

MEMORIA DEL PROYECTO DE FIN DE GRADO EN DESARROLLO DE APLICACIONES WEB

Autor: David Cortijo Traba

Tutor: Sergio de Mingo

Introducción

La aplicación que he creado es un aula virtual que toma el nombre de NoodleMoodle.

Es una app que en resumen, sirve para que un administrativo del centro educativo que la use, introduzca en ella los módulos que se cursan en el mismo, cree las asignaturas que imparte el centro y las asocie entre si. De tal manera que cada usuario pueda matricularse en un determinado módulo y en sus correspondientes asignaturas. El profesor podrá subir allí los ficheros que considere oportunos para impartir el temario de cada asignatura y los alumnos podrán tener acceso a él.

Pese a la falta de tiempo, que ha desembocado principalmente en un diseño de web bastante escaso, creo que es una aplicación intuitiva y es algo que buscaba, ya que normalmente los aulas virtuales con los que me he topado eran algo engorrosos con flujos un poco liosos.

Es cierto que no he podido implementar todas las funcionalidades que tiene un aula virtual convencional y que, por ello, es posible que mi app sea algo más fácil e intuitiva. No obstante, creo que por ello también es un aula virtual apta para cualquier rango de edad y curso superior a Infantil.

Puesta en contexto

La idea de esta aplicación está motivada por la pandemia actual que estamos viviendo. Está dirigida a todos esos centros educativos que se han visto obligados a cerrar sus aulas y que necesitaban una forma de comunicar a profesores y alumnos fácilmente para seguir con los temarios correspondientes y posibilitar la continuación del año lectivo 2020-2021.

Creo que se enmarca bien dentro del contexto empresarial por varios motivos expuestos a continuación:

- El uso de este software por parte de un centro educativo (nuestro cliente), otorga al mismo un control total sobre el aula virtual sin que haya una empresa o entidad intermediando entre datos y centro. De este modo, la entrega de credenciales, tramitaciones y demás, serán administradas por el centro y por tanto, no habrá prácticamente complicaciones.
- Comparado con otros aulas virtuales, NoodleMoodle es muy intuitiva y prácticamente es un repositorio en el cual el profesor podrá dejar los archivos del temario que considere oportunos de una manera muy sencilla y organizada y los alumnos podrán consumirlos fácilmente. Podrán navegar entre sus módulos y asignaturas de un modo ágil y los profesores tendrán control sobre ello.

Requerimientos legales

En términos legales no tendría ningún requerimiento especial o adicional a los típicos requerimientos protocolarios de una matriculación en el centro educativo.

Riesgos

En un primer análisis si el centro educativo quisiera usar este software no correría ningún tipo de riesgo (ajeno a seguridad informática de la que hablaré más adelante). Y yo como programador tampoco ya que al ser una aplicación de

intranet (más adelante hablaré del por qué de esta decisión) la empresa se hará cargo del despliegue de la misma.

Consideraciones de seguridad

La aplicación será desplegada en una Intranet (más adelante explicaré por qué).

Por falta de tiempo no he hecho muchas comprobaciones de seguridad. Desde mi punto de vista la seguridad en una aplicación ya sea web o con otros protocolos es lo más importante de la misma y probablemente me hubiera tomado mucho tiempo. No obstante nunca voy a desplegar esta aplicación en ningún lado. Simplemente la hice por y para el proyecto final y por ello también su interfaz deja mucho que desear.

Para su seguridad habria tenido en cuenta ciertos factores, como los siguientes:

Para acceder a una aplicación lanzada en una intranet, o directamente a la intranet en si, tendríamos que lidiar con los *pentesters*. Son piratas informáticos que atacan constantemente al servidor para encontrar vulnerabilidades y que programan exploits para conseguir un uso inesperado o inadecuado de la misma con innumerables fines.

Una intranet es una red privada que utiliza la tecnologia de protocolo de internet. Al ser privada puede parecer segura pero ni mucho menos.

Lo primero que tendrían que conseguir los pentesters es acceder a la red interna y una vez ahí tratar de acceder a la intranet. Esto se podría hacer de varias formas:

- Usando la vpn que use la intranet.
- Consiguiendo hacer algún backdoor al servidor, que consiste en dejar un hueco en el servidor para volver a tomar control del mismo fácilmente en otra ocasión.

- Interceptando algunas peticiones accediendo al proxy interno que actúa de intermediario entre cliente y servidor.
- Por XSRF (Cross Site Request Forgery). Algo más peligroso y difícil de detectar que el XSS (Cross Site Scripting), y lo que hace es explotar la vulnerabilidad de un servidor falsificando una petición de un cliente a otro servidor completamente distinto. El servidor web confía en un usuario el cual sin darse cuenta está ejecutando código malicioso al hacer click por ejemplo en un enlace o descargarse una foto del propio servidor y hace una petición a otro servidor el cual hace una acción inesperada. Es difícil detectar quién fue el pirata informático detrás de ese exploit ya que por así decirlo fue un usuario normal el cual lo hizo sin darse cuenta.

De las alternativas expuestas anteriormente la más compleja sería la de hackear la vpn. Ya que se tendría que conseguir acceso al hardware diseñado para dar vpn y obtener las credenciales de un usuario que tenga acceso al mismo sería algo complicado. Otra forma sería la de buscar algunas vulnerabilidades tales como buffer overflow, root access, etc.

La alternativa de acceder al dispositivo del proxy interno e interceptar alguna petición sería parecida a la anterior.

La alternativa más fácil sería la del backdoor al servidor web. Si los pentesters encontraran alguna inyección sql en el portal de acceso al servidor web, podrían acceder al sistema como administrador y subir una shell propia. Con ella podrían hacer un bouncer web, que es un software que permite a un usuario ocultar la fuente original de la conexión del usuario para darle privacidad frente a otros usuarios y también es capaz de enrutar el tráfico a través de un lugar específico. Con ello, los pentesters podrían redirigir o más bien replicar paquetes que van a la ip del servidor a otra ip que especifiquen (las suyas) y así obtener los datos.

He querido comentar estas alternativas anteriores simplemente por dar una idea de la cantidad de formas que hay de acceder a una intranet.

Si bien es cierto que parte de ellas conciernen en gran medida al centro educativo que nos compra el software, nosotros también tenemos que tomar ciertas medidas para intentar evitar ciertos ataques.

Por ejemplo una de las anteriores alternativas que podríamos evitar en cierto modo y que he implementado en NoodleMoodle son los XSRF a nuestra app. Con Laravel (el framework de php que he utilizado) podremos hacerlo fácilmente ya que con su sistema de vistas Blade, puedes implementar un token para cada usuario con sesión activa, de tal manera que sólo dicho usuario pueda hacer solicitudes a su nombre. Con ello evitaremos ataques XSRF.

Como comentaba anteriormente, me habría gustado dedicarle más tiempo a este apartado ya que considero que es el más importante de todos. Podría haber incluido comprobaciones en las extensiones de los archivos subidos por los profesores para rechazar algunos no deseados, ya que un profesor es probable que no tenga ninguna acción malintencionada (podría tenerla), pero podrían robarle las credenciales y otra persona si que podría querer hacer alguna acción de ese tipo.

También me faltó algo básico como hacer una comprobación cada vez que se mostraba una vista que solo podía ver un administrador, un profesor o un alumno. Así un alumno o profesor modificando la url no podría ver las vistas destinadas al administrador o viceversa.

Descripción técnica

Para montar el proyecto decidí usar las siguientes tecnologías:

- Php

- Javascript
- MySql
- Laravel
- Bootstrap

El lenguaje que he utilizado para el backend es php. Realmente no le he elegido frente a otros por nada en especial, simplemente porque tengo más fluidez por haber sido el lenguaje que utilizamos en el curso en servidor.

Javascript lo he utilizado en contextos muy específicas en el cliente y me ha sido de gran ayuda. Con él he podido hacer llamadas AJAX al servidor y conseguir ejecutar ciertas consultas en la base de datos modificando la vista sin necesidad de hacer una recarga. Esto ha servido para mejorar en términos de experiencia de usuario.

MySql lo escogí como Sistema Gestor de Base de Datos (SGBD) por varias razones.

En primer lugar era también la tecnología que más usamos en la asignatura de servidor y con la que más familiarizado estaba, por lo tanto esto tuvo bastante peso en mi decisión final.

En un primer momento tuve la intención de usar MongoDB, ya que me interesa bastante su funcionamiento y quería aprenderlo. Finalmente no lo escogí por la curva de aprendizaje que podía tener y porque habría tenido ciertas complicaciones.

