**Web**

**ForoUpna**

Jhonny F., Chicaiza Palomo

Proyecto de IWE

**Indice**

**Presentación y objetivos**

Este proyecto consiste en la creación de una página web del tipo Foro que sirva para la interactuación entre los estudiantes de la UPNA. En esta página se podrá registrar y subir una foto para su perfil, cada usuario podrá crear un foro y podrá interactuar con los demás foros creado, habrá un buscador para facilitar la búsqueda de los foros por el título. En la creación de un foro de podrá subir una foto para que acompañe a este.

Los objetivos principales para desarrollar son:

* Creación de usuario
* Modificación de los datos de un usuario
* Creación de foro
* Buscador de foros
* Creación de comentarios dentro de un foro
* Fácilmente usable para las diferentes plataformas web, móvil.
* Se implementarán reglas que permitan una mejor a accesibilidad ya se desde el diseño y desde el código.

**Requisitos y prototipos**

En primer lugar, se deja el link del wireframe realizado para el diseño de la página web: [lnik: wireframe](https://www.figma.com/proto/MK9BGlXYbzMnQzO0BpFAKn/Lo-fi-Wireframe-Kit-(Community)?node-id=102%3A2&scaling=min-zoom&page-id=101%3A5639)

1. Barra de Navegación:

Esta constara con links de acceso a los diferentes sitios de la pagina web, y se adaptara según si el usuario esté logueado o no. Estará fija en todas las páginas de la web.

1. Sistema de login y registro:

Para que el usuario se registre es necesario un email, contraseña y un nombre de usuario que será único. Para el login se podrá realizar con el nombre de usuario y la contraseña.

Para que el sistema se mas seguro se ha limitado el tiempo de la sesión si el usuario esta inactivo a 500 segundos.

1. Página principal:

Consta de dos partes, la parte superior donde aparecerán los foros más comentados y la parte inferior donde aparecerán los últimos comentarios que se han hecho. Desde las tarjetas del foros mas comentados o comentarios el usuario podrá acceder a este.

1. Perfil:

El usuario podrá visualizar sus datos de perfil aquí junto a su foto. Además, podrá modificarlos si este lo desea.

1. Buscador:

El usuario podrá buscar el foro por el título, se listarán y se podrá acceder a estos con un click, se realizará una paginación si no caben en la página, para cargar las paginas se utiliza Ajax..

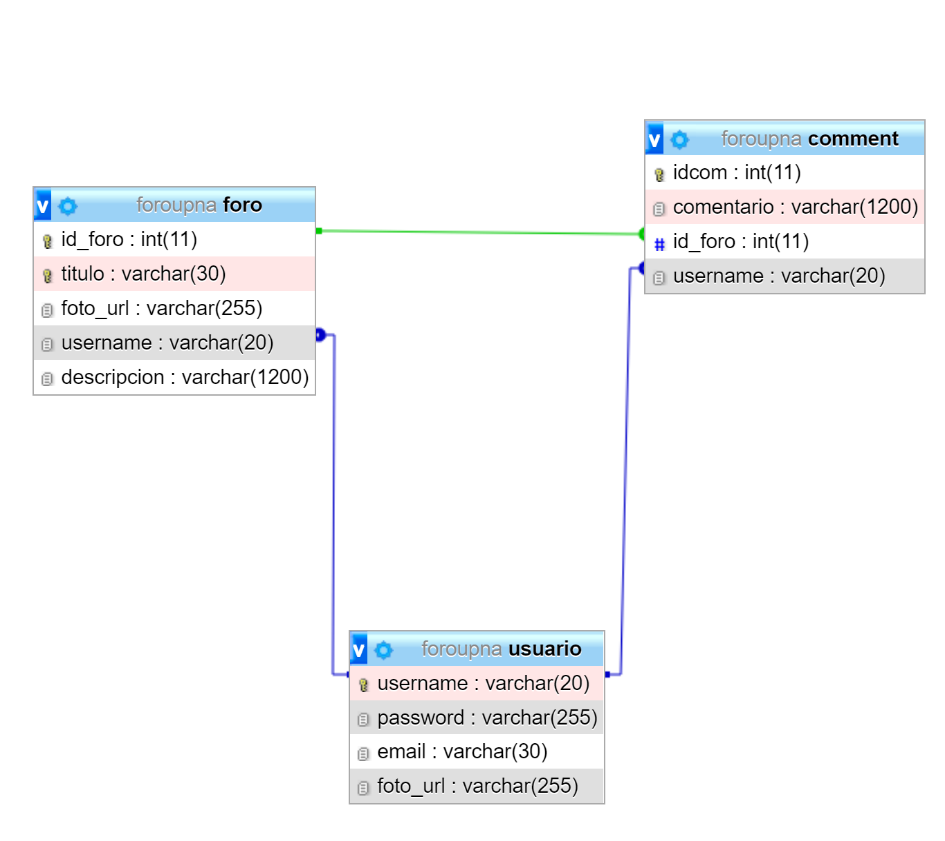
1. Foro:

La parte principal de la pagina web. En primer lugar, se visualizará el nombre del foro y la foto, si no subió foto se pondrá una por defecto. Después aparecerá el comentario que puso el creador del foro, posteriormente aparecer un textarea donde el usuario logueado podrá escribir su comentario y añadirlo al foro. Finalmente se listaran los comentarios del foro, si son demasiados se realizara la paginación de estos, se podrán cargar los distintos comentarios mediante llamadas Ajax.

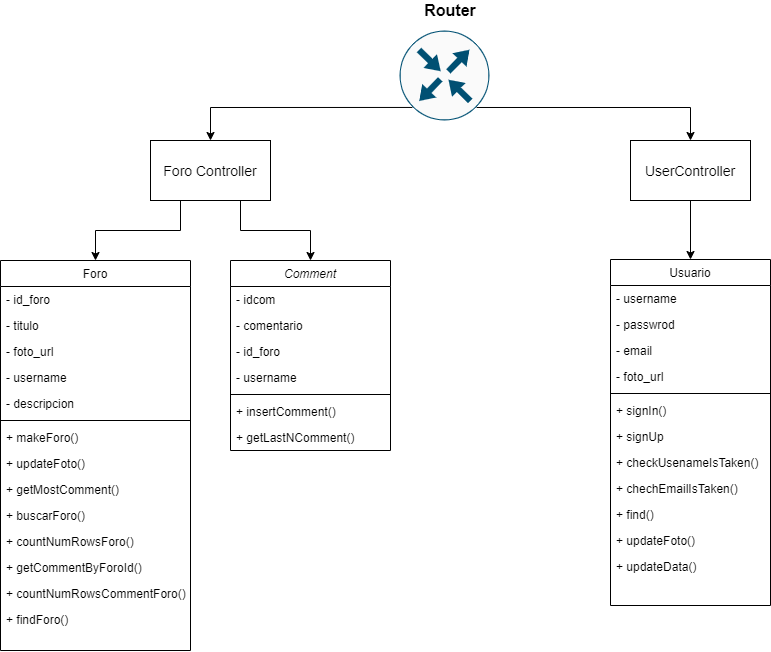
Todas estas funcionalidades están implantadas y correctamente funcionales.

**Análisis y diseño**

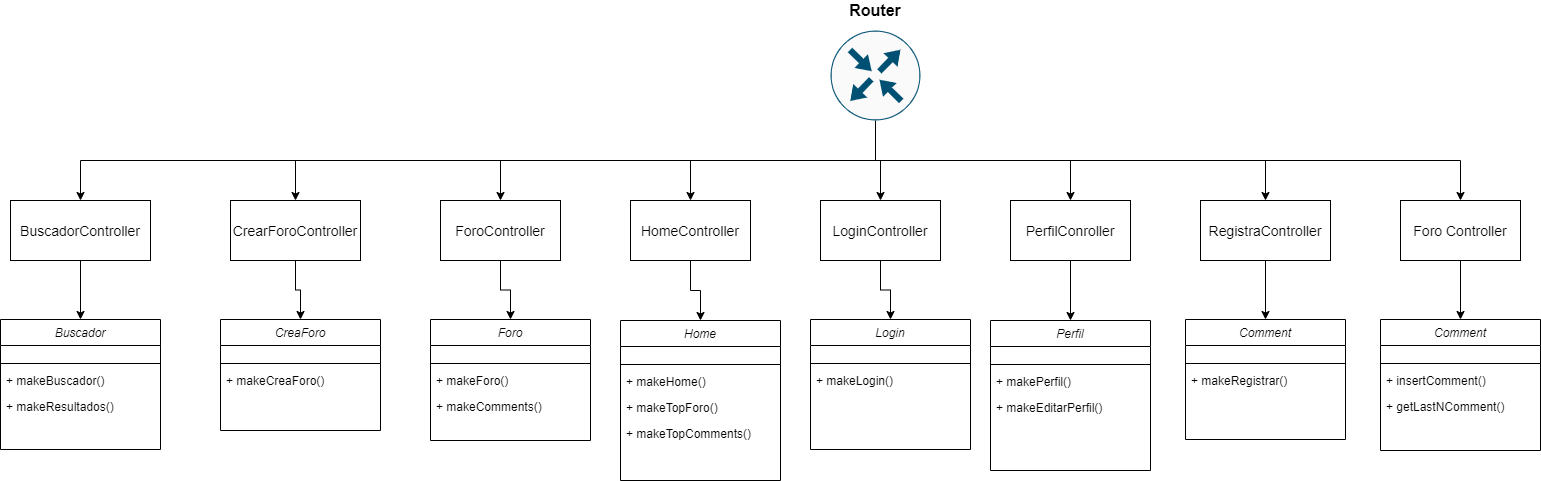
La base de datos se ha realizado con MySql, es decir se ha utilizado una base de datos relacional, la estructura de esta es la siguiente:



Para el control de esta se ha realizado un api que controla las llamadas al backend, con la siguiente estructura:



Con lo que respecta al frontend se ha utilizado la modelo vista controlador MVC facilitar la construcción de las vistas, y una mayor estabilidad. Con el mismo diseño de router, controlador y modelo que el api del backend:



Como en el frontend necesita realzar llamadas al api se ha realizado en este un controlador de las llamadas al api, para que sea mas seguro. **BackendConxController.php.**

Para la paginación se ha realizado un modelo común que es capaz de realizar la paginación tanto como para el buscador o como para los comentarios del foro.

Para la carga dinámica de datos en el buscador se ha utilizado JavaScript con Ajax y para el estilo se ha utilizado css.

**Implementación**

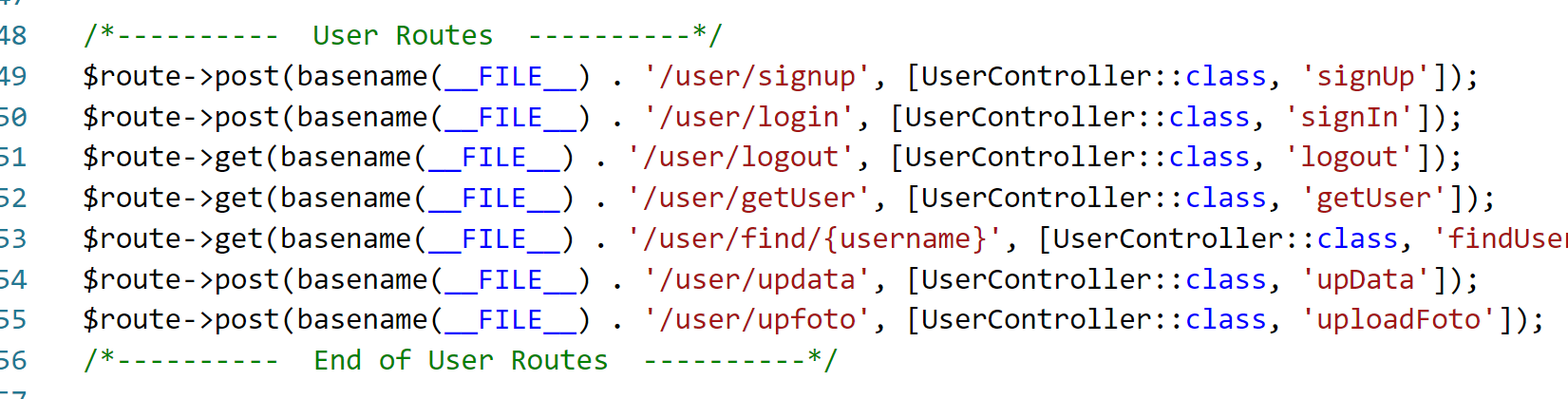
**Backend:**

Para la implementación se ha utilizado el PHP orientado a objetos y se ha aplicado el patrón MVC.

Se ha replicado el diseño de Laravel, mediante el modelado de rutas y Composer para añadir las dependencias necesarias.

Router.php:

dir: WebForoUpna/backend/public/router.php

 Este fichero consta de todas las rutas accesibles con las que se puede interactuar con el api, ejem:

Controladores:

dir: WebForoUpna/backend/src/Controller

En esta carpeta se han implementado los controladores listados en la descrita en la sección anterior, estos se encargan de sanear los datos y verificar que son válidos, verificado además que el cliente tenga los privilegios necesarios si es requerido, como para crear foro o comentar.

