3. Modelo de datos

En este apartado se van a incorporar al proyecto las clases del modelo de datos. Se crearán los objetos de acceso a datos para los mensajes y contactos de un usuario. Realizaremos cinco tareas.

La primera tarea a realizar es crear la base de datos que va a contener la información de contactos y mensajes. Para ello ejecutamos la aplicación WampServer. En la Figura 3.4 podemos ver el panel de control. Debemos asegurarnos que están iniciados todos los servicios (icono en verde).

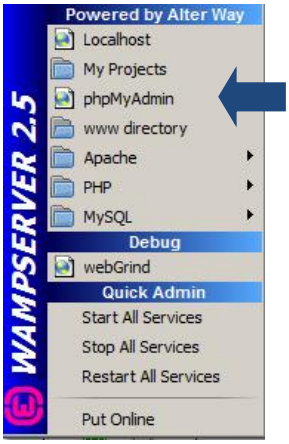

Figura 3.4. Panel de control de WampServer. Inicio de los servicios.

Al iniciar WampServer se pone en marcha un servidor HTTP Apache y un gestor de bases de datos ***MySQL***. Si pulsamos sobre phpMyAdmin en el panel de control (Figura 3.4), se abre el navegador y aparece una interfaz como la mostrada en la Figura 3.5. Para crear la base de datos, seleccionamos la pestaña “Bases de datos” y en “Crear base de datos” introducimos el nombre de nuestra base de datos, que es ***whatsat2019***.

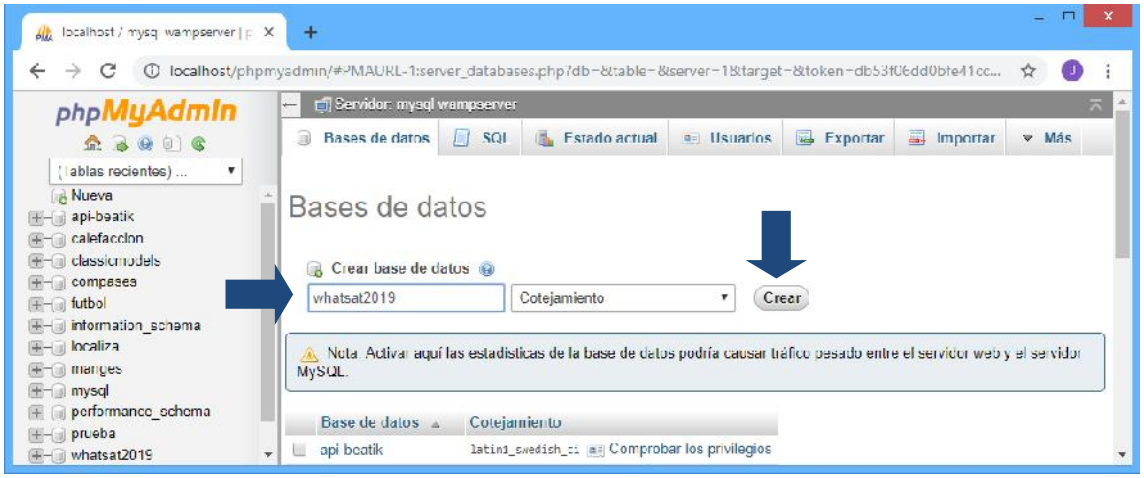

Figura 3.5. Crear una base de datos con phpMyAdmin.

La segunda tarea será importar la estructura y datos a la base de datos que hemos creado. Para ello descargaremos el archivo ***whatsat2019.sql*** y utilizando la interfaz phpMyAdmin (Figura 3.6), seleccionamos la base de datos ***whatsat2019*** (parte izquierda). A través de unos pasos recogidos en las figuras 3.7, 3.8, y 3.9, llevamos a cabo la importación.

