

MANUAL TECNICO DEL SISTEMA

**APLICACIÓN WEB PARA GESTIONAR ACTIVIDADES, CITAS Y MEDICAMENTOS DE
PACIENTES CON CÁNCER DE ESTÓMAGO**

**WILLIAM ALBERTO PATERNINA ROMO
LUIS GUILLERMO SALGADO SALGADO**

**PROGRAMA DE INGENIERIA DE SISTEMAS
FACULTAD DE INGENIERIAS**

**UNIVERSIDAD DEL SINÚ ELIAS BECHARA
ESCUELA DE INGENIERÍA DE SISTEMAS
CARTAGENA DE INDIAS D.T. Y C.**

2018

MANUAL TECNICO DEL SISTEMA

**APLICACIÓN WEB PARA GESTIONAR ACTIVIDADES, CITAS Y MEDICAMENTOS DE
PACIENTES CON CÁNCER DE ESTÓMAGO**

**WILLIAM ALBERTO PATERNINA ROMO
LUIS GUILLERMO SALGADO SALGADO**

ASESOR DISCIPLINAR

EUGENIA ARRIETA RODRÍGUEZ

ASESOR METODOLÓGICO

EUGENIA ARRIETA RODRÍGUEZ

UNIVERSIDAD DEL SINÚ ELIAS BECHARA

ESCUELA DE INGENIERÍA DE SISTEMAS

CARTAGENA DE INDIAS D.T. Y C.

2018

TABLA DE CONTENIDO

OBJETIVOS.....	4
OBJETIVO GENERAL.....	4
OBJETIVOS ESPECIFICOS.....	4
INTRODUCCIÓN.....	5
REQUISITOS TECNICOS MINIMOS DE HARDWARE.....	6
REQUISITOS TECNICOS MINIMOS DE SOFTWARE.....	6
HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO.....	7
INSTALACIÓN DE SERVIDOR WEB LOCAL.....	7
ADJUNTAR BASE DE DATOS.....	¡Error! Marcador no definido.

OBJETIVOS

OBJETIVO GENERAL

Brindar la información necesaria para poder realizar la instalación y configuración del aplicativo.

OBJETIVOS ESPECIFICOS

- ❖ Ilustrar las funcionalidades técnicas y estructurales en el diseño y definición del aplicativo.
- ❖ Definir el proceso de instalación del aplicativo.
- ❖ Detallar los requisitos de las especificaciones de hardware y software necesarios para la instalación de la aplicación.
- ❖ Describir las herramientas utilizadas para el desarrollo y diseño de la aplicación.

INTRODUCCIÓN

Este manual describe los pasos necesarios para la instalación y administración del aplicativo creado para la gestión de actividades, citas y medicamentos de pacientes con cáncer de estómago.

Es importante considerar los requisitos mínimos de hardware y software para la correcta instalación del aplicativo.

REQUISITOS TECNICOS MINIMOS DE HARDWARE

Procesador	Memoria RAM	Disco duro
Core	1 gigabyte (GB)	500 gigabyte (GB)

REQUISITOS TECNICOS MINIMOS DE SOFTWARE

- ❖ Privilegios de administrador.
- ❖ Sistema operativo: Windows Versión: Xp, 7, 8, 8.1, 10 32/64 BIT

HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

- ❖ PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.
- ❖ MySQL Server: MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.
- ❖ Apache Server: El Proyecto Apache HTTP Server es un esfuerzo para desarrollar y mantener un servidor HTTP de código abierto para los sistemas operativos modernos, incluidos UNIX y Windows. El objetivo de este proyecto es proporcionar un servidor seguro, eficiente y extensible que proporcione servicios HTTP en sincronización con los estándares HTTP actuales.

INSTALACIÓN DE SERVIDOR WEB LOCAL

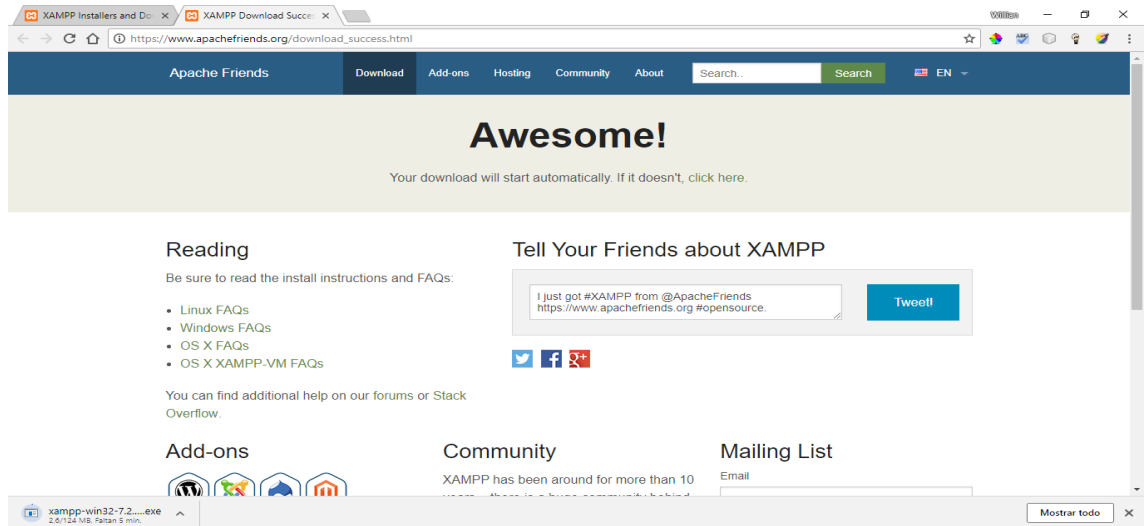
XAMPP: Es una herramienta para “emular” un servidor en nuestro propio ordenador, de forma local, desarrollada por Apache Friends. Lo que hace XAMPP es instalar un servidor Apache con MySQL como gestor de base de datos, PHP como lenguaje de script del lado del servidor y también cuenta con el lenguaje de programación Perl (en este proyecto no es usado). Gracias a esto, es posible desarrollar el proyecto al completo de manera local, sin necesidad de disponer de servidores remotos. Dicho todo esto nos disponemos a instalar XAMPP en Windows.

Descargamos el archivo de instalación de XAMPP en la ruta www.apachefriends.org donde podemos encontrar la versión más reciente y para todos los sistemas operativos, aunque en esta ocasión utilizaremos Windows.

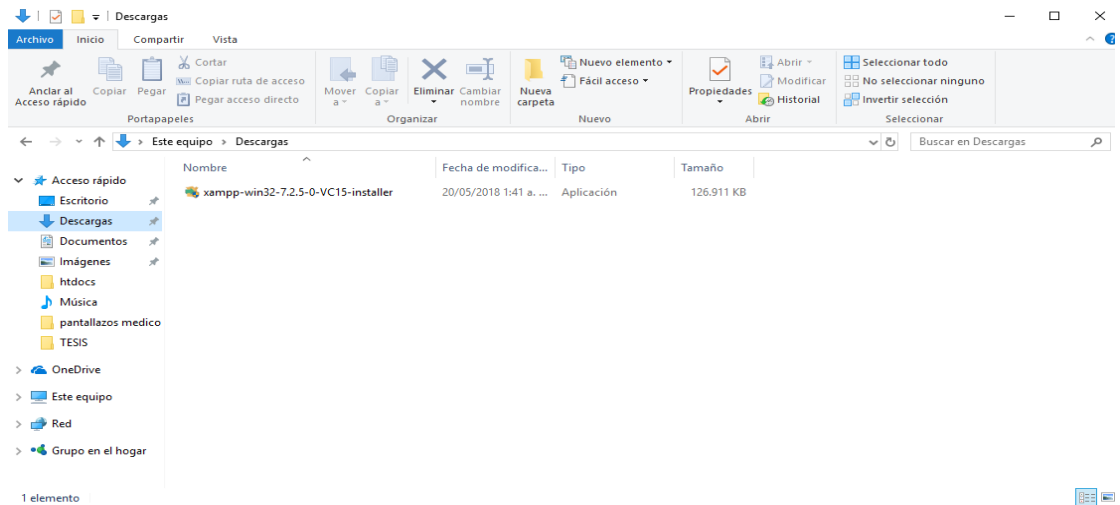


The screenshot shows the Apache Friends website for XAMPP. The page title is "XAMPP Apache + MariaDB + PHP + Perl". Under the heading "What is XAMPP?", it states that XAMPP is the most popular PHP development environment and is a completely free, easy-to-install Apache distribution containing MariaDB, PHP, and Perl. Below this, there are three download buttons: "Download" (with a link to other versions), "XAMPP for Windows 7.2.5 (PHP 7.2.5)", "XAMPP for Linux 7.2.5 (PHP 7.2.5)", and "XAMPP for OS X XAMPP-VM (PHP 7.2.5)".

