

**INSTRUCTIVO DE DESPLIEGUE**

**[****Sistema para un Blog]**

# OBJETIVO

Este documento tiene el objetivo de describir los pasos requeridos para el correcto despliegue del proyecto en un ambiente local.

# COMPONENTES

**Requisitos Previos**

* Servidor local, APACHE
* PHP
* Composer
* Scoop
* IDE o editor de Texto

**NOTA IMPORTANTE:** Asegúrese de estar usando PHP 7, para ello. Para ello la instalación de composer debió realizarse sobre PHP 7.

Si no se realizó así o se duda, desinstalar y reinstalar nuevamente.

**Instalación del Servidor Local**

**Apache:** el servidor web de código abierto es la aplicación más usada globalmente para la entrega de contenidos web. Las aplicaciones del servidor son ofrecidas como software libre por la Apache Software Foundation.

**Empezando a instalar un servidor local en XAMPP.**

* Logotipo

  Descripción generada automáticamenteDescargamos el instalable desde su página web. Según la versión de vuestro Windows, seleccionaremos una u otra opción.
* Ejecutamos el instalador. Selecciona la carpeta de instalación (normalmente c:/xampp64/).
* Para iniciar el XAMPP., podemos hacerlo desde el menú de Inicio -> Todos los programas -> XAMPPcontrolpanel -> Start.
* Seguramente aparezca un aviso de seguridad. Podéis decir que Sí, y se abrirá el programa.

**Instalación de Composer**

**Composer** es un sistema de gestión de paquetes para programar en PHP el cual provee los formatos estándar necesarios para manejar dependencias y librerías de PHP.

**Para la instalación en Windows**

* Descargar Composer (<https://getcomposer.org/download/>) de este enlace y lo instalaremos siguiendo los pasos que se indiquen.
* Para comprobar su correcta instalación es mejor que se cierre el terminal si se lo tiene abierto, volver a ejecutar y poner lo siguiente en el terminal:

composer -v

**Scoop:**

Este administrador de paquetes instala los programas que conoce, desde la línea de comandos con una cantidad mínima de fricción.

Pasos para la instalacion:

Abrir Powershell y ejecutar:

* Set-ExecutionPolicy RemoteSigned -scope CurrentUser
* Invoke-Expression (New-Object System.Net.WebClient).DownloadString('https://get.scoop.sh')
* scoop install symfony-cli

**Persistencia**

**Doctrine**

Logotipo, Icono

Descripción generada automáticamenteUna de las tareas más comunes y a la vez más complejas de la programación web consiste en la persistencia de la información en una base de datos. Afortunadamente, Symfony incluye la librería Doctrine, que proporciona herramientas para simplificar el acceso y manejo de la información de la base de datos. En este capítulo aprenderás la filosofía de trabajo de Doctrine y lo fácil que puede ser trabajar con bases de datos. Doctrine no tiene ninguna relación con Symfony y su uso es totalmente opcional. Este capítulo se centra en el ORM, que te permite manejar la información de la base de datos como si fueran objetos de PHP. También puedes realizar consultas SQL directamente, para lo cual tienes que utilizar la librería DBAL de Doctrine en vez del ORM.

**Instalación y Configuración**

**Instalación**

Para instalar Doctrine usamos el siguiente comando:

composer require doctrine maker

**Configuración**

La información de conexión de la base de datos se almacena como una variable de entorno llamada DATABASE\_URL.

Para el desarrollo, podemos modificarlo dentro de. env:

###> doctrine/doctrine-bundle ###

# Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url

# IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml

#

# DATABASE\_URL="sqlite:///%kernel.project\_dir%/var/data.db"

 DATABASE\_URL="mysql://admin:admin@127.0.0.1:3306/sistema\_blog?serverVersion=8&charset=utf8mb4"

# DATABASE\_URL="postgresql://app:!ChangeMe!@127.0.0.1:5432/app?serverVersion=15&charset=utf8"

###< doctrine/doctrine-bundle ###

**Framework utilizados:**

Imagen que contiene Icono

Descripción generada automáticamente**Symphony**

Es un framework diseñado para desarrollar aplicaciones web basado en el patrón Modelo Vista Controlador. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web.

# PASOS DESPLIEGUE

Realizar un git clone del repositorio: https://gitlab.com/sasf\_ec/prueba-tecnica-symfony.git

Instalar composer para crear carpetas var y vendor: composer install

Crear la base de datos: DATABASE\_URL="mysql://admin:admin@127.0.0.1:3306/sistema\_blog?serverVersion=8&charset=utf8mb4"

Realizar una migración: php bin/console doctrine:migrations:migrate

Actualizar los datos en la base de datos: php bin/console doctrine:schema:update --force

# CONTROL DE CAMBIOS

|  |  |  |  |
| --- | --- | --- | --- |
| VERSION | DESCRIPCION DE CAMBIOS | FECHA | AUTOR |
| 1.0 | **CREACIÓN DE DOCUMENTO** | **16/02/2023** | **ISAAC GÓMEZ** |
|  |  |  |  |
|  |  |  |  |