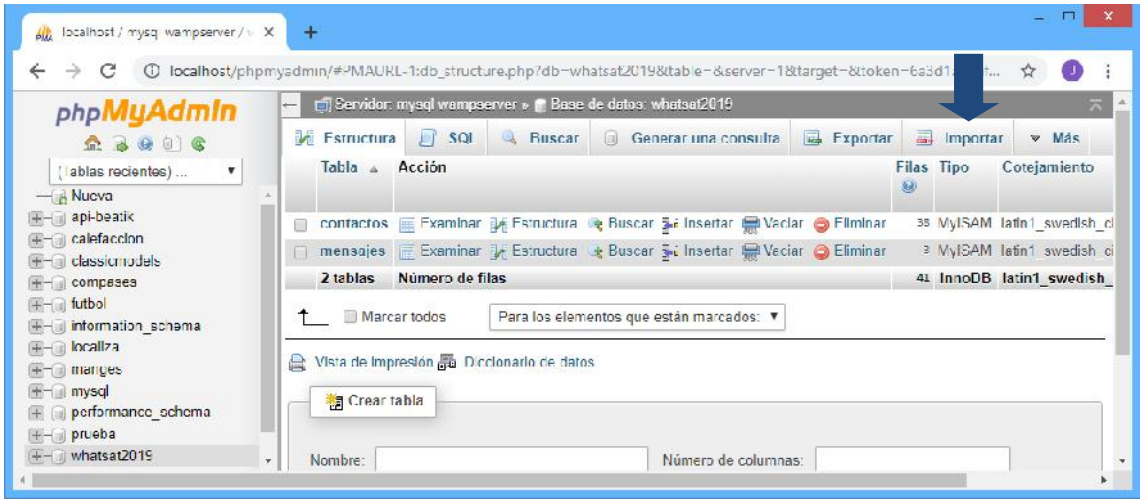

Figura 3.6. Importar la estructura y datos de una base con phpMyAdmin.

Figura 3.7. Elección del archivo SQL a importar.

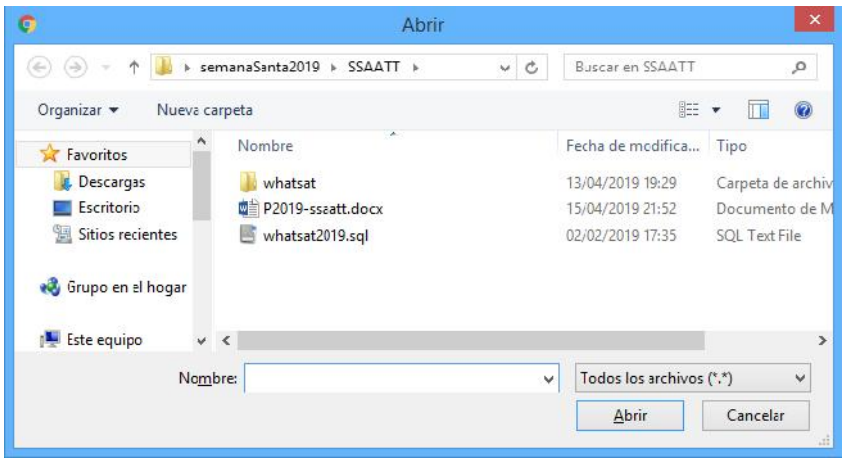

Figura 3.8. Búsqueda del archivo SQL.

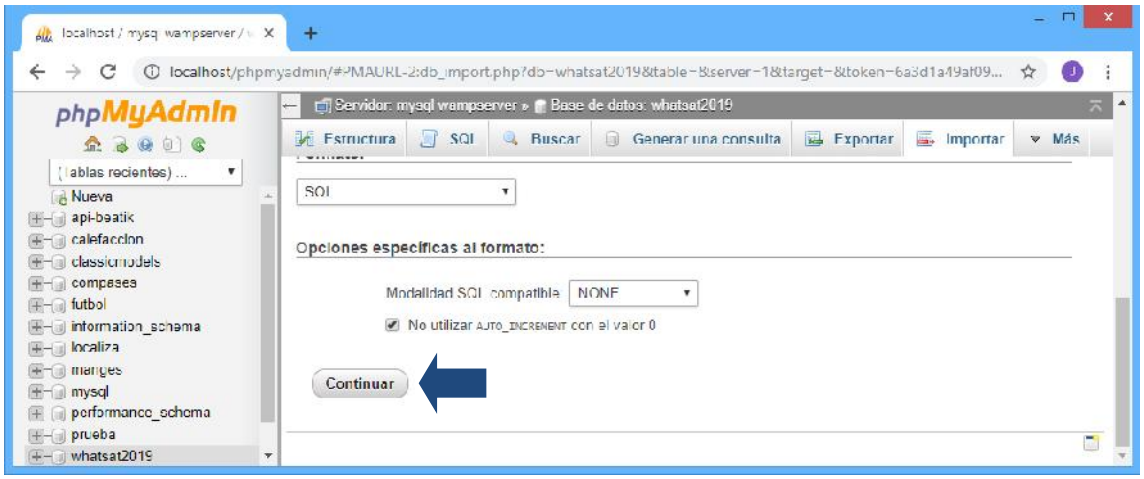

Figura 3.9. Confirmación del archivo SQL.