MongoDB es un gran SGBD NoSQL y tiene muchas ventajas. Tales como la escalabilidad que aporta, la flexibilidad que tiene dado que sus datos se guardan en colecciones y los documentos dentro de las mismas no tienen por qué seguir el mismo esquema. No requiere potentes recursos para su uso y sus consultas están muy bien optimizadas teniendo también mucha documentación a nuestra disposición.

A pesar de todas sus ventajas, no vi apropiado usarlo por lo siguiente:

- En mi aplicación iba a usar el framework de php Laravel y, si bien es cierto que se puede usar MongoDB, ambos sistemas no encajan del todo bien debido a que SQL está prácticamente incrustado en Laravel y proporciona una gran experiencia a los nuevos usuarios de Laravel por su simplicidad en consultas usando el propio ORM que proporciona llamado Eloquent.
- Por otra parte, también me venía bien MySql porque realmente la aplicación que he creado no va a escalar o al menos no tengo intención de que lo haga, ya que no voy a continuar con su desarrollo. Por tanto MySql funciona muy bien en este tipo de casos con el almacenamiento de datos que ofrece totalmente organizado y jerarquizado con claves primarias y foráneas.
- Además, con MongoDB no se pueden hacer ciertas consultas que con SQL se podrían hacer, como por ejemplo los JOIN (sí se pueden hacer pero se tendrían que hacer de otra forma algo más complicada). Así pues en mi aplicación habría ciertas duplicaciones innecesarias que con MySql me he ahorrado debido a la posibilidad de realizar consultas más elaboradas.

Escogí el uso de Bootstrap para que me facilitara la vida a la hora de crear un diseño de la interfaz algo vistoso, a pesar de no haberle dedicado mucho tiempo para finalizar el resto de funcionalidades.

Y finalmente hablaré del framework de php que utilicé, Laravel.

Laravel

Antes de comenzar con el proyecto quería aprender un framework de php e implementarlo en el mismo. Estuve dudando entre los dos más usados actualmente que son Symfony y Laravel y finalmente me decanté por este último por lo siguiente:

Laravel es menos robusto que Symfony y realmente, es prácticamente su hijo. Yo para este proyecto no necesitaba un gran entorno de trabajo puesto que con el tiempo limitado no iba a conseguir aprenderme uno de los dos al 100% y la curva de aprendizaje de Symfony es mucho mayor al tener tantas particularidades.

Por otro lado, Laravel es algo menos estricto que Symfony en prácticamente todo, pero lo que me hizo decantarme en este sentido fue su ORM Eloquent. Es mucho más sencillo configurar un modelo en este ORM y con la documentación que ofrece se entiende más o menos rápido el funcionamiento del mismo y de su Query Builder.

También tiene un buen motor de plantillas (Blade) que, si bien es cierto que es algo más escaso que el de Symfony (Twig), es mucho más sencillo a la hora de incluir variables o de establecer estructuras de control por medio de sintaxis propias del motor Blade.

Me gustó también la interfaz de línea de comandos que ofrecía Laravel, artisan. Con ella me pude ahorrar mucho trabajo a la hora de crear controladores, modelos, etc. Y es bastante ágil para crear rápidamente el esqueleto del modelo vista controlador que usé.

Algo que aprendí con el desarrollo de esta app que antes no conocía son las migraciones entendidas desde el punto de vista de Laravel, no las migraciones de datos. Este tipo de migraciones son una especie de control de versiones de la base de datos incluso algo más que eso, ya que nos permite crear tablas, establecer relaciones, modificarlas y eliminarlas mediante la línea de comandos.

Otra herramienta que incluye Laravel y de la cual me di cuenta tarde es la clase Validator. Con ella se pueden validar formularios con reglas muy dinámicas que nos ahorran mucho trabajo a la hora de ir haciendo comprobaciones a mano sobre determinados campos.

Y también incluye un token muy sencillo de implementar con Blade (@csrf), que comenté anteriormente en el apartado de seguridad, el cual nos protege de ataques XSSRF. Se utiliza para verificar que el usuario autenticado es quien en realidad está haciendo la petición al servidor.

En líneas generales mis impresiones de Laravel han sido muy buenas y hace mucho más dinámico el desarrollo facilitando el centrarse al 100% en la programación de la aplicación. Obviamente no lo he explotado todo lo que lo podía explotar ya que me dejo muchas cosas que ofrece y no he comentado porque realmente no las conozco, pero seguiré aprendiendo sobre él porque es muy interesante.