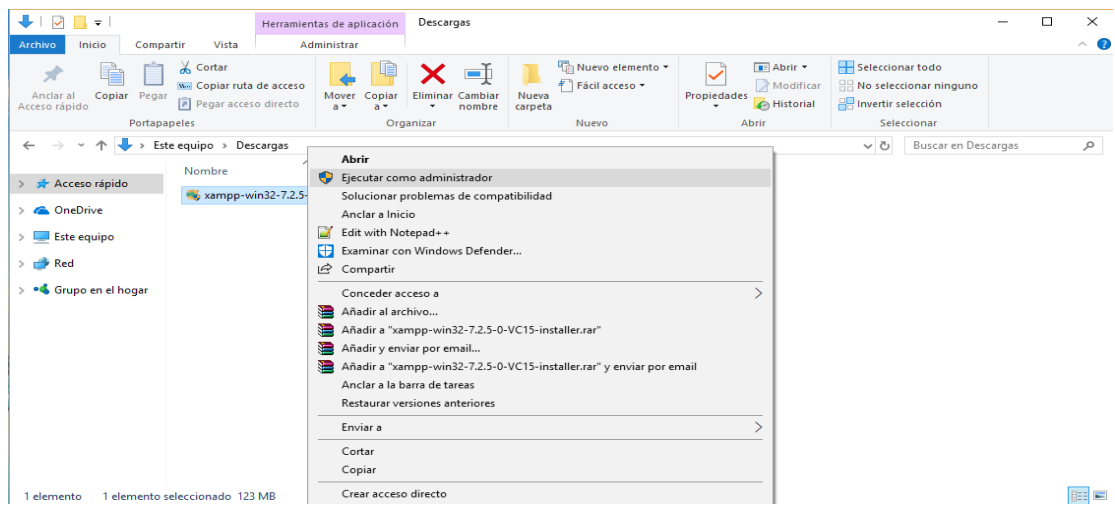
Escogemos alguno de los tres sistemas operativos coincidiendo con el sistema operativo que tenemos instalado en nuestro computador, en mi caso el sistema operativo Windows 10 pro de 64 bit.



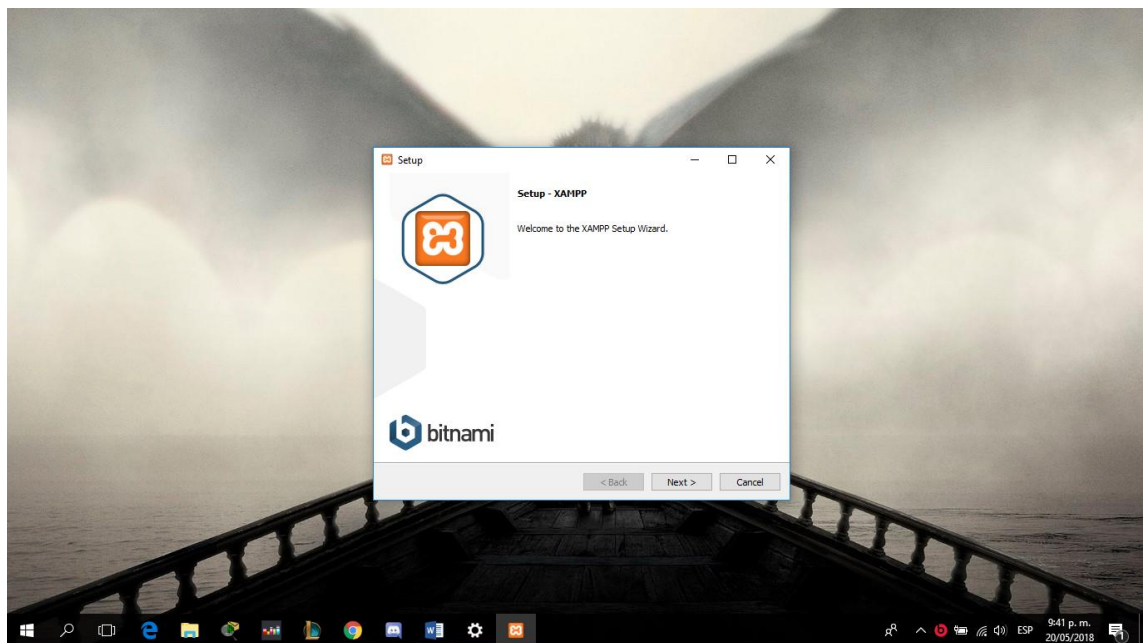
Una vez descargado el archivo de instalación de XAMPP, generalmente lo encontramos en la carpeta de descargas de cualquier versión de Windows.



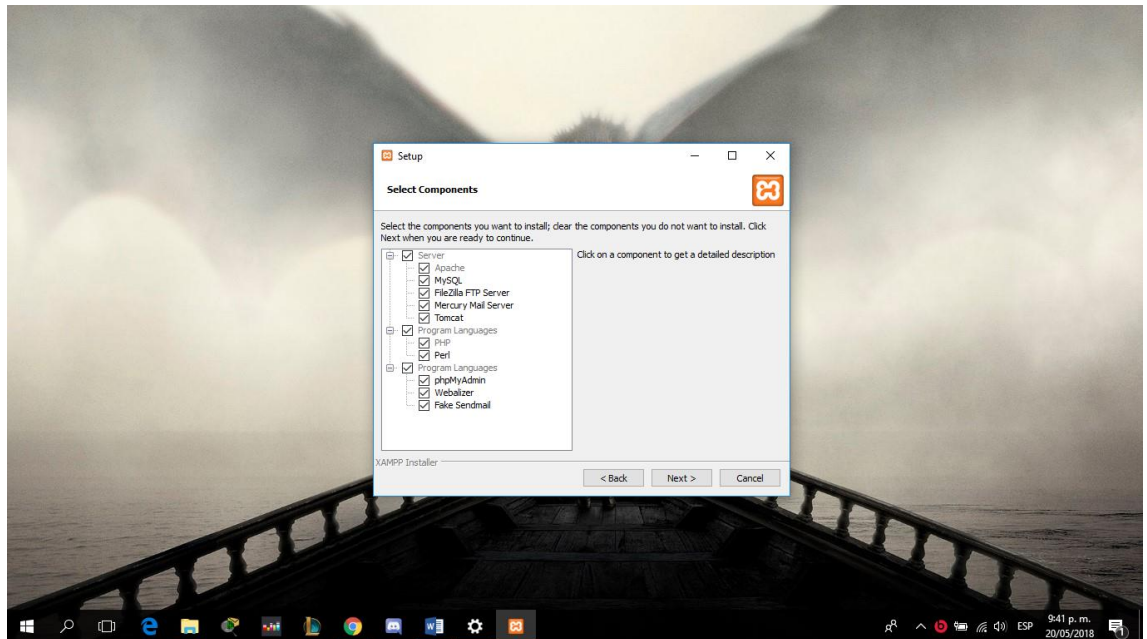
Luego de descargar xampp ejecutamos el instalador como administrador.



