



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Alcoy

Desarrollo Full-stack de un inventario para una entidad
educativa

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Sirvent Juan, Antonio Manuel

Tutor/a: Alemany Mollà, Josep Tomàs

Cotutor/a: Guerola Navarro, Vicente

CURSO ACADÉMICO: 2021/2022

Resumen

El siguiente trabajo se centra en el desarrollo Full-stack de una aplicación web para un inventario de una entidad educativa, teniendo en cuenta otras funciones como pueden ser las de tener anotaciones de desperfectos en las aulas para su posterior solución y que de alguna forma queden anotadas para el usuario de la aplicación.

En este caso, para el desarrollo de la aplicación, se utilizarán los siguientes lenguajes de programación: JavaScript (en este caso, para toda la parte de las funciones), HTML5 y CSS (para el front-end). Y para el desarrollo de la parte de back-end, se utilizará MySQL.

La aplicación permitirá al usuario tener un control de todos los objetos que se encuentren registrados en todas las aulas del centro, de manera que no haya ningún problema a la hora de tener que hacer inventarios. También se permitirá incluir nuevos objetos en dicho control, en el caso de que aparezcan nuevas tecnologías y/o materiales escolares a utilizar.

Es un trabajo que se centra tanto en la parte de back-end con su propia base de datos, como en la parte de front-end, que tiene una aplicación cliente para poderse utilizar correctamente y sin ninguna dificultad.

Como se ha mencionado en el título del trabajo, las tecnologías que se utilizarán son principalmente los lenguajes JavaScript, HTML5 y CSS, además de utilizar MySQL.

Palabras clave: Full-stack, Front-end, Back-end, JavaScript, HTML5, CSS, MySQL.

Resum

El següent treball es centra en el desenvolupament Full-stack de una aplicació web per a un inventari d'una entitat educativa, tenint en compte altres funcions com poden ser les de tindre anotacions de desperfectes en les aules per a la seua posterior solució, i que de alguna forma queden anotades per a l'usuari de l'aplicació.

En aquest cas, per al desenvolupament de l'aplicació, s'empraran els següents llenguatges de programació: JavaScript (en aquest cas, per a la part de funcions), HTML5 i CSS (per al front-end), i finalment, per al desenvolupament de la part de back-end, s'utilitzarà MySQL.

L'aplicació permetirà a l'usuari tenir un control de tots els objectes registrats en totes les aules del centre, de manera que no hi haja cap problema a l'hora de tindre que fer inventaris. També es permetrà incloure nous objectes en els controls, en el cas de que apareguen noves tecnologies i/o materials escolars a utilitzar.

Es un treball que es centra tant en la part de back-end, amb la seua base de dades propia, com en la part de front-end, que té una aplicació client per a poderse emprar correctament i sense cap dificultat.

Com s'ha mencionat en el títol del treball, les tecnologies principalment emprades serán els llenguatges JavaScript, HTML5 i CSS, además de emprar MySQL.

Paraules clau: Full-stack, Front-end, Back-end, JavaScript, HTML5, CSS, MySQL.

Abstract

The following academic job will be centered in the Full-stack development of a web application to have a control on the objects used in an academic entity, having in mind other functions such as having anotations of flaws in the objects, or in the rooms of the entity itself, so there's a way to track them for the user of the application.

In this case, to the development of the aplication, there will be used several programming languages such as: JavaScript (for the function parts), HTML and CSS (for the front-end), and finally, for the development of the back-end part, MySQL will be used.

The application will let the user have a control of every object registered in every room of the entity, in a way that there won't be any problem in tracking them when the user wants to do their inventory function. Also it will also be possible to include new objects in the control, in the case that new technologies or new materials are added to the educational entity.

This academic job, is centered on the full-stack development of the entire application, having it's own database, that's related to the back-end part, and it's front-end, with his own client application so it can be use correctly and efficiently.

As we have mentioned before in the title, the technologies used will be mostly the following: JavaScript, HTML5, CSS, and MySQL aswell.

Keywords: Full-stack, Front-end, Back-end, JavaScript, HTML5, CSS, MySQL.

Palabras clave

- **Desarrollo Full-stack:** este tipo de desarrollo, se refiere a un Desarrollo general de la aplicación en cuestión, haciendo referencia tanto a la parte del Back-end, como a la parte del Front-end.
- **Back-end:** es la parte de la aplicación a la que el usuario no suele tener acceso directo, se suele encargar de almacenar y manipular los datos.
- **Front-end:** es la parte de la aplicación a la que el usuario accede directamente, se podría decir también que es la parte visual de la aplicación.
- **JavaScript:** es un lenguaje de programación orientado a objetos, se suele utilizar para el Desarrollo de aplicaciones web.
- **HTML5:** es un estándar que sirve como referencia del software que conecta con la elaboración de páginas web, funciona mediante etiquetas.
- **CSS (Cascading Style Sheets):** es el lenguaje que utilizan las hojas de estilo en cascada, se utilizan para estilizar los elementos de, en nuestro caso el HTML5.
- **MySQL:** es un lenguaje de gestión de bases de datos, normalmente funciona con un modelo cliente-servidor.
- **Cliente:** un usuario de internet que pide un servicio. En MySQL son los programas que permiten al usuario acceder al servidor para consultar o manipular la información.
- **Servidor:** para MySQL, son los dispositivos dónde se almacena toda la información que el cliente puede añadir, observar o modificar.

Agradecimientos

A mi madre, la cual siempre me ha estado apoyando durante todo lo que me he ido proponiendo durante toda mi vida, y por estar siempre a mi vera animándome.

A mis mejores amigos, que han estado ahí desde el momento en el que más los necesitaba y hasta día de hoy siguen conmigo.

Por último, agradecer especialmente a Vicente Guerola Navarro la gran ayuda que me ha proporcionado durante mis últimos años de carrera.

Contenido

1.	Introducción	11
1.1.	Motivación del Proyecto	12
1.2.	Antecedentes y descripción del Proyecto	13
1.3.	Objetivos	14
1.4.	Fases del Desarrollo	14
1.4.1.	Investigación y análisis	14
1.4.2.	Creación de la parte del Back-end y estructura del proyecto	14
1.4.3.	Datos en MySQL y la conexión con el propio MySQL	17
1.4.4.	Creación de las distintas vistas de la aplicación	18
1.4.5.	Manipulación de datos en MySQL.....	18
1.4.6.	Comunicación con el usuario de la aplicación mediante connect-flash y navegación. 19	
1.4.7.	Registro de usuarios en MySQL	20
1.4.8.	Log in de usuarios en MySQL	21
1.4.9.	Protección de las rutas del servidor	21
1.4.10.	Datos privados.....	22
1.5.	Dificultades Encontradas.....	22
1.6.	Estructura de la memoria final	23
2.	Estado del arte.....	24
2.1.	IDE (Entorno de desarrollo integrado).....	24
2.2.	Tecnologías actuales	25
2.3.	Base de datos.....	28
3.	Análisis del problema	29
3.1.	Plan de Trabajo.....	30
4.	Desarrollo de la solución propuesta	32
4.1.	Requisitos.....	32
4.2.	El Sistema.....	34
4.3.	Sistema de usuarios	37

4.4.	Función principal (visualización del inventario).....	39
4.5.	Envío y recogida de datos a la base de datos	44
4.5.1.	Usuarios.....	45
4.5.2.	Aulas	45
4.5.3.	Objetos	46
5.	Pruebas	48
5.1.	Seguridad de la aplicación.....	48
5.2.	Guardado de datos en la base de datos	50
5.3.	Conclusiones de las pruebas realizadas.....	56
6.	Conclusiones generales	57
6.1.	Problemas encontrados.....	58
6.2.	Solución de los problemas encontrados	58
6.3.	Tecnologías aprendidas	59
6.4.	Relación del trabajo desarrollado con los estudios cursados.....	60
7.	Futuras líneas de Desarrollo	61
7.1.	Ingesta y obtención de aulas/departamentos mediante protocolos de ETL.....	61
7.2.	Ingesta y obtención de objetos mediante archivos protocolos de ETL.	62
7.3.	Creación de elementos buscadores de aulas/departamentos y/o objetos.....	62
7.4.	Mejorar la parte visual de la aplicación.....	63
8.	Bibliografía	64
A.	Anexo.....	67
A.1.	Manual de usuario	67
A.1.1.	Pantalla de inicio	67
A.1.2.	Inicio de sesión.....	68
A.1.3.	Visualización de perfil	69
A.1.4.	Visualización de aulas/departamentos.....	70
A.1.5.	Visualización de objetos.....	70
A.1.6.	Visualización de los formularios	72

Tabla de figuras

Figura 1. Código de conexión a la base de datos	15
Figura 2. Código para "encender" el servidor	15
Figura 3. Estructura interna del proyecto	16
Figura 4. Estructura interna del proyecto	17
Figura 5. Estructura interna del proyecto	17
Figura 6. Logotipo Visual Studio Code	24
Figura 7. Logotipos HTML5, CSS, JavaScript	25
Figura 8. Logotipo Handlebars.....	26
Figura 9. Logotipo de NodeJS.....	27
Figura 10. Logotipo de MySQL.....	28
Figura 11. Diagrama de Gantt con el plan de desarrollo.....	30
Figura 12. Diagrama EER de la base de datos	35
Figura 13. Flujo de la aplicación del cliente	36
Figura 14. Logotipo IES Pare Arques	37
Figura 15. Log in de la aplicación web desarrollada.....	38
Figura 16. Visualización de usuarios y contraseñas en MySQL.....	38
Figura 17. Formulario para registrar nuevos usuarios.....	39
Figura 18. Visualización del perfil de la aplicación web	40
Figura 19. Barra de navegación para una persona no autenticada	40
Figura 20. Barra de navegación para una persona autenticada	40
Figura 21. Visualización de las funcionalidades del inventario.	41
Figura 22. Carta de texto con acceso al inventario de aulas.....	41
Figura 23. Carta de texto con acceso al formulario de registro de usuarios.....	42
Figura 24. Visualización de las aulas guardadas en la base de datos.	42
Figura 25. Botón inferior derecho para crear nuevas aulas/departamentos.....	43
Figura 26. Carta de texto con información de un objeto en buen estado	43
Figura 27. Carta de texto con información de un objeto en mal estado	44
Figura 28. Carta de texto con información de un objeto dañado.....	44
Figura 29. Información de los usuarios en la base de datos	45
Figura 30. Información aulas/departamentos en la base de datos	46
Figura 31. Formulario de creación de una nueva aula	46
Figura 32. Información de la tabla objeto en la base de datos	47
Figura 33. Formulario de creación de un objeto nuevo.....	47
Figura 34. Log in correcto dentro de la aplicación.....	49