Modelo de datos

Monté mi base de datos con la herramienta phpmyadmin y mi objetivo principal era intentar estructurar todo de manera organizada, para así no duplicar datos innecesariamente, o incluir otros que no son realmente relevantes en nuestra aplicación.

Para ello me serví de la creación de estas tablas con estas relaciones entre sí que explicaré a continuación:

- *Asignaturas*

Esta tabla es la encargada de representar la entidad asignatura en la base de datos. Para ello necesité 3 campos necesarios.

Id: Es el identificador de cada fila. Un entero autoincrement como clave primaria que me sirve para diferenciar una asignatura entre todas las demás y poder acceder a cada una de ellas ya que son únicas. Es también una clave foránea que está relacionada con el campo IdAsignatura tanto en la tabla AsignaturasModulos como en la tabla UsuariosAsignaturas.

NombreAsignatura: Campo encargado de, como su propio nombre indica, guardar el nombre de cada una de las asignaturas.

CodigoActivacion: Más adelante se entenderá mejor este campo cuando explique el funcionamiento de la app. Básicamente cada asignatura tendrá su propio código de activación, con el cual los alumnos podrán matricularse.

- Usuarios

Esta tabla es la encargada de representar la entidad Usuario. Es la tabla con más campos ya que contiene información personal del mismo, la cual será relevante para el centro educativo.

Id: Es el identificador de cada fila. Un entero autoincrement como clave primaria. Con el podré dar exclusividad a cada uno de los usuarios y podré acceder a cualquier fila fácilmente.

Nombre: Representa el nombre del usuario.

Apellidos: Representa los apellidos de la persona.

Email: Representa el email de la persona y es el campo que usaré para guardar en la sesión de cada usuario ya que los emails son únicos. Pude haber utilizado un login en lugar del email pero no lo vi adecuado ya que el usuario podía elegir el login que quisiera y eso dotaría a la aplicación de cierto carácter informal que no quería que tuviese.

Telefono: Representa el teléfono del usuario.

Ciudad: Representa la ciudad en la que vive el usuario.

ComunidadAutonoma: Representa la comunidad autónoma a la que pertenece la ciudad en la que vive el usuario.

IdRol: Es una clave foránea relacionada con el campo Id de la tabla Roles. Con este campo puedo obtener el rol del usuario y así saber si es un alumno, un profesor o el administrador.

FechaPrimerAcceso: Representa la fecha del primer inicio de sesión del usuario que coincide con el registro del mismo.

FechaUltimoAcceso: Representa el último inicio de sesión de un usuario. Me sirve para que así los profesores tengan un cierto control sobre los alumnos para saber si van accediendo al aula o no. También es útil para el administrador y para los alumnos, para ver si un profesor de una determinada asignatura ha podido subir algún archivo nuevo o no.

Password: Representa la contraseña de un usuario a la cual le pasamos una función hash antes de insertarla a la tabla.

- *Modulos*

Esta tabla es la encargada de representar la entidad Módulo. Un módulo es un curso escolar determinado al que se le van a asociar determinadas asignaturas.

Id: Es el identificador de cada fila. Un entero autoincrement como clave primaria. Diferencia a todos los módulos entre si. Es también una clave foránea que está relacionada con el campo IdModulo tanto en la tabla AsignaturasModulos como en la tabla UsuariosModulos.

NombreModulo: Campo encargado de, como su propio nombre indica, guardar el nombre de cada uno de los módulos.

Esta tabla es bastante parecida a la de Asignaturas. La única diferencia es que esta no tiene el campo CodigoActivacion. Esto es debido a que no vi necesario implementarlo. Un usuario podría unirse a cualquier módulo sin ningún tipo de restricción, ya que no hay ningún tutor de módulo en mi aplicación que pueda monitorear o controlar esas matriculaciones (entiendo que esto lo haría una minoría de alumnos, tan solo algún gracioso, o algún pirata informático en busca de alguna vulnerabilidad). No obstante, necesitan unirse a una asignatura para poder hacer algo útil dentro del módulo y ahí ya sí es necesario un código de activación y a parte la confirmación del profesor de la asignatura.

- Roles

Esta tabla es la encargada de representar los roles que puede tener un usuario dentro de la aplicación. En este caso solo pueden tener 3: alumno, profesor o administrador.

Id: Es el identificador de cada fila. Un entero como clave primaria insertado a mano. Diferencia a todos los roles entre si. Tiene una relación con la tabla usuarios para que con el campo IdRol de la tabla usuarios se pueda acceder a una determinada fila de la tabla Roles con el campo Id y así saber el nombre del rol que tiene ese usuario.

NombreRol: Representa el nombre de cada uno de los roles insertados en la tabla.

- AsignaturasModulos

Esta tabla es la encargada de establecer una relación entre las asignaturas y los módulos de tal forma que un módulo pueda tener varias asignaturas y una misma asignatura pueda estar en varios módulos a la vez. Por ejemplo, puede haber una asignatura llamada Programación que se curse tanto en DAM como en DAW. Es la misma asignatura pero puede ser impartida de muchas maneras, pero a pesar de ello sigue siendo la misma asignatura y en esencia conforma una entidad total que es Programación. Por tanto no he introducido un campo en la tabla Asignaturas que se llamara IdModulo y he creado esta tabla para relacionar de muchos a muchos.

IdAsignatura: Forma parte de la clave primaria junto con el campo IdModulo y representa el identificador de una asignatura en concreto. Establece una relación con el campo Id de la tabla Asignaturas.

IdModulo: Forma parte de la clave primaria junto con el campo IdAsignatura y representa el identificador de un módulo en concreto. Establece una relación con el campo Id de la tabla Modulos.

- UsuariosModulos

Esta tabla es la encargada de establecer una relación entre los usuarios y los módulos de tal forma que un módulo pueda tener muchos usuarios matriculados y un usuario pueda estar matriculado en muchos módulos. Estableciendo así una relación de muchos a muchos.

IdUsuario: Forma parte de la clave primaria junto con el campo IdModulo y representa el identificador de un usuario en concreto. Establece una relación con el campo Id de la tabla Usuarios.

IdModulo: Forma parte de la clave primaria junto con el campo IdUsuario y representa el identificador de un módulo en concreto. Establece una relación con el campo Id de la tabla Modulos.

- UsuariosAsignaturas

Esta tabla es la encargada de establecer la relación entre los usuarios y las asignaturas de tal manera que un usuario pueda cursar o impartir muchas asignaturas y una asignatura pueda ser cursada o impartida por muchos usuarios. Estableciendo así una relación de muchos a muchos.

En ella también tuve que añadir otro campo más que es el id del módulo de cada asignatura insertada en la tabla. El motivo por el que lo hice fue porque un usuario puede pertenecer a dos módulos en los cuales esté la misma asignatura, que aunque se enseñe de manera diferente, es en esencia la misma asignatura.

IdUsuario: Forma parte de la clave primaria junto con el IdAsignatura y el IdModulo. Representa el identificador de un usuario en concreto. Establece una relación con el campo Id de la tabla Usuarios.

IdAsignatura: Forma parte de la clave primaria junto con el campo IdUsuario y el IdModulo. Representa el identificador de una asignatura en concreto. Establece una relación con el campo Id de la tabla Asignaturas.

IdModulo: Forma parte de la clave primaria junto con el campo IdUsuario y el IdAsignatura. Representa el identificador de un módulo en concreto. Establece una relación con el campo Id de la tabla Modulos.

- Avisos

Esta tabla fue creada para representar las notificaciones que recibe un profesor cuando algún alumno quiere matricularse en su asignatura y ha completado el formulario de la misma introduciendo el código de activación correspondiente. Aquí el profesor tiene la potestad para aceptar la solicitud de inserción del alumno o denegarla.

Id: Es un identificador autoincrement como clave primaria. Sirve para diferenciar cada una de las filas que se insertan en la tabla.

IdAlumno: Es el identificador del alumno que ha solicitado la inserción a la asignatura. Establece una relación con el Id de la tabla Usuarios.

IdAsignatura: Es el identificador de la asignatura a la cual el alumno quiere acceder. Establece una relación con el Id de la tabla Asignaturas.

IdModulo: Este campo es necesario por lo explicado anteriormente en la tabla usuariosasignaturas. Es el identificador del módulo de la asignatura en cuestión. Establece una relación con el Id de la tabla Modulos.