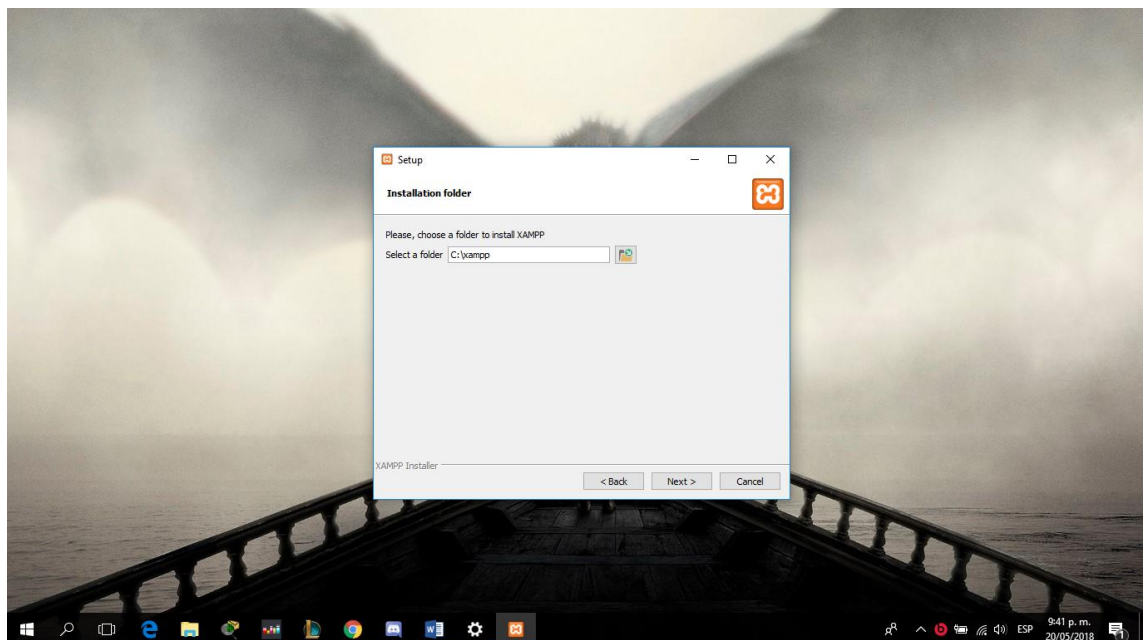
En la imagen se puede observar que se ha iniciado el asistente de instalación en la que procederemos en presionar “Next”.



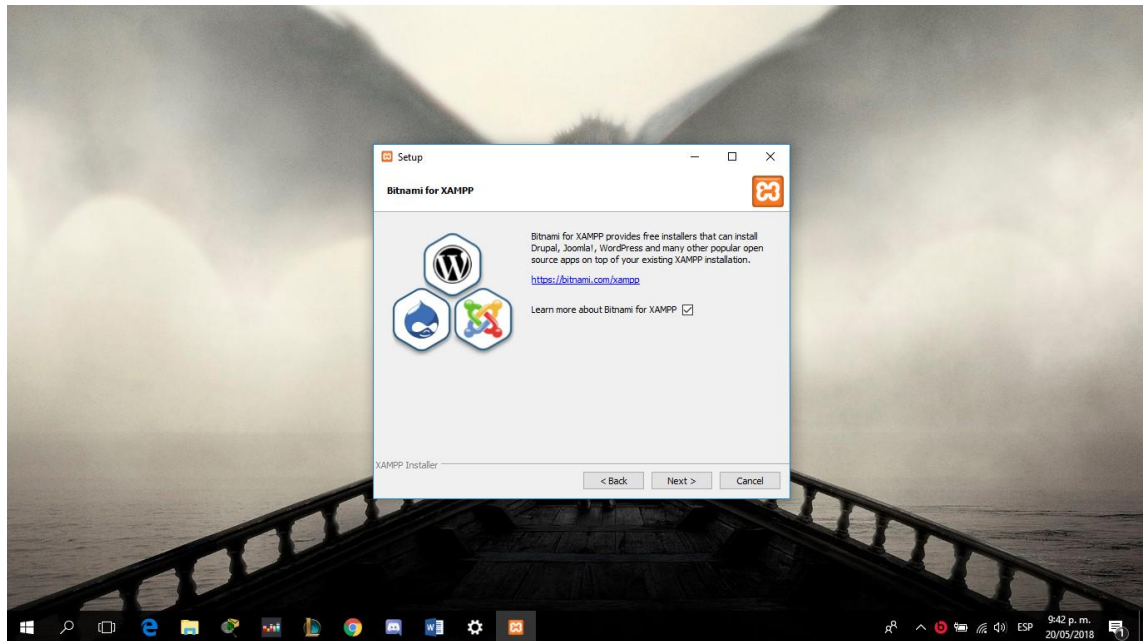
Seguimos con seleccionar cada uno de los elementos o paquetes que desea instalar, en este caso los seleccionamos todos.



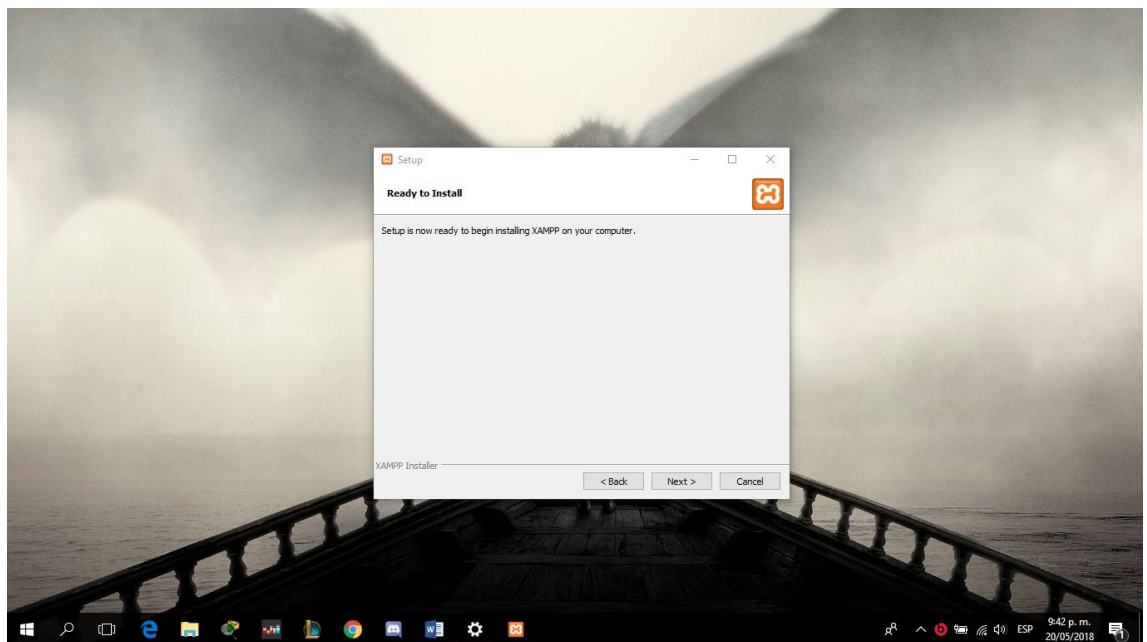
Procedemos en buscar el directorio en el cual va a quedar instalado el programa en este caso "C:\xampp".



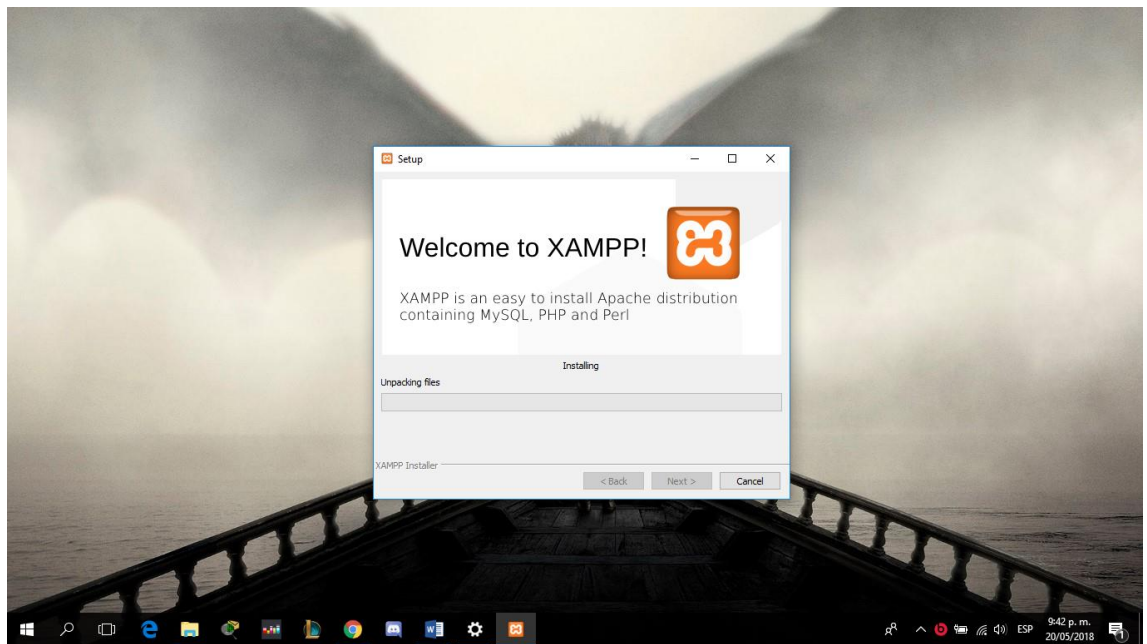
En la anterior pantalla la instalación nos ofrece información sobre los instaladores de aplicaciones para XAMPP creados por Bitnami, presionaremos en el botón “next” para continuar.



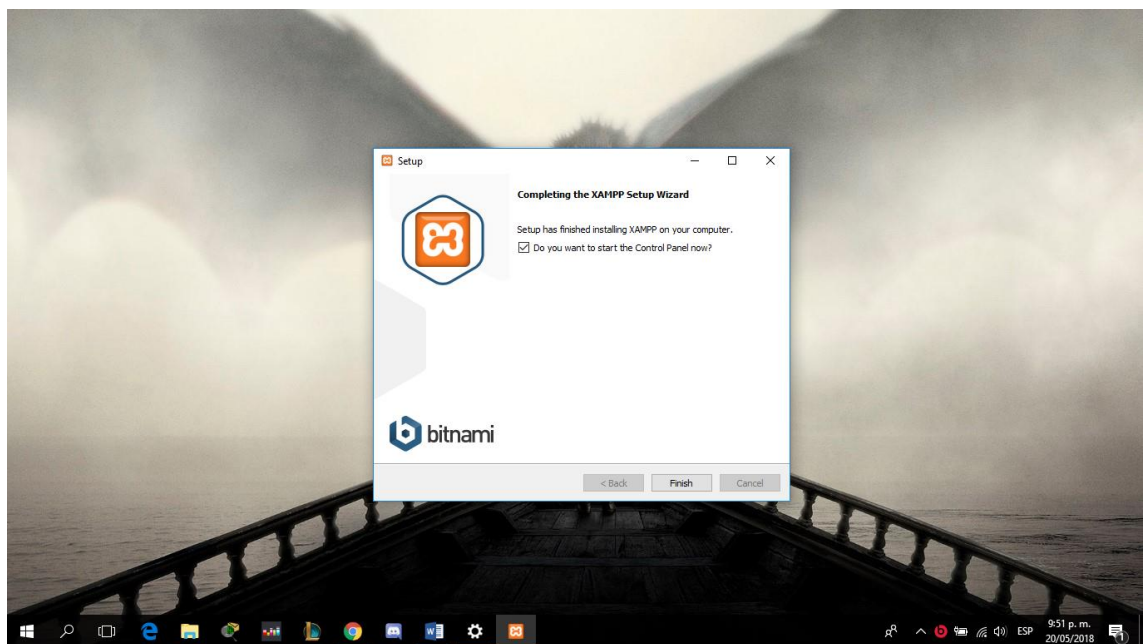
Seguimos en confirmar la instalación dando clic en el botón “next”.



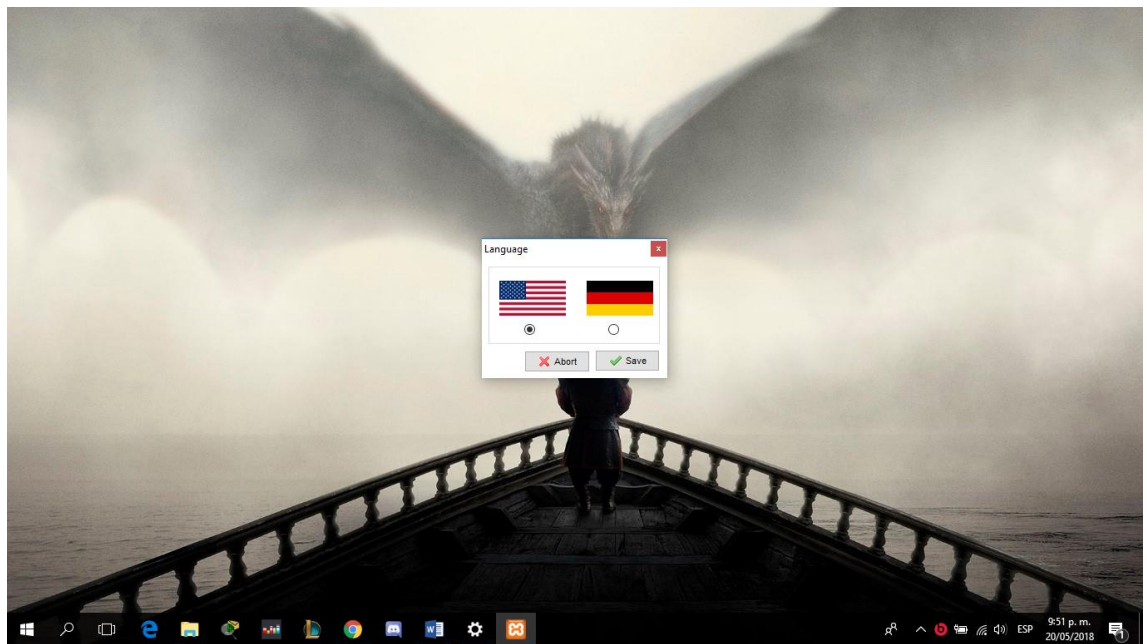
Comienzo de la etapa de instalación.



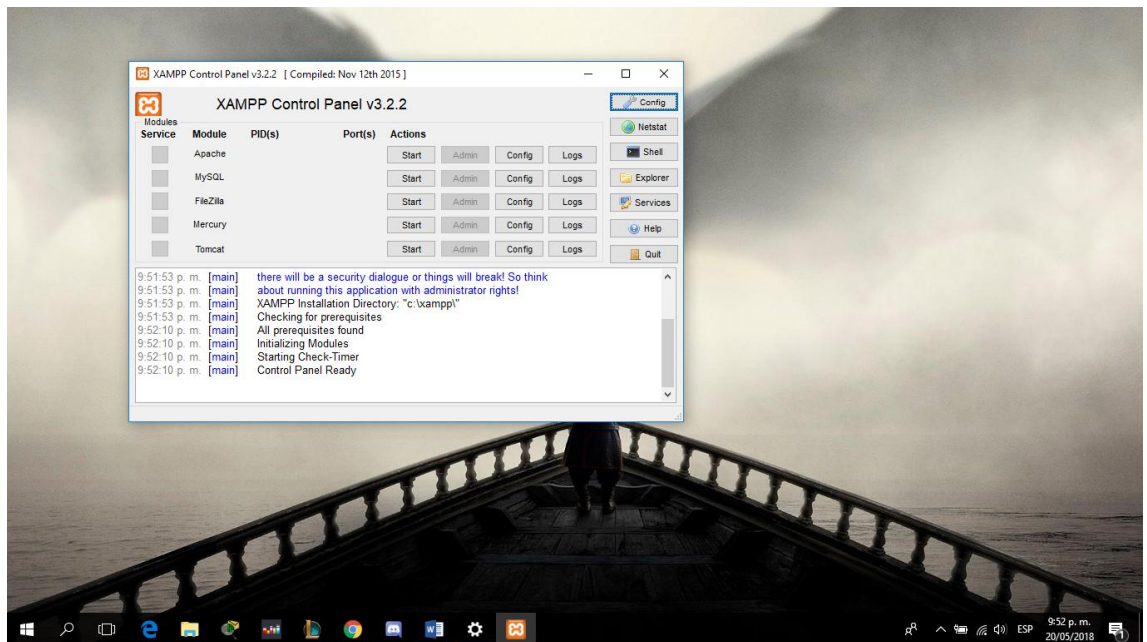
Una vez terminado el proceso de instalación continuamos con presionar en el botón "finish".



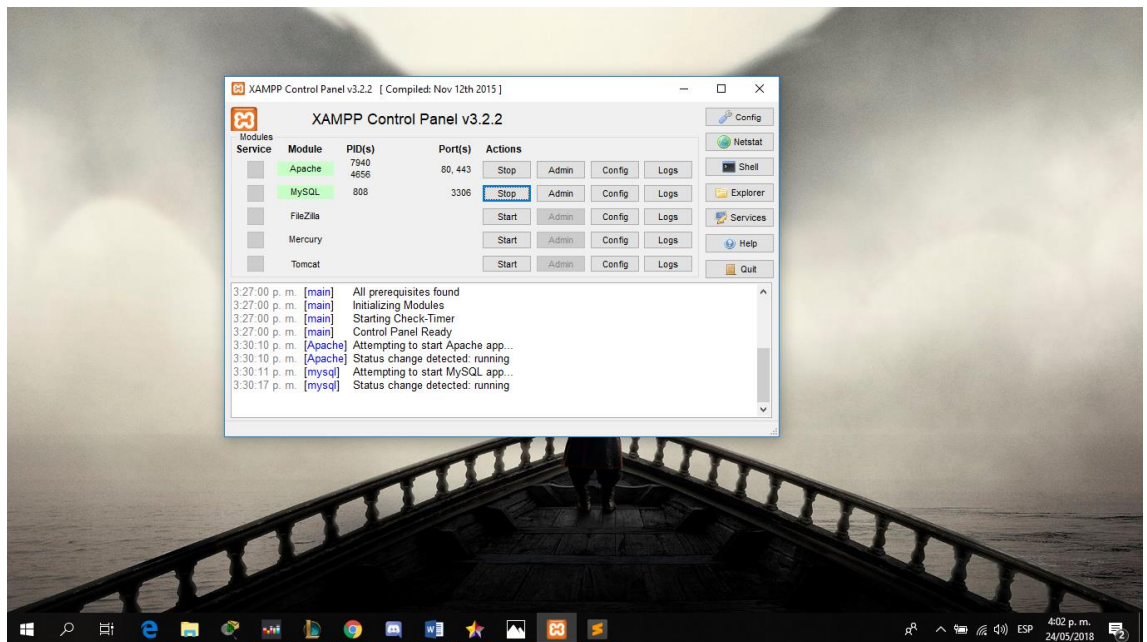
Elegimos nuestro idioma de preferencia y presionamos “save”.



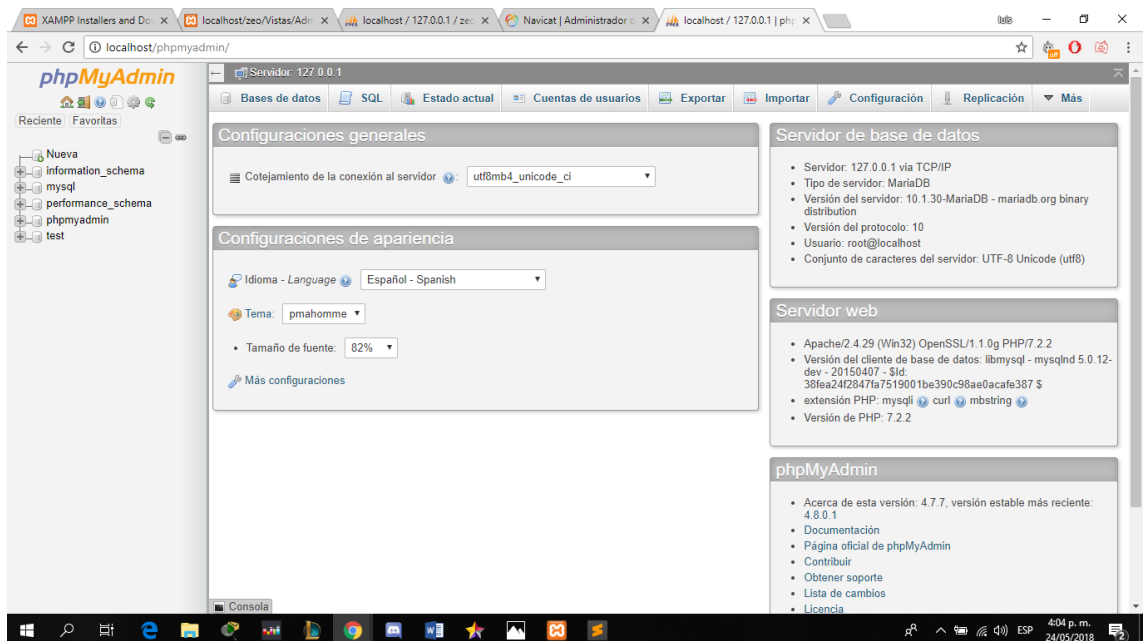
Para la muestra de nuestra aplicación se es necesario iniciar los servicios “Apache” y “MySQL”.



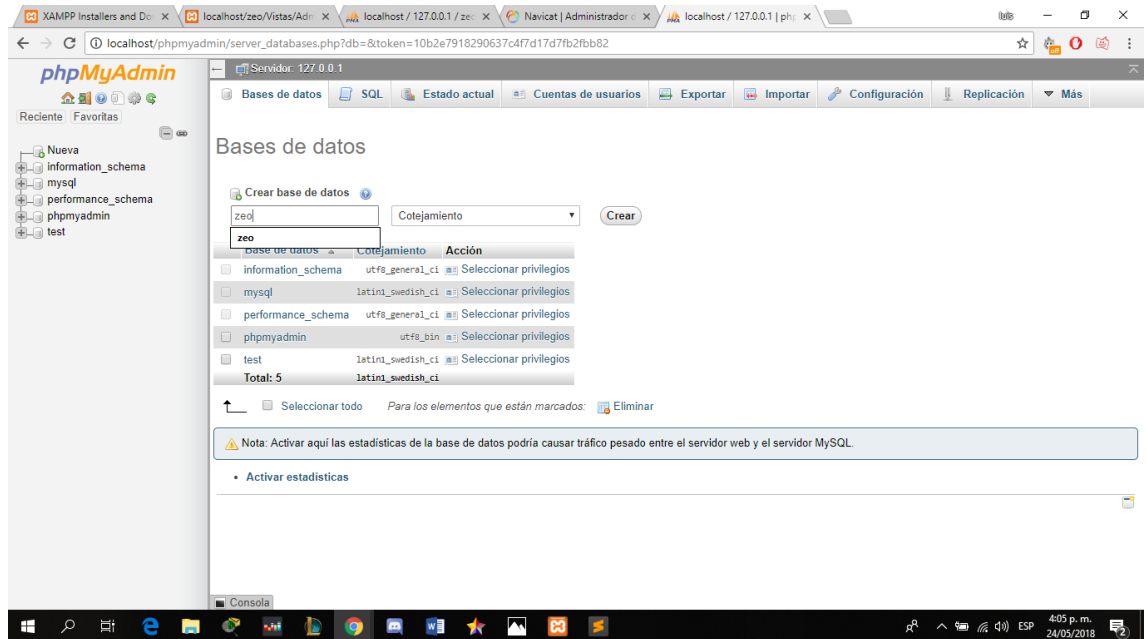
Activar servicios "Apache" y "MySQL"