Figura 35. Log in incorrecto dentro de la aplicación	49
Figura 36. Código referente al método de autenticación.....	50
Figura 37. Cabecera de la petición POST de editar un aula.....	50
Figura 38. Guardado de aula (BIO1) correctamente	51
Figura 39. Guardado de aula (BIO1) correctamente en la base de datos	52
Figura 40. Intento de actualizar datos de (BIO1 a MATE1)	52
Figura 41. Visualización del mensaje de éxito al borrar BIO1	53
Figura 42. Visualización de las aulas/departamentos en la base de datos al borrar BIO1	53
Figura 43. Objeto añadido al aula MUS1	54
Figura 44. Visualización del objeto añadido al aula MUS1	54
Figura 45. Intento de duplicar el objeto con ID SAX1 en la base de datos.....	55
Figura 46. Visualización de la eliminación del objeto SAX1	55
Figura 47. Visualización de la eliminación del objeto SAX1 en la base de datos	55
Figura 48. Visualización de los usuarios en la base de datos	56
Figura 49. Diagrama de flujo de como funcionaría el protocolo ETL	61
Figura 50. Página de inicio de la aplicación (no autenticado).....	67
Figura 51. Página de inicio de la aplicación (usuario autenticado).....	68
Figura 52. Página de inicio de sesión.....	68
Figura 53. Página de visualización de perfil	69
Figura 54. Página de visualización de aulas/departamentos	70
Figura 55. Página de visualización de objetos en aula CAS1	71
Figura 56. Página con formulario para añadir un usuario	72
Figura 57. Página con formulario para añadir un aula/departamento	73
Figura 58. Página con formulario para añadir un objeto a CAS1	73

1. Introducción

En el siguiente Trabajo de fin de grado se va a exponer el desarrollo Full-stack de una aplicación web, con el fin de digitalizar el inventario de una entidad educativa, en este caso, la entidad educativa sobre el que se va a hacer el proyecto es el instituto IES Pare Arques, situado en Cocentaina, justo en Av. Ferrocarril 29, 03820.

Se van a utilizar distintos tipos de lenguajes y tecnologías con tal de conseguir esta digitalización, ya que, al ser un desarrollo completo, se deben de tener conocimientos en todos los ámbitos del desarrollo del proyecto, principalmente conocimientos en la parte del back-end, para manipulación de datos en general, se va a utilizar NodeJS y MySQL.

Por lo que respecta a la parte del front-end, se van a utilizar distintos lenguajes y tecnologías también, ya bien sea HTML5, Handlebars, y CSS para el desarrollo de la parte gráfica de la web, y JavaScript para todo lo referente a las consultas a las bases de datos y distintas funciones de la web como podremos ver más adelante.

1.1. Motivación del Proyecto

Con la idea previamente planteada, se ha estado buscando un tipo de desarrollo que permita al alumno obtener experiencia en tanto el desarrollo de una aplicación web, como de un planteamiento de desarrollo que nace de una necesidad de una entidad, en este caso, el instituto IES Pare Arques, el cual presenta un problema, y mediante el desarrollo de esta aplicación web, se propone una solución al mismo, así existen dos motivaciones distintas a la hora de hacer este proyecto

- **Motivaciones con respecto al desarrollo del proyecto:**

Posibilidad de desarrollarse en los dos ámbitos referentes en cuanto a desarrollo de aplicaciones, que principalmente son la combinación del desarrollo de la parte de manipulación de datos (back-end), y del desarrollo de la parte respectiva a la visualización de información en la aplicación (front-end). Todo el desarrollo, hará que se puedan adquirir conocimientos en los distintos ámbitos, y distintos lenguajes:

- MySQL
- HTML5
- CSS
- JavaScript
- NodeJS

- **Motivaciones con respecto al planteamiento de un problema real:**

Respecto al planteamiento de un problema real, la principal motivación de esto es llegar a simular una relación empresa-cliente, en el que la empresa ofrece sus servicios al cliente. En este caso, el alumno hará el papel de empresa, y el cliente en este caso será el instituto IES Pare Arques, el cual tiene una necesidad, y busca la ayuda del alumno, que mediante un desarrollo de una aplicación web, va a solucionar, o plantear una solución a este problema.

Para llegar a esta solución, durante todo el proyecto, se han planteado distintas reuniones con el cliente, para ver sus necesidades y cómo se pueden satisfacer mediante el desarrollo de dicha aplicación.

1.2. Antecedentes y descripción del Proyecto

- **Antecedentes**

A principios del curso 2021/2022, me encontraba buscando una opción para desarrollar mi trabajo de final de grado, y me presentaron la opción de poder desarrollar esta aplicación, para el instituto IES Pare Arques.

En este instituto no se cuenta con una tecnología avanzada, de manera que todos los procesos de inventario, gestión de bibliotecas, y procesos parecidos, se suelen hacer de forma manual, y nunca existe nada informatizado que pueda agilizar estos procesos. De modo que, una vez planteado el problema, se vio la viabilidad de sus necesidades, que principalmente era la digitalización del inventario de este instituto, con mi necesidad, que era desarrollar una aplicación web para así obtener experiencia tanto en desarrollo como en relaciones con clientes.

- **Descripción del proyecto**

Una vez se ha detectado la necesidad del cliente, se han pactado distintas reuniones para la realización del desarrollo, en la que se han ido introduciendo las distintas necesidades del cliente, así como posibles sugerencias por parte de la empresa:

- Necesidades por parte del cliente:
 - Poder visualizar todas las aulas/departamentos/almacenes del instituto
 - Poder visualizar todos los objetos dentro de las distintas aulas/departamentos/almacenes del instituto
 - Poder llevar una identificación de los posibles desperfectos en los objetos, en este caso distinguir entre objeto en buen estado, dañado o que se encuentre en mal estado y que se pueda visualizar de forma rápida y eficiente.
- Sugerencias por parte de la empresa:
 - Tener un sistema de autenticación que nos permita asegurar todas las direcciones url para que no se pueda acceder a ellas sin una identificación
 - Creación de usuarios base, y a partir de ahí, la posibilidad de añadir distintos usuarios dentro de la aplicación, además de toda la parte de seguridad de cifrado de contraseñas y etc..
 - Que el entorno en el que se desarrolle sea un entorno de aplicación web, para así facilitar la portabilidad a cualquier otro dispositivo que se desee utilizar

1.3. Objetivos

Los principales objetivos de este proyecto son:

- Analizar las tecnologías de las que se dispone hoy en día para decidir que tecnologías serán las más adecuadas para el desarrollo de la aplicación.
- Desarrollar la aplicación por ambas partes, tanto por la parte que respecta a la manipulación de datos (back-end), como la parte respectiva a funcionalidades/visualización de contenido.
- Satisfacer en todo momento las necesidades del cliente de cara al uso de la aplicación.
- Ampliar conocimientos en todo el ámbito del desarrollo y evolución profesional trabajando con casos y problemáticas reales.

1.4. Fases del Desarrollo

El proyecto se ha realizado durante todo el curso 2021-2022, haciendo reuniones con el cliente, y viendo las necesidades progresivamente durante el curso, en los últimos meses del curso es cuando más ímpetu se le ha dado a la parte del desarrollo y la memoria, una vez ya se han tenido todos los objetivos y necesidades del cliente fijados.

1.4.1. Investigación y análisis

En esta fase, se busca información y se analiza los distintos lenguajes de programación y posibles frameworks que se podrían utilizar en el proyecto. Se tendrán en cuenta las ventajas e inconvenientes que puedan ocurrir con cada uno, y se verá el funcionamiento con estos. Una vez analizada toda esta información, se escogerá de las opciones la que mejor se adapte a nuestras necesidades.

1.4.2. Creación de la parte del Back-end y estructura del proyecto

Una vez definido con el cliente las necesidades de la aplicación, y de haber seleccionado los lenguajes de programación y frameworks adecuados se procederá a instalar toda la tecnología necesaria y crear la parte del back-end y el servidor.

```

pool.getConnection((err, connection) => {
  if (err) {
    if (err.code === 'PROTOCOL_CONNECTION_LOST') {
      console.error('DATABASE CONNECTION WAS CLOSED');
    }
    if (err.code === 'ER_CON_COUNT_ERROR') {
      console.error('DATABASE HAS TOO MANY CONNECTIONS');
    }
    if (err.code === 'ECONNREFUSED') {
      console.error('DATABASE CONNECTION WAS REFUSED');
    }
  }

  if (connection) { connection.release(); }
  console.log('Se ha conectado a la base de datos');
  return;
});

```

Figura 1. Código de conexión a la base de datos

```

// Encender el servidor
app.listen(app.get('port'), () => {
  console.log('Server on port', app.get('port'));
});

```

Figura 2. Código para "encender" el servidor

Mediante este código, encenderemos tanto el servidor, como la conexión a la base de datos, y podremos observar la web en localhost, en este caso, la variable 'port' se ha definido como 4000, así que la dirección principal de la página estará en localhost:4000

Una vez instalado todo el software necesario, se procederá a crear toda la estructura del proyecto, básicamente aquí se define dónde se guardarán todos los archivos, las carpetas principales son:

- **Carpeta con el nombre de proyecto** (nodejs-mysql-app): es la carpeta donde se van a guardar todos los módulos de NodeJS que vayamos a necesitar para la aplicación web.
- **Carpeta con el nombre "src"**: es una carpeta dónde se va a guardar todo el código, para que no se mezcle con las carpetas de la base de datos y de los módulos que estamos utilizando con NodeJS. Es dónde vamos a almacenar posteriormente nuestras vistas (HTML5, CSS y Handlebars) y principales funciones (JavaScript)
- **Carpeta con el nombre "database"**: es una carpeta donde guardaremos el código respectivo a la generación de la base de datos.

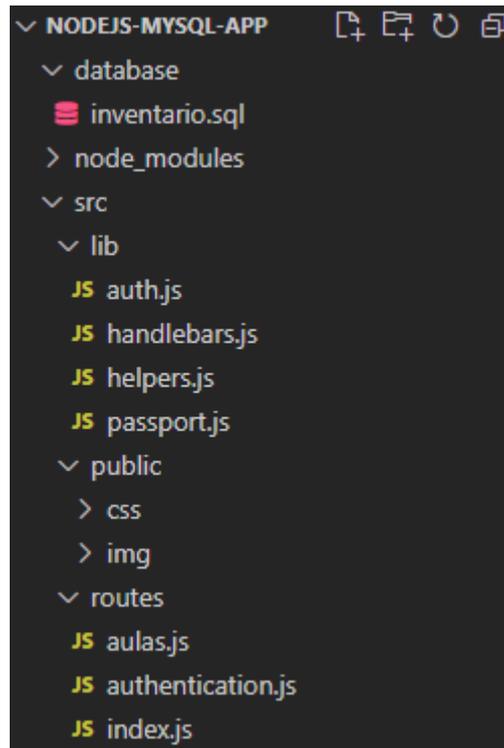


Figura 3. Estructura interna del proyecto

Dentro de la carpeta de “src” tendremos distintos archivos, entre ellos los códigos de funcionalidades en JavaScript (todos los archivos acabados en .js contienen código en JavaScript). Los archivos de la carpeta “lib” dentro de “src” nos van a servir como librerías que creemos para utilizar en las distintas rutas y en los archivos generales del proyecto, los vamos a referenciar en los archivos de la [Figura 3. Estructura interna del proyecto](#) específicamente en el archivo index.js, ya que este archivo es el “cuerpo” de la aplicación, el cual contiene toda la información del proyecto, y cuando se quiera referenciar cualquier cosa, o se quieran incluir nuevas rutas, se deberán de incluir en este archivo.

La carpeta de public en este caso, contendrá los distintos elementos multimedia que vayamos a utilizar en nuestro proyecto.

La carpeta de routes, contiene las distintas rutas que utiliza la aplicación para hacer peticiones POST y GET, y en esto se basa el funcionamiento de la aplicación.

La carpeta de views, contiene todas las vistas de la aplicación, aquí es donde se guarda todo el código HTML5, no obstante los archivos se guardan con formato .hbs, porque también se utiliza el lenguaje de template Handlebars, ya que nos permite incluir distintas lógicas en nuestros archivos HTML5 y nos da un mayor abanico de posibilidades a la hora de desarrollar.

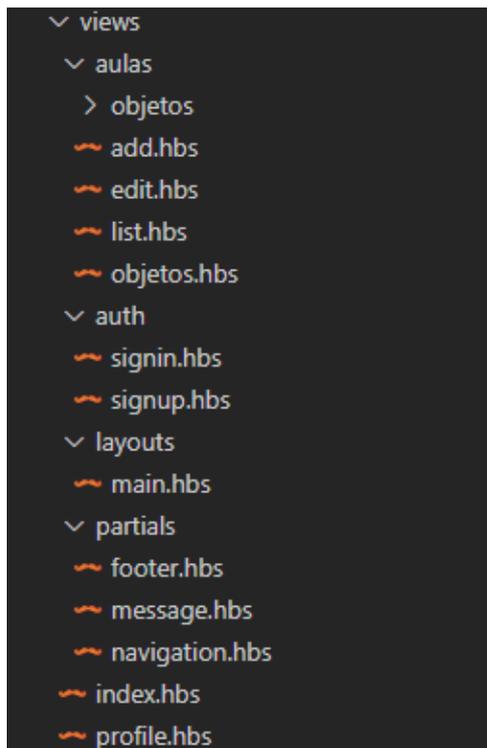


Figura 4. Estructura interna del proyecto

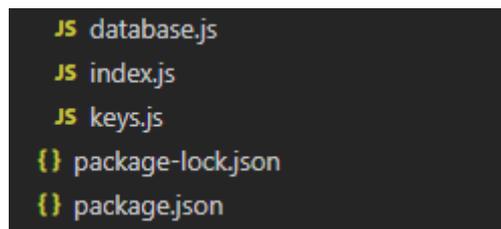


Figura 5. Estructura interna del proyecto

Por último, tanto los archivos .json, como la carpeta de node-modules, son carpetas que se han creado mediante la herramienta de Node Package Manager, en la que se guardan todos los datos de los paquetes que se han utilizado en el proyecto, se especifica más sobre ellos en [“2.2Tecnologías actuales”](#).

1.4.3. Datos en MySQL y la conexión con el propio MySQL

En esta fase, se creará un diagrama EER mediante MySQL, y luego mediante las posibilidades que nos ofrece MySQL, lo traeremos a código en forma de query, para poder utilizarlo posteriormente en la creación de la base de datos. Una vez tengamos este código, lo guardaremos en la carpeta “database” que habíamos creado en la Fase 1.4.2, y ejecutaremos el código en MySQL Workbench, el cual nos creará la base de datos, donde vamos a ir almacenando nuestra información.

1.4.4. Creación de las distintas vistas de la aplicación

En esta fase, nos vamos a centrar en el diseño de las vistas de la aplicación, y cómo de esta forma, vamos a poder procesar toda la información necesaria y navegar de un lado a otro de la aplicación. En la carpeta “views” es el lugar donde se va a almacenar toda esta información referente a las vistas de la aplicación, o interfaz de usuario.

La organización de las rutas que va a tener la aplicación se va a gestionar mediante peticiones GET y POST al servidor, de manera que podamos tanto visualizar información mediante peticiones GET, y modificar, eliminar o crear nueva información mediante peticiones POST.

Además, las vistas de la aplicación van a estar ordenadas mediante direccionamiento en la URL, esto quiere decir, que vamos a poder visualizar perfectamente dónde nos encontramos simplemente echando un vistazo a la URL de la aplicación.

1.4.5. Manipulación de datos en MySQL

Para lograr la manipulación de los datos que vamos a utilizar junto con las bases de datos que vamos a utilizar en MySQL, principalmente vamos a necesitar distintas funciones, que básicamente son las siguientes:

- a. Guardado de datos.
- b. Obtención de datos.
- c. Borrado de datos.
- d. Actualización de datos.

Estas funciones nos van a permitir cumplir con todas las necesidades del proyecto, ya que las necesidades del cliente principalmente son la correcta visualización de las distintas aulas/departamentos, y dentro de ellas poder ver los objetos y sus respectivos estados.

Así que principalmente en esta fase, vamos a incluir estas funciones en el proyecto. Para recoger datos de la base de datos lo vamos a conseguir mediante peticiones GET a nuestro servidor. Mientras que para añadir, modificar y eliminar los distintos objetos y aulas/departamentos vamos a utilizar peticiones POST al servidor, ya que las peticiones GET se suelen utilizar para recoger datos, y las POST se utilizan para modificar, añadir, o eliminar datos existentes.

También, durante esta fase, se contemplan todos los posibles errores que pueden surgir a la hora de manipular datos en MySQL, que en este caso principalmente las inconsistencias de datos pueden surgir cuando:

- Se pretende borrar un aula existente que contenga objetos dentro.
- Se pretende editar un aula que contenga objetos dentro.

Ambos errores anteriormente mencionados se contemplarán dentro del código para que a la hora de borrar un aula existente, también se borren con ella los objetos que tiene dentro. Además para la edición de aulas con objetos dentro, se modificarán todos los objetos que haya dentro al nuevo ID del aula asignada.

- Se pretende modificar un objeto cuyo ID ya exista.
- Se pretenda modificar un aula/departamento cuyo ID esté repetido.

Y estos errores también se tendrán en cuenta a la hora de montar el código. Se utilizarán mensajes flash en la aplicación para avisar de que los cambios se han realizado correctamente en caso de que no haya habido ningún problema. No obstante cuando nos topemos con un problema mencionado anteriormente, no podremos actualizar, y al cliente le aparecerá un mensaje informando del error que ha sucedido, y que así pueda rectificar dentro de la aplicación.

1.4.6. Comunicación con el usuario de la aplicación mediante connect-flash y navegación.

En esta fase, se van a implementar una serie de mensajes para que el usuario sepa en todo momento, a la hora de hacer cambios, añadir, o eliminar datos en la base de datos, que sepa lo que está pasando. También se darán avisos al usuario cuando se autentique en la aplicación, o cuando cree un usuario nuevo por ejemplo.

De esta manera, nos aseguraremos de que el cliente sabe qué está haciendo en todo momento, y cómo puede actuar para que la aplicación funcione correctamente. Principalmente se suele enviar este tipo de mensajes cuando:

- El usuario se autentica correctamente en la aplicación
- El usuario, una vez dentro de la aplicación, crea distintos usuarios para los distintos usuarios de la misma, esto es simplemente un método de autenticación y no de distintos perfiles con distintos permisos.
- El usuario crea, borra, o actualiza un aula/departamento.

- El usuario crea, borra, o actualiza un objeto dentro de un aula.

Con esta fase del desarrollo vamos a conseguir que el usuario tenga toda la información necesaria a la hora de hacer distintas acciones en la aplicación y que no haya problemas a la hora de seguir con el flujo de información en la misma.

La navegación en la página web, se hará mediante direcciones URL, es decir, todo el desarrollo está orientado a que el usuario pueda reconocer exactamente dónde se encuentra simplemente viendo a la URL. Aquí existen algunos ejemplos de cómo se van a estructurar las direcciones en la aplicación:

- Inicio de la aplicación: localhost:4000/
- Log in de la aplicación: localhost:4000/signin
- Perfil de la aplicación: localhost:4000/profile
- Visor de aulas de la aplicación: localhost:4000/aulas
- Visor de objetos de la aplicación: localhost:4000/aulas/:id_aula, donde el id_aula será el identificador de cada aula que nos lo traeremos de la base de datos para utilizarlo como parámetro en la URL.

Aclarar que se está utilizando la dirección de localhost porque el proyecto aún no se ha subido a ningún servidor. La aplicación está preparada para subir a servidor, no obstante, de momento cuando queramos referenciar a cualquier tipo de dirección, utilizaremos localhost:4000, haciendo referencia a la máquina dónde se está ejecutando el proyecto y el puerto por dónde se están transmitiendo los datos.

Para el proyecto a desarrollar, se ha pensado que esta es una forma bastante sencilla también de que el cliente sepa en cualquier momento donde se encuentra en la aplicación, ya que por ejemplo, en el caso del visor de objetos, es bastante intuitivo saber que te encuentras en el visor, si la ruta que has seguido hasta llegar ahí es aulas – seleccionar aula – ver los objetos del aula.

1.4.7. Registro de usuarios en MySQL

Para esta fase, dentro de una de las páginas de la aplicación web, vamos a incluir un formulario para que se puedan registrar nuevos usuarios dentro de la aplicación, de esta manera, junto con nuestro sistema de Login, (que vamos a crear posteriormente en la Fase 1.4.8. Login de usuarios en MySQL) podremos trabajar en una forma de asegurar las rutas del servidor para que nadie externo que no tenga ningún tipo de identificación correcta pueda acceder a la aplicación web.

Así que, como en este caso solamente estamos buscando credenciales de acceso, solamente vamos a pedir en el formulario los datos mínimos e indispensables para cumplir con esto.

1.4.8. Log in de usuarios en MySQL

Una vez ya hemos creado el registro de nuevos usuarios en MySQL, podremos empezar a trabajar en la fase de generar el Login de usuarios en nuestra base de datos.

Para formalizar esto, vamos a utilizar las credenciales que hemos generado en el formulario anterior para cumplir con los requisitos del login. Una vez las credenciales coincidan con cualquier credencial de la base de datos, tendremos el acceso a la aplicación.

1.4.9. Protección de las rutas del servidor

En esta fase, se trabajará en un método para que todas las rutas del servidor se encuentren protegidas, y que no se pueda acceder a la aplicación sin ningún tipo de credencial.

Para esto vamos a necesitar que un usuario con unas credenciales correctas acceda a la aplicación, y hacer una función en JavaScript que verifique que esta condición se está cumpliendo. Luego solamente tendremos que utilizar este método en cada petición tanto GET como POST a la hora de cargar las páginas para que, en caso de que el usuario esté loggeado se le redirija a la pagina correcta. No obstante, en caso de que este método no funcione (lo cual quiere decir que el usuario no tiene las credenciales correctas, o no se encuentra loggeado) se le redirigirá a la página de log in para que el usuario se pueda loggear con las credenciales correctas.

1.4.10. Datos privados

Por último, con el desarrollo de las funcionalidades de la aplicación terminadas y el diseño visual de la aplicación, vamos a proteger los únicos datos que pueden ser más sensibles para que, junto con la fase 9, se pueda asegurar una seguridad más que aceptable para el caso que el cliente está proponiendo.

Entonces, en esta fase, se procederá a que, una vez se cree el nuevo usuario en la base de datos, en vez de crearlo con la contraseña original, que se encripte esta contraseña y que no se pueda leer externamente. Así, en caso de que alguien consiguiera los datos de los usuarios, tendría la contraseña encriptada, y no la contraseña que necesitaría para acceder a la aplicación.

1.5. Dificultades Encontradas

En el desarrollo de la aplicación se han ido encontrando distintas dificultades, que han supuesto bastante consumo de horas, principalmente las dificultades han sido respecto a:

- Conexión de las distintas tecnologías
- Estructura del proyecto
- Conocimiento en las distintas tecnologías utilizadas a lo largo del desarrollo

En la carrera de ingeniería informática se han ido cursando distintas asignaturas, unas más relacionadas con el desarrollo web, y otras menos relacionadas. Así pues, tampoco existe una asignatura que abarque el desarrollo completo de una aplicación web, de manera que en este caso se ha tenido que recurrir a buscar bastante información, dicha información, principalmente se encuentra en las documentaciones proporcionadas por los distintos lenguajes que se utilizan.

Generalmente al hacer desarrollos de aplicaciones y demás, se tiene que tener muy en cuenta la documentación de los lenguajes y tecnologías que se utilizan en los proyectos, ya que a groso modo cumplen con las funcionalidades que se le podrían atribuir a un manual de usuario, en el que un usuario con conocimientos básicos en una herramienta, utiliza este manual para guiarse en el uso del mismo, y así conseguir un resultado satisfactorio para el cliente.

De manera que, teniendo conocimientos generales de programación proporcionados en las distintas asignaturas que se han ido cursando a lo largo de la carrera, se ha adquirido un conocimiento general a la hora de tratar con distintos lenguajes de programación.

1.6. Estructura de la memoria final

Así pues, una vez construida la introducción, se ha dividido la memoria final del desarrollo de la aplicación web en las siguientes partes:

1. Introducción
2. Estado del arte
3. Análisis del problema
4. Desarrollo de la solución propuesta
5. Pruebas
6. Conclusiones
7. Futuras líneas de Desarrollo
8. Bibliografía
9. Anexo

2. Estado del arte

Para el estado de arte, vamos a explicar principalmente las distintas tecnologías que se han utilizado en el desarrollo de la aplicación, empezando por el IDE, y luego las distintas tecnologías dentro del desarrollo. Por último, explicaremos un poco sobre el gestor de base de datos que hemos escogido para este desarrollo.

2.1. IDE (Entorno de desarrollo integrado)



Figura 6. Logotipo Visual Studio Code

- **Visual Studio Code [1][2]**, es el editor de texto que se ha escogido para este proyecto, existen diversos editores de texto distintos, que nos proporcionan distintas funciones dependiendo de lo que queramos hacer en el desarrollo del proyecto. No obstante, este editor es uno de los mejores que existen a día de hoy y es utilizado tanto a nivel amateur como a nivel profesional por toda la gente del sector de la programación.

Es una herramienta que no pesa nada en nuestro ordenador, y tiene soporte para todo tipo de lenguajes, además de tener una infinidad de extensiones que nos permiten hacer prácticamente de todo. Esto es esencial en un desarrollo como el que se ha propuesto en este Trabajo de Fin de Grado.

2.2. Tecnologías actuales



Figura 7. Logotipos HTML5, CSS, JavaScript

- **HTML5** [3][4], también conocido como **HyperText Markup Language, HyperText**, ya que no es más que un texto que se utiliza para enlazar unos contenidos con otros, ya sean textos o archivos. A groso modo es la base del funcionamiento de la web en este caso. Markup significa que en este tipo de lenguaje, se escribe mediante etiquetas. Por último Language, se refiere a que es un lenguaje de que tiene unas reglas fijas. Destacar que esto no es un lenguaje de programación, pero sí un lenguaje. Se distinguen en que HTML5 no puede hacer cosas como bucles, condiciones, funciones, etc..

Este lenguaje lo vamos a utilizar para construir nuestros componentes, además nos da mucha libertad a la hora de “importar” distintas funciones, como por ejemplo estilizar rápidamente nuestros componentes con herramientas como Bootstrap.

- **CSS** [5][6], también conocido como **Cascading Style Sheets**, esto nos va a permitir añadir estilos a uno o varios documentos de forma masiva.

- **JavaScript [7][8]**, es un lenguaje de programación, con el que a grandes rasgos podemos “decirle” a nuestro navegador que tareas debe realizar, en que orden y cuántas veces, por ejemplo.

Utilizaremos este lenguaje para dotar a nuestra web de funciones, y “darle vida”.



Figura 8. Logotipo Handlebars

- **Handlebars [9]**, se asemejaría a lo que se llama un lenguaje de plantillas. Este simple lenguaje, nos va a permitir, dentro de nuestros archivos HTML5, añadir “placeholders” que nos permitirán añadir variables directamente, de manera que no tenga que ser todo estático en la web, y que el contenido dependa de lo que vamos a traernos con las variables que tengamos disponibles. Además de que también se pueden conseguir distintas funcionalidades que soporten bucles, condicionales y demás dentro del mismo archivo HTML5.

Con la combinación de la tecnología anterior mencionada, podremos dar una cobertura completa a nuestro proyecto, añadiendo todos los elementos necesarios, desde funcionalidades, hasta distintos estilos generales que podríamos utilizar en el desarrollo de la aplicación web.

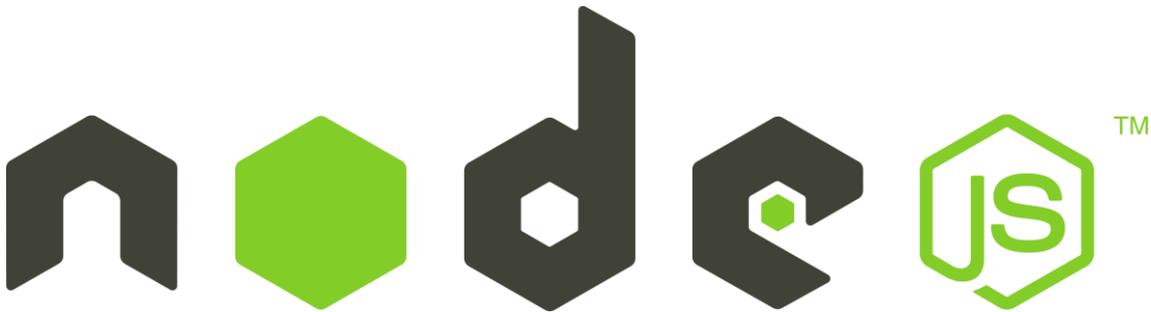


Figura 9. Logotipo de NodeJS

- **NodeJS [10][11]**, es un entorno de tiempo de ejecución de JavaScript. Es un entorno de ejecución en tiempo real, y incluye todo lo que necesitamos para ejecutar un programa escrito en JavaScript, además de que aporta muchos beneficios al proyecto y soluciona muchos problemas en el mismo.

Por otra parte, NodeJS brilla en la creación de aplicaciones de red rápidas, ya que es capaz de manipular una gran cantidad de conexiones simultáneas con un alto nivel de rendimiento.

Para este proyecto se han instalado bastantes paquetes mediante la herramienta **Node Pack Manager**, que nos van a ayudar en la mayoría de implementaciones que vayamos a utilizar dentro del proyecto. Un breve listado de los paquetes utilizados:

- express: framework bastante popular para la creación de aplicaciones back-end
- express-handlebars: integración del motor de plantillas handlebars en express
- express-session: administra la parte de sesiones activas de nuestra aplicación, se utiliza en métodos de autenticación de usuario
- mysql: es un módulo que nos permite conectarnos a la base de datos
- express-mysql-session: este módulo guardará las sesiones activas en la base de datos
- bcryptjs: módulo que se utilizará para cifrar las contraseñas de los usuarios antes de entrar en la base de datos.
- passport: módulo con soporte a las autenticaciones
- passport-local: módulo que permite autenticar a los usuarios de nuestra propia base de datos
- connect-flash: se utilizará para mandar mensajes de error y éxito cuando distintos mensajes realicen operaciones

- express-validator: módulo para validar los datos que el usuario nos envía desde el servidor

2.3. Base de datos



Figura 10. Logotipo de MySQL

- **MySQL [12][13]**, es un gestor de bases de datos relacional, destacar que es uno de los más utilizados en la actualidad debido a que está basado en código abierto. Es utilizado en el ámbito empresarial para gestionar toda la parte del back-end, así que se ha considerado que es un buen lugar donde guardar los datos del proyecto.

3. Análisis del problema

El desarrollo del proyecto nace de un problema planteado por la entidad educativa IES Pare Arques, en el que buscan la digitalización del inventario del instituto.

Para la solución del problema, se ha considerado que se puede desarrollar una aplicación con las funciones que debe de tener un inventario normal:

- Añadir aulas/departamentos.
- Modificar y borrar aulas/departamentos.
- Añadir objetos dentro de las distintas aulas.
- Modificar y borrar objetos dentro de las distintas aulas.

Se ha considerado pertinente, además, que la aplicación tenga una forma de conseguir que sea segura frente a posibles ataques de exteriores, para ello se ha considerado que exista en el desarrollo del proyecto una parte referente a la seguridad:

- Log in de usuarios existentes en la base de datos.
- Asegurar cada ruta posible dentro de la aplicación web para evitar accesos indeseados.
- Protección de las contraseñas mediante encriptación.

Para todo esto, se ha seguido un plan de trabajo con el fin de desarrollar el proyecto satisfactoriamente, que comentaremos a continuación. La parte de implementación se verá más extensamente en “[4. Desarrollo de la solución propuesta](#)”, en la que se dividirá y explicará el desarrollo de la solución propuesta, pero siguiendo las fases en: “[1.4. Fases del Desarrollo](#)”.

- Requisitos.
- El Sistema.
- Sistema de usuarios.
- Función principal.
- Envío y recogida de datos a la base de datos.

Así pues, el desarrollo del proyecto, girará en torno a las funciones que se ha deseado para la aplicación, y teniendo en cuenta todo el ámbito referente a seguridad planteado por el alumno.

3.1. Plan de Trabajo

Para organizar el desarrollo y que se pueda cumplir de una forma satisfactoria, se ha organizado un diagrama de Gantt para la distribución de las tareas y de los tiempos que va a seguir el plan de trabajo del desarrollo del proyecto.



Figura 11. Diagrama de Gantt con el plan de desarrollo

Como podemos ver, se ha distribuido toda la carga de trabajo los meses del segundo cuatrimestre, a modo de que el proyecto termine a finales de junio.

En el mes de marzo se va a centrar en hacer distintas reuniones con el cliente, para que éste explique su problemática y que plantee unas funcionalidades para el desarrollo de la aplicación. Las Reuniones con el cliente posteriores en el mes de marzo se harán para definir el alcance de la aplicación, y fijar los requisitos mínimos de la misma, que podemos ver en el apartado “[4.1. Requisitos](#)”.

Durante los meses de abril y mayo se ha planeado desarrollar la aplicación general, siguiendo los pasos explicados en el apartado “[1.4. Fases del Desarrollo](#)”. Así pues, durante estas fases se implementa tanto la interfaz de usuario, el front-end y el back-end.

Por último, durante el mes de junio, se ha pensado conveniente hacer una parte de pruebas de la aplicación, donde podamos comprobar que todo funciona correctamente. Se ha creído oportuno hacer dos fases de pruebas, una primera, de la que seguramente vayan a surgir problemas, y una vez estos problemas estén solucionados, hacer una fase de pruebas final.

Una vez se termine este período, la aplicación estará desarrollada completamente, y disponible para utilización del cliente.

La intención de todo esto es acercarse lo más posible, como ya se ha comentado anteriormente, para obtener experiencia en el desarrollo de aplicaciones, y de poder tener alguna experiencia de cómo se trabaja en el mundo laboral, ya que se trabaja de forma organizada. En este caso, hemos utilizado un diagrama de Gantt para establecer los tiempos de entrega de cada parte del desarrollo, ya que esta es una herramienta que se suele utilizar en el ámbito empresarial, y muchas consultorías de desarrollo, los utilizan para aclarar los distintos tiempos de entrega de los proyectos.

4. Desarrollo de la solución propuesta

Antes de empezar a desarrollar la aplicación se han establecido con el cliente distintos requisitos que debe cumplir la aplicación para especificar el alcance de la misma.

En el siguiente apartado se van a listar las distintas funcionalidades mínimas que tendrá la aplicación.

4.1. Requisitos

La aplicación constará en las siguientes funcionalidades:

1. Log in de usuarios existentes:

- Req 1.1, log in de usuarios para acceder a la aplicación.
- Req 1.2, el formulario de log in, tendrá los campos de usuario y contraseña.

2. Perfil de usuario:

- Req 2.1, visualización de la página de perfil del usuario que se ha logeado.
- Req 2.2, visualización de la funcionalidad de las aulas disponibles. El botón que aparezca redirigirá a la página de visualización de aulas.
- Req 2.3, visualización de la funcionalidad de crear nuevos usuarios para la aplicación. El botón que aparezca redirigirá a un formulario para crear usuarios.

3. Visualizador de aulas/departamentos:

- Req 3.1, visualización de la página de aulas/departamentos, con todas las aulas disponibles. En la parte inferior derecha aparecerá un botón para poder añadir nuevas aulas/departamentos.
- Req 3.2, en cada aula/departamento mostrar un botón para poder editar el objeto seleccionado. Se mostrará un formulario para cambiar los campos editables del objeto. Se mostrará un mensaje rojo con información en caso de error, y un mensaje verde en caso de que los cambios se hayan efectuado correctamente.
- Req 3.3 en cada aula/departamento mostrar un botón para poder eliminar el aula/departamento seleccionado/a. Una vez eliminada el aula, se eliminarán los objetos de dentro de la misma. Se mostrará un mensaje con información sobre la acción realizada.

- Req 3.4, cada aula/departamento tendrá un enlace a la página de la aplicación donde se muestren todos los objetos del aula/departamento.

4. Visualizador de objetos:

- Req 4.1, visualización de la página de objetos, para acceder a esta, se tiene que acceder desde las distintas aulas que existen dentro de la aplicación. En la parte inferior derecha aparecerá un botón para añadir nuevos objetos. Se mostrará entonces un formulario con los campos que se guardan de cada objeto.
- Req 4.2, en cada objeto, mostrar un botón para poder editar el objeto seleccionado. Se mostrará un formulario con los campos editables del objeto. Se mostrará un mensaje rojo con información en caso de error, y un mensaje verde en caso de que los cambios se hayan efectuado correctamente.
- Req 4.3, en cada objeto, mostrar un botón para poder eliminar el objeto seleccionado. Una vez eliminado el objeto, se mostrará un mensaje con información sobre la acción realizada.
- Req 4.4, para cada objeto, poder mostrar de forma visual el estado de cada uno de ellos, buen estado (verde), mal estado (amarillo), dañado (rojo).

5. Creación de nuevos usuarios (una vez se tiene una sesión iniciada):

- Req 5.1, en el perfil de usuario, en la barra de navegación habilitar la opción para poder registrarse nuevos usuarios en la base de datos.
- Req 5.2, una vez se entre en esta dirección, mostrar un formulario con el nombre de usuario, nombre completo y contraseña. Se mostrará un mensaje con información realizada una vez se confirme el formulario.

6. Elementos externos a las funcionalidades:

- Req 6.1, visualización de una pantalla previa al log in de usuario.
- Req 6.2, componentes comunes a todas las vistas para mejorar la navegación (barra de navegación superior, y información general en el footer de la página).

4.2. El Sistema

A continuación, se explica todas las partes que interactúan en la aplicación y su funcionamiento para tener una idea general de la misma.

1. Sistema de usuarios.

Para empezar con el proyecto, se ha considerado apropiado crear un sistema de usuarios sencillo, más que nada para poder utilizarlo de referencia y así, con un simple formulario de usuario, solucionar casi la mayoría de problemas relevantes con respecto a la seguridad.

Esto es bastante importante, pues nos ayuda a tener una sesión abierta, y la podemos utilizar para comprobar que existe un usuario conectado con las credenciales correctas. Una vez el usuario cierra el navegador, la sesión se cierra y no se puede acceder a ninguna de las rutas que existen dentro de la aplicación, así consigue la aplicación impedir que los usuarios no deseados se metan dentro de la aplicación.

Mencionar que es un simple sistema de autenticación. No es una aplicación para que cada usuario tenga permisos distintos y vistas distintas.

2. Sistema de gestión de base de datos

En este caso, para la aplicación se ha decidido utilizar MySQL.

Aquí es donde nos guardaremos todos los datos relevantes que vayamos a necesitar en la aplicación:

- Aulas/departamentos
- Objetos
- Usuarios
- Sesión

Estas serán las 4 tablas que vamos a utilizar en nuestra base de datos para tener toda la información relevante que utilizamos en todos los requisitos de la aplicación.

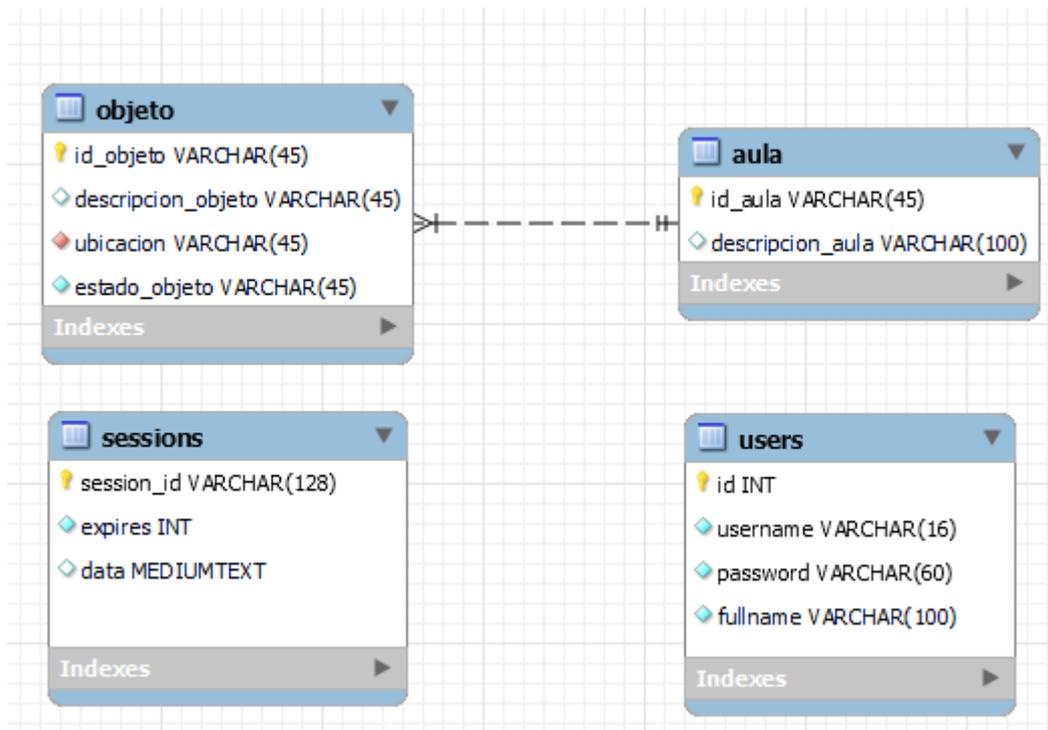


Figura 12. Diagrama EER de la base de datos

3. Gestión de datos

Para la gestión de datos de nuestra aplicación se van a realizar comunicaciones entre la propia aplicación y la base de datos, todo esto utilizando NodeJS, el cual nos permite “conectar” la base de datos con la aplicación mediante JavaScript.

Se utilizarán peticiones dos tipos de peticiones principalmente al servidor:

- **Peticiones GET**, nos permitirá obtener datos de la base de datos.
- **Peticiones POST**, nos permitirán añadir, modificar y eliminar información de la base de datos.

Así pues, todos los formularios que rellenemos, a la hora de crear nuevas aulas/departamentos, modificarlas/os o eliminarlas/os tendrán peticiones POST para hacer la modificación de estos datos. Lo mismo para los objetos, que también cuentan con formularios que permiten añadir, modificar o borrar los objetos que existen.

Por otra parte, a la hora de visualizar una página, o necesitar información, se utilizarán las peticiones GET. Esto se utiliza por ejemplo, a la hora de visualizar las distintas aulas/departamentos, o los objetos que existen dentro de estos.

4. Visualización de datos (aulas/departamentos y sus objetos)

Por último, la página se estructura de una forma bastante sencilla, para que el usuario no tenga ningún problema a la hora de orientarse en la aplicación.

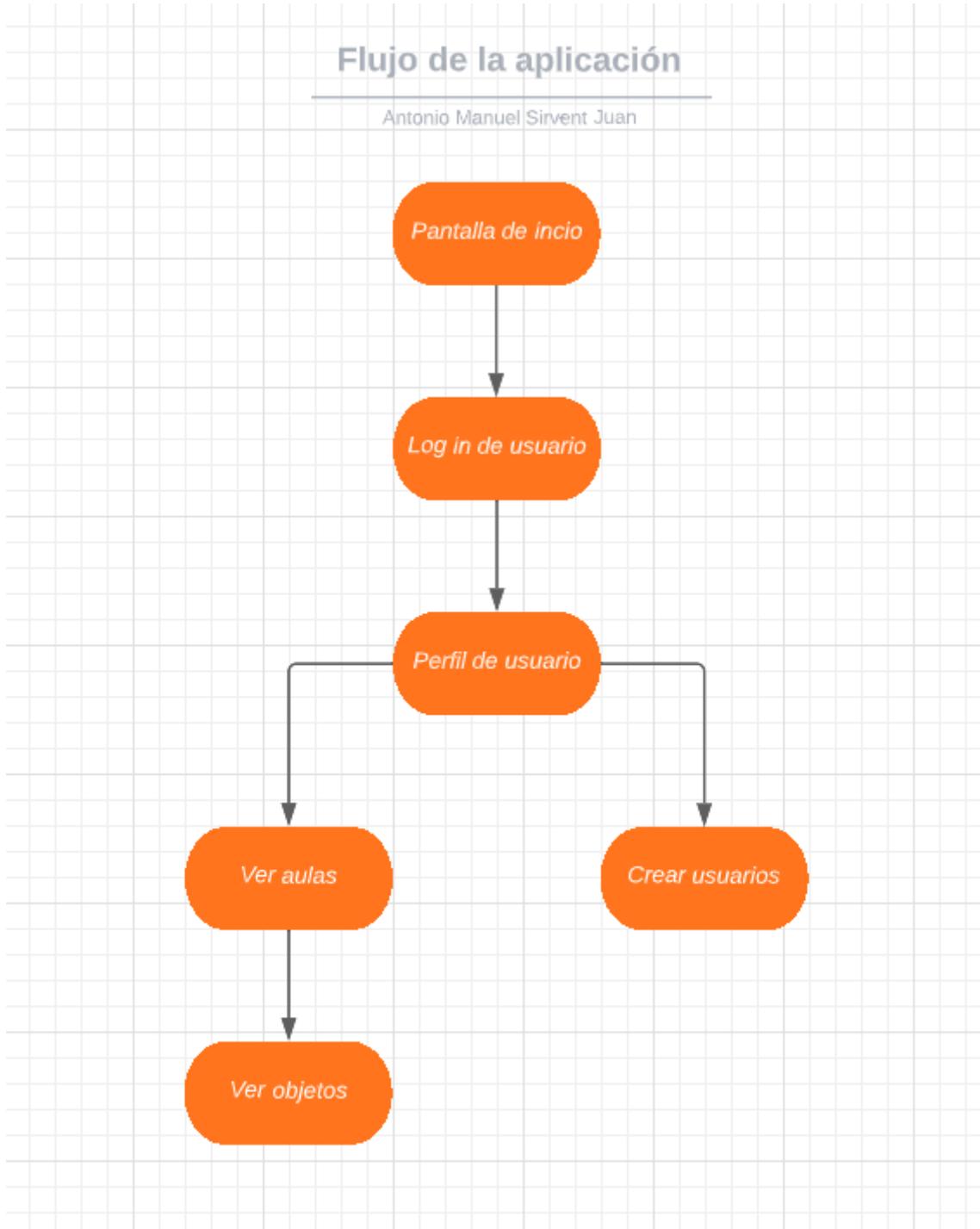


Figura 13. Flujo de la aplicación del cliente

4.3. Sistema de usuarios

La idea con el desarrollo de este proyecto es para que sea utilizado por el personal encargado de gestionar los temas de inventarios dentro del instituto IES Pare Arques.



Figura 14. Logotipo IES Pare Arques

Antes de empezar con el desarrollo del proyecto, se hicieron varias reuniones con el cliente, en este caso la entidad educativa IES Pare Arques para definir el alcance que tendría la misma.

Con todo esto en mente, se volvieron a hacer distintas reuniones, en la que se han ido haciendo actualizaciones del estado del proyecto, obteniendo el visto bueno por la entidad.

- **Datos necesarios para acceder a la aplicación;** al usuario se le mostrará una pantalla con un botón que proporcionará acceso a la ventana en la que el usuario se podrá logear, una vez en esta pantalla, necesitará de un nombre de usuario y una contraseña para poder acceder.



Figura 15. Log in de la aplicación web desarrollada

En este caso, ya que el sistema de usuarios es más por sistemas de protección que de funcionalidad, los datos necesarios de los usuarios son simples (usuario, nombre completo y contraseña), aunque las contraseñas se encuentren protegidas en la base de datos mediante métodos de encriptación también por distintas razones de seguridad.

```
mysql> SELECT * FROM users;
+----+-----+-----+-----+
| id | username | password | fullname |
+----+-----+-----+-----+
| 6 | IESPA | $2a$10$66M1DtYHd50d/GEJh5FJcui.V7VKJ/wlVU1Jnx9QA62Jd3mk6QHvq | IES Pare Arques |
+----+-----+-----+-----+
1 row in set (0.04 sec)
```

Figura 16. Visualización de usuarios y contraseñas en MySQL

Por otra parte, una vez identificados dentro de la aplicación, el usuario podrá crear nuevos usuarios según necesite, a pesar de que todos tendrán la misma visualización de la página, ya que los usuarios se utilizan como una forma de asegurar las rutas URL más que de temas de permisos/visualización de vistas diferentes.

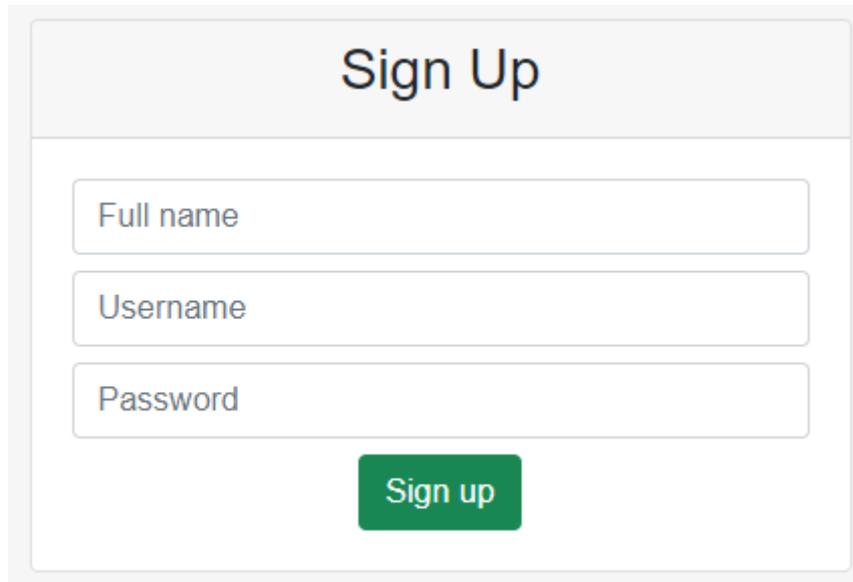
The image shows a 'Sign Up' form. At the top, the text 'Sign Up' is centered in a large, dark font. Below this, there are three input fields stacked vertically: 'Full name', 'Username', and 'Password'. Each field is a simple rectangular box with a thin border. At the bottom center of the form is a green rectangular button with the text 'Sign up' in white.

Figura 17. Formulario para registrar nuevos usuarios

4.4. Función principal (visualización del inventario)

Ahora vamos a pasar a la parte de la función principal, que es donde vamos a poder visualizar todas las aulas/departamentos y sus respectivos objetos dentro de cada una de ellas.

Una vez el usuario se haya autenticado correctamente, entrará en la pantalla del perfil, donde tendrá distintas opciones, una de ellas será ver el inventario de aulas, y como se ha comentado anteriormente tener la posibilidad también de crear nuevos usuarios a placer del cliente.

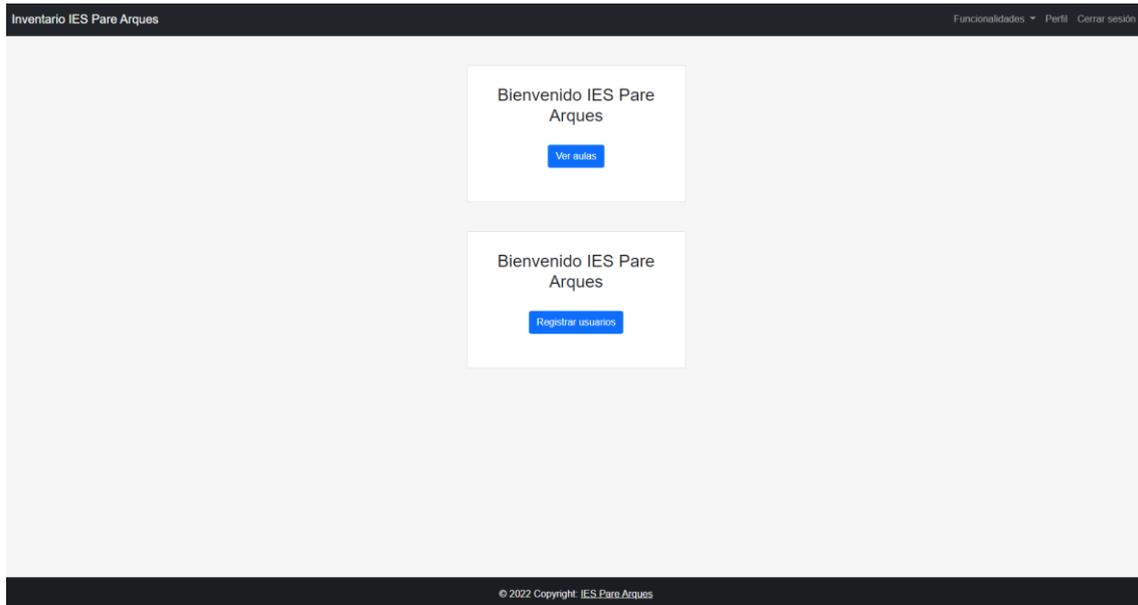


Figura 18. Visualización del perfil de la aplicación web

Una vez el usuario se encuentra en esta página, podemos ver distintos accesos a páginas dentro de la misma aplicación:

- Acceso a la página inicial, que es la página previa al Log in. Se accede a ella en la barra de navegación superior, en el nombre “Inventario IES Pare Arques”
- Acceso a las funcionalidades de la aplicación, en este caso son añadir nuevas aulas, y mostrar las aulas disponibles. (Añadir objetos también cuenta como una funcionalidad, pero por cómo se ha construido la aplicación no se ha considerado viable añadirlas aquí también). La barra de navegación que también tiene acceso al perfil es común para toda la aplicación, y cambia dependiendo si un usuario se encuentra autenticado, o aún no ha hecho log in en la página.



Figura 19. Barra de navegación para una persona no autenticada



Figura 20. Barra de navegación para una persona autenticada

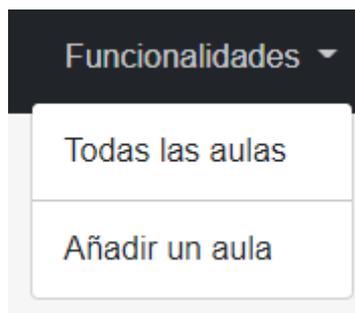


Figura 21. Visualización de las funcionalidades del inventario.

- En el cuerpo de la aplicación, se puede visualizar una carta de texto, que incluye una bienvenida con el nombre del usuario y el acceso a la página de visualización de aulas.



Figura 22. Carta de texto con acceso al inventario de aulas

- También se puede ver un poco más abajo, la carta de texto que incluye una bienvenida con el nombre del usuario y el acceso al registro de nuevos usuarios.

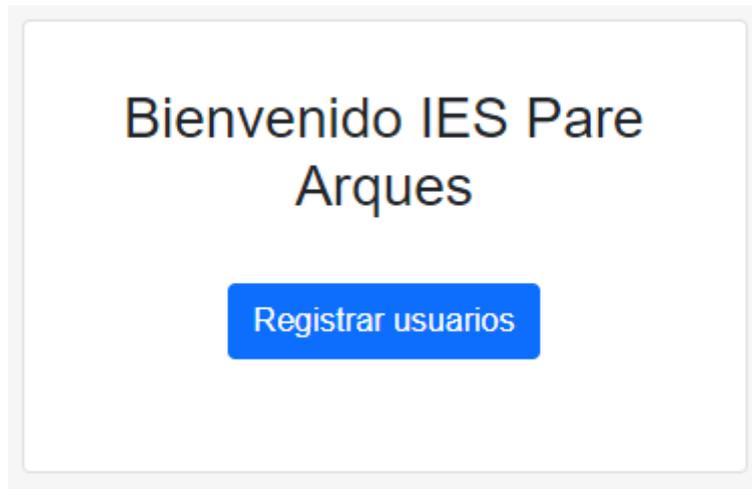


Figura 23. Carta de texto con acceso al formulario de registro de usuarios

Ahora, una vez ya hemos identificado todos los elementos de la pantalla del perfil, en la carta de texto con acceso al inventario de aula, podremos visualizar también, mediante cartas de texto las distintas aulas que hayan guardadas en la base de datos.



Figura 24. Visualización de las aulas guardadas en la base de datos.

Dentro de cada carta de texto, vamos a tener un enlace, en este caso el enlace se ha puesto justamente en cada ID de aula, para así hacerlo más intuitivo (perfil -> aulas/departamentos -> objetos).

También podemos observar que se han añadido botones tanto para editar como para borrar cada aula/departamento.

Podemos acceder al menú de crear nuevas aulas tanto mediante la barra de navegación superior, como de un botón que se ha proporcionado en la pantalla inferior derecha de la pantalla que añadirá nuevas aulas si nos encontramos en la página de aulas aula/departamento, y si nos encontramos en la página de objetos, nos creará un objeto dentro del aula/departamento en la que estemos.

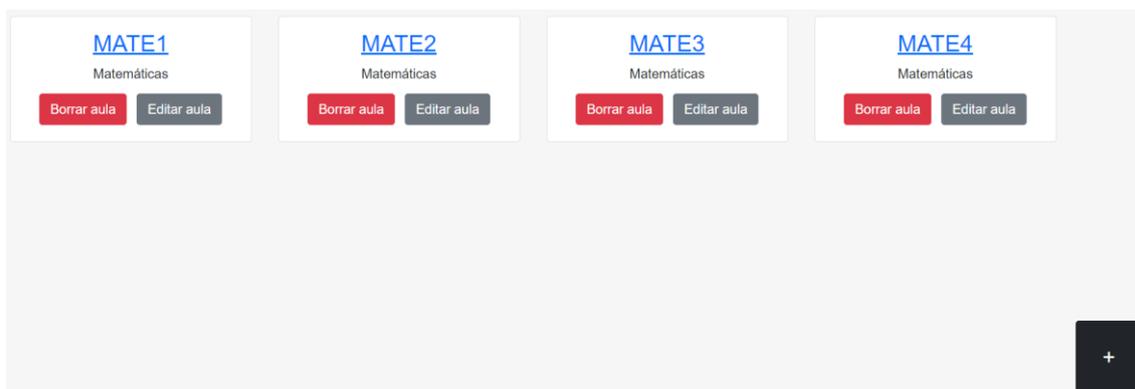


Figura 25. Botón inferior derecho para crear nuevas aulas/departamentos

Por último, la visualización de los distintos objetos ocurrirá cuando un usuario entre dentro del enlace en las cartas de texto de cada aula/departamento. Se necesita visualizar tanto los objetos como su estado actual. También se han añadido botones para editar y borrar los objetos. Además del botón de añadir objetos nuevos, que en este caso funciona como el botón que se ha explicado en la “[Figura 25. Botón inferior derecho para crear nuevas aulas/departamentos](#)”.

El botón quedará situado en la misma zona de la pantalla, solamente que la función de este cambia si se encuentra en la pantalla de visualización de aula, o se encuentra dentro de un aula para visualizar los objetos.



Figura 26. Carta de texto con información de un objeto en buen estado



Figura 27. Carta de texto con información de un objeto en mal estado



Figura 28. Carta de texto con información de un objeto dañado

4.5. Envío y recogida de datos a la base de datos

Para toda la parte de envío y recogida de datos sobre nuestra base de datos, se ha utilizado NodeJS, que con su herramienta npm (Node Package Management) nos permite instalar paquetes para utilizar nuestra base de datos en MySQL con nuestra aplicación que se encuentra programada con HTML5, CSS, Handlebars y JavaScript.

En este caso, se está trayendo y enviando datos constantemente a la base de datos mediante peticiones GET (que nos sirven para traernos información de la base de datos) como peticiones POST (que nos sirven para modificar datos de la base de datos).

Así pues, y teniendo una idea general de cómo funcionará nuestra aplicación, podremos mostrar más claramente su funcionamiento.

La recogida de datos ocurre en las siguientes situaciones:

- Cuando en el perfil visualizamos las aulas.
- Cuando en las aulas, dentro de una de ellas, visualizamos objetos.
- Cuando un usuario se autentica en la aplicación

La modificación de datos ocurre a la hora de:

- Modificar datos de aulas/departamentos.
- Modificar datos de objetos.
- Añadir nuevos objetos.
- Añadir nuevas aulas/departamentos.
- Eliminar datos de aulas/departamentos.
- Eliminar datos de objetos.
- Añadir nuevos usuarios a la aplicación.

4.5.1. Usuarios

Para la parte de usuarios, cuando se quiera enviar y recoger información, se hará cuando el usuario se autentique y se creen nuevos usuarios, el formulario para ver nuevos usuarios lo tenemos en la [“Figura 17. Formulario para registrar nuevos usuarios”](#).

Y para la parte del Log in, la tenemos en la [“Figura 15. Log in de la aplicación web desarrollada”](#).

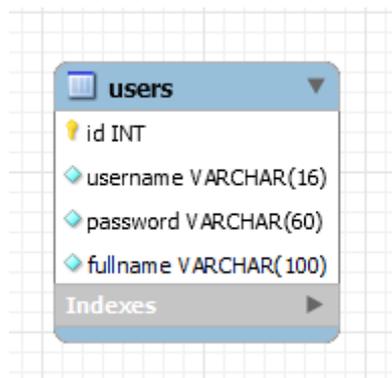


Figura 29. Información de los usuarios en la base de datos

4.5.2. Aulas

En el caso de las aulas, la única información que se va a guardar es la que se va a mostrar en las cartas de texto:

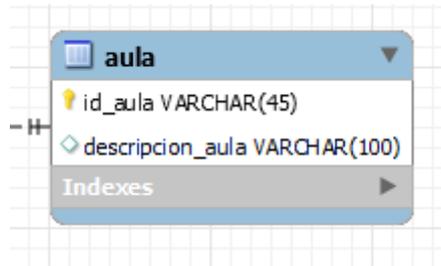


Figura 30. Información aulas/departamentos en la base de datos

No necesitamos guardar nada más sobre las aulas/departamentos, ya que simplemente con una breve descripción y un ID que identifique las distintas aulas/departamentos es suficiente.

El formulario para añadir nuevas aulas entonces será bastante simple:

Figura 31. Formulario de creación de una nueva aula

4.5.3. Objetos

Para los objetos, sí que vamos a necesitar traernos un poco más de información. Para este desarrollo se ha pensado conveniente que los objetos estén ligados a las aulas/departamentos, de manera que solamente puedan haber objetos creados dentro de aulas. En el desarrollo también se ha tenido en cuenta todos los problemas de inconsistencias de datos que puede ocurrir a la hora de eliminar aulas que contengan objetos.

Así que, para los objetos se ha decidido guardar la siguiente información:

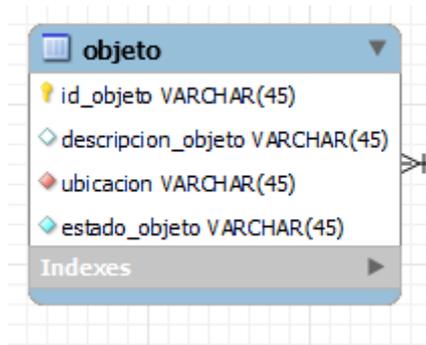


Figura 32. Información de la tabla objeto en la base de datos

En este caso, necesitamos, además del id del objeto y de la descripción del objeto, vamos a necesitar guardar la ubicación, y el estado del objeto. Como los objetos se crean siempre desde dentro de las aulas, esto nos va a ayudar a tener un parámetro para incluir dentro del campo que hace referencia a la ubicación, en este caso, la [Figura 33. Formulario de creación de un objeto nuevo](#)” muestra un objeto nuevo que se está creando dentro del aula CAS1.

A screenshot of a web form for creating a new object. The form is enclosed in a light grey border. It contains four input fields stacked vertically: 'Nombre del objeto', 'Descripcion del objeto', 'Buen estado', and 'CAS1'. The 'CAS1' field is highlighted with a grey background. At the bottom left of the form, there is a green button with the text 'Save' in white.

Figura 33. Formulario de creación de un objeto nuevo

5. Pruebas

Para este apartado se ha pensado conveniente crear una parte de “testing” de la aplicación, en la que se compruebe que todas las funcionalidades de la aplicación funcionan perfectamente, y que no vamos a tener ningún problema cuando el cliente vaya a utilizar la aplicación.

Se ha considerado dividir el apartado de pruebas en distintos subapartados que cubren todo el alcance de la aplicación.

Por otra parte, todas las pruebas se van a realizar viendo los datos tanto en la parte del back-end, como en la parte visual del usuario de la aplicación. De esta forma también nos vamos a asegurar de que todo esté funcionando correctamente, y en caso de cualquier posible error, saber dónde se está produciendo exactamente. Para comprobar la parte visual, como se ha comentado en el punto “1.4.6. Comunicación con el usuario de la aplicación mediante connect-flash y navegación” se utilizarán mensajes de comunicación visual, y para comprobar la parte del back-end, veremos que todo se está modificando en la base de datos correctamente mediante distintas consultas en un terminal que podremos abrir en nuestro proyecto.

5.1. Seguridad de la aplicación

Para comprobar la seguridad de la aplicación, en la Introducción, “[1.2. Antecedentes y descripción del Proyecto](#)”, se ha creído conveniente dotar a la aplicación de una parte de seguridad, ya que en el caso de que haya que proteger cualquier información que se posea en la aplicación, si no se le dota de una seguridad mínima, la aplicación no será viable.

Entonces se ha pensado que sería adecuado tener distintas formas de asegurar la aplicación, como ya se ha comentado brevemente, la seguridad que le vamos a dar a la aplicación vendrá dada por:

- Un sistema de autenticación que nos permita asegurar todas las direcciones url para que no se pueda acceder a ellas sin una identificación.
- Creación de usuarios base, y a partir de ahí, la posibilidad de añadir distintos usuarios dentro de la aplicación, además de toda la parte de seguridad de cifrado de contraseñas y etc..
- Que el entorno en el que se desarrolle sea un entorno de aplicación web, para así facilitar la portabilidad a cualquier otro dispositivo que se desee utilizar.

Para generar el sistema de autenticación vamos a tener que generar unas funciones en JavaScript que nos permitan esta comprobación. Para comprobar que todo funciona correctamente vamos a comprobarlo mediante el uso normal de la aplicación, de manera que nos encontraremos en la carta de texto referente al log in (o sign in, en este caso) y introduciremos las credenciales. Hay dos casos esperados:

- Log in correcto: en este caso, ocurrirá una redirección a la página del perfil de usuario, donde veremos un mensaje de bienvenida, al producirse el log in correcto, y las cartas de texto con las opciones de visualizar las aulas y poder registrar nuevos usuarios.

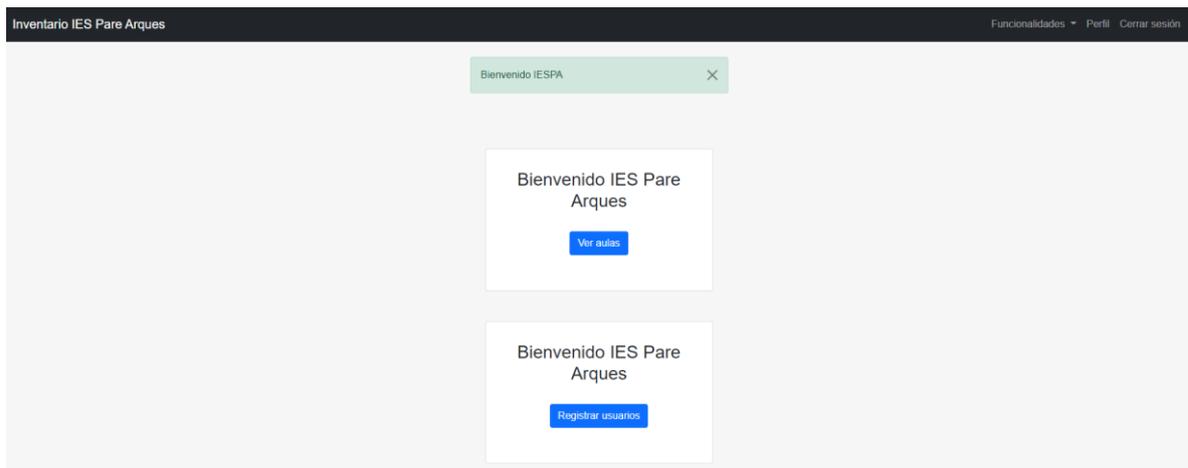


Figura 34. Log in correcto dentro de la aplicación

- Log in incorrecto: en este caso, ocurrirá una redirección a la página de log in otra vez, para que el usuario se vuelva a autenticar correctamente con unas credenciales válidas, además de mostrar por pantalla un mensaje de error.

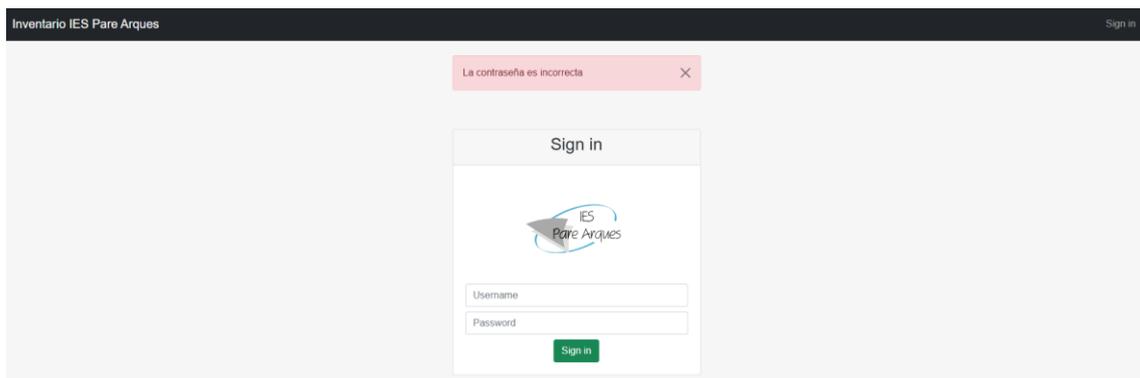


Figura 35. Log in incorrecto dentro de la aplicación

También se ha definido un método para asegurar de que todas las rutas de la aplicación se aseguren de que el usuario no se pueda meter a cualquier enlace de la aplicación, que incluiremos en todas las cabeceras de los métodos que utilicemos para cargar las páginas.

```

1  module.exports = {
2    isLoggedIn(req, res, next) {
3      if (req.isAuthenticated()) {
4        return next();
5      }
6      return res.redirect('/signin');
7    },
8
9    isLoggedIn(req, res, next) {
10     if (!req.isAuthenticated()) {
11       return next();
12     }
13     return res.redirect('/profile');
14   }
15 };

```

Figura 36. Código referente al método de autenticación

Es importante mostrar esto, porque es el código en el que se va a basar la seguridad de nuestra aplicación, ya que añadiendo el método `isLoggedIn` a cualquier cabecera de cualquier petición GET, si un usuario no está autenticado, se le redirija a la página para autenticarse con unas credenciales correctas. Sin esto, las URL de la aplicación quedarían descubiertas, y teniendo el enlace de la dirección que cualquier usuario indeseado pueda acceder a la aplicación.

```

48  router.post('/edit/:id_aula', isLoggedIn, async (req, res) => {

```

Figura 37. Cabecera de la petición POST de editar un aula

5.2. Guardado de datos en la base de datos

Para la comprobación de guardado de datos en la base de datos, y que toda la información aplicación – base de datos está viajando correctamente se van a utilizar los mensajes de error y éxito, como se ha hecho anteriormente, además de ver mediante consultas a la base de datos la información en todo momento. Así pues, como se ha definido anteriormente en el alcance de la aplicación, los casos en los que se envía y modifica datos en la base de datos es en los siguientes casos:

- Añadir, modificar o eliminar aulas/departamentos.

Para comprobar que este apartado funciona correctamente, vamos a utilizar los formularios creados para las aulas/departamentos, y ver su comportamiento en distintas casuísticas para ver que todo funciona correctamente. De manera que primero, vamos a añadir una nueva aula/departamento, mediante el formulario que podemos observar en “[Figura 31. Formulario de creación de una nueva aula](#)” crearemos una aula nueva. Se pueden esperar dos casuísticas distintas:

- Creación correcta del aula (no hay datos repetidos): nos redireccionará al visualizador de aulas, donde podremos ver la nueva aula creada. También podremos observar un mensaje de éxito de la aplicación, indicando que el aula/departamento se ha creado correctamente. En la parte del back-end, podremos ver que los datos se encuentran en la base de datos. Se toma como referencia, antes de la modificación el visor de aulas de la imagen “[Figura 24. Visualización de las aulas guardadas en la base de datos.](#)”

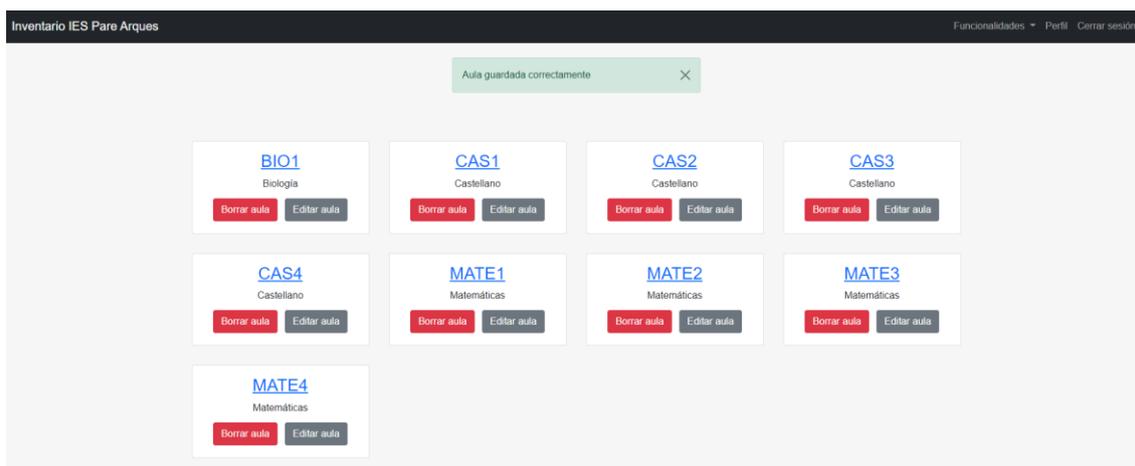


Figura 38. Guardado de aula (BIO1) correctamente

```
mysql> SELECT * FROM aula;
+-----+-----+
| id_aula | descripcion_aula |
+-----+-----+
| BIO1    | Biología         |
| CAS1    | Castellano       |
| CAS2    | Castellano       |
| CAS3    | Castellano       |
| CAS4    | Castellano       |
| MATE1   | Matemáticas     |
| MATE2   | Matemáticas     |
| MATE3   | Matemáticas     |
| MATE4   | Matemáticas     |
+-----+-----+
9 rows in set (0.00 sec)
```

Figura 39. Guardado de aula (BIO1) correctamente en la base de datos

- Creación errónea del aula (hay datos repetidos): esta casuística se dará cuando un id_aula se encuentre repetido, ya que por consistencia de datos esto no puede ser así, se ha adecuado el código del proyecto para que en estos casos, se vuelva a redireccionar al visualizador de aulas con un mensaje de error. Así que vamos a intentar cambiar el nombre del id_aula de BIO1 a MATE1. Mencionar que para añadir una nueva aula/departamento funcionaría de la misma forma que para editar, también daría un error, y volvería a redireccionar a la página de visualización de aulas/departamentos.

En este caso, el resultado en la base de datos es el mismo que se puede observar en [“Figura 32. Información de la tabla objeto en la base de datos”](#)

