# Descargo de Responsabilidad

El presente Proyecto está escrito en PHP solo esta diseñado para fines educativos, aunque en un futuro pudiera ser perfeccionado y preparado para proyectos reales, la idea de este es concientizar en la utilización de un framework o por lo menos organizar el código creado en PHP.

Tengo la firme idea que la mejor forma de aprender es haciendo es por eso que en algún momento me decidí a hacer mi propio framework, la idea no es competir con otros ya existentes que gozan de buena “reputación” pero la verdad que no existe ninguno perfecto y a veces no se adaptan a todas las condiciones, por tanto, aquí les va,

**JADME PHP**

**Por Maxwell Sotomayor Ramos**

**Es importante aclarar que este código es vulnerable a ataques,**

1. **Pues los request están expuestos,**
2. **No cuenta con un ORM todavía**
3. **No maneja sesión todavía**

# Por qué veo esto?

Bueno primero que todo hace unos días se me ocurrió escribir este pequeño microfrawork que no es mas que resumen a muy bajo nivel del verdadero jadmephp, el cual uso en uno de mis proyectos, medsabia.com, lo que estaré aquí escribiendo le colocamos el nombre de framework, por llamarle de algún modo ya que la idea principal es organizar el código, créanme que soy de los que creo una clase en cualquier lugar cuando llega una idea a la mente pues es la carpeta que estoy ahí mismo creo la clase o el método que este implemento la funcionalidad.

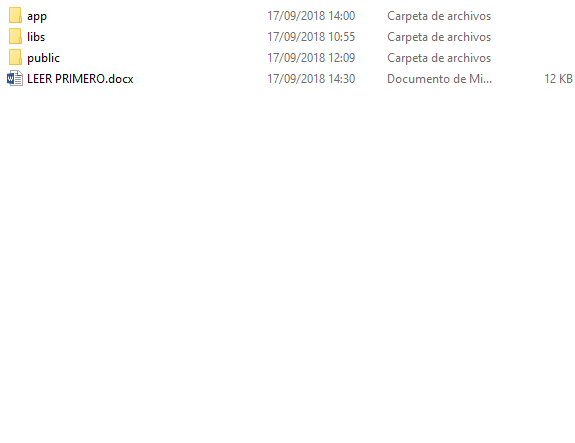
La experiencia me ha demostrado que no debería hacerlo ahí, pues ya mucha gente ante que yo creo patrones de diseños y responsabilidad, porque de seguro al igual que lo hacia y e dio cuenta que tenía que escribir un montón de código muchas veces duplicado porque no encontraba forma de reutilizarlo o a la hora de hacer crecer la aplicación todo e complica, porque esa es otra cosa, mucha vece nos imaginamos un proyecto súper fácil, creemos que solo será un login, dos querys y dos tablas y las dos meses hemos escritos más código que phpmyadmin.

En la actualidad existen muchos framework de este tipo, pero la verdad me he animado a escribir este porque tengo la intención de ensenarte a entender como funcionan, una vez que entres el este mundo veras y te darás cuenta, al igual que me paso a mi, como otros como Symfony tienen desventajas o cosas que preguntas “¿Por qué lo hicieron asi? Si así es más fácil”.

Antes que nada, te recomendaría que comenzaras estudiando patrones de diseño, patrones de responsabilidades, e muy importante que entiendas que es MVC, por motivos propios por ahora no explicaré nada sobre eso pues la idea es la practica en código pero seria bueno leyeras sobre ellos son muy importantes,

# Distribución básica de directorios

A continuación, explicare o tratare de justificar la distribución de directorios y que contiene cada uno. También hablare un poco sobre algunos archivos muy importantes, cuáles serían sus funciones.



**App**

En esta carpeta colocamos nuestra aplicación, por ahora solo usaremos MVC y no otros más populares como HMVC, es importante aclarar esto. En este puntos esta el código del “framework” y el código de nuestra aplicación.

**Libs**

Nuestras librerías que podemos crear o utilizar, por ahora esta vacia.

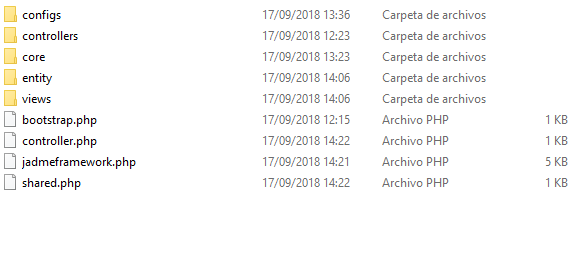
**Public**

El inicio de todo comienza por aquí, es la carpeta publica de nuestro proyecto nuestro ROOT

# Si ya te aburriste de tanta explicación y no eres de leer mucho como yo, bueno ver [aqui](#_Un_ejemplo_practivo)

### Carpeta App

Dentro de esta carpeta encontramos los siguientes recursos



**Configs**

**Controllers**

**Core**

**Entity**

**Views**

#### Algunos achivos que también se encuentran son los siguientes

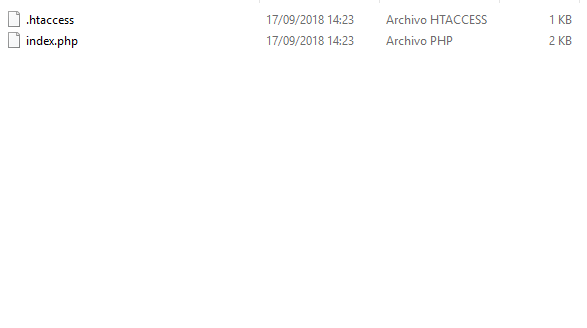
Boostrap.php

Controller.php

Jadmeframework.php

Shared.php

### Carpeta Public



# Un ejemplo práctico

1. Descarga el proyecto y colócalo en tu carpeta raíz, por ejemplo diremos que esta en jadmePhp,

Vamos a suponer que colocaste en C:\xampp\htdocs\ y ahí copiaste la carpeta del Proyecto.

1. Ves a la carpeta **app**/**controllers**, crea un controlador por ejemplo PersonController.php, recuerda que es importante,
   1. Este bajo el namespace correcto, o sea, namespace App\Controllers; fíjate en el controlador Default
   2. Herede de la clase Controller, debe haber instanciado antes use App\Controller;
   3. Tenga por lo menos la **function** index(), que tenga esta function es muy importante por es el action inicial en caso que no se especifique nada a este recurso
2. Colocate en la carpeta app/views y crea un nueva carpeta puede ser por ejemplo Person y dentro crea una vista index.php.html, asegúrate de agregarle algun contenido, como por ejemplo el clásico “Hello world”
3. Ahora en la function index del controlador antes creado -que te dije que era obligatoria no pienses que se me había olvidado- vas a retornar esta vista, ¿pues como lo haces? Que no se ensuciar ese controller con un require por favor, fíjate encima que ese controller ya hereda de otro que implenta al asi, dentro de la function index() coloca el código siguiente;

$this->render("index@person");

Pués este fragmento de código a devolver la vista index.php.html antes creada en la carpeta app/views/person