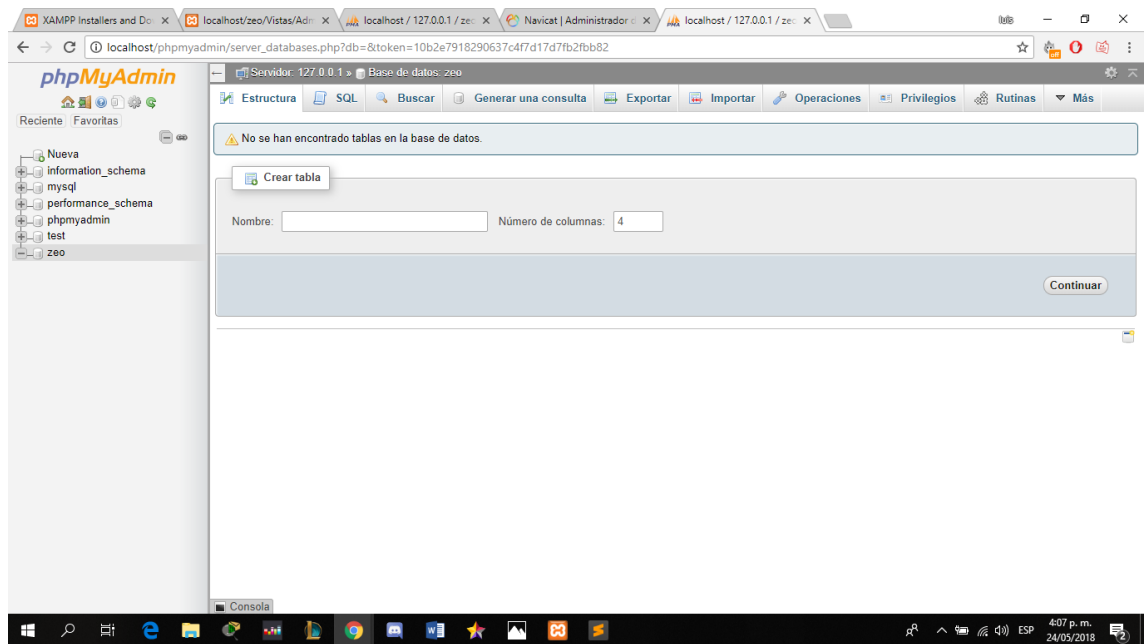
Una vez activo nuestros servicios entramos al link <http://localhost/phpmyadmin/>



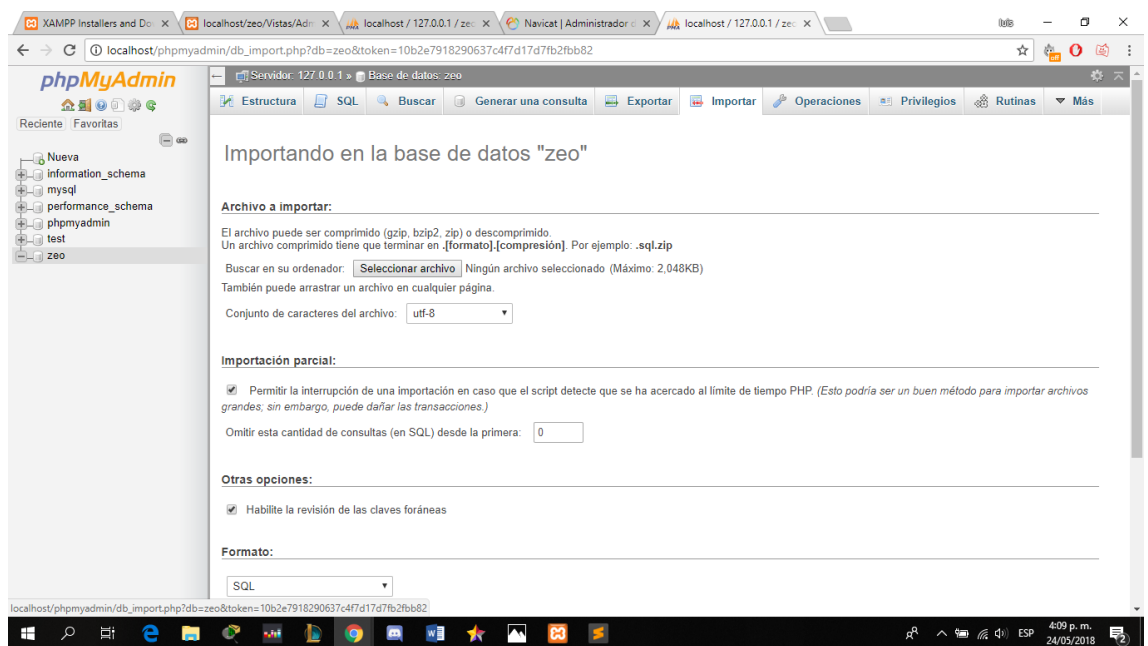
Continuamos con crear nuestra base datos, con el nombre de “zeo” en minúscula, así está configurado en el aplicativo.



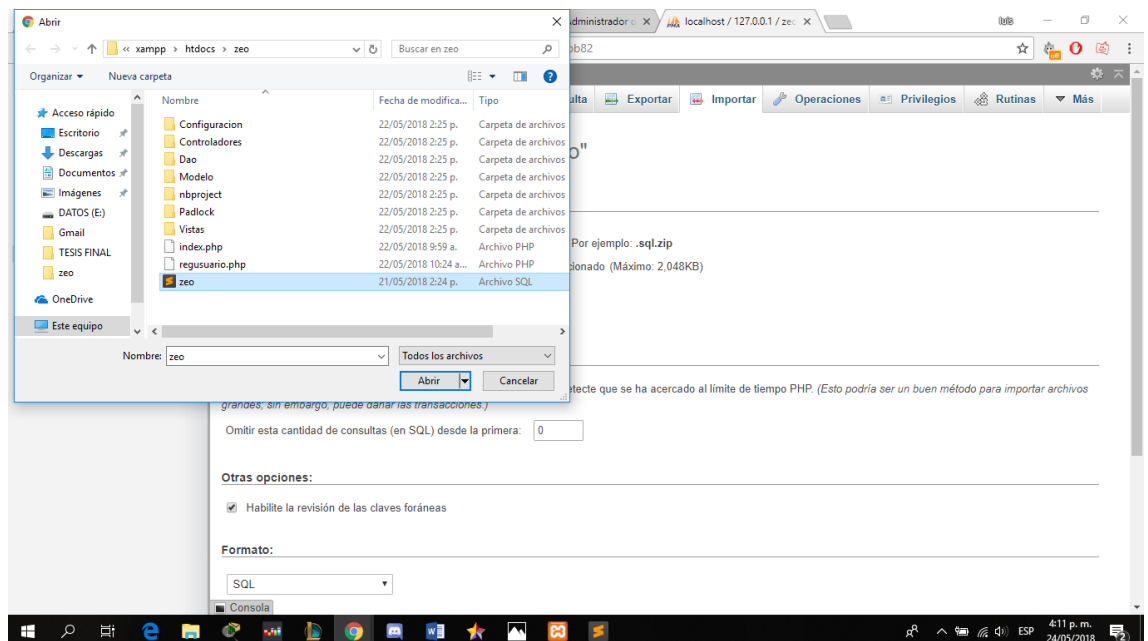
Seleccionamos la base de datos creada con el nombre de zeo.



