

DESARROLLO DE APLICACIONES WEB
WEB ACADEMIA DE ESPAÑOL CON AULA VIRTUAL
PROYECTO FINAL

Antonio Jesús Perea Aguilar

ÍNDICE

1. Título	2
2. Antecedentes.....	2
3. Definición del problema a solucionar	2
4. Justificación del proyecto	3
5. Objetivos y alcance	3
6. Procedimiento y tecnología a usar.....	4
7. Presupuesto	5
8. Product Backlog.....	6
9. Backlog.....	9
10. Descripción del trabajo	10
11. Documentación.....	13
12. Retrospectiva.....	23
13. Sprint 2 Backlog.....	24
14. Descripción del trabajo Sprint 2	24
15. Documentación Sprint 2.....	27
16. Retrospectiva Sprint 2.....	39
17. Backlog Sprint 3.....	40
18. Descripción del trabajo Sprint 3	40
19. Documentación Sprint 3.....	42
20. Retrospectiva Sprint 3.....	48
21. Manual de uso y puesta en marcha	48
22. Conclusiones globales acerca del proyecto	55
23. Webgrafía	55

1. Título

Título del proyecto: Web academia de español con aula virtual

Tutor del proyecto: Alejandro Romero

Autor del proyecto: Antonio Jesús Perea Aguilar

Ciclo: DAW

2. Antecedentes

Sabemos que la enseñanza ha estado con nosotros desde el inicio de la humanidad, desde como enseñaban a las nuevas generaciones como se debía de recolectar, cazar y sobrevivir de la mejor manera posible, hasta de que manera construir grandes edificios, realizar investigaciones, descubrir nuevas tecnologías e incluso crecer en pensamiento y filosofía. Todo esto ha sido posible gracias a la enseñanza de grandes maestros que impartían sus grandes dotes de maestría a las nuevas generaciones.

El nacimiento de las aulas virtuales como las conocemos hoy día, fue posible gracias a las tecnologías de la información y la comunicación (TIC) que permiten a los usuarios acceder, almacenar, transmitir y manipular la información.

En 1960 Donald Bitzer desarrolla lo que hoy día denominamos como aula virtual. Donde los alumnos tenían la oportunidad de acceder a documentos del curso y escuchar conferencias que habían sido grabadas previamente

Por lo que, gracias a esta aula virtual además de poder aprender en escuelas, colegios, centros de enseñanza o academias, podemos aprender gracias a las tecnologías que nos brindan la posibilidad de conectarnos desde nuestros hogares y que tanto alumno como profesor no tengan la necesidad de presentarse en un mismo lugar, si no que nos ofrezcan la oportunidad de aprender a través de aulas virtuales donde se puede disponer de todo el contenido para aprender cualquier oficio o idiomas.

3. Definición del problema a solucionar

Como se acaba de mencionar, nos encontramos en pleno auge tecnológico, donde actualmente la mayoría de las empresas disponen de una web online. Por lo que es de vital importancia crear una web académica que disponga de aula virtual, donde

los usuarios registrados puedan acceder a información de carácter educativo y el profesorado puedan compartir diferente tipo de documentos, videos, imágenes, conferencias, exámenes y tareas para que cada uno de los alumnos de los que disponen acceso al aula puedan empaparse de todo ese contenido.

4. Justificación del proyecto

Visto los antecedentes, se plantea la creación de una página web que publicite la academia y además disponga de una sección con aula virtual para los usuarios registrados. Como se trata de una web online, no se hace necesario instalar ningún software en nuestros equipos y esto brinda la posibilidad de tener mayor acceso para cualquier usuario que disponga de una conexión a internet y un navegador web.

Hoy en día existen un gran número de academias y centro de enseñanzas que disponen de sitio web, pero no todos disponen de aula virtual para que el alumno pueda tener el contenido impartido del día a su disposición en la web, por ejemplo.

Por eso se hace necesario crear actualmente webs académicas que brinden información publicitaria como, por ejemplo: localización, ofertas de las que dispone, información de contacto, hacia quién va dirigido, niveles de enseñanza, etc. Además, disponga de un aula virtual donde existan 2 roles diferentes: alumno y profesor. Los alumnos podrán acceder a todo el contenido educativo (temario impartido en clase, exámenes y videos explicativos) y los profesores podrán ingresar nuevos alumnos, ver un listado de los alumnos a los que imparten clase y poder subir archivos al aula virtual.

En conclusión, se propone la creación y desarrollo de una página web académica de español donde se ofrezca información de la academia misma y que disponga de aula virtual, donde existirán dos tipos de usuarios (alumno y profesor). El alumno podrá tendrá permisos de lectura y el profesor permisos todos los permisos para la subida, modificación y lectura de los archivos.

5. Objetivos y alcance

Objetivos

El objetivo se trata de realizar una página web de una academia de español que disponga de un aula virtual y que funcione en cualquier navegador web.

Alcance

- Análisis e investigación previa:
 - Previa investigación sobre los antecedentes del proyecto.
 - Justificación del proyecto.
 - Requisitos del proyecto.
 - Definición del proyecto a crear.
- Diseño e implementación:
 - Estudio previo del diseño de la web en su conjunto.
 - Diseño de un modelo de la web a crear.
 - Documentar el diseño de la web.
 - Implementación de las conexiones back-end necesarias.
 - Implementación del front-end para el diseño de la web.
 - Documentar resultados y dificultades obtenidas en la implementación.
- Comprobación y pruebas para verificar el funcionamiento de la aplicación web a implementar:
 - Pruebas unitarias.
 - Pruebas de integración.
 - Prueba del sistema.
 - Documentación de las pruebas realizadas y los resultados obtenidos.
- Entrega de documentación con metodología SCRUM:
 - Idea y anteproyecto: **08/04/2022**.
 - Versión básica documentada: **22/04/2022**.
 - Versión mejorada documentada: **06/05/2022**.
 - Entrega final: **20/05/2022**.

6. Procedimiento y tecnologías a usar

Tecnologías y software	Empleo
Atom	Editor de código fuente
Mysql	SGBD relacional
Framebox	Creación de mockups
phpMyAdmin	Base de datos relacional

PHP	Lenguaje de programación back-end
HTML	Lenguaje para estructurar web
Javascript/JQuery	Lenguaje para añadir interactividad
CSS	Lenguaje para dar estilo a la web
GitHub	Para almacenar el código del proyecto
PHPUnit/Codeception	Realización de pruebas
Apache	Servidor que aloja la web creada mediante protocolo http

7. Presupuesto

Presupuestado en horas:

Procesos	Subprocesos	Horas
Análisis e investigación previa	Previa investigación sobre los antecedentes del proyecto	2
	Justificación del proyecto	1
	Requisitos del proyecto	1
	Definición del proyecto a crear	2
Total análisis e investigación previa		6
Diseño e implementación	Estudio previo del diseño de la web en su conjunto	2
	Diseño de un modelo de la web a crear	4
	Implementación de las conexiones back-end necesarias	6
	Implementación del front-end para el diseño de la web	10

	Documentar el diseño de la web	2
	Resultados y dificultades obtenidas	2
Total diseño e implementación		26
Comprobación y pruebas para verificar el funcionamiento de la aplicación web a implementar	Pruebas unitarias	2
	Pruebas de integración	2
	Prueba del sistema	2
	Documentación de las pruebas realizadas y los resultados obtenidos	2
Total comprobación y pruebas para verificar el funcionamiento de la aplicación web a implementar		8
Resultado horas totales		40

8. Product Backlog

Prioridad	Requisito	Funcionalidad
Obligatorio.	Compatibilidad con los navegadores más usados (Chrome, Firefox, Edge e Internet Explorer).	La página web se podrá visualizar desde cualquier navegador web.
Obligatorio.	Página de inicio atractiva.	La página web deberá de disponer de un inicio al sitio atractivo, publicitando la empresa.
Obligatorio.	Menús desplegables.	Deberá de tener una serie de menús y submenús desplegables que den acceso a otras secciones de la web (página de contacto, página de inicio, cursos, etc).

Obligatorio.	Aula virtual.	En la web habrá una sección denominada aula virtual, en la cual los alumnos que han sido registrados, podrán acceder a diferentes archivos para aprender.
Obligatorio.	Log in con dos tipos de usuarios (alumno, profesor).	La sección de aula virtual estará compuesta de dos usuarios diferentes. Uno solo con permisos de lectura de archivos y otro con permisos para registrar usuarios; subir, eliminar y editar archivos.
Obligatorio.	Acceso a archivos (usuario alumno).	Una vez ingrese el usuario podrá visualizar en la página al curso que está inscrito y una serie de archivos que el usuario profesor irá subiendo para que puedan visualizarlos.
Obligatorio.	Registro de usuarios; subida, edición y lectura de archivos (usuario profesor).	Los usuarios con rol de profesor tendrán la posibilidad de registrar alumnos a su curso, subir archivos y documentos, además de poder eliminarlos y visualizarlos.
Obligatorio.	Conexión con la base de datos.	Los datos de los usuarios serán almacenados en una

		base de datos relacional, con la cuál se establecerá conexión cuando sea preciso hacer log in o añadir/eliminar usuarios.
Obligatorio.	Página compuesta por secciones.	Desde un menú se podrá acceder a cada una de las partes de la página web (contacto, cursos, aula virtual, quiénes somos y página de inicio.)
Obligatorio.	Texto en varios idiomas	El texto será escrito originalmente en español, pero se tiene que poder traducir a cualquier otro idioma.
Obligatorio.	Cierre de sesión.	Los usuarios que han ingresado con sus respectivas claves y usuarios tendrán la posibilidad de cerrar la sesión en cualquier momento con un botón/click.
Obligatorio.	Formulario de contacto.	La web tendrá un formulario en la página de contacto para que cualquier usuario que acceda a la web pueda mandar un mensaje de

		correo electrónico para solicitar información.
Opcional.	Web responsive.	Web adaptable a dispositivos.
Opcional.	Cambios de contraseña.	Los usuarios al ser registrados por el usuario profesor, obtendrán una contraseña definida por el usuario que los registra. Una vez ingresan tendrán la posibilidad de cambiar la contraseña.
Opcional.	Contraseña olvidada.	Si al usuario se le olvida la contraseña, tendrá la opción de generar una nueva contraseña enviándole la contraseña nueva al correo electrónico con el que ha sido registrado.
Opcional.	Usuarios registrados listados.	El usuario con rol de profesor tendrá la opción de poder visualizar todos los alumnos de los que dispone, con iconos en el lado derecho de los datos de los alumnos con los que podrá editar campos y eliminar registros.

9. Backlog

Requisito	Sprint 1
-----------	----------

Compatibilidad con los navegadores más usados (Chrome, Firefox, Edge e Internet Explorer).	Realizado
Página de inicio atractiva.	Realizado (Mejorable)
Menús desplegables.	Realizado
Aula virtual.	Realizado
Log in con dos tipos de usuarios (alumno, profesor).	Realizado
Acceso a archivos (usuario alumno).	Pendiente
Registro de usuarios; subida, edición y lectura de archivos (usuario profesor).	Pendiente
Conexión con la base de datos.	Realizado
Página compuesta por secciones.	Realizado
Texto en varios idiomas	Pendiente
Cierre de sesión.	Realizado
Formulario de contacto.	Realizado
Web responsive.	Pendiente
Cambios de contraseña.	Pendiente
Contraseña olvidada.	Pendiente
Usuarios registrados listados.	Pendiente

10. Descripción del trabajo

Preparación del software a emplear

Antes de comenzar a trabajar con el proyecto y empezar a manejar códigos y documentos se hace necesario comenzar a preparar los softwares que se van a emplear. Como instalarlos, actualizarlos y comprobar que funcionan en nuestro sistema operativo.