IdProfesor: Es el identificador del profesor que imparte la asignatura a la cual el alumno quiere acceder. Si hubieran más de un profesor impartiendo esa asignatura en el mismo módulo, la lógica que he seguido ha sido la de enviar el aviso al primer profesor que se matriculó, entendiendo que ese podría ser el tutor de la misma o el principal del resto de profesores porque imparta más horas, etc. Establece una relación con el Id de la tabla Usuarios.

Funcionamiento de la aplicación y casos de uso

El funcionamiento de la aplicación en líneas generales es muy sencillo de entender e intuir. Me he centrado en eso debido a que con la situación actual, muchos padres de alumnos de infantil o primaria quieren estar al tanto del aula virtual para ayudar a sus hijos con las materias o simplemente controlarles un poco. Normalmente estos padres no suelen estar muy familiarizados con las tecnologías y creo que es una razón de peso a tener en cuenta. Por otro lado, si bien es cierto que la

tecnología está muy extendida entre las nuevas generaciones, es posible que aún haya alguien con unas habilidades menos desarrolladas en lo que a manejo de nuevas tecnologías se refiere, y también hay que tenerlo en cuenta.

Para explicar el funcionamiento de la aplicación de una manera más organizada, dividiré la sección en tres subsecciones que serán los roles de cada usuario.

- *Alumno*

En primer lugar, deberá rellenar un formulario de registro, si es que no está registrado, introduciendo sus credenciales. Una vez registrado o logueado si ya lo estaba, el menú de navegación cambia y le brinda otras opciones.

Home

Esta sección es solo visual en la que se expone un slider con tres opciones que, aunque ya estén en el menú de navegación, están en el Home para facilitar al usuario el moverse entre secciones de una manera más cómoda. Las tres opciones serían el acceso a sus módulos propios, el acceso a sus asignaturas y el acceso a su perfil de alumno.

Mis módulos

Esta sección expone de una manera visual los módulos en los que el alumno se ha matriculado. Cada módulo se muestra en un card de bootstrap y en su interior hay un botón de acceso que posibilita al usuario acceder al detalle del módulo para ver más datos. Esta sección también incluye un botón que permite al usuario ver un listado de módulos a los cuales puede unirse.

Al acceder a un módulo existente o al clicar en un módulo del listado de todos los módulos para unirse al mismo, se verá, como expongo anteriormente, el módulo en detalle.

En esta vista nos aparecen 3 listados.

El primero nos muestra todas las asignaturas asociadas al módulo en el que estamos. Si pertenecemos al módulo, tendremos la posibilidad de solicitar acceso a una si no estamos matriculados, acceder a ella si lo estamos o esperar confirmación por parte del profesor de la misma si ya hemos solicitado acceso pero aún no ha sido tramitado. Si queremos solicitar acceso a una asignatura que aún no tiene profesor, ese botón no hará nada. Sin embargo si esa asignatura si lo tiene, nos enviará a un formulario de activación en el cual deberemos introducir el código de activación y, si es correcto, se enviara un aviso al profesor indicando que queremos acceder a la asignatura.

El segundo listado muestra a todos los profesores asociados al módulo y las asignaturas que imparten en el mismo.

El tercero muestra a todos los alumnos asociados al módulo y las asignaturas que cursan en el mismo.

Dentro de la vista en detalle del módulo es donde podremos unirnos al módulo si no lo estamos o abandonarlo.

Mis asignaturas

Esta sección muestra un listado de todas las asignaturas que cursa el alumno en cuestión y la posibilidad de acceder a una en concreto.

Al acceder a una asignatura tanto por esta vista como por la del detalle del módulo, se mostrará una nueva vista que corresponde a la asignatura en cuestión.

En esta nueva vista tendremos dos nuevas subsecciones. Una de ellas será el temario, que es la que se muestra de manera predeterminada al acceder a la asignatura.

En el temario se listarán todos los archivos que ha subido el profesor y se podrán descargar. En esta vista también se podrá abandonar la asignatura si el alumno quisiera.

La otra subsección es la de participantes. En ella se muestra un listado con todos los integrantes de la asignatura, tanto profesores como alumnos, y su último inicio de sesión. Aquí también se podrá abandonar la asignatura.

Mi perfil

Esta sección muestra de un modo vistoso todos los datos personales del alumno o del usuario que hayamos clickeado para ver su perfil. Su nombre completo, teléfono, email, ciudad, su primer acceso, su último acceso, sus módulos, sus asignaturas, etc.