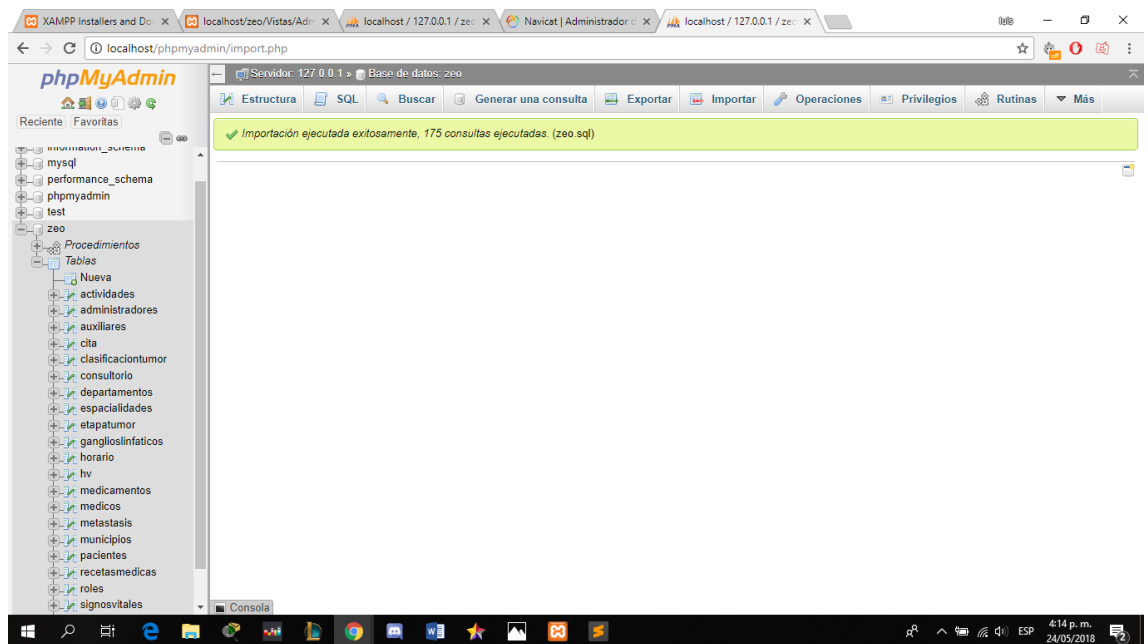
Seguimos con presionar la pestaña donde dice "Importar"



Una vez que estemos en el ítem "importar" procederemos a buscar nuestra base de datos, la cual se encuentra en el proyecto zeo, con el nombre zeo.sql



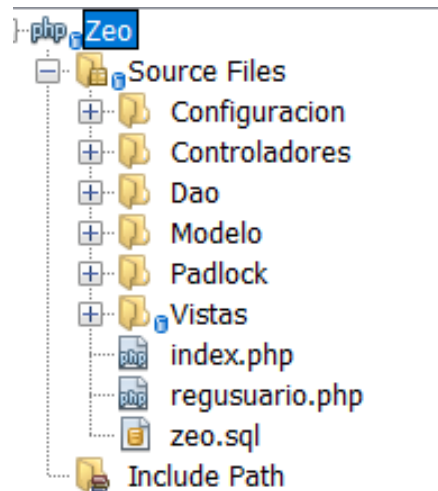
La base de datos zeo.sql se importó de una manera correcta.



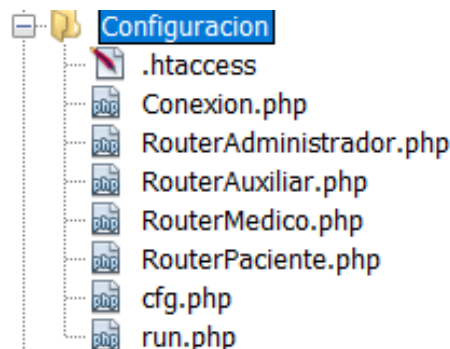
Nuestro proyecto está desarrollado bajo el patrón de diseño modelo, vista y controlados. Las ventajas de utilizar este patrón de diseño para nuestro proyecto son las siguientes:

- ❖ La implementación se realiza de forma modular.
- ❖ Sus vistas muestran información actualizada siempre. El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
- ❖ Cualquier modificación que afecte al dominio, como aumentar métodos o datos contenidos, implica una modificación sólo en el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
- ❖ Las modificaciones a las vistas no afectan al modelo de dominio, simplemente se modifica la representación de la información, no su tratamiento.
- ❖ MVC está demostrando ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

En la siguiente ilustración nos muestra como está estructurado el proyecto. Tenemos 6 capas de desarrollo, pero el patrón de diseño es modelo, vista y controlador. Las 3 capas restante solo complementan las capas de modelo, vistas y controladores.



Comenzaremos con la capa de configuración. En esta capa es donde configuramos las conexiones con la base de datos y también donde declaramos las políticas de seguridad y configuraciones de roles por vistas.



Una de las políticas de seguridad es ejecutar el archivo .htpasswd si se quiere ejecutar las conexiones con la base de datos.

```
.htaccess x
Source History
1 AuthUserFile /xampp/htdocs/Zeo/Padlock/.htpasswd
2 AuthName Development
3 AuthType Basic
4
5 require user wpaternina
```

```
.htpasswd x
Source History
1 wpaternina:$apr1$Yb46bTs2$vT0Gx8SZSh5s3pt4eIZZ/0
```

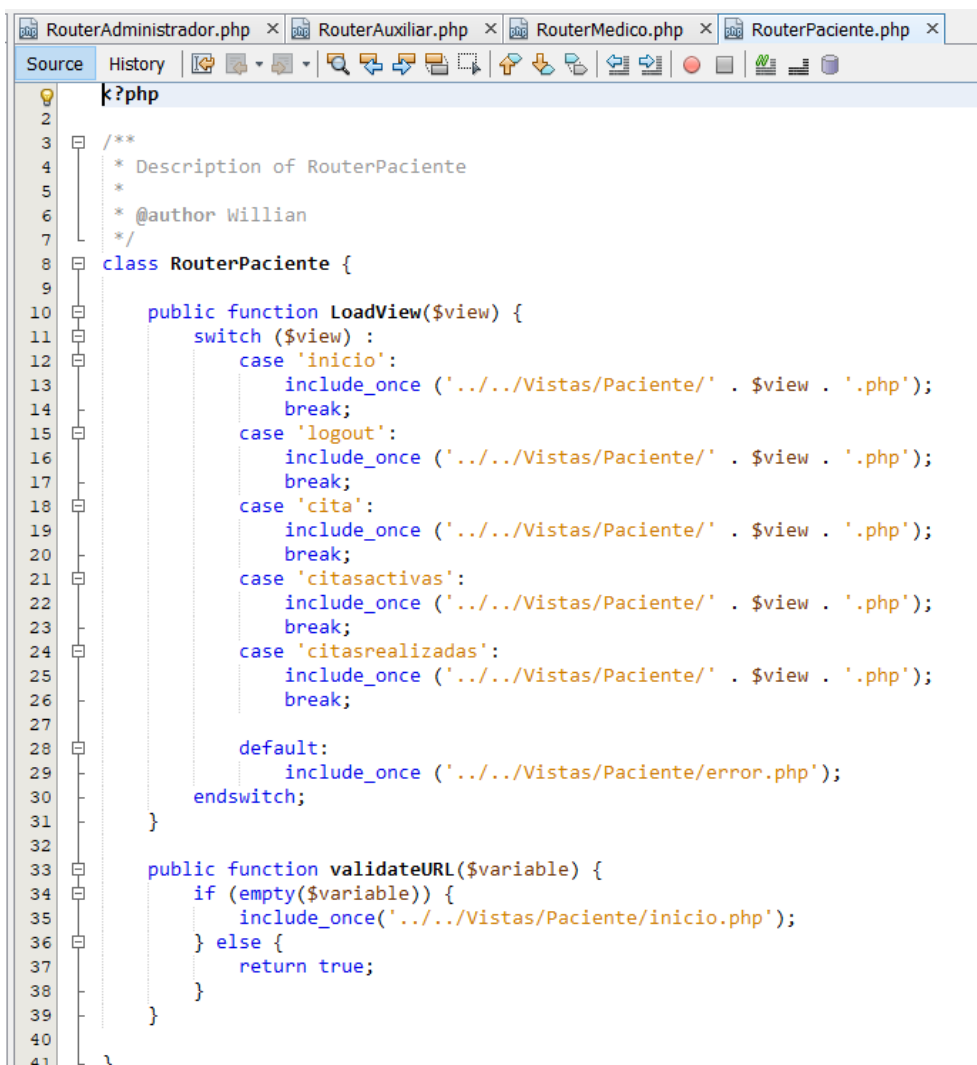
Tenemos una clase Conexión donde declaramos una función donde nos conectamos por PDO con la base de datos zeo.

```
Conexion.php x
Source History
1 <?php
2
3 /**
4  * Esta clase me conecta con la base de datos
5  *
6  * @package Configuracion
7  * @author Willian Paternina <willian.paternina@unisinu.edu.co>
8  * @copyright 2017-2018 Proyecto de grado para optar el titulo de ingeniero de sistemas
9  * @version 1.0
10 */
11 include_once 'cfg.php';
12 class Conexion extends PDO {
13
14     protected $cnn;
15
16     public function ConexionMySQLServer() {
17         try {
18             $this->cnn = new PDO(SERVER, USERNAME, PASSWORD);
19             $this->cnn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
20         } catch (PDOException $e) {
21             echo $e->getMessage();
22         }
23     }
24 }
25
26 ?>
27
```

Tenemos un archivo llamado cfg.php donde declaramos los parámetros de conexión.

