**Navegador web:** URL: localhost:8000/mis\_pruebas/entidades. Alquien escribe la URL para ver una pagina de nuestra aplicación por ejemplo un listado de entidades

**Ngnix:** location /mis\_pruebas { rewrite ^(.\*)$ /mis\_pruebas/index.php?$1 last;}

Se redirecciona cualquier URL que contenga el sufijo /mis\_pruebas a index.php

**Config.php**

Constantes de conexión base datos y ruta de index.php

**Index.php**

Add (GET, ruta URL, func(controller->list())

Dispatch()

**\*Router**.**php** class router {

metod add: añade en una tabla bidimensional con indice asociativo routes[metodo][ruta URL]=funcion ejecutar;}

metod dispatch{ejecuta funcion guardada que coincida con la URL actual}

**EntidadController.php**

Class entidadController{

*New EntidadeService*

*New page*

Llama a EntidaService\_list() para sacar los datos de la consulta deseada.

Llama a pages->render para mostrarlos al cliente

A render se la pasa empaquetada una tabla con objetos class entidad. **(ver explicacion del patron empaquetar que hice en un world**

**Pages.php**

class Pages{

render} render desempaqueta la tabla de objetos y crea una variable tabla con los objetos, para que el cuerpo.php pueda mostrar los datos de la consulta

header.php

cuerpo.php

footer.php

aquí si muestra la pagina html con los datos de la consulta

**EntidadServicio.php**

Class EntidadService { findAll()}

Esta clase no hace nada, esta para poder crear varios tipos de conexiones a base de datos de otros tipos

**BaseDatos.php**

class BaseDatos

Aquí esta la conexión con la base de datos sql, la consulta y la conversion de los datos de consulta a un tabla asociativa fetch\_all(MYSQLI\_ASSOC)

**Entidad.php**

Class entidad{

Esta es la clase que crea los objetos de tipo entidad para contener los datos de la consulta sql

**EntidadRepository.php**

class EntidadRepository

Aquí es donde se crea la conexión con la class base de datos, se envia la consulta (select)a la class base de datos y se convierte la tabla con la consulta en una tabla de objetos de tipo class entidad

**Router**.**php: aquí tenemos una clase router con dos metodos, el metodo add sirve para añadir en una tabla bidimensional routes statica, con indices aasociativos,** routes[metodo, GET, POS, …][ruta URL /mis\_pruebas/entidades]=funcion ejecutar;, la funcion a ejecutar es una consulta sql para mostrar al cliente, es el desencadentante de todo.

Index.php es el que guarda las rutas. Al guardar en add estos datos estamos indicando que rutas van a tener una accion asociada, solo esas rutas podran ser ejecutadas por el cliente.

Despues de almacenar las rutas que vamos a usar (habra que hacer varios add, uno con cada ruta que necesitemos ejecutar, index.php llama a dispatch que esta em router y aquí se recorre la tabla router en busca de alguna funcion que se haya guarada y coincidad con la URL que acaba de llamar a index, es decir, index se carga porque alguien escribe en la url del navegador localhost:8000/mis\_pruebas/xxxxxx, pues cada vez que eso ocurre se viene a index, index almacena lo que tiene configurado en add, y luego lo ejecuta, pero solo ejecuta la URL que acaba de llamarlo, de todos los add almacenados, solo ejecuta el de ahora, y si lo encientra en la tabla routes, porque si escribimos una URL no valida no hara nada.

**COMO FUNCIONAN LOS LINK href de los HTML o PHP**

**Hay** dos formas de poner un link, absoluta o relativa.

***Absoluta:*** supongamos que estamos en la URL **localhost:8000/mis\_ pruebas/entidades** y pongo un boton con **href=”/vehiculos”** esto lo que hace es quita todo el sufijo despues de **localhost:8000** y pone el sufijo indicado en **href=/vehiculos**, saltando a la URL **localhost:8000/vehiculos**. Al quitar el sufijo **/mis\_pruebas/** el servidor ngnix no va a redireccionar a index.php y como encima **/vehiculos** no existe pues dara error. El interprete web detecta que hay una **/** antes del la ruta indicada y lo toma como absoluto.

***Relativa***: supongamos que estamos en la URL **localhost:8000/mis\_pruebas/entidades** y pongo un boton con **href=”vehiculos”** esto lo que hace es saltar a la URL que tengo pero cambiando entidades por vehiculos, **localhost:8000/mis\_ pruebas/vehiculos**. Si en router hay añadida una accion para vehiculos pues cargara algo, sino hay nada añadido dar error porque vehiculos no es una pagina, para que fura pagina que sepudiera cargar tendria que ser vehiculos.html o php. Pero nosotros todo lo pasamos por el router, por eso no usamos .php o .html, el router junto con la clase Pages se encargan de hacer la composicion nombre de pagina.extension.

Otro caso de relativa es si estoy en **localhost:8000/mis\_ pruebas/entidades** y pongo **href=”pages/vehiculos”** cargara **localhost:8000/mis\_ pruebas/pages/vehiculos** ya que al no haber **barra /** delante de pages lo toma como relativa y cambia entidades por **pages/vehiculos.**

Despues de lo explicado podemos ver que si estoy en **localhost:8000/mis\_ pruebas/entidades** , un **href=”vehiculos”** y un **href=”/mis\_ pruebas/vehiculos”** me llevaran al mismo sitio, ya que el 1º caso, **href=”vehiculos”** , es relativo ysaltara a **localhost:8000/mis\_ pruebas/vehiculos, y el 2º caso, href=”/mis\_ pruebas/vehiculos”** al llevar la **/** delantede **mis… s**e toma como abosluto y quitará todo el sufijo y pondrá **/mis\_ pruebas/vehiculos** quedando **localhost:8000/mis\_ pruebas/vehiculos**

**Las sentencias que hay en la clase Pages:**

**require\_once "pages/header.php";**

**require\_once "pages/$pageName.php";**

**require\_once "pages/footer.php";**

no tienen nada que ver con lo explicado sobre **href. Require\_once** lo que hace es insertar y ejecutar el codigo del fichero **.php,** indicado en la ruta relativa a donde esta **index.php,** (“pages/fichero.php”).

**Nota**: cada vez que pincho en un boton con un href entidades o nueva entidad se va a index.php por el enrutamiento de ngnix. Asi que al pinchar en un href, se sale de la url actual eliminado todas las variables del programa, la tabla routes incluida y se ejecuta de nuevo el index.php, creando de nuevo la tabla routes y ejecutando todo de nuevo, por eso por muchas veces que le pinche a los link href de entidades para adelante o para atrás no se lleva la tabla routes de datos.