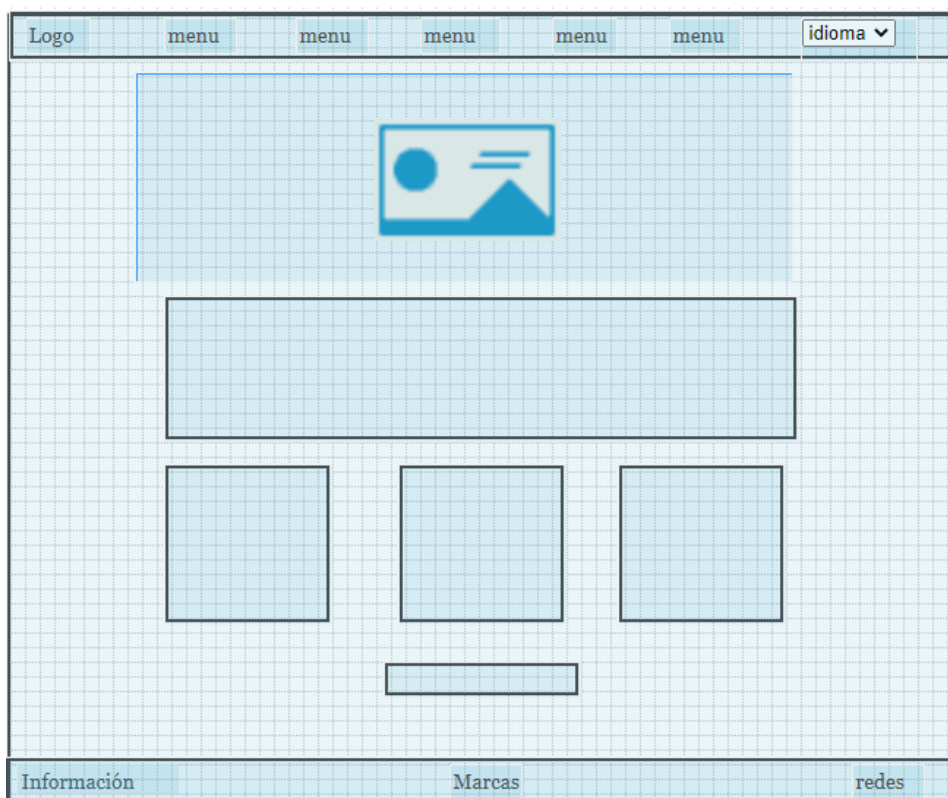
Por lo que antes de comenzar analizo los softwares de los que dispongo, tales como: Atom (para manejar HTML, Javascript, PHP y CSS), servidor Apache instalado con el paquete de XAMPP (este paquete me permite interactuar en un servidor local para comprobar los funcionamientos de códigos y comenzar a montar la web), la base de datos phpmyadmin también administrada por XAMPP y por último Gitbash con

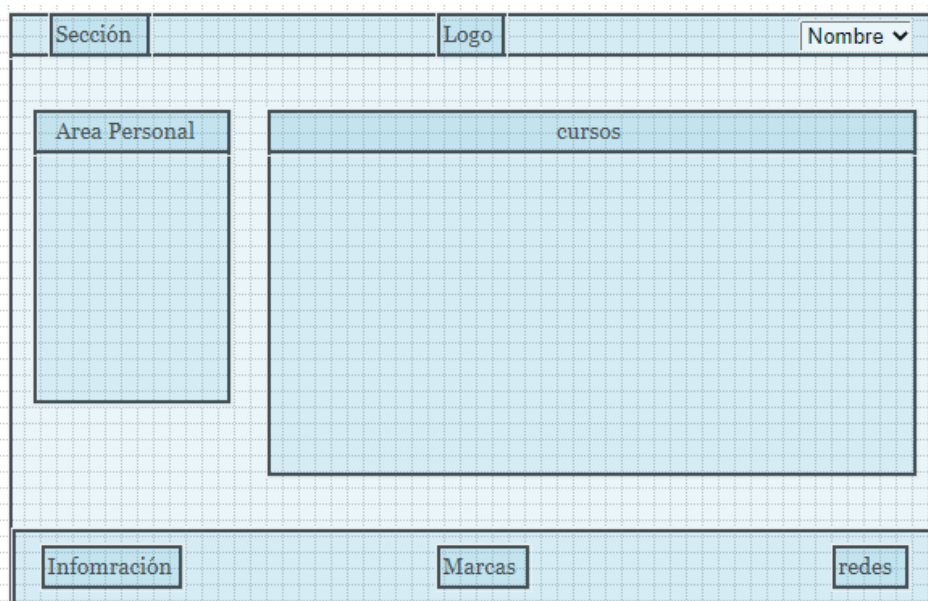
GitHub y git para alojar todo el código del proyecto en un repositorio, en el cuál podrá ser accesible públicamente.

Mockup

Otro aspecto a tener en cuenta antes de comenzar a manejar código, es realizar un diseño base de como va a ser visto el resultado de la web y el aula virtual. Para ellos realizo dos Mockups creados en la web Framebox:

El primero de ellos representa el diseño que va a tener la página principal, en la cual se van a basar las demás secciones y páginas de la web (contactos, sobre nosotros, cursos). Lo único que cambia comparada con las demás secciones de la web es el contenido del texto y la imagen solo aparece en el índice.





Desarrollo de la aplicación

Se trata tanto del desarrollo en la parte de diseño de la página, implementación de la base de datos y la conexión que realiza la aplicación con esta:

Diseño de la web

Se implementa la estructura de la página mediante HTML y el diseño de la misma con CSS (hojas de estilo) que proporciona a la web un diseño atractivo, colorido y con formato. También se emplea Javascript para permitir que el formulario de contacto sea enviado como correo electrónico al correo de la academia, de tal manera, que cualquier usuario que entre en la web podrá enviar un correo electrónico para solicitar información.

Implementación de la base de datos

Se utiliza la base de datos de phpmyadmin, la cual se trata de una base de datos relacional que nos permitirá guardar los datos de los usuarios registrados, para que puedan acceder al log in del aula virtual con los usuarios y contraseñas que se encuentran alejados en la base de datos. Además, cuando podamos acceder con el rol de profesor podremos registrar nuevos usuarios a través de un formulario que alojará nuevos datos en la tabla de alumnos.

Conexión back end con la base de datos

Se realizará en PHP un código que nos permitirá conectar la web con la base de datos. Nos permitirá ingresar con dos roles de usuario distintos: alumnos y profesores, mostrar datos de los usuarios en cada sección de aula virtual y editar, subir o descargar archivos y documentos. En el primer sprint se realiza la conexión con la base de datos, nos permitirá mostrar datos del usuario en la web y nos dará la opción de cerrar sesión. Cada tipo de usuario será redirigido a una página diferente, los alumnos a la página de alumnos y los profesores a la sección de profesores.

Resolución de problemas encontrados

En el primer sprint me he encontrado con dos problemas principales:

- Una pequeña falta de organización con el código HTML que finalmente he solucionado, estructurando cada una de las etiquetas en contenedores (div) principales, que alojarían el contenido correctamente. Esto me ha evitado el problema de que cuando avanzaba en el siguiente código, se produjese algún desajuste en el código anteriormente escrito, como puede ser por ejemplo que se muevan de posición los elementos del HTML y ha sido solucionado con una combinación tanto de divs como del CSS responsive organizado.

- Otra dificultad encontrada ha sido con el inicio de sesión de dos roles de usuario diferente, ya que según se inicie con un correo u otro nos llevará a una sección de la web u otra. El principal problema en esto ha sido que con un rol de usuario si podía iniciar y cerrar sesión correctamente, pero con dos usuarios no me dejaba iniciar sesión. Esto ha sido solucionado con una serie de condicionales if. Si el resultado de la consulta (es decir, el número de filas) es mayor a 0 se inicia sesión con uno de los roles de usuario, mientras el otro no y viceversa.

11. Documentación

Diseño y Front end de la web

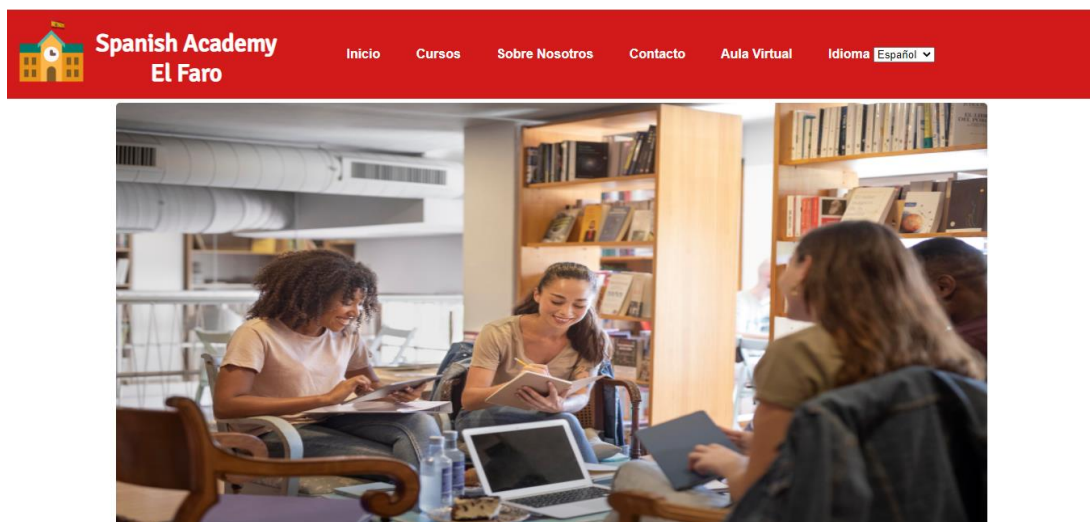
El **HTML** está estructurado en contenedores, para poder distribuir los elementos que lo componen de manera uniforme cada uno de ellos. Aquí un ejemplo de cómo está estructurada la página principal:

```
<body>

<div class="Main-contains-header">

  <header>
    <a href="index.html" class="logo">
      
      <h2 class="company-name">Spanish Academy <br> El Faro</h2>
    </a>
    <nav>
      <a href="index.html" class="nav-link">Inicio</a>
      <a href="courses.html" class="nav-link">Cursos</a>
      <a href="aboutus.html" class="nav-link">Sobre Nosotros</a>
      <a href="contact.html" class="nav-link">Contacto</a>
      <a href="virtual.html" class="nav-link">Aula Virtual</a>
      <a class="languages">Idioma <select name="Idiomas" class="nav-link">
        <option value="">Español</option>
        <option value="">English</option>
        <option value="">Deutsch</option>
        <option value="">Français</option>
      </select></a>
    </nav>
  </header>
</div>
```

El resultado obtenido en la página de inicio de la web:





Las demás secciones de la página web que se indican en el menú del header han sido organizadas y estructuradas de igual manera, empleando contenedores y display flex (css) para la organización de los elementos

Todas las imágenes e iconos han sido descargados de páginas webs tales como: pexels, pixabay y freepng. Las webs mencionadas alojan imágenes e iconos sin derechos de autor.

En cuanto al estilo de la página, ha sido realizado mediante hojas de estilos (css) y se ha seguido una organización flexible, utilizando sobre todo el atributo en los contenedores de display: flex. Esto permite adaptar la pantalla y hacer una web más responsive, donde los elementos se estructuran según el contenido de la pantalla. Esto es un ejemplo de estilos realizados para las imágenes y contenido:


```

9  .main-contain-aboutus{
10  display: flex;
11  align-items: center;
12  justify-content: space-between;
13  margin-left: 10%;
14  margin-right: 10%;
15  margin-top: 40px;
16  }
17  .main-contain-aboutus li{
18  color: black;
19  margin-bottom: 30px;
20  }
21  .main-contain-aboutus ul{
22  list-style-type: "►";
23  }
24
25
26  .contain-description-aboutus{
27  display: flex;
28  align-items: center;
29  justify-content: center;
30  margin-left: 10%;
31  margin-right: 10%;
32  background-color: #E7E7E7;
33  margin-top: 20px;
34  padding: 15px;
35  border-radius: 5px;
36  }

```

Con la hoja de estilos también se pueden realizar animaciones, como cambiar el color de un elemento cuando se coloca el cursor encima de éste. En este caso se utiliza la función `:hover` en css para un elemento dado, que aplique esos cambios en el elemento. Como sucede en el siguiente caso:

El elemento en el momento en el cual todavía no se ha colocado el cursor encima de él:

Más información

Cuando se pasa el cursor por encima del elemento:

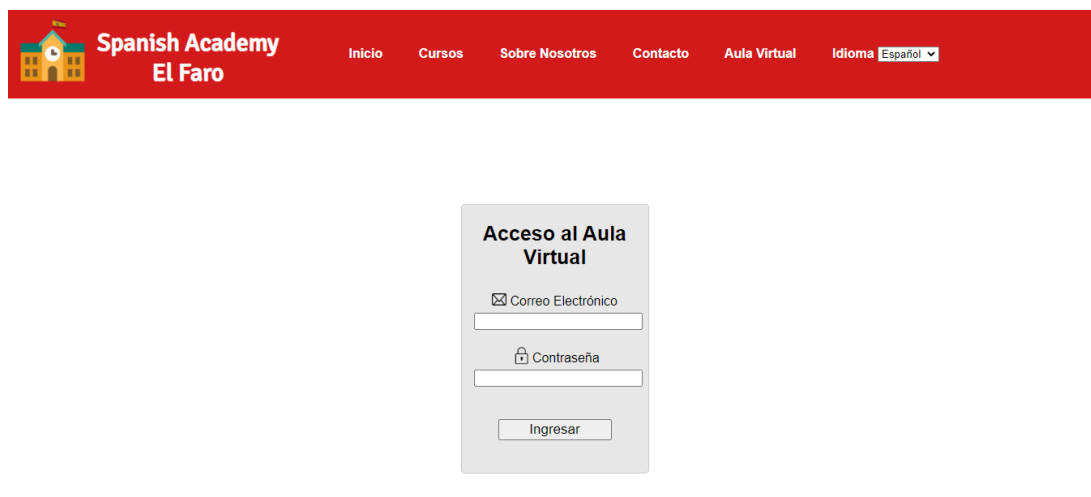
Más información

En cuanto a los enlaces de la web, cada elemento del header enlaza a cada una de las secciones que conforman la web (Inicio, Contacto, Sobre nosotros, Cursos y aula virtual).

Formularios

En el primer script se crean dos formularios:

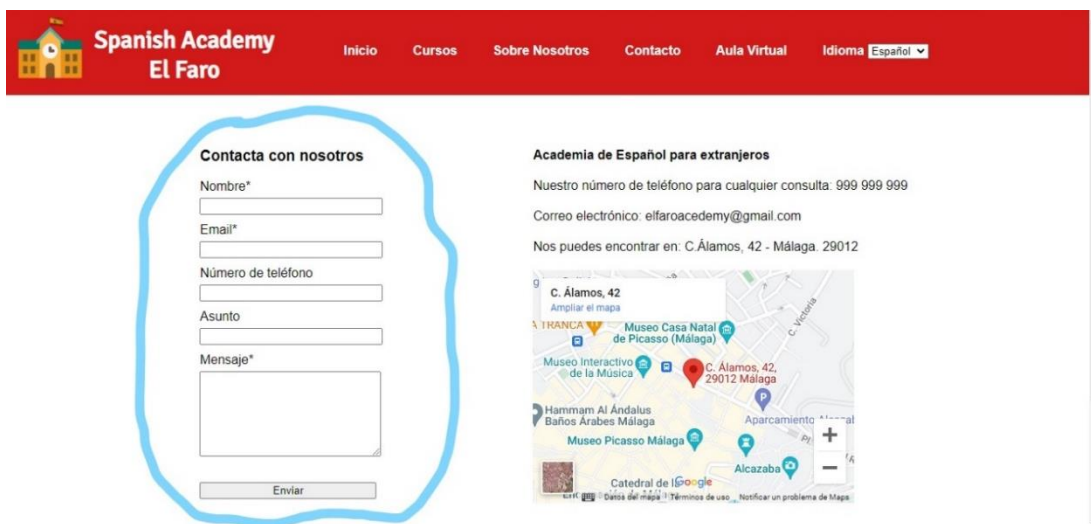
El principal de ellos es el del aula virtual, con el cual se realiza un POST y envía los datos del formulario a una serie de variables que recogen esa información. La estructura del formulario:



The screenshot shows the 'Acceso al Aula Virtual' login form. It features a red header with the 'Spanish Academy El Faro' logo and navigation links: Inicio, Cursos, Sobre Nosotros, Contacto, Aula Virtual, and Idioma (Español). The login form itself is a light gray box with the title 'Acceso al Aula Virtual'. It contains two input fields: 'Correo Electrónico' (with an envelope icon) and 'Contraseña' (with a lock icon). Below these fields is an 'Ingresar' button.

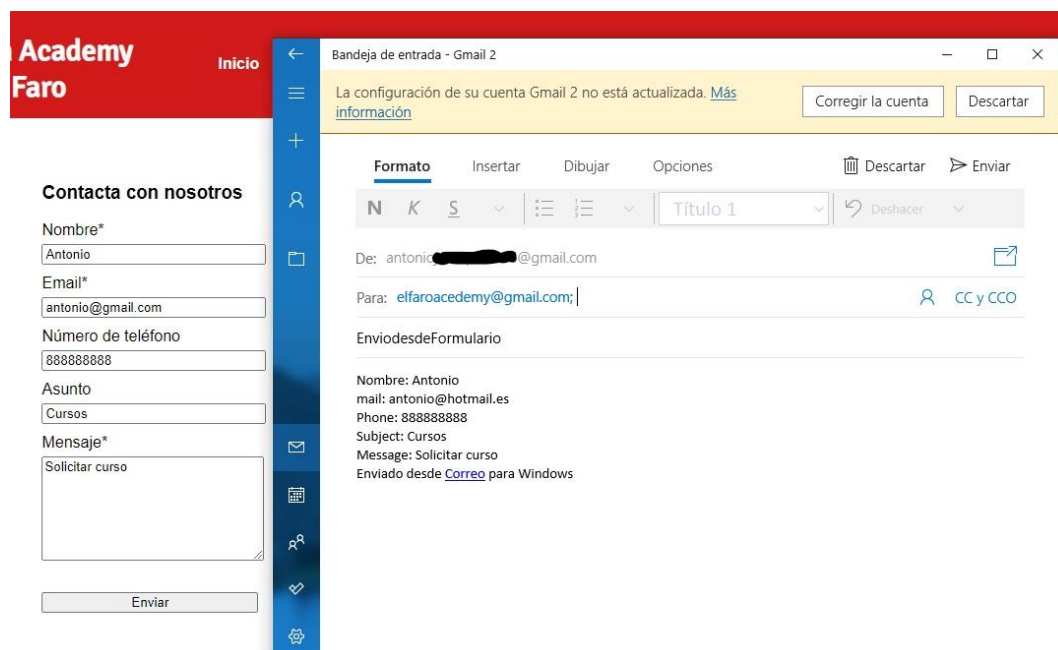
Introduciendo los parámetros correctos nos llevará a una u otra sección del aula virtual, dependiendo del tipo de usuario introducido.

El segundo formulario es utilizado, dentro de la sección de contacto, para enviar un mensaje por correo electrónico al correo de la empresa. Al pulsar el botón enviar nos dirige hacia la web de nuestro email donde, si disponemos de una sesión iniciada en el correo electrónico del cliente redactará directamente el mensaje y destinatario solo con la información introducida en el formulario, hacia un mensaje desde nuestro perfil de correo electrónico; si no tenemos nuestro correo electrónico iniciado, nos indicará que iniciemos sesión previamente. El aspecto del formulario es el siguiente:



The screenshot shows the 'Contacta con nosotros' form and contact information. The form is on the left, with fields for 'Nombre*', 'Email*', 'Número de teléfono', 'Asunto', and 'Mensaje*'. A blue hand-drawn circle highlights the form fields. To the right, the contact information for 'Academia de Español para extranjeros' is displayed, including the phone number '999 999 999', email 'elfaroacademy@gmail.com', and address 'C. Álamos, 42 - Málaga. 29012'. Below the text is a map showing the location of the academy in Málaga, with a red pin marking the address 'C. Álamos, 42, 29012 Málaga'.

Muestro la web completa para que se pueda visualizar el aspecto. El formulario señalado en la imagen es el formulario de contacto mencionado. Al redactar un mensaje y hacer click en enviar aparece lo siguiente:



Los campos rellenados en el formulario se envían al correo electrónico de la empresa como un JSON, como campo y valor.

Javascript

En el primer sprint solo se emplea en una ocasión y es en el caso anteriormente mencionado del formulario de contacto (en los próximos sprint será más utilizado para dar interactividad a la web). Mediante el siguiente script se recoge la información que hay en los formularios (values) que introducimos al escribir en él y estos valores los asocia a los elementos captados como `getelementbyid`, por lo que para esa variable se le asocia un valor.

Esos valores se capturan en una url de envío dentro de un mailto, por lo que escribe el contenido dentro de un mensaje de formulario:

```

const button_send = document.getElementById('send');
button_send.addEventListener('click', function(e) {
  //Prevenir que se actualice la página para que no se actualice de nuevo una vez enviado el formulario.
  //Por lo que el evento puede ser llamado de nuevo.
  e.preventDefault()
  //Se definen las constantes que recogerán el valor del formulario
  const name = document.getElementById('Nombre').value;
  const email = document.getElementById('mail').value;
  const phone = document.getElementById('Phone').value;
  const subject = document.getElementById('Subject').value;
  const message = document.getElementById('Message').value;
  /*Para enviar el formulario al correo electrónico definido en el siguiente código, al cuál se le pasan los parámetros con
  las constantes obtenidas anteriormente*/
  window.location.href = `mailto:elfaroacademy@gmail.com?subject=EnviodesdeFormulario&body=Nombre%3A%20${name}%0Amail%3A%20${email}%0Aphone%3A%20${phone}%0ASubject%3A%20${subject}%0AMessage%3A%20${message}`;
});

```

En el enlace se puede apreciar que se introduce para cada variable (cerrada entre corchetes y que captura el valor de la variable) un título y su valor.

Base de datos

Se emplea la el sistema de gestión de base de datos phpmyadmin que se trata de una base de datos distribuidas (MySQL). En dicha base de datos se alojan los datos de los usuarios que van a ingresar en el aula virtual, de tal forma que mediante el back-end se accederán a estos datos.

Se crea una base de datos a la cuál llamo academia_el_faro y que está compuesta de dos tablas (los dos tipos de usuarios para ingresar en el aula virtual). De momento tiene pocos usuarios. En los siguientes script se alojarán más usuarios.

La tabla alumnos:

				id	Name	Mail	Password	Curso
<input type="checkbox"/>	Editar	Copiar	Borrar	1	Javier Castilla Fuentes	javiercastilla@gmail.com	usuario1	A2
<input type="checkbox"/>	Editar	Copiar	Borrar	3	Marcelo Gonzalez Rodriguez	gonzalezrodriguez@gmail.com	usuario2	B1

Y la tabla profesores con más privilegios:

Back end y PHP

Para realizar la conexión con la base de datos se emplea el lenguaje de programación del lado del servidor PHP, que nos permitirá realizar todos los ajustes del back end.

Primeramente, en el primer sprint se llevan a cabo la conexión con la base de datos, el inicio de sesión de dos tipos de usuarios y el cierre de sesión de dichos usuarios.

Conexión del aula virtual con la base de datos

Antes de nada, se debe de conectar el código con el que vamos a interactuar y la base de datos donde se encuentran alojados los datos. Para ello se crea el objeto de conexión, se introducen los datos de nuestra base de datos creada en phpmyadmin y posteriormente se enviará un mensaje de error en caso de que la conexión no se realice con éxito.

```

conexion.php
1  <?php
2  //Se crea el objeto de conexión ("localhost", "usuario", "contraseña", "nombre de base de datos creada en phpmyadmin")
3  $mysqli = new mysqli("localhost", "root", "", "academia_el_faro");
4  //Se utiliza la estructura condicional if else para saber si se ha realizado la conexión con la BD con éxito.
5  if ($mysqli->connect_errno) {
6      //Si existe algún tipo de error al conectar con la base de datos aparece el siguiente mensaje
7      echo "Fallo al conectar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->connect_error;
8  }else {
9
10 }
11 ?>

```

Inicio de sesiones según usuario

Según iniciemos sesión con el usuario de una tabla o con el de otra, estás nos redirigirán a una sección u otra del aula virtual. Para ello se crearán dos variables que recibirán como valor el introducido en el formulario, de tal manera que con la super global: \$_POST y el formulario con método post y el valor de action, alojará los valores en las variables del archivo de sesiones.