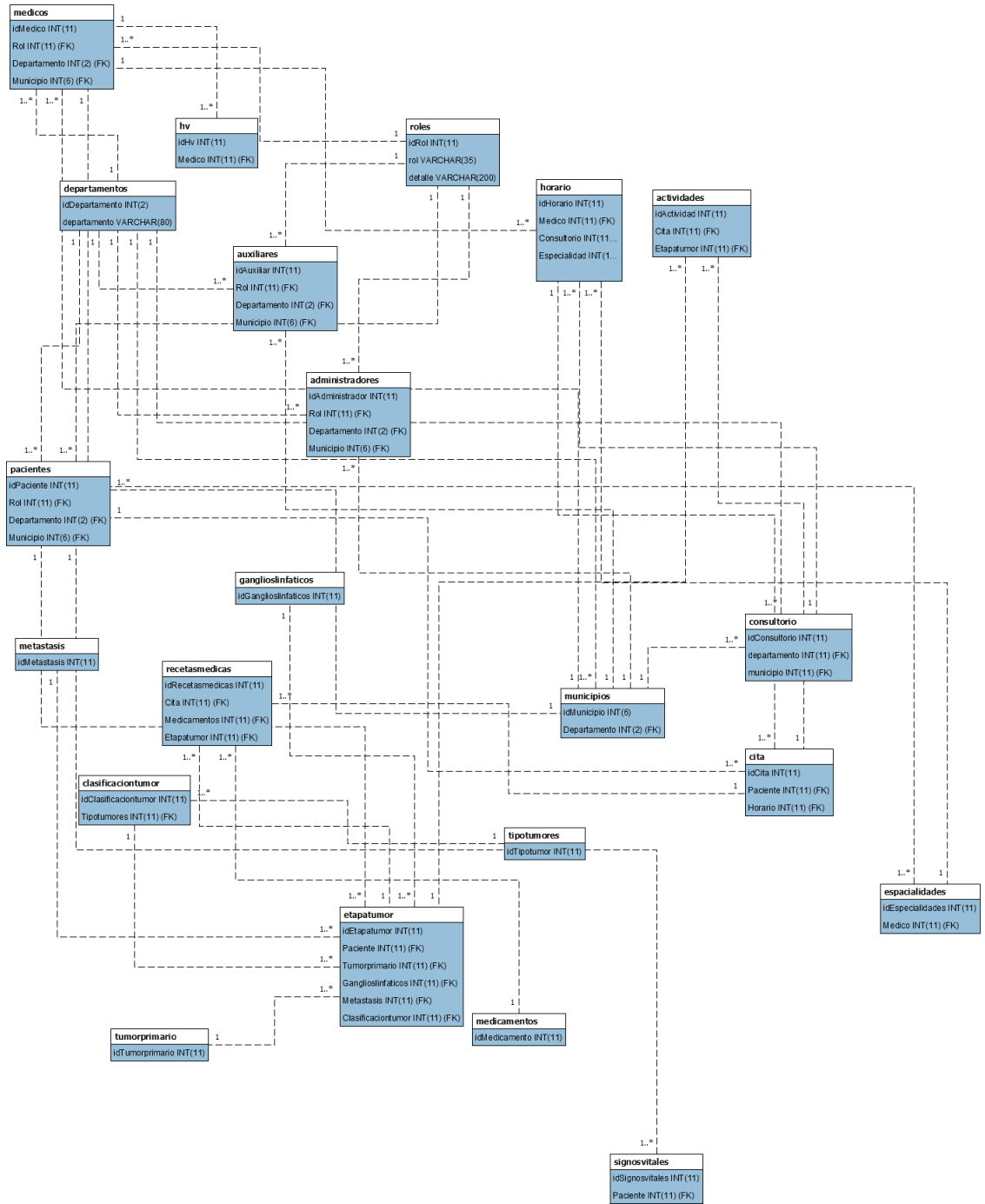
```
Conexion.php x cfg.php x
Source History
1 <?php
2 /**
3  * Este archivo contiene los parametros de conexion con la base de datos
4  *
5  * @package Configuracion
6  * @author Willian Paternina <willian.paternina@unisinu.edu.co>
7  * @copyright 2017-2018 Proyecto de grado para optar el titulo de ingeniero de sistemas
8  * @version 1.0
9  */
10 define("SERVER", "mysql:dbname=zeo;host=localhost");
11 define("USERNAME", "root");
12 define("PASSWORD", "");
13
14 ?>
```

Tenemos los archivos RouterAdministrador, RouterAuxiliar, RouterMedico y RouterPaciente, que básicamente son lo mismo en cuanto estructura, estos archivos nos permiten manejar roles y configurar las vistas de nuestro proyecto.

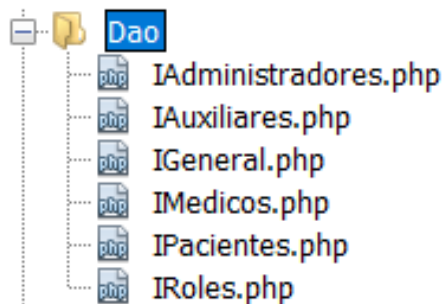


```
1 k?php
2
3 /**
4  * Description of RouterPaciente
5  *
6  * @author Willian
7  */
8 class RouterPaciente {
9
10     public function LoadView($view) {
11         switch ($view) :
12             case 'inicio':
13                 include_once ('../../Vistas/Paciente/' . $view . '.php');
14                 break;
15             case 'logout':
16                 include_once ('../../Vistas/Paciente/' . $view . '.php');
17                 break;
18             case 'cita':
19                 include_once ('../../Vistas/Paciente/' . $view . '.php');
20                 break;
21             case 'citasactivas':
22                 include_once ('../../Vistas/Paciente/' . $view . '.php');
23                 break;
24             case 'citasrealizadas':
25                 include_once ('../../Vistas/Paciente/' . $view . '.php');
26                 break;
27
28             default:
29                 include_once ('../../Vistas/Paciente/error.php');
30         endwhile;
31     }
32
33     public function validateURL($variable) {
34         if (empty($variable)) {
35             include_once ('../../Vistas/Paciente/inicio.php');
36         } else {
37             return true;
38         }
39     }
40
41 }
```

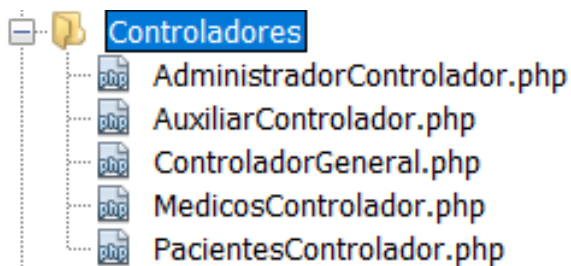

Tenemos un diagrama relacional donde nos centramos mas en entender las diferentes relaciones que hay entre tablas.



La capa dao es la capa que nos permite declarar las funciones para acceder a los datos de nuestra base de datos y relacionarlos con nuestro modelo de negocio por medio de objetos.



La capa de controladores accede a las funciones de la capa dao e incluye dichas funciones abstractamente que posteriormente serán utilizadas en las diferentes vistas para gestionar las diferentes operaciones entre la base de datos y el aplicativo.



Finalmente tenemos la capa de vistas donde por cada rol tenemos una carpeta.

