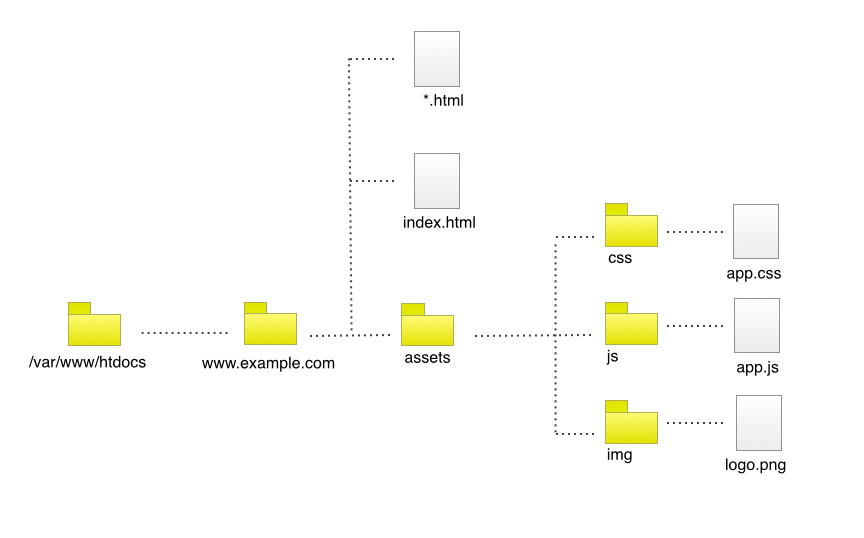
**Organización de Proyectos**

Si ya has hecho [sitios o aplicaciones web](https://blog.eamexicano.com/consejos/conceptos/) es probable que tengas una manera de organizar tus archivos. Si apenas estás empezando o quieres tener otra referencia de cómo puedes organizar tus proyectos esta entrada te puede servir.

**Nota:**  
Las organizaciones mencionadas a continuación son para proyectos simples. No está considerado el uso de herramientas como [grunt](https://blog.eamexicano.com/herramientas/instalar-grunt/" \t "_blank), [yeoman](https://blog.eamexicano.com/herramientas/yeoman/" \t "_blank) o de algún framework.

**Sitio web**

*Organización de archivos para sitios web base*  


Referencia de las carpetas y / o archivos:

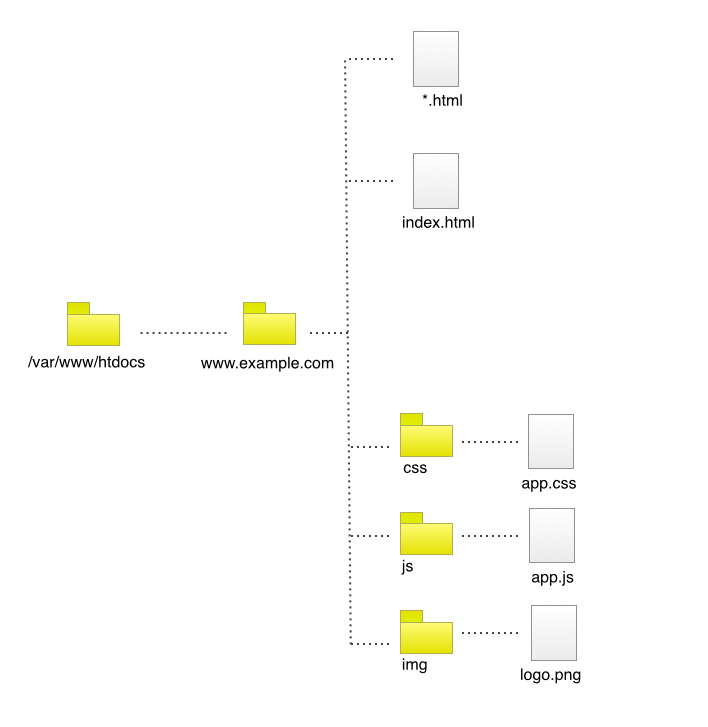
www.example.com: Carpeta que contiene el proyecto. Es probable que quieras nombrar esta carpeta como el URL que tiene asociado el sitio – www.example.com – para que identifiques más rápido cada sitio.

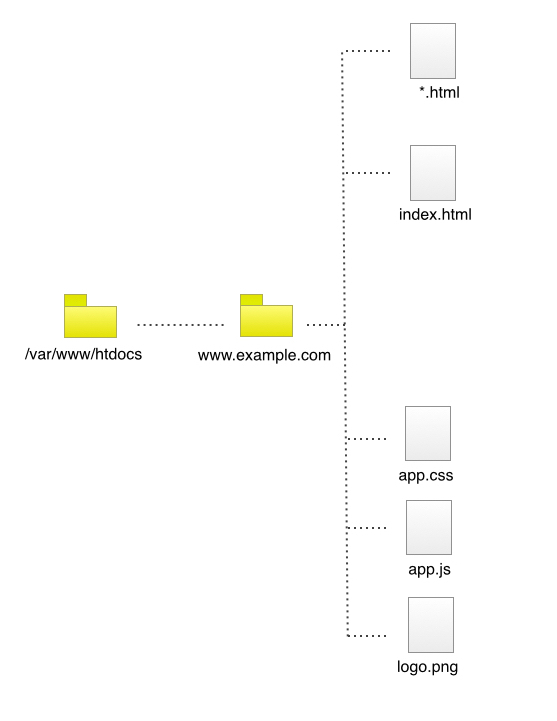
index.html: Documento – predeterminado – inicial. Generalmente el servidor web lo toma como archivo predeterminado y se envía si no se incluye como parte del URL – dirección web -.

