiene bootstrap.

Foundation: uno de los frameworks más empleados y, por tanto, al igual que bootstrap cuenta con una comunidad amplia detrás y mucho material de apoyo en internet. Su filosofía es “mobile-first” de manera que su punto fuerte es el desarrollo de productos para este tipo de plataformas, aunque evoluciona bien para otras resoluciones. Cuenta además con un set de iconos ya precargados. El principal problema, al igual que bootstrap y otros frameworks más elaborados, es que una actualización puede suponer bastante trabajo para “traducir” el código de la versión anterior a la actual.

Semantic UI: su principal punto fuerte es que se orienta a la claridad del código para el usuario final (o el desarrollador, mejor dicho). Requiere de un dominio de JavaScript más profundo que en el caso de Bootstrap o foundation. Ofrece más temas que bootstrap (más riqueza en este sentido) a cambio de hacer más compleja la inclusión de Semantic UI en nuestro proyecto.

Min: siguiendo la ideología de PureCSS de buscar el minimalismo. Como punto positivo es que su tamaño es mínimo y que funciona prácticamente en cualquier navegador. Además, el tamaño de la librería agiliza la carga de la página. No está pensado para proyectos complejos, pero es una alternativa muy intersante para proyectos simples.

Conclusión: realmente, esto es solo una muestra de los diferentes frameworks que vamos a encontrar, pero podemos hacernos una idea de por dónde se mueven. Tendremos que tomar decisiones basándonos principalmente en:

* La complejidad del proyecto y velocidad para escribir el código, para lo cuál frameworks más complejos (Semantic, Bootstrap, Foundation…) son más “simples” de usar, a costa de un mayor consumo de recursos, tiempo de carga
* Legibilidad de nuestro código
* Capacidad de nuestro servidor, tanto a la hora de servir páginas al cliente (lo que nos puede hacer decantarnos por soluciones más o menos pesadas) como de almacenamiento
* Curva de aprendizaje (en este caso, seguramente sea más sencillo adaptarnos a la nueva versión de bootstrap que empezar de cero)
* Dinamismo de la página
* Dispositivos clave a los que nos dirijamos, no es lo mismo diseñar principalmente para móviles que para ordenadores de sobremesa.

Por lo que no se puede hablar de forma general de una solución mejor, si no que es dependiente del proyecto en cuestión. La recomendación sería seguir empleando Bootstrap pero comenzar también a documentarnos en soluciones más ligeras como PureCSS o Min, ya que pueden ser interesantes.