Modelos:

dir: WebForoUpna/backend/src/Models

En esta carpeta se encuentran todos los modelos listados en la gráfica de la sección anterior. Estos se encargan de realizar las acciones requeridas contra la base de datos, insertar, actualizar, obtener datos o verificar.

Database.php:

dir: WebForoUpna/backend/src/Models/Database.php

Para la conexión con la base de datos se ha utilizado el modelo singelton que establece una única conexión para no sobrecargar el api.

Config.php:

dir: WebForoUpna/backend/src/config/config.php

Este fichero contiene los datos para autentificarse con la base de datos.

Sanitazer.php:

dir: WebForoUpna/backend/src/Controllers/Sanitaizer.php

Este se encarga de limpiar los datos con los que se consultan a la api de condigo malicioso.

Session.php:

dir: WebForoUpna/backend/src/Models/Session.php

Se encarga de mantener los datos de sesión y controlarlos.

FileController.php:

dir: WebForoUpna/backend/src/Controller/FileController.php

Este se encarga de guardar las fotos cunado se van a actualizar. Para el almacenamiento de las fotos se ha utilizado un sistema de fichero, dado que es más rápido, se guardan en la carpeta backend/public/imgs, dentro de esta podemos ver las fotos de perfil o foro en /perfil o / foro respectivamente.

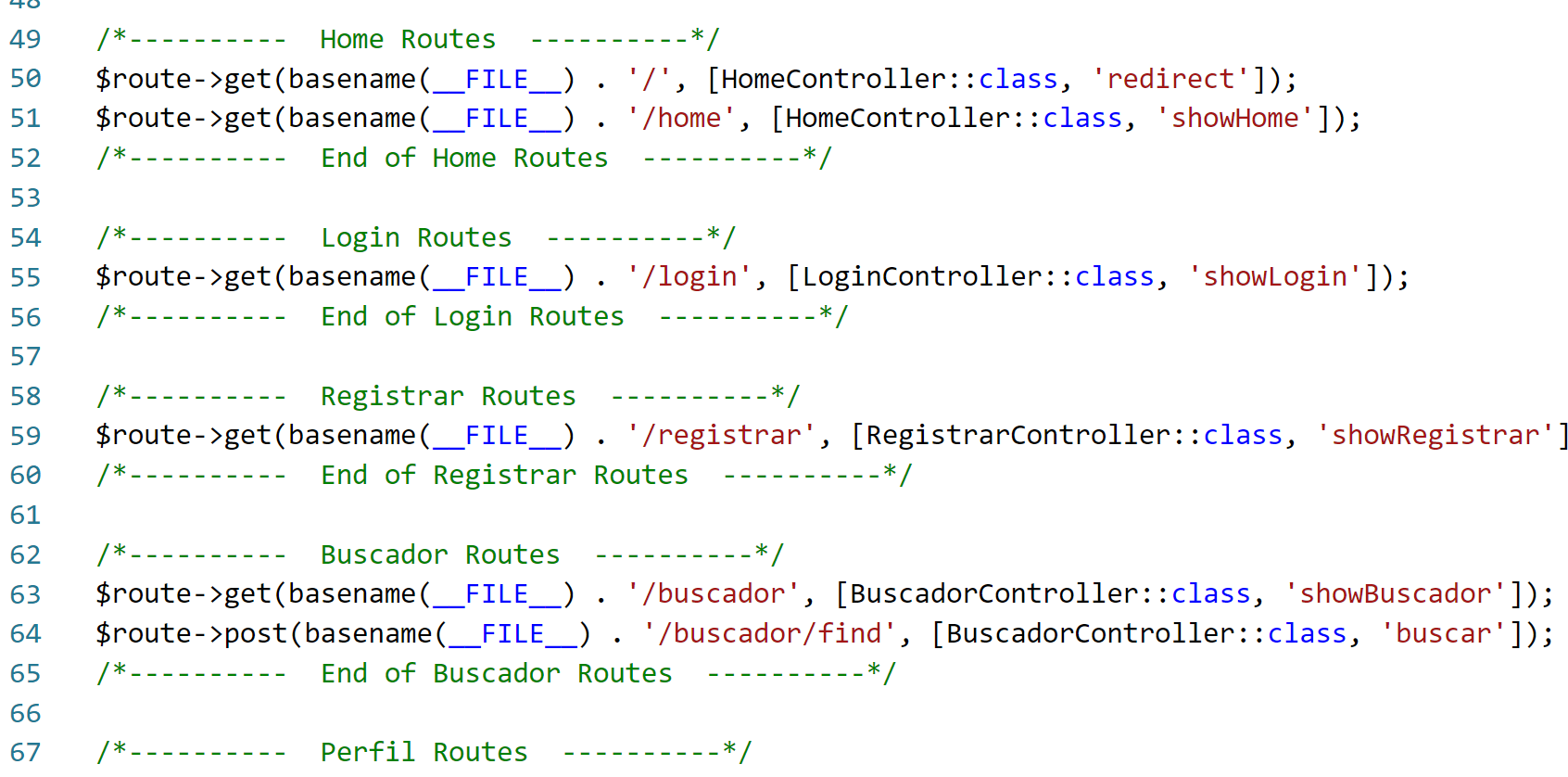
**FrontEnd:**

Al igual que en le backend se ha utilizado para la implementación el diseño PHP orientado a objetos y se ha aplicado el patrón MVC.

Para la construcción de la vistas se replicado el funcionamiento de Angular.

Router.php:

dir: WebForoUpna/frontend/public/router.php

 Este fichero consta de todas las rutas accesibles con las que el cliente puede interactuar para navegar por la web, ejem:

Controladores:

dir: WebForoUpna/fronend/src/Controller

En esta carpeta se han implementado los controladores listados en la descrita en la sección anterior, estos se encargan de sanear los datos y verificar que son válidos, además si es necesario datos de la base de datos mediante llamadas a la api.

Modelos:

dir: WebForoUpna/frontend/src/Models

En esta carpeta se encuentran todos los modelos listados en la gráfica de la sección anterior. Estos modelos se encargan de crear todas las vistas que se visualizan, modificándolas según si el usuario este logueado o no, por lo que se han creado plantillas fácilmente modificables al estilo de Angular, cada modelo consta de un template, css y js si es necesario.

/Templates:

dir: WebForoUpna/frontend/src/Models/templates/

En esta carpeta se pueden encontrar todas las vistas html que se utilizan en la web, para que sean fácilmente modificables por los modelos.

Dentro tenemos la carpeta /res/:

Aquí están tantos los css como los js en las carpetas /css y js respectivamente.

Si es requerido como en el caso de buscador se han descompuesto los html en componentes, al estilo de Angular, para la composición iterativa de los elementos repetidos.

Paginado.php:

dir: WebForoUpna/frontend/src/Models/Paginado.php

Como este es un componente que se repite tano en diseño y funcionalidad en dos partes de a web se ha realizado este modelo que permite construir el html de esta sección independientemente de la pagina a la que pertenezca.

Session.php:

dir: WebForoUpna/frontend/src/Models/Session.php

Al igual que en el backend hacemos en el frontend un control de los datos de sesión con esta implementación.

BackendConx.php & BackendConxController.php:

dir: ../frontend/src/Controllers/BackendConxController.php

dir: ../frontend/src/Models/BackendConx.php

Como es necesario hacer llamadas a la api se han creado estos elementos que permiten realizar una conexión con la api, el controlador sanea los datos y el modelo permite realizar una única conexión a la api mediante el modelo singelton, para no sobrecargar el servicio del frontend.

Resurce.php & ResurceController.php:

dir: ../frontend/src/Controllers/ResurceController.php

dir: ../frontend/src/Models/Resurce.php

Estos elementos se han realizado debido a que la web necesita acceder a los recursos que están en carpetas del servidor, como lo css y js. Para que tener un acceso seguro a ellos se ha realizado eso, que da únicamente acceso a estos ficheros.

Header.php:

dir: WebForoUpna/frontend/src/Models/Header.php

Al estilo de angular esta es la vista principal donde se montan todas las demás, dado que el Header es un elemento persistente.

Sanitazer.php:

dir: WebForoUpna/frontend/src/Controllers/Sanitaizer.php

Este se encarga de limpiar los datos con los que se consultan al servicio del frontend de condigo malicioso.

**Despliegue**

**Auditoria de Rendimiento**

Para esto se ha utilizado la herramienta de Google Chrome LightHouse, accesible desde el menú F12, en una de estas pestañas.

Se ha hecho la prueba en versión móvil, debido a que la herramienta valora en primer lugar la versión móvil, es decir si en el móvil tiene un buen rendimiento en la versión web el rendimiento será mucho mejor.

Se ha pasado esta prueba a todas la paginas de la web y se ha obtenido este resultado, +-1 dependiendo de la página:

