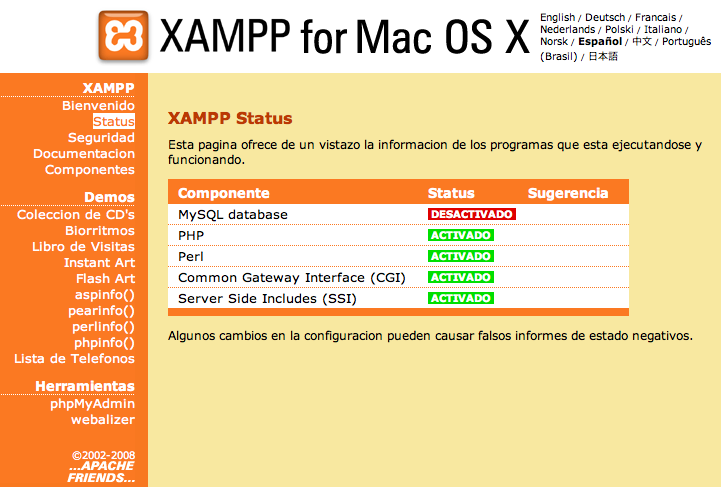
# Preparación e instalación del entorno de desarrollo

En todo proyecto web de complejidad media o alta debes distinguir claramente entre tres entornos de trabajo:

1. **Entorno de desarrollo:** normalmente en un equipo local, a veces en un servidor de desarrollo, es el entorno en el que trabajarás para crear, probar, revisar y mejorar tu código.
2. **Entorno de preproducción:** normalmente un servidor lo más parecido posible al que alojará definitivamente tu aplicación, sobre todo en lo que a software y versiones de software se refiere: PHP, Apache, MySQL, librerías, etc. Aquí a veces podrás dar algún retoque para que la aplicación termine de funcionar, pero los cambios en el código debes hacerlos siempre en el entorno de desarrollo.
3. **Entorno de producción:** es el entorno que aloja definitivamente tu aplicación y la base de datos, ya sea en uno o varios servidores. Quita las manos de este servidor, cualquier error que detectes debes intentar reproducirlo en preproducción para no interrumpir el servicio a tus usuarios y no comprometer sus datos.

Puedes desarrollar el código de tu programa en la máquina donde normalmente trabajes. Para ir probando la aplicación conviene que instales una serie de servicios que, dependiendo de tus habilidades como administrador de sistemas te resultarán más o menos fáciles. Una solución cómoda y efectiva en este terreno es instalar un paquete integrado tipo **LAMP**, **WAMP** o **MAMP**. La primera letra de estos acrónimos corresponde al sistema operativo sobre el que corren: Linux, Windows o Mac; el resto de las iniciales son para Apache, MySQL y PHP. Hay una estupenda distribución disponible para los sistemas operativos más frecuentes llamada **XAMPP**, que además incorpora otros interesantes paquetes como phpMyAdmin, Webalizer, OpenSSL, etc. Puedes descargar XAMPP gratuitamente desde <http://www.apachefriends.org/es/xampp.html>



## Descargando Cáscara

Una vez instalados y configurados estos servicios descarga el paquete de Cáscara desde <https://github.com/juananruiz/cascara>

Descomprime el fichero descargado en el directorio raíz del servidor web, copia la carpeta **app\_sample** a otra carpeta al mismo nivel que ella pero con el nombre de tu proyecto y ya estás listo para empezar.

Si utilizas **git** para control de versiones, algo que te recomiendo vivamente, puedes bajarte el código con [git@github.com:juananruiz/cascara.git](mailto:git@github.com:juananruiz/cascara.git) ???

# La estructura de directorios de Cáscara

Cuando te descargas Cáscara obtienes una carpeta llamada cascara que a su vez contiene tres carpetas:

* **cascara\_core**: con el núcleo común
* **app\_sample**: una plantilla de aplicación, de la que puedes sacar tantas copias como aplicaciones necesites.
* **icons**: una completa librería de iconos libre de derechos de autor para utilizarlos en tus aplicaciones

Dentro de **app\_sample** tienes las siguientes carpetas:

* **app\_code:** el directorio donde están todos los controladores (\*.php) y plantillas (\*.tpl) de la aplicación. Si tu aplicación es muy grande o tiene partes muy diferenciadas puedes ir separando estos ficheros en subcarpetas o usar prefijos en los nombres de los ficheros, tu mismo.
* **cache, config y templates\_c:** son directorios que smarty necesita
* **class:** aquí van todas las clases, mayoritariamente representan la parte del "modelo". Se podría haber llamado "model" pero estoy muy acostumbrado a "class", puedes cambiar el nombre si lo prefieres
* **public:** aquí van todos los archivos que tienen que ser accesibles desde el exterior. El "index.php" que sería "El Gran Controlador", los "themes", el directorio javascript ("js") y una carpeta "upload" por si los usuarios tienen que subir ficheros.

Por último, dentro de **cascara\_core** están:

* **lib:** con todas las librerías de terceros que se utilicen en la aplicación. En la versión actual tienes: adodb5, pChart, pExcel, phprtflite y smarty. La primera y la última son imprescindibles (o no) para que funcione el tinglao. El resto puedes eliminarlas si no las vas a usar o dejarlas vivir.
* **function:** aquí irán las funciones de código a las que se llame desde distintos lugares dentro de la aplicación y no tengan cabida en los modelos, por defecto viene sólo con **sanitize.php**.

Podéis verlo más claro con estas dos capturas de pantalla:

