Objeto

El objeto del proyecto es investigar sobre la arquitectura hexagonal, cuáles son sus ventajas, inconvenientes y como poder implementarlo, para ilustrar la implementación, se harán unos ejemplos para cada una de las partes más significativas.

Dichas partes son, puertos, adaptadores, comunicación entre las capas.

Los lenguajes elegidos para la realización de este proyecto son PHP para la parte del servidor y en JavaScript para la parte de cliente.

Se utilizara el framework Slim para abstraer el manejo de las peticiones HTTP.

Queda fuera de este proyecto otro tipo de arquitecturas como Onion Architecture, Screaming Architecture, Lean Architecture, etc…, otros tipos de patrones que los indicados anteriormente y la utilización del framework Slim en profundidad.

Tampoco se van a realizar ejemplos en otros lenguajes de programación que los especificados, ni se va a entrar en el detalle de cómo funcionan las herramientas elegidas en este proyecto.

Dada la naturaleza del proyecto,

Motivación.

Aprender una nueva arquitectura del software, de las consideradas limpias, que intenta solucionar los problemas de acoplamiento para que las aplicaciones se adapten mejor al cambio.

Tecnologías

Como base de datos se ha optado por MySQL, muy potente adecuada para almacenar la información, para gestionar la base de datos se utilizara MySQL Workbench es una aplicación de escritorio, que permite trabajar con la base de datos de manera visual, se ha preferido esta opción en vez de PHPMyAdmin.

En cuanto a los lenguajes de programación se ha optado por dos, una para realizar la parte que se ejecuta en el servidor, PHP y otro en la parte cliente, JavaScript. El motivo para esta elección es por su sencillez y rapidez con la que se desarrollaran aplicaciones Web.

Se ha optado como servidor web Apache que con diferencia el servidor Web más usado de la actualizad.

La aplicación web correrá dentro de una máquina virtual, montada con VirtualBox, uno de los softwares de virtualización más usados, se opta por esta composición para poder controlar mejor el entorno donde se va a ejecutar la aplicación. Para automatizar el proceso de montaje de la máquina virtual se va utilizar Vagrant, este software funciona definiendo un archivo de configuración con las características que tiene que tener la máquina virtual y el programa se encarga de montarla, también provee de herramientas para gestionar la máquina virtual.

La máquina virtual correrá con Ubuntu sin entorno gráfico, cualquier modificación de la de la configuración de la maquina virtual, abra que definirla en el archivo de configuración, para modificaciones puntuales se puede utilizar una conexión ssh que configura Vagrant automáticamente para ese propósito. La programación se realizara en la maquia anfitrión, dentón de una carpeta compartido con la maquina virtual.

La gestión de librerías se realizara a través de Composer, esta aplicación nos permite a través de un archivo de configuración, definir las librerías que sean necesarias utilizar, y se encarga de gestionar la descarga y resolver las dependencias entre librerías, Composer implementa el estándar PSR-4 para realizar esta tarea, este estándar está orientado a los nombres de los Namespaces, Clases y Ficheros, autolouder, para así evitar colisiones entre las librerías y nuestra aplicación.

Se ha escogido GIT como sistema de control de versiones, nos permite tener versionado el código sin tener que montar un servidor para ello, otro de los motivos es la cantidad de información que existe en la red y la gran comunidad que lo apoya.

El entorno de desarrollo que se va utilizar es PHPStorm, este ID integra algunas de las herramientas que se pretenden utilizar para la realización del proyecto, de esta manera no hay que ir cambiando entre aplicaciones, para realizar tareas repetitivas como hacer un commit, levantar la máquina virtual o añadir una librería.