Luego se utilizan otras variables para almacenar el resultado de las consultas de datos de la base de datos y entonces indico que si el valor de la base de datos coincide con el valor que se le pasa a la variable del post, inicie sesión mediante la super global \$_Session.

```

<?php
//Se incluye el objeto de conexión.
include('conexion.php');

//Recibimos las variables del formulario Post.
$Mail = $_POST["Mail"];
$Password = $_POST["Password"];

//Se realizan unas consultas en la base de datos para verificar los usuarios que tienen permitido acceder.
$alumnos = $mysqli->query("SELECT * FROM alumnos WHERE Mail = '$Mail' AND Password = '$Password'");
$profesores = $mysqli->query("SELECT * FROM profesores WHERE Mail = '$Mail' AND Password = '$Password'");

/*Ahora se comprueban las variables que tienen los usuarios y profesores y para cada tipo de usuario, se accede a un documento distinto*/
//Con este if se verifica que el que intenta acceder es un alumno.
if(mysqli_num_rows($alumnos) > 0){

    session_start();
    $_SESSION['alumnos'] = "$Mail";

    //Con header nos redirigirá al documento deseado, en este caso la sección de alumnos del aula virtual.
    header("Location: Alumnos.php");
    exit();

    //Y para verificar que el que ingresa es el profesor.
}elseif (mysqli_num_rows($profesores) > 0) {

    session_start();
    $_SESSION['profesores']="$Mail";

    //Nos redirige a la sección de profesores.
    header("Location: Profesores.php");
    exit();

    //Si no se encuentra ni al usuario registrado como profesor ni como alumno, que aparezca el siguiente mensaje.
}else {
    header("Location: templates/UsuarioNoEncontrado.html");
}

?>

```

Entonces indico mediante un condicional if que si el número de filas de las consultas realizadas es mayor a 0 inicie sesión y por lo tanto en la otra consulta como será 0 no iniciará sesión. De esta manera se indica que cuando inicie un usuario sesión dirija hacia una página u otra según el caso que se dé.

Mostrar datos en web

Para mostrar los datos de los usuarios que han iniciado sesión, se mantiene la sesión iniciada en las páginas que redirige la sesión donde los usuarios se inician. Entonces se deja la sesión iniciada mediante `session_start` y mediante un condicional if indico que si el usuario no coincide con el usuario de la sesión, ésta se cierre. Para mostrar datos se utiliza un for que recoge los datos de la consulta del usuario iniciado y los muestra mediante un echo indicado en las etiquetas html respectivas.

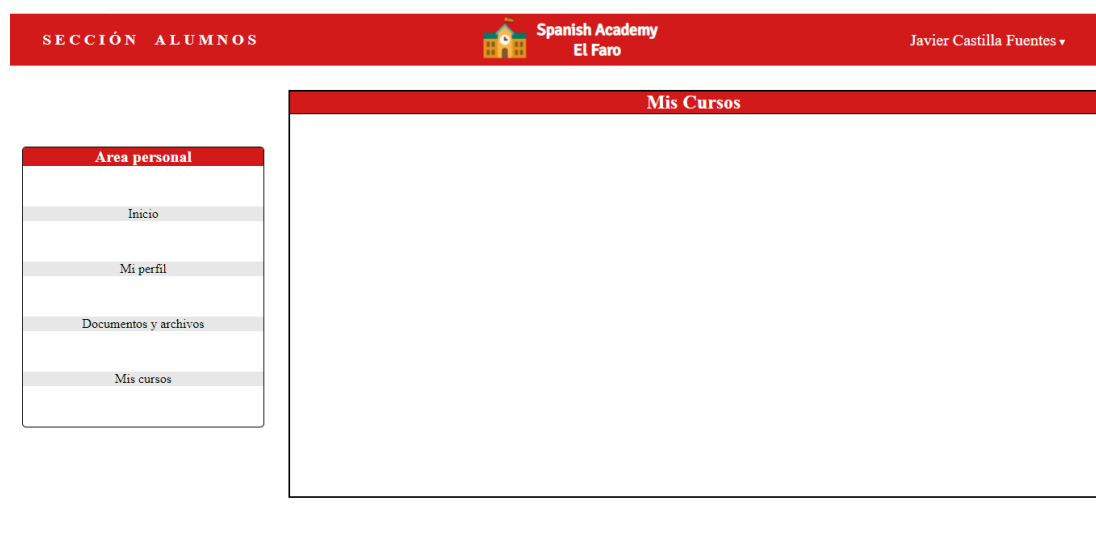
Muestra de código de usuario alumno:

```

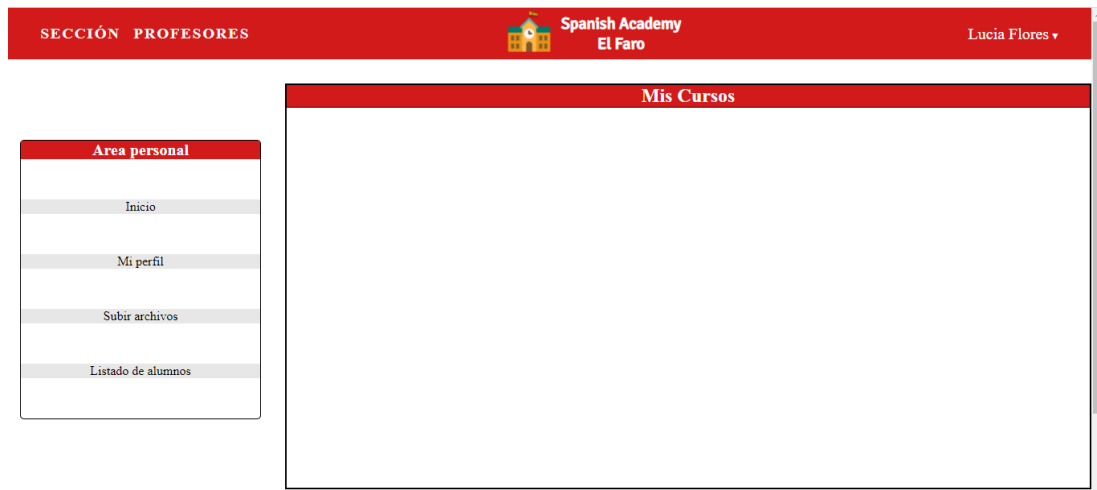
8  <body>
9
10  <?php
11  include('conexion.php');
12
13  //Se abre la sesión del alumno.
14  session_start();
15
16  //Si no existe la variable de la sesión del alumno nos regresará al formulario del aula virtual indi
17  if (!isset($_SESSION['alumnos'])) {
18      header("location: templates/virtual.html");
19  }
20
21  //Se realiza una consulta mysql a la base de datos para volcar la información del usuario con la ses
22  $alumnos = $mysqli->query("SELECT * FROM alumnos WHERE Mail = '$_SESSION[alumnos]' ");
23
24  //Se crea un bucle para guardar la información de la consulta en un array asociativo que mostrará la
25  for ($i=0; $i < $alumnos->num_rows; $i++) {
26      $fila=$alumnos->fetch_assoc();
27
28  }
29  ?>
30
31
32  <div class="contain-header">
33
34  <div class="virtual-tittle">
35  <h3>SECCIÓN ALUMNOS </h3>
36

```

Resultado obtenido:



En ella se muestra el diseño obtenido en la sección. Con el nombre del usuario que ha iniciado sesión en la esquina superior derecha. En el caso de la sección profesores muestra el siguiente resultado:



Cerrar la sesión

Finalmente, para cerrar la sesión se emplea en la web un elemento que contiene un enlace al archivo php que cierra sesión, de manera que cuando es pulsado se cierra sesión, ya que el archivo contiene la sesión iniciada pero si es pulsado se invocan los métodos que destruyen la sesión (`session_unset` y `session_destroy`) y nos dirigen a la página de inicio de sesión nuevamente (mediante `header` y la url).

Ejemplo cierre de sesión del alumno:

```
<?php
include('conexion.php');

session_start();

//Se comprueba que es el usuario que ha ingresado quien se encuentra activo en la sesión para finalizarla.
if (!isset($_SESSION['alumnos']))
{
    //Al lugar donde redireccionará si no hay una sesión activa
    header("location: templates/virtual.html");
}

//Para liberar la sesión que se encuentra activa.
session_unset();

//Para destruir la sesión.
session_destroy();

//Lugar que redireccionará cuando se finalice la sesión.
header("location: templates/virtual.html"); ?>
```

12. Retrospectiva

El tiempo me ha dificultado realizar otros aspectos de diseño que implementaré en los siguientes sprints. También habría estado más que bien emplear una plantilla de diseño previa que mostrará como quedaría el diseño completo de HTML con CSS. Además

hay otros aspectos que podría haber implementado ya en el primer sprint como pueden ser que el usuario profesor añada alumnos y suba archivos. Por último, añadir más elementos a las secciones de alumno y profesor respectivamente primeramente y haberle dedicado más tiempo al diseño del aula virtual, ya que he dedicado demasiado tiempo en este sprint al diseño y maquetación de toda la web.

13. Spring 2 Backlog

Requisito	Sprint 1
Compatibilidad con los navegadores más usados (Chrome, Firefox, Edge e Internet Explorer).	Realizado
Página de inicio atractiva.	Realizado (Mejorable)
Menús desplegables.	Realizado
Aula virtual.	Realizado
Log in con dos tipos de usuarios (alumno, profesor).	Realizado
Acceso a archivos (usuario alumno).	Realizado
Registro de usuarios; subida, edición y lectura de archivos (usuario profesor).	Realizado
Conexión con la base de datos.	Realizado
Página compuesta por secciones.	Realizado
Texto en varios idiomas	Pendiente
Cierre de sesión.	Realizado
Formulario de contacto.	Realizado
Web responsive.	Pendiente
Cambios de contraseña.	Realizado
Contraseña olvidada.	Realizado
Usuarios registrados listados.	Realizado

14. Descripción del trabajo Sprint 2

Desarrollo de la aplicación

En el segundo sprint me he centrado sobre todo en el desarrollo de las funcionalidades back end, dejando para el último sprint tareas orientadas más a mejorar el diseño de la misma.

Recuperar contraseña

Permite al usuario recuperar la contraseña introduciendo el correo electrónico con el que se le registró, si el correo es incorrecto envía un mensaje de error a la página indicando que el correo introducido es erróneo y si el correo introducido es correcto se realiza un envío de la contraseña de la que dispone a dicho correo electrónico que ha sido introducido por el usuario. Puede realizar esta acción los dos tipos de usuarios existentes

Editar perfiles de usuarios

Los dos tipos de usuarios tendrán la posibilidad de editar los datos que están alojados en la base de datos mediante un formulario. En dicho formulario tendrán la opción tanto de cambiar el correo y la contraseña. Si los datos de usuarios introducidos son los mismo que los que se encuentran alojados en la base de datos, se mostrará un mensaje de que los datos ya existen y por lo tanto deberá de introducir otros datos diferentes a los existentes y, en el caso contrario se enviará un mensaje de que se han modificado los datos en la base de datos.

Mostrar datos

Se mostrarán los datos de los usuarios que han iniciado sesión en el aula virtual. Se muestran datos como el nombre, el curso en el que están registrados o el email.

Listado de alumnos

El usuario con rol de profesor tendrá a su disposición un listado con los datos de los usuarios que se encuentran registrados y, además el número de alumnos. Tendrá la opción de eliminar alumnos mediante un botón de eliminar y podrá registrar nuevos alumnos mediante un formulario. Si registra un usuario con un correo ya existente, lo indicará mediante un mensaje. Si se registra un nuevo usuario aparecerá también un mensaje de usuario registrado con éxito.

Subida y lectura de archivos

Los usuarios con rol de profesor tendrán la opción de subir archivos (con formato .jpg y .pdf) de tal manera que los usuarios con rol de alumno podrán visualizarlos una vez inicien sesión. Los usuarios profesor podrán visualizar dichos archivos y tendrán la opción de eliminarlos también. Dichos archivos dependiendo de la opción que se marque (por ejemplo, opción de subir archivo a sección de comprensión auditiva) serán subidos a una sección de la página u otra. No se podrá subir el mismo archivo duplicado.

Los alumnos solo podrán visualizar los archivos de su curso. Por ejemplo, si el usuario pertenece al curso A2 no podrá visitar la página que contiene archivos de otros cursos, en cambio el usuario profesor podrá visitar todos los cursos existentes y subir archivos en dichos cursos.

Empleo de Javascript para mensajes de alerta

Se emplea Javascript para crear funciones que permitan, por ejemplo, que pulsando un botón se envía un mensaje de alerta que presionando aceptar se active una función y pulsando en cancelar no se realice ninguna acción.