Al finalizar los pasos anteriormente descritos, dentro de la base de datos aparecen las siguientes entidades (Figura 3.10):

- *Mensajes*. Contiene la información de los mensajes intercambiados entre los distintos usuarios. Sus atributos aparecen recogidos en la Figura 3.11 y son: origen, destino, tipo, contenido y leído. Este último se utiliza para leer una sola vez un mensaje recibido. Cuando está pendiente de ser leído su valor es 0. Cuando se lee, pasa a valor 1.
- *Contactos*. Contiene la información de los usuarios que existen en el servicio de intercambio de mensajes. Sus atributos aparecen en la Figura 3.12: id (identificador de usuario), Surname (apellidos), name (nombre) y profileImageUrl (enlace a la foto de perfil).

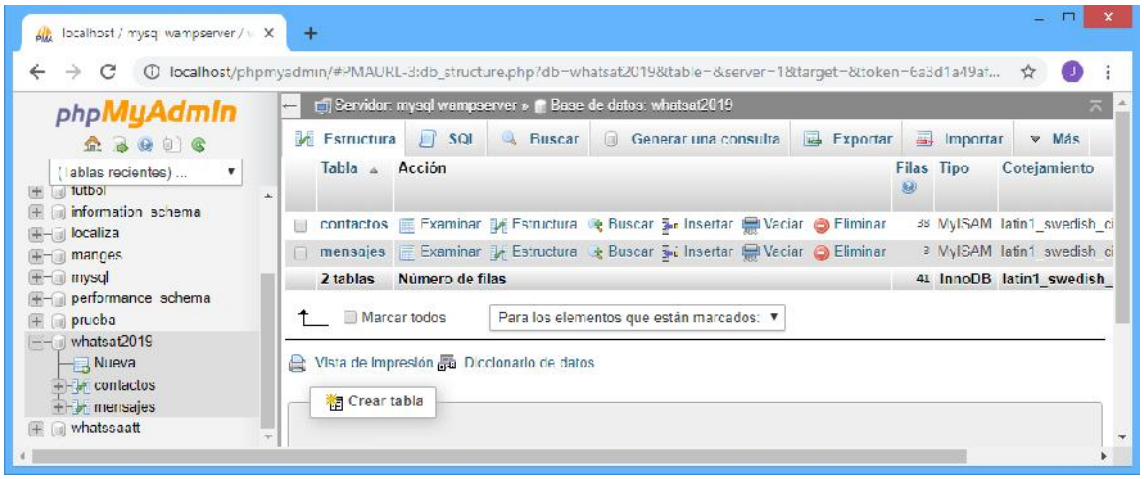

Figura 3.10. Entidades y datos importados.

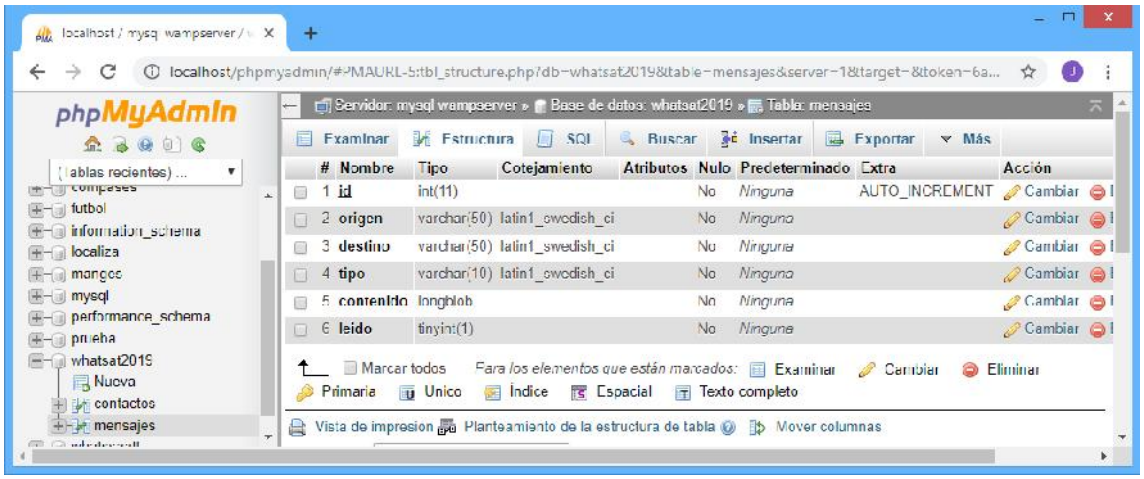

Figura 3.11. Atributos de la entidad mensajes.