\*.html: Documento(s) con más contenido del sitio.

assets: Carpeta que contiene archivos adicionales como hojas de estilo – css -, código en JavaScript – js – o imágenes – img -. Si utilizas otro tipo de contenido puedes agregar carpetas adicionales – audio, video, pdf, swf, -.

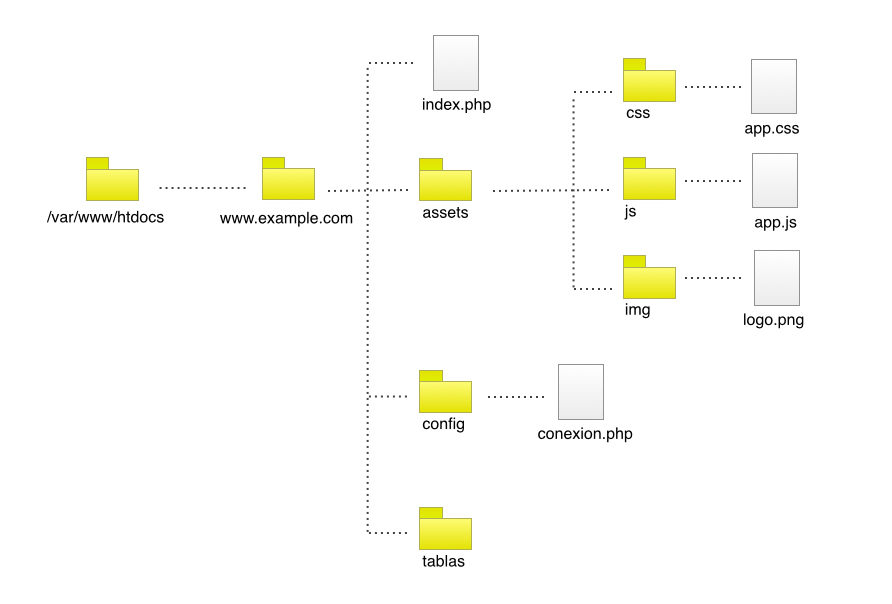
Utilizo la carpeta assets porque me gusta tener una estructura similar entre los sitios y aplicaciones – básicas – pero puedes eliminar la carpeta assets y utilizar sólo las carpetas css, js, img – y adicionales – o, si solo utilizas un archivo de cada tipo también los puedes incluir en el directorio raiz.

*Sin carpeta assets*  


*Archivos dentro del directorio principal*  


**Aplicaciones web**

Esta organización la tomé “*prestada*” de las primeras versiones de [Rails](http://rubyonrails.org/" \t "_blank) / [CakePHP](http://cakephp.org/" \t "_blank). Esta organización representaría únicamente la V – de la arquitectura MVC – con un poco de [Código espagueti](https://es.wikipedia.org/wiki/C%C3%B3digo_espagueti).



Referencia de las carpetas y / o archivos:

www.example.com: Carpeta que contiene el proyecto. Es probable que quieras nombrar esta carpeta como el URL que tiene asociado el sitio – www.example.com – para que identifiques más rápido cada aplicación.

index.php: Parte inicial de la aplicación. Puedes utilizar index.html si la página principal de tu proyecto no necesita preprocesamiento.

assets: Al igual que en la organización de archivos para sitios web, esta carpeta contiene archivos adicionales a la aplicación – CSS, JS, imágenes -.

config: Aquí se encuentran los archivos de configuración de la aplicación entre los cuales está la conexión a la BD.

tablas: Esta carpeta – generalmente son más de una – tiene el nombre de una tabla que tengas en tu base de datos – con excepción de las tablas de unión en las relaciones M:M pero ese caso también lo vamos a ver – . Los archivos que se van incluir dentro de cada una de las carpetas van a servir para para administrar los registros de esa tabla – patrón CRUD -.

Una vez que veamos cómo conectar un archivo PHP con MySQL voy a retomar esta sección para describir cada uno de los archivos.

**Recuerda:**  
Cuando hay orden es más fácil encontrar y solucionar errores – y desarrollar -.

https://developer.mozilla.org/es/docs/Learn/Getting\_started\_with\_the\_web/Dealing\_with\_files

[**¿Qué estructura debe tener tu sitio web?**](https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/Dealing_with_files#%C2%BFqu%C3%A9_estructura_debe_tener_tu_sitio_web)

A continuación, veamos qué estructura debería tener tu sitio de prueba. Las cosas más comunes que tendrás en cualquier proyecto de sitio web que crees son un archivo de índice HTML y directorios para contener imágenes, archivos de estilo y archivos de script. Crea estos ahora:

1. **index.html**: Este archivo generalmente tendrá el contenido de tu página de inicio, es decir, el texto y las imágenes que las personas ven cuando visitan tu sitio por primera vez. Usando tu editor de texto, crea un nuevo archivo llamado index.html y guárdalo dentro de tu directorio pruebasitio.
2. Directorio **images**: Este directorio contendrá todas las imágenes que utilices en tu sitio. Crea un directorio llamado images, dentro de tu directorio pruebasitio.
3. Directorio **styles**: Este directorio contendrá el código CSS que se utiliza para aplicar estilo al contenido (por ejemplo, configurar el texto y los colores de fondo). Crea un directorio llamado styles, dentro de tu directorio pruebasitio.
4. Directorio **scripts**: Este directorio contendrá todo el código JavaScript utilizado para agregar funcionalidad interactiva a tu sitio (por ejemplo, botones que cargan datos cuando se hace clic en ellos). Crea un directorio llamado scripts, dentro de tu directorio pruebasitio.