Resolución de problemas encontrados

Siguientes problemas encontrados en sprint 2:

- Problemas al subir archivos y alojarlos directamente en la página web. Debido a este problema decidí alojar dichos archivos en carpetas que se encuentran en el servidor con php y después para visualizar dichos archivos mediante las funciones php: opendir y readdir y luego un enlace que hacía coincidir el final de la página con el nombre del archivo u carpeta, un ejemplo es en el enlace: `".$archivo4."` Donde archivo4 es igual al nombre del archivo y de esta manera se puede visualizar correctamente.

- Otro problema encontrado con relación al anterior ha consistido en que dichos archivos subidos fuesen dirigidos a alumnos de un curso u otro respectivamente, es decir, los archivos de un alumno del curso B1 podía visualizar los del curso B2. Este

problema ha sido solventado mediante condicionales if/elseif donde, dependiendo del usuario, podría acceder a unos cursos u otros (y por lo tanto a dichos archivos)

- Por último, para comentar un típico fallo común es el de olvidar colocar un punto y coma al final de una función y que todo el código se venga abajo y no encontrar el problema, hasta que revisas el código y realizas las pruebas unitarias pertinentes donde aparece dicho error tan común.

15. Documentación Sprint 2

Como se ha comentado, en este segundo sprint, el trabajado se ha centrado más en realizar todas las conexiones correctamente y todas las funcionalidades del aula virtual, dejando el diseño para el tercer sprint.

Recuperar contraseña

Se implementa la funcionalidad de recuperar la contraseña en caso de que los usuarios (para ambos tipos de usuario) hayan olvidado la contraseña con la cuál fueron registrados. Para dicha funcionalidad se realiza un formulario que recoge el correo electrónico mediante un post y envía dicha información a una serie de variables que permiten consultar a la base de datos y realizar un for each para recorrer dichos datos:

```

33     if(isset($_POST['submit'])){
34
35         $mail = $_POST['Mail'];
36
37         $notfound = "";
38
39         $email_exist = $mysqli->query("SELECT * FROM `alumnos`,`profesores`");
40         $email_query_alumnos = $mysqli->query("SELECT Password FROM `alumnos` WHERE Mail = '$mail'");
41         $email_query_profesores = $mysqli->query("SELECT Password FROM `profesores` WHERE Mail = '$mail'");
42
43
44         for ($y=0; $y < $email_exist->num_rows; $y++) {
45             $exist_email = $email_exist->fetch_assoc();
46         }
47     }

```

Dependiendo de los resultados obtenidos, se obtienen una serie de fila de resultados (se comprueba si en las variables de las consultas hay resultados) y dependiendo de un usuario u otro se accederán a las contraseñas de unos tipos de usuarios u otros. Si existe el correo en la base de datos, se realiza un envío por email de la contraseña que se encuentra en dicha base de datos. Este envío se realiza mediante la función email de php, en la cuál se añaden una serie de datos(el correo electrónico que envía dicha

información, el título y el contenido del mensaje, que envía una variable con la contraseña de la base de datos de dicho usuario).

Para antes de realizar dicha acción debía de modificar el archivo de php.ini, en la cual se han modificado información importante:

El protocolo para transferencia de correo (SMTP) y la variable sendmail_from en la cuál introduzco la dirección del correo desde donde quiero que se mande el mensaje.

```

if (mysqli_num_rows($email_query_alumnos) > 0) {

    for ($i=0; $i < $email_query_alumnos->num_rows; $i++) {
        $Password_alumnos=$email_query_alumnos->fetch_assoc();
    }
    $content_alumnos = "Su contraseña es:: ".$Password_alumnos['Password'];
    mail($mail,"Recuperación de contraseña",$content_alumnos);
    $message_password_alumnos = "Contraseña enviada al correo introducido";

}elseif (mysqli_num_rows($email_query_profesores) > 0) {

    for ($i=0; $i < $email_query_profesores->num_rows; $i++) {
        $Password_profesores=$email_query_profesores->fetch_assoc();
    }
    $content_profesores = "La contraseña con la cuál se registró es: ".$Password_profesores['Password'];
    mail($mail,"Recuperación de contraseña",$content_profesores);
    $message_password_profesores = "Contraseña enviada al correo introducido";

}elseif(!empty($mail)){
    $notfound = "Correo no encontrado";
}else{
}
}

```

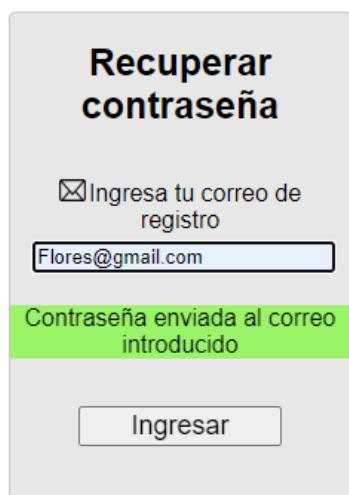
Aspecto:

Acceso al Aula Virtual

☐ Correo Electrónico

Contraseña

[¿Olvidaste la contraseña?](#)



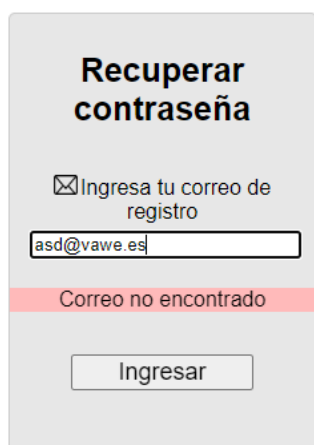
Recuperar contraseña

✉ Ingresa tu correo de registro

Flores@gmail.com

Contraseña enviada al correo introducido

Ingresar



Recuperar contraseña

✉ Ingresa tu correo de registro

asd@vawe.es

Correo no encontrado

Ingresar

Editar perfiles de usuario

Los usuarios pueden modificar la información de sus cuentas (correo y contraseña). Se realiza mediante la instrucción SQL UPDATE, en la cual se actualizan los datos de la base de datos. En php se recogen las variables de una consulta y luego se actualizan con la siguiente consulta query que modifica los datos al pulsar el botón de submit del formulario.

```
?php

if (isset($_POST['edit'])) {
    $email_update = $_POST['user_email'];
    $password_update = $_POST['user_password'];

    $sql_db = $mysqli->query("SELECT * FROM profesores WHERE Mail ='$email_update' AND Password = '$password_update'");
    $exist = $sql_db->num_rows;

    if($exist > 0){
        $update_exist = "<p>Ya existen los datos introducidos</p>";
    }else{

        $db_edit = "UPDATE profesores SET Mail='$email_update', Password='$password_update' WHERE id=".$_fila['id'];
        if (mysqli_query($mysqli, $db_edit)) {
            $update = "<p>Datos actualizados</p>";
        }
    }
}
```

Si los datos que se quieren modificar ya existen se muestra el siguiente mensaje:

Editar perfil

Cambiar email

javiercastilla@gmail.com

Cambiar contraseña

.....

Guardar cambios

Ya existen los datos introducidos

Si los datos introducidos son diferentes a los de la base de datos se realiza la modificación y se muestra el siguiente mensaje:



Al abrir la página en la cual se encuentra dicho formulario, los datos ya se encuentran alojados en los input, por lo que el usuario ya aprecia cuáles son los datos introducidos y solo necesita modificar el contenido que ya aparece si lo desea.

Datos del usuario visibles en partes de la web

Cuando los usuarios inician sesión dispondrán de un área personal en la cuál se muestran los datos de los usuarios. Dichos datos se muestran con una consulta query a la base de datos mediante PHP, en la cuál mediante bucle for each se recorre los datos de los usuarios y con variables se permiten mostrar en la web:

```

nav class= menu-personal-area >
<ul>
  <div class="main-title-nav"><li class="tittle-nav">Area personal</li>
  <a href="/ProyectoFinal/AlumnosPHP/Alumnos.php"><li><b>Inicio</b></li>
  <li><b>Mi perfil</b></li>
  <ul class="submenu-personal-area">
    <li><b>Nº expediente alumno:</b> <?php echo $fila['id']; ?></li>
    <li><b>Alumno:</b> <?php echo $fila['Name']; ?></li>
    <li><b>Email:</b> <?php echo $fila['Mail']; ?></li>
    <li><b>Cursos:</b> <?php echo $fila['Curso']; ?></li>
  </ul>
  </ul>

```

Resultado obtenido:

Area personal
Inicio
Mi perfil
Nº expediente alumno: 1
Alumno: Javier Castilla Fuentes
Email: javiercastilla@gmail.com
Cursos: A2
Documentos y archivos

Listado, registro y eliminación de alumnos

Para listar los alumnos se emplea el mismo método que en el apartado anterior donde se realiza una consulta sql y se muestran los datos en una tabla (<table>) utilizando un bucle for each para recorrer uno a uno cada uno de los datos de los usuarios que se desean mostrar.

La consulta empleada: \$alumnos = \$mysqli->query("SELECT * FROM alumnos");

El for each que recorre los datos y los muestra en celdas de la tabla:

```
<div class="alumnos-list">
  <div class="main-alumnos-list">
    <table>
      <tr>
        <th>Nº alumno</th>
        <th>Nombre</th>
        <th>Email</th>
        <th>Curso</th>
        <th>Acciones</th>
      </tr>
    </table>
  </div>
</div>
<?php
for ($i=0; $i < $alumnos->num_rows; $i++) {
    $fila_table=$alumnos->fetch_assoc();
    echo "<tr><td>".$fila_table['id']. "</td>";
    echo "<td>".$fila_table['Name']. "</td>";
    echo "<td>".$fila_table['Mail']. "</td>";
    echo "<td>".$fila_table['Curso']. "</td>";
    echo "<td> <a href='delete_alumno.php?id=".$fila_table['id']. "</td>";
}
}
```

El resultado obtenido:

Nº alumno	Nombre	Email	Curso	Acciones
1	Javier Castilla Fuentes	javiercastill@gmail.com	A2	Eliminar
3	Marcelo González Rodríguez	gonzalezrodriguez@gmail.com	B1	Eliminar
4	Antonio Perea	antoniojesusperea96@gmail.com	B2	Eliminar
13	Kevin Jhonson	JhonsonK@gmail.com	B1	Eliminar
20	Anne Dubois	DuboisAnne@hotmail.es	B2	Eliminar

También dispone de una opción para eliminar los usuarios. Al hacer clic en eliminar, nos dirige a un enlace que se trata de un archivo PHP que realiza una función de eliminación de usuarios según la id del usuario. Como en el for each se recorre los usuarios se muestra en un lateral la opción de eliminar que contiene el id de cada usuario que va siendo mostrado en la tabla. Dicho id se envía a través de la url del documento: `echo ("<td> Eliminar</td></tr>");`.

Dentro del código de dicho archivo de eliminación se crea un **script** para redireccionar a la página en la cual se listan los usuarios, para que no nos envíe a un documento vacío (página en blanco, pero con código php).

Se realiza una consulta query con instrucción DELETE que permite eliminar a los usuarios:

```

1 <?php
2 include('../conexion.php');
3
4 session_start();
5
6 if (!isset($_SESSION['profesores'])){
7     header("location: /ProyectoFinal/templates/virtual.html");
8 }
9
10 $alumn_id = $_GET['id'];
11
12 $delete_query = mysqli_query("DELETE FROM alumnos WHERE id = '$alumn_id'");
13
14 if ($delete_query) {
15     header("location: /ProyectoFinal/ProfesoresPHP/ListadoAlumnos.php");
16     echo "<script>alert('No se pudo eliminar el alumnos seleccionado');</script>";
17 }else{
18     echo "<script>alert('No se pudo eliminar el alumnos seleccionado'); window.history.go(-1);</script>";
19 }
20 ?>

```

En la misma página se dispone de un formulario que permite registrar usuarios, de tal manera que al completar los campos del formulario nos registrará un nuevo usuario en

nuestra base de datos. Dicha acción se realiza mediante un query con función INSERT INTO que nos permite introducir valores nuevos en la base de datos. Los valores a introducir se recogen con formulario comentado anteriormente con método POST y lleva ese valor a unas variables en PHP que permite insertar los datos en la base de datos:

```
<?php
if(isset($_POST['Registrar'])) {
    $name = $_POST['Name'];
    $pass = $_POST['Password'];
    $email = $_POST['Mail'];
    $course = $_POST['Curso'];

    $email_exist = $mysqli->query("SELECT * FROM alumnos WHERE Mail = '$email'");

    if($_POST['Name'] == '' || $_POST['Password'] == '' || $_POST['Mail'] == '' || $_POST['Curso'] == '') {
        $Unfilled = 'Por favor rellene todos los campos.';
    } elseif($email_exist->num_rows >= 1) {
        $same_user = "Usuario ya existe";
    } else {
        $registry = $mysqli->query("INSERT INTO `alumnos` (`Name`, `Mail`, `Password`, `Curso`) VALUES ('$name",
        $user_registry = "Alumno registrado exitosamente";
    }
} else {
    echo "";
}
```

Si el correo electrónico coincide con el de algún otro alumno ya registrado, nos devuelve un mensaje de usuario ya registrado. Ya que no puede haber dos usuarios con el mismo correo electrónico:



Registrar nuevo alumno

Nombre
Usuario

Contraseña
....

Email
javiercastill@gmail.com

Curso
A2 ▼

Usuario ya existe

Registrar

Si el correo del usuario que se va a registrar no existe en la base de datos, nos devuelve un mensaje de usuario registrado, nos almacena los datos en la base de datos y nos muestra el usuario en la tabla:



El formulario tiene un encabezado rojo con el texto "Registrar nuevo alumno". Contiene los siguientes campos:

- Nombre:** Un campo de texto con el valor "Usuario".
- Contraseña:** Un campo de texto con cuatro puntos para ocultar el contenido.
- Email:** Un campo de texto con el valor "Usuarioprueba@gmail.es".
- Curso:** Un menú desplegable con "A2" seleccionado.

Debajo de los campos, hay un mensaje "Alumno registrado exitosamente" y un botón "Registrar".

Solo podrá ser realizado por el usuario profesor.

Subida y lectura de archivos

Los usuarios con rol de profesor podrán subir, eliminar y visualizar archivos. Los usuarios con rol de alumno solo podrán visualizar los archivos.

Para la subida de archivos se emplea un formulario con un campo tipo file que permite subir archivos. Creo un formulario para alojar archivos en páginas diferentes según el curso al que se quieran subir dichos archivos. Así que tomando de referencia por ejemplo la página del curso de A2. Dicho formulario seleccionará la sección de la página a la que se subirá el archivo. Dicha selección (<select>) se recoge mediante una variable donde se almacena el archivo mediante post del formulario. Dependiendo de si el archivo se sube exitosamente, no se sube o el formato del archivo no es válido se muestra un mensaje u otro.

```

</div>
<div class="content-form-file">
  <form action="" method="post" enctype="multipart/form-data">
    <h3>Subir Archivo</h3>
    <input type="file" name="file"><br><br>
    <select class="" name="select-skill">
      <option value="Comprensión-auditiva">Comprensión auditiva</option>
      <option value="Comprensión-lectora">Comprensión lectora</option>
      <option value="Producción-escrita">Producción escrita</option>
      <option value="Producción-oral">Producción oral</option>
    </select>
    <input type="submit" name="upload-button" value="Subir Archivo">
    <p><br> <?php
      if (isset($archivosubido)) {
        echo $archivosubido;
      }elseif(isset($error_archivo)){
        echo $error_archivo;
      }elseif(isset($Notaccept)){
        echo $Notaccept;
      }else{

```

Al hacer clic en subir archivo, se nos abre una ventana que nos permitirá escoger un archivo de nuestro equipo.

Aspecto del formulario:

Se crea una variable que recoge solo dos tipos de formato de archivos (jpg y pfd). Si se intenta subir cualquier archivo con otro tipo de formato nos muestra el siguiente mensaje:

Dicha variable luego es ubicada en un if que nos permite indicar que si el archivo contiene dichos formatos, nos muestre el mensaje de error. Esto es posible indicarlo gracias a substr (que nos permite obtener la cadena de caracteres con el nombre del archivo) y strrpos (que nos permite devolver los caracteres indicados a partir de lo indicado, que en este caso es el un punto), de tal manera que nos analiza un archivo a partir de su extensión. Es decir, si el archivo es archivo.pdf, nos analiza el nombre a partir del punto y así poder compararlo con la variable donde tenemos los formatos permitidos:

```
$formats = array('.jpg', '.pdf');
if (isset($_POST['upload-button'])) {
    $name_file = $_FILES['file']['name'];
    $TMPname_file = $_FILES['file']['tmp_name'];
    $extension = substr($name_file, strrpos($name_file, '.'));
    $skills = $_POST['select-skill'];

    if (in_array($extension, $formats)) {
        if($skills == "Comprensión-auditiva"){
            if (move_uploaded_file($TMPname_file, "../archivosA2/ComprensiónAuditiva/$name_file")) {
                $archivosubido = "Archivo subido correctamente";
            }else{
                $error_archivo = "Error al subir archivo";
            }
        }elseif($skills == "Comprensión-lectora"){
            if (move_uploaded_file($TMPname_file, "../archivosA2/ComprensiónLectora/$name_file")) {
                $archivosubido = "Archivo subido correctamente";
            }else{
                $error_archivo = "Error al subir archivo";
            }
        }elseif($skills == "Producción-escrita"){
            if (move_uploaded_file($TMPname_file, "../archivosA2/ProduccionEscrita/$name_file")) {
                $archivosubido = "Archivo subido correctamente";
            }
        }
    }
}
```

El usuario profesor podrá visualizar los archivos que ha subido en dichas secciones. Los archivos son almacenados en directorios, desde los cuales se podrán mostrar a los usuarios empleando funciones tales como opendir (para abrir el directorio del archivo) y readdir que devuelve el nombre del archivo, que se encuentra en dicho directorio. Mediante un while se recorre archivo por archivo del directorio, por lo que se mostrará todo lo que esté alojado en dicha carpeta. Además, se le añade a cada archivo un enlace, donde podrá ser visualizado desde el navegador. Se añade un enlace con el directorio y al final la variable que contiene el nombre del archivo:

```

<div class="main-Courses">
  <div class="main-content-courses">
    <div class="tittle-main-Courses"><h2>Nivel: A2</h2></div>
    <div class="A2-content-courses">
      <div class="file-content">
        <h3>COMPRESIÓN AUDITIVA</h3>
        <?php $dir1 = "C:/xampp/htdocs/ProyectoFinal/archivosA2/ComprensiónAuditiva";
        if($folder1 = opendir($dir1)){
          while ($archivo1 = readdir($folder1)) {
            if($archivo1 != "." && $archivo1 != ".."){
              echo "&nbsp;<a href='/ProyectoFinal/archivosA2/ComprensiónAuditiva/$archivo1'>".$archivo1."</a>&nbsp;";
            }
          }
        }
      }
    }
  }
}
?>

```

Resultado obtenido:

<u>COMPRESIÓN AUDITIVA</u>
-A2_MODELO_0_v2.pdf >Eliminar<
<u>COMPRESIÓN LECTORA</u>
-A2_MODELO_0_v2.pdf >Eliminar<
<u>PRODUCCIÓN ESCRITA</u>
-A2_MODELO_0_v2.pdf >Eliminar<
<u>PRODUCCIÓN ORAL</u>
-A2_MODELO_0_v2.pdf >Eliminar<

El botón eliminar nos direcciona a un enlace PHP desde el cual habrá una función que elimine el archivo. Dicha función PHP es unlink que permite eliminar archivos del directorio y por lo tanto al eliminarlos del directorio, ya no se muestran, tanto al profesor como al usuario con solo permisos de lectura:

```

<?php
$file = $_GET['item'];
if (!unlink($file))
{
    echo ("Error deleting $file");
}
else
{
    header("Location: /ProyectoFinal/ProfesoresPHP/ProfesoresA2.php");
}

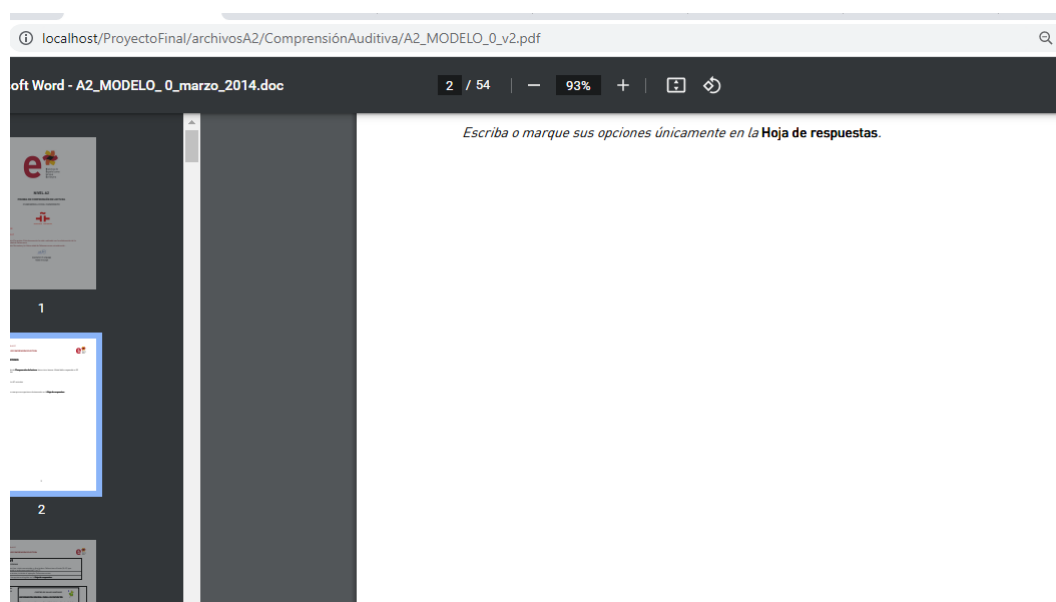
?>

```

Como se muestra al usuario alumno:

Nivel: A2	
<u>COMPRENSIÓN AUDITIVA</u>	-A2_MODELO_0_v2.pdf
<u>COMPRENSIÓN LECTORA</u>	-A2_MODELO_0_v2.pdf
<u>PRODUCCIÓN ESCRITA</u>	-A2_MODELO_0_v2.pdf
<u>PRODUCCIÓN ORAL</u>	-A2_MODELO_0_v2.pdf

Si hacemos clic en alguno de los archivos mostrados:



16. Retrospectiva Sprint 2

Debido a la falta de experiencia en el trabajo con los archivos en php, se me ha hecho difícil desarrollar esta parte, pero finalmente y paso a paso se han podido solventar los errores que he podido ir encontrando. Debido a la gran cantidad de funcionalidades de las que dispone el aula virtual, me he visto en la situación de tener que dejar la parte del diseño para el último sprint, dejando todas las funcionalidades del aula virtual listas.

17. Backlog Sprint 3

Requisito	Sprint 1
Compatibilidad con los navegadores más usados (Chrome, Firefox, Edge e Internet Explorer).	Realizado
Página de inicio atractiva.	Realizado
Menús desplegables.	Realizado
Aula virtual.	Realizado
Log in con dos tipos de usuarios (alumno, profesor).	Realizado
Acceso a archivos (usuario alumno).	Realizado
Registro de usuarios; subida, edición y lectura de archivos (usuario profesor).	Realizado
Conexión con la base de datos.	Realizado
Página compuesta por secciones.	Realizado
Texto en varios idiomas	Realizado
Cierre de sesión.	Realizado
Formulario de contacto.	Realizado
Web responsive.	Realizado
Cambios de contraseña.	Realizado
Contraseña olvidada.	Realizado
Usuarios registrados listados.	Realizado

18. Descripción del trabajo sprint 3**Enfoque front-end**

En el último sprint se ha trabajado sobre todo aspectos de diseño de la web y del aula virtual, para hacer la web más atractiva y dinámica. Se han empleado sobre todo código en css y Javascript.

Además, se ha incluido un menú responsive en el caso que el usuario no pueda acceder desde una computadora o las dimensiones de la misma no sean las adecuadas para cómo se ha maquetado el proyecto.

Página multilenguaje

Se han realizado traducciones a 4 idiomas diferentes: alemán, inglés, francés y español. Se ha llevado a cabo mediante la creación de nuevas páginas HTML enlazadas todas con enlaces y los archivos ubicados en subcarpetas. A dicha traducción se puede acceder desde un submenú desplegable ubicado en la esquina superior derecha.

Pruebas del software

Por último, se han realizado diferentes pruebas, para comprobar el funcionamiento completo del software. Se han elaborado pruebas unitarias conforme se elaboraba el código, pruebas de integración para comprobar el funcionamiento de los diferentes módulos que se compone la aplicación y pruebas de sistema para comprobar el funcionamiento en su conjunto.

Repositorio GitHub

Todo el código ha sido alojado en un repositorio GitHub (<https://github.com>) por lo que, asimismo puedo disponer de dicho código en mi repositorio local personal, se ha subido al repositorio en la nube, para poder acceder al código en cualquier momento y lugar.

Problemas encontrados

En este caso han sido pocos los problemas encontrados debido a que se han realizado cambios en el diseño en mayor medida, con casi ningún cambio ejecutado a nivel de back-end. Un problema que me ha retrasado bastante y es lo que me ha provocado disponer de menos tiempo para la realización de las pruebas del software, se ha producido a la hora de traducir las páginas a diferentes lenguajes, ya que empecé a realizarlo mediante archivos json, donde se alojaban las traducciones de los textos en español y mediante jquery al pulsar el botón del idioma se traduciría sin cambiar de página. Pero este proceso era demasiado enrevesado y finalmente decidí cambiar de método.

Problemas internos con PHPUnit que no me han permitido poder avanzar demasiado en las pruebas unitarias principalmente debido al tiempo requerido en ello, pues finalmente me decidí a realizar este procedimiento de manera manual.

19. Documentación Sprint 3

Enfoque front-end

Se ha mejorado parte del diseño de la web mediante varias funciones en Javascript y JQuery. Se ha habilitado la opción de cambiar de imagen en la página principal, realizando botones funcionales con JQuery. Este sería el código utilizado en HTML, donde se emplean dos imágenes y dos botones, cada imagen corresponde a uno de los botones, de tal manera que dependiendo de que botón se pulse una imagen se ocultará y la otra aparecerá:

```
<div class="contain-image-home" id="changeimages">
  <ul class="slide-images">
    <li id="slide1"></li>
    <li id="slide2"></li>
  </ul>
</div>

<div class="contain-slide-buttons">
  <ul class="slide-buttons">
    <li id="button1"></li>
    <li id="button2"></li>
  </ul>
</div>
```

Código JQuery empleado:

```
//Al cargar el documento
$(document).ready(function(){
  /*Realizar las funciones de click en botones. Al hacer clic
  Y cambiar colores de botones al realizar click en dicho bot
  $("#button1").click(function(){
    $("#button1").css("background-color", "#D31A1B");
    $("#button2").css("background-color", "white");
    $("#slide2").hide();
    $("#slide1").show();
  });
  $("#button2").click(function(){
    $("#button1").css("background-color", "white");
    $("#button2").css("background-color", "#D31A1B");
    $("#slide1").hide();
    $("#slide2").show();
  });
});
```

Resultado obtenido:



Por otro lado, se ha empleado la función css @media (medida en píxeles) para adaptar el estilo de la página según el ancho establecido en la pantalla o cuando maximizamos/minimizamos una ventana en la web. Por lo que cuando la ventana

alcanza la dimensión indicada en @media, se aplicarán todos los estilos que deseemos de manera anidada. Un ejemplo es el famoso menú responsive “hamburguesa”, que cuando minimizamos la pantalla, todo el contenido del menú de un header, se aloja en un submenú. Ejemplo de código @media de un menú “hamburguesa”:

```
@media (max-width:1000px){
  .final-contain, .logo, .main-contain-courses{
    flex-direction: column;
  }
  .button-burguer-menu img{
    display:block;
    margin-right: 200px;
  }
  .header-menu{
    display:none;
  }
  .mostrar, .header-menu-responsive ul{
    display:block;
  }
  .header-menu-responsive{
    display: none;
  }

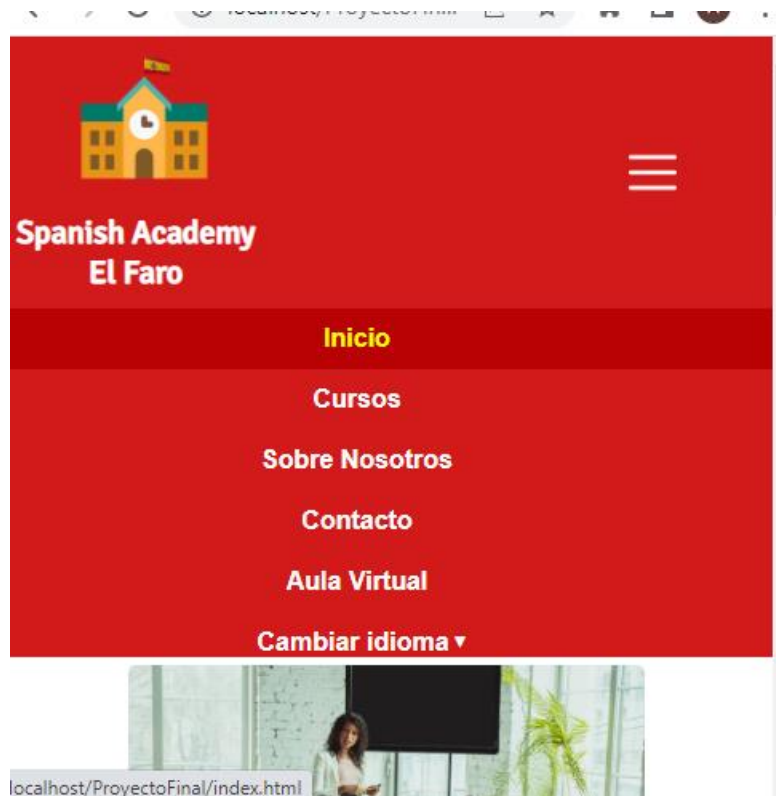
  .header-menu ul ul li{
    float: left;
  }
}
```

Como podemos observar, se indica que, si el ancho de la pantalla alcanza un valor inferior a los 1000 píxeles, se aplican los estilos dados. En este caso se oculta un menú para mostrar otro. Aplico dicho método en todas las páginas, para que se vean de la mejor forma posible:



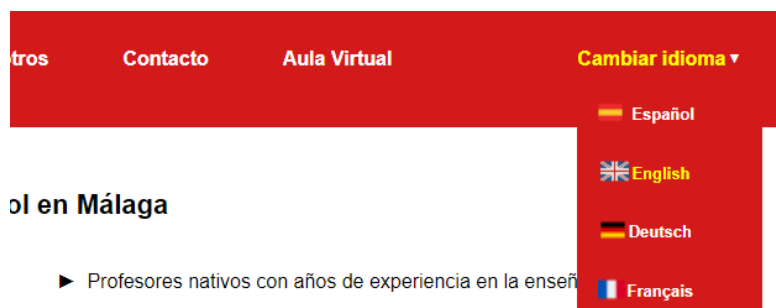
Como se puede apreciar, el menú que se encontraba en la cabecera de la página, desaparece y mediante el empleo de JQuery, se habilita un botón para mostrar el contenido del menú de manera vertical:

```
//Efecto de slide (de arriba a abajo y viceversa) al hacer click en el elemento con la clase indi
$("#buttonburguermenu").click(function(){
    $(".header-menu-responsive").slideToggle(200).addClass('.mostrar');
});
```



Página multilinguaje

Se emplea un menú desplegable con varios idiomas, de tal manera que enlazan a páginas cuyo texto ha sido traducido y alojado en el HTML:

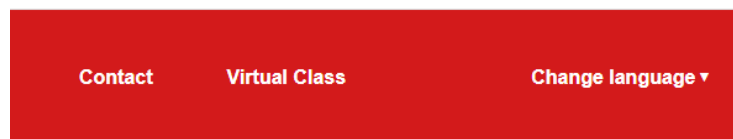


Escuela en Málaga

- ▶ Profesores nativos con años de experiencia en la enseñanza.
- ▶ 90% de alumnos aprobados en los diferentes niveles.
- ▶ Alumnos de todas las edades y de habla variada.

Escuela "El Faro"

Las primeras **galardonadas** de nuestra ciudad. Dispone de un espacio por las líneas de autobuses **EMT 22, 34, 12 y 5**, y de metro está a esta por **3 amplias aulas**. En cada una de ellas se imparte los recursos y el material docente necesario para llevar a cabo



Escuela in Málaga

- ▶ Native teachers with years of teaching experience.
- ▶ 90% of students passed at different levels.
- ▶ Pupils of all ages and from a wide range of speech backgrounds.

Escuela "El Faro"

The **award-winning schools** in our city. It has spacious premises, located near the bus lines **EMT 22, 34, 12 and 5**, and the metro is less than 10 minutes walk from where we teach a different level of Spanish (**A2, B1 y B2**) and

Pruebas del software

Para las pruebas unitarias se ha realizado un poco de análisis con el framework PHPUnit, pero la mayor parte del tiempo se han realizado pruebas personalmente analizando cada línea de código que se iba introduciendo, de tal manera que se reviva el funcionamiento más rápidamente y sin tener que depender de software.

Cuando cada una de esas líneas de código formaban un módulo completo, se comprobaba el funcionamiento de dichos módulos. De tal manera que por ejemplo se comprobaban las comunicaciones con las bases de datos, los arreglos creados para el muestreo de datos o para que los usuarios registrados pudiesen cerrar la sesión correctamente sin ningún tipo de error en el código del módulo creado.

En las pruebas del sistema del software creado, se ha comprobado el funcionamiento completo de la aplicación, una vez que se ha considerado terminado todo el proceso de creación del código, con todos los módulos acoplados. De esta manera podemos comprobar si hemos dejado pasar algún tipo de error y si el resultado obtenido es el esperado.

20. Retrospectiva Sprint 3

En este último Sprint, es cuando más he sufrido por el tiempo esperado para finalizar el proyecto. Esto ha sido debido a problemas con PHPUnit para la realización de diferentes pruebas, por la decisión de escoger el mejor método para fabricar una web multilinguaje. Esto me ha llevado a la decisión definitiva de no alojar la web en algún hosting gratuito debido a que debía de cambiar gran parte del código para que funcionase correctamente con las bases de datos de los hostings gratuitos disponibles. Por lo que se ha optado por subir el código al repositorio GitHub y poder acceder a la web de manera local.

21. Manual de uso y puesta en marcha

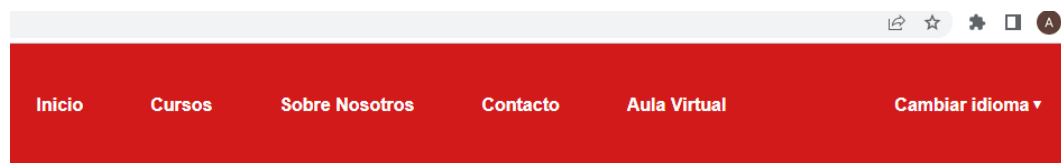
Para acceder a la aplicación disponemos de la opción de introducir en el navegador, (con el código en alguna de las carpetas en nuestro dispositivo, donde vayamos a visualizar el contenido) al escribir localhost en alguno de nuestros navegadores,

podremos acceder a todo el contenido de la web. Hay que tener en cuenta que si queremos ingresar en el aula virtual, deberemos de haber sido registrados previamente por un usuario con rol de “Profesor”. También existe la opción de alojar el código en un hosting, para acceder a la web mediante un dominio web como por ejemplo, www.google.es.

En nuestro caso accederemos por localhost y la carpeta donde se encuentra alojado el proyecto: localhost/ProyectoFinal.

El funcionamiento de la web es bastante sencillo y dispone de un sistema de acceso a páginas como disponen otro tipo de webs. A través de un menú se acceden a las diferentes áreas de la web, como puede ser: contactos, inicio, sobre nosotros...

También disponemos de la opción de traducir nuestras páginas mediante el submenú de cambiar idioma:



Estas páginas disponen de la información de los servicios que se ofrecen.

Por otro lado, disponemos del aula virtual, donde los usuarios podrán acceder a contenido exclusivo de la web, pero solamente si éstos se encuentran registrados por el usuario con más permisos como, por ejemplo, de registro y eliminación de usuarios o permisos de subida y eliminación de archivos.

Para poder acceder disponemos de un formulario en la sección de aula virtual donde podremos acceder, con el usuario y contraseñas que nos han indicado o hemos sido registrados:



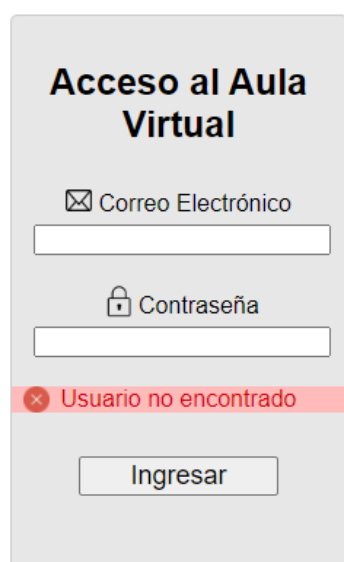
Acceso al Aula Virtual

☒ Correo Electrónico

☐ Contraseña

[¿Olvidaste la contraseña?](#)

Si no introducimos los datos correctos nos aparecerá el siguiente mensaje:



Acceso al Aula Virtual

☒ Correo Electrónico

☐ Contraseña

☒ Usuario no encontrado

También se dispone de la opción de recuperar la contraseña, en el caso que la hayamos olvidado. Para recuperarla introducimos un correo electrónico que haya sido registrado y se nos enviará al correo un mensaje con nuestra contraseña:

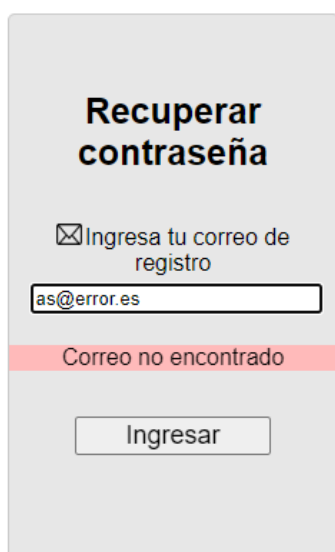


Recuperar contraseña

✉ Ingresa tu correo de registro

Contraseña enviada al correo introducido

En el caso que introduzcamos un correo erróneo no se enviará la contraseña al correo introducido, ya que no está registrado:



Recuperar contraseña

✉ Ingresa tu correo de registro

Correo no encontrado

Al introducir el correo y contraseña correctos en el aula virtual, podremos acceder a los cursos. Hay que tener en cuenta que la aplicación detecta automáticamente si el correo introducido es de un alumno o de un profesor, por lo que accederemos a páginas distintas:

Si accedemos como alumnos, accederemos a la siguiente página, con las diferentes opciones:

La página de inicio del aula virtual o el area de los cursos, donde se muestran los cursos de los que dispone la academia. El alumno solo podrá acceder al curso en el que se

encuentra registrado. A la izquierda nos aparece la información del alumno es la denominada área personal.

Tendremos la posibilidad de cerrar la sesión y volver al formulario de inicio de sesión.

The screenshot shows the 'SECCIÓN ALUMNOS' (Students Section) of the 'Spanish Academy El Faro'. On the left is a sidebar for the user 'Javier Castilla Fuentes' with a dropdown menu containing 'Inicio', 'Editar perfil', 'Archivos', and 'Cerrar sesión'. The main content area is titled 'Cursos' and displays two course cards: 'A2' and 'B1'. Below the 'Cursos' title, there are three columns of course cards. The first column contains a card for 'A2'. The second column contains a card for 'B1'. The third column contains a card for 'A2' and a card for 'B1'. The sidebar also includes a section for 'Área personal' with links to 'Inicio', 'Mi perfil', 'Nº expediente alumno: 1', 'Alumno: Javier Castilla Fuentes', 'Email: javiercastill@gmail.com', 'Cursos: A2', 'Documentos y archivos' (listing several PDF files), and 'Mis cursos' (listing 'A2').

Dispone de una sección de edición de perfil, en donde el usuario (tanto profesor como alumno) podrá cambiar su email o contraseña.

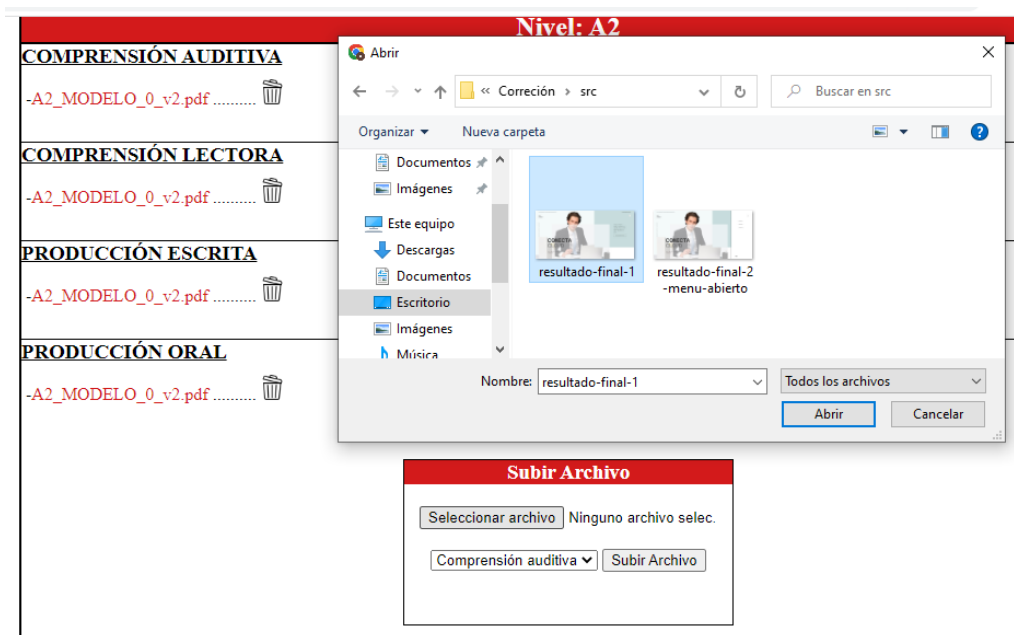
También dispone de una sección de archivos, donde se encuentran los archivos que han sido subidos por el usuario profesor (o lo que es lo mismo acceder pulsando en el o los cursos donde nos encontremos matriculados). El alumno solo podrá visualizar los archivos:

The 'Editar perfil' form contains two sections. The first section, 'Cambiar email', has a text input field containing 'javiercastill@gmail.com'. The second section, 'Cambiar contraseña', has a password input field with masked characters. Below these sections is a 'Guardar cambios' button.

Area personal		Nivel: A2
Inicio		COMPRENSIÓN AUDITIVA
Mi perfil		-A2_MODELO_0_v2.pdf
Nº expediente alumno: 1		
Alumno: Javier Castilla Fuentes		COMPRENSIÓN LECTORA
Email: javiercastill@gmail.com		-A2_MODELO_0_v2.pdf
Cursos: A2		PRODUCCIÓN ESCRITA
Documentos y archivos		-A2_MODELO_0_v2.pdf
A2_MODELO_0_v2.pdf		
A2_MODELO_0_v2.pdf		PRODUCCIÓN ORAL
A2_MODELO_0_v2.pdf		-A2_MODELO_0_v2.pdf
A2_MODELO_0_v2.pdf		
Mis cursos		
A2		

En cuanto al usuario profesor, éste podrá acceder a cualquier curso para poder subir o eliminar archivos. Al pulsar en cualquier curso se accede al área del curso específico, en el cual nos encontramos con un formulario donde podremos subir archivos a distintas áreas, marcadas en las opciones. Dichos archivos solo podrán tener formato .jpg y .pdf.

Subir Archivo	
<input type="button" value="Seleccionar archivo"/>	Ninguno archivo selec.
<div> <div>Comprensión auditiva ▼</div> <div> Comprensión auditiva Comprensión lectora Producción escrita Producción oral </div> </div>	<input type="button" value="Subir Archivo"/>



El profesor dispone de otra sección en la cual puede registrar o eliminar alumnos y se muestra una tabla con los usuarios que se encuentran registrados:

Hay que tener en cuenta que no podemos registrar a más de un usuario que dispongan de un mismo correo electrónico, ya que nos aparecerá un mensaje de error. En caso de que se registre correctamente nos mostrará un mensaje y nos mostrará el usuario en la tabla.

Nº alumno	Nombre	Email	Curso	Acciones
1	Javier Castilla Fuentes	javiercastill@gmail.com	A2	Eliminar
4	Antonio Perea	antoniojesusperea96@gmail.com	B2	Eliminar
13	Kevin Jhonson	JhonsonK@gmail.com	B1	Eliminar
20	Anne Dubois	DuboisAnne@hotmail.es	B2	Eliminar
22	Jessie Ferber	FerberJessie@gmail.com	A2	Eliminar
23	Robert Kennedy	Kennedy123456@hotmail.es	B1	Eliminar

Registrar nuevo alumno

Nombre

Contraseña

Email

Curso
A2 ▼

Registrar

Enlace repositorio GitHub:

<https://github.com/sk8antoniosk8/ProyectoFinalAntonio2022.git>

22. Conclusiones globales acerca del proyecto

Tanto en el comienzo, elaboración y finalización del proyecto he llegado a las siguientes conclusiones:

En cuanto al tiempo, me lo tendría que haber administrado mejor ya que me sentía bastante presionada en las finales de entregas de cada sprint. Es posible que esto haya sido debido a que quizás no tuve en cuenta los posibles fallos, errores o contratiempos que me he ido encontrando a lo largo de todo el proyecto y todo esto me ha ido obligando a aplazar tareas que tenía previstas para ciertos días.

En relación con los objetivos alcanzados, estoy bastante satisfecho con la parte del back-end y el código ejecutado en PHP ya que, he podido reflejar todos los conocimientos adquiridos en todo el curso. A demás he cumplido con todos los objetivos en este aspecto y no me han quedado ideas pendientes, sin embargo, si hablamos respecto al diseño me hubiese gustado haber mejorado bastantes aspectos que si me han quedado pendientes, como por ejemplo efectos y animaciones. También me hubiese gustado haberle dedicado más tiempo a las pruebas, ya que creo que es un aspecto muy importante a la hora de programar.

Finalmente, teniendo en cuenta todos los aspectos buenos como malos puedo decir que me siento bastante orgulloso con este proyecto ya que he ampliado en mayor medida los conocimientos con PHP, Javascript y sobre todo CSS. Teniendo esto en cuenta es de seguro que continúe desarrollando esta aplicación y me sea de ayuda para futuros proyectos y ampliar mis conocimientos que es lo más importante.

23. Webgrafía de recursos empleados en el proyecto

- <https://catalogo-decursos.com/academico/aula-virtual-todo-lo-que-necesitas-saber/>
- https://www.youtube.com/watch?v=0xYoGZtCq0I&ab_channel=GermanSol%C3%B3rzano
- https://www.youtube.com/watch?v=DOD_tLSkD58&ab_channel=ErickSantiago
- https://www.youtube.com/watch?v=OuoLQiXPitk&ab_channel=deivchoi

- https://www.youtube.com/watch?v=qP_FDTv9hcQ&ab_channel=Cursania-Cursosdeprogramaci%C3%B3nwebym%C3%B3vil
- <http://www.formacionwebonline.com/crear-sesiones-php-en-intranet/>
- https://www.youtube.com/watch?v=LL66QTP3txE&ab_channel=VidaMRR-Dise%C3%B1oydesarrolloweb
- https://www.youtube.com/watch?v=zXMcoTRTswQ&ab_channel=lasfito
- <https://www.urlencoder.org/>
- https://www.youtube.com/watch?v=aZ8Alp7kKlo&ab_channel=ElmerRodriguez
- https://www.youtube.com/watch?v=QpnUD3MJLYo&t=321s&ab_channel=JoseDeconstruye
- <https://desarrolloweb.com/faq/configurar-php-ini-smtp>
- <http://www.falconmasters.com/web-design/menu-de-navegacion-responsive/>
- https://www.youtube.com/watch?v=9QIq5LM2NXY&ab_channel=JaqueRomero-Dise%C3%B1o
- <https://pixabay.com/es/>
- <https://www.freepng.es/hd-png/copyright-free.html>
- <https://www.php.net/>
- <https://jquery.com/>
- <https://www.w3schools.com/css/>

DESARROLLO DE APLICACIONES WEB
WEB ACADEMIA DE ESPAÑOL CON AULA VIRTUAL
PROYECTO FINAL

Antonio Jesús Perea Aguilar