DOCUMENTACIÓN CMS-BUILDER

# 1. Requisitos Previos

- XAMPP instalado.  
- Editor de texto (recomendado VS Code).  
- PHP 7.4+.  
- phpMyAdmin disponible.  
- Sistema operativo: Windows.

# 2. Configuración de Dominios Locales

Paso 1: Editar httpd-vhosts.conf en C:\xampp\apache\conf\extra

<VirtualHost \*:80>  
 DocumentRoot "C:/xampp/htdocs/cms-builder/api"  
 ServerName cms-builder-api.com  
</VirtualHost>  
  
<VirtualHost \*:80>  
 DocumentRoot "C:/xampp/htdocs/cms-builder/cms"  
 ServerName cms-builder-dash.com  
</VirtualHost>  
  
<VirtualHost \*:80>  
 DocumentRoot "C:/xampp/htdocs/cms-builder/web"  
 ServerName cms-builder-web.com  
</VirtualHost>

Paso 2: Editar archivo hosts en C:\Windows\System32\drivers\etc

127.0.0.1 cms-builder-api.com  
127.0.0.1 cms-builder-dash.com  
127.0.0.1 cms-builder-web.com

# 3. Estructura de Archivos

* index.php: Carga la plantilla base y depura errores.
* .htaccess: Permite rutas amigables con mod\_rewrite.
* template.php: Captura parámetros de URL, carga hojas de estilo y plugins.
* install.php: Formulario de instalación del CMS.
* forms.js: Validación de formularios con Bootstrap 5.
* custom.css: Hoja de estilos personalizada.
* login.php: Implementa login post-instalación.

# 4. Instalación del Sistema

1. Crear una base de datos en phpMyAdmin con codificación utf8\_general\_ci.  
2. Editar install.controller.php para ingresar el nombre de la base de datos y, si aplica, la contraseña del hosting.  
3. Ejecutar install.php para iniciar la instalación automática mediante la API.  
4. Una vez instalado, se activa el sistema de autenticación y login de administradores.

# 5. Lógica del Backend

install.controller.php: Crea todas las tablas necesarias a través de la API.

curl.controller.php: Realiza peticiones a la API (GET, POST, PUT, DELETE).

admins.controller.php: Control de acceso y validación de administradores.

# 6. Uso de Postman

Se utilizó Postman como herramienta de pruebas para realizar solicitudes a la API del CMS.   
Esto incluye la verificación de rutas, métodos de autenticación, validación de datos y pruebas de endpoints   
para la creación, edición y eliminación de registros.

# 7. Recomendaciones

- Leer toda la documentación de la API para crear nuevas funcionalidades.  
- Nunca generar tablas directamente: usar los scripts SQL automatizados por la API.  
- Si se migra a producción, actualizar install.controller.php con credenciales del hosting.