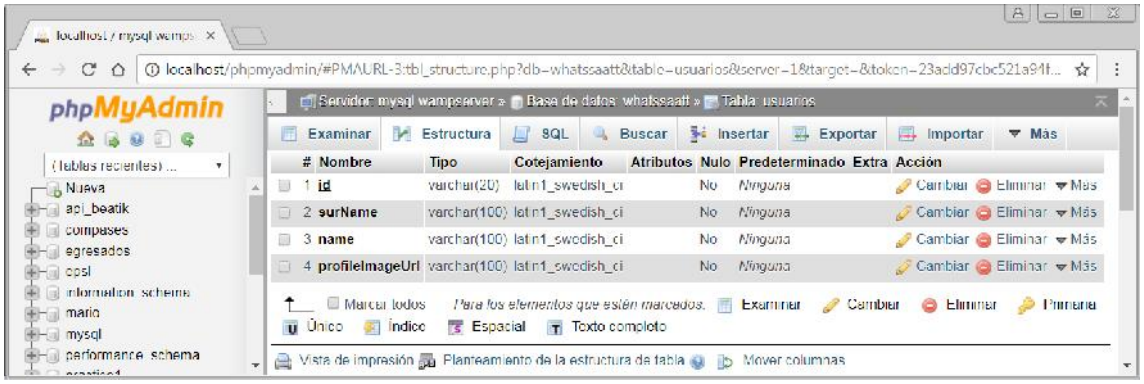

Figura 3.12. Atributos de la entidad contactos.

La tercera tarea consiste en la copia, dentro de la aplicación, de las siguientes clases del modelo de datos:

- ***model.js***. Contiene las clases ***DaoContactos*** y ***DaoMensajes***. En ellas existen los métodos para obtener los contactos o mensajes y el método para crear un nuevo contacto. Contiene los métodos CRUD.
- ***Db.js***. Es la clase que contiene los métodos para manipular los datos del origen de datos. En este caso, el origen es una base de datos gestionada por un gestor MySQL.

Estas clases se pueden incluir donde el alumno estime, pero se propone copiarlas en el directorio raíz, tal y como muestra la Figura 3.12.

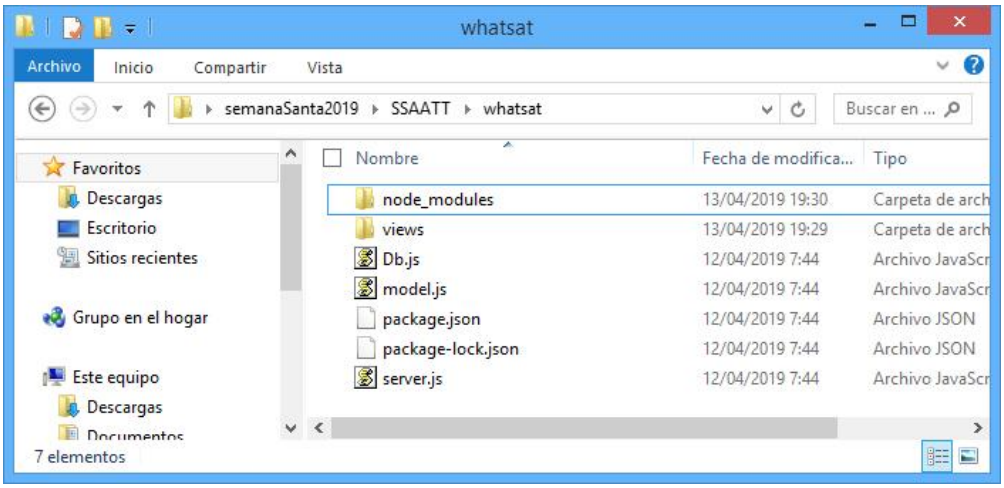

Figura 3.12. Clases del modelo dentro de la aplicación web.

Como cuarta tarea, se completarán las clases existentes en ***model.js*** para crear y obtener (leer) los mensajes de usuario. Con estas ya tenemos todas las clases del modelo de datos.

Finalmente, se deben realizar las modificaciones en el controlador (a partir de las clases del modelo) para manipular la información del origen de datos (base de datos). Por ejemplo, si queremos pasar a la vista la lista de contactos que posee un usuario, debemos incluir las siguientes líneas en nuestro controlador:

```
var model=require('./model.js');
router.get("/contactos/:id", function(req, res) {
    var dao=new model.DaoContactos();
    dao.read(req.params.id).then(dtoContactos => {
      res.setHeader('Access-Control-Allow-Origin','*');
      res.setHeader('Content-Type', 'application/json');
      res.render("contactos",{"listaContactos":dtoContactos});
    });
});
```