Cerrar sesión

Este enlace del navbar no es una sección propiamente dicha, si no que nos sirve para cerrar nuestra sesión y desconectarnos de la aplicación.

- *Profesor*

En primer lugar (igual que el alumno), deberá rellenar un formulario de registro, si es que no está registrado, introduciendo sus credenciales. Una vez registrado o logueado si ya lo estaba, el menú de navegación cambia y le brinda otras opciones.

Home

Esta sección es igual que la expuesta anteriormente en el alumno.

Mis módulos

Esta sección es prácticamente igual que la sección del alumno, ya que tan solo muestra una diferencia.

El profesor no necesita solicitar acceso a nadie para ser nombrado profesor de una asignatura en concreto y, por tanto, no necesita introducir el código de activación correspondiente de las asignaturas que vaya a impartir. No obstante, deberá pedir verbalmente al administrador el código de esa asignatura para dársela a sus alumnos.

Mis asignaturas

Esta sección muestra una diferencia importante con la del alumno.

En la vista de cada asignatura impartida por el profesor, en la sección del temario podrá subir los archivos que estime oportuno para que los alumnos lo descarguen. También podrá borrar los archivos subidos.

Mi perfil

Esta sección es igual que la expuesta anteriormente en el alumno.

Avisos

En esta sección se muestran todas las solicitudes de los alumnos que quieren acceder a la asignatura o asignaturas impartidas por el profesor en un determinado módulo o módulos. En ella, el profesor podrá aceptar cada solicitud o denegarla según estime oportuno.

Cerrar sesión

Este enlace del navbar no es una sección propiamente dicha, si no que nos sirve para cerrar nuestra sesión y desconectarnos de la aplicación.

- *Administrador*

El administrador es el encargado de gestionar toda la aplicación. Correspondería con un trabajador del centro educativo el cual tenga la potestad para dirigir y organizar.

Home

Esta sección es igual que la expuesta anteriormente en el alumno y el profesor.

Módulos

Esta sección muestra un listado de todos los módulos que tiene la aplicación y las asignaturas asociadas al mismo. En la misma vista, el administrador podrá crear un nuevo módulo y asignar las asignaturas correspondientes a cada módulo de una manera muy sencilla.

También podrá eliminar el módulo seleccionado rompiendo así también la relación con las asignaturas que estuvieran asociadas al mismo.

Asignaturas

Esta sección muestra un listado de todas las asignaturas impartidas en el centro.

El administrador podrá crear nuevas asignaturas generando aleatoriamente su código de activación y también podrá eliminar las asignaturas que estime oportuno.

Usuarios

Esta sección muestra un listado de todos los usuarios registrados en la aplicación. Es una vista que da mucha información sobre cada usuario y el administrador tendrá la potestad de eliminar a un usuario del sistema si lo estima oportuno. Esta vista tiene la opción de filtrar por roles a los usuarios para que sea más sencillo encontrar a algún usuario en concreto.

Mi perfil

Esta sección es igual que la expuesta anteriormente en el alumno y el profesor.

Cerrar sesión

Este enlace del navbar no es una sección propiamente dicha, si no que nos sirve para cerrar nuestra sesión y desconectarnos de la aplicación.

Despliegue de la aplicación

Para desplegar la aplicación es necesario clonar el repositorio de la misma en la carpeta pública del servidor web del centro educativo. Una vez clonado, se tendrán que instalar todas las dependencias con el gestor de dependencias de php Composer.

Al haberse instalado todo, ya estaría lista para su uso.

He decidido hacerla para intranet debido a que no he tenido en cuenta el registro de cada centro educativo. Ya que de esta manera habría muchísimos más datos y con mysql no sería una gran opción si la aplicación escalara cada vez más y tendría que

reprogramar muchas cosas. Además no podía abarcar un desarrollo tan grande ahora mismo.

Además de esta manera, tendría menos problemas de seguridad (aunque los habría) y el centro educativo tendría mucho más control sobre la aplicación si en esencia, es suya propia.

No habria problemas de tramitación de credenciales, registros, ya que todo queda centralizado en el propio centro.

De esta manera podría quizás sacar más beneficio económico que si mi aplicación estuviera en internet disponible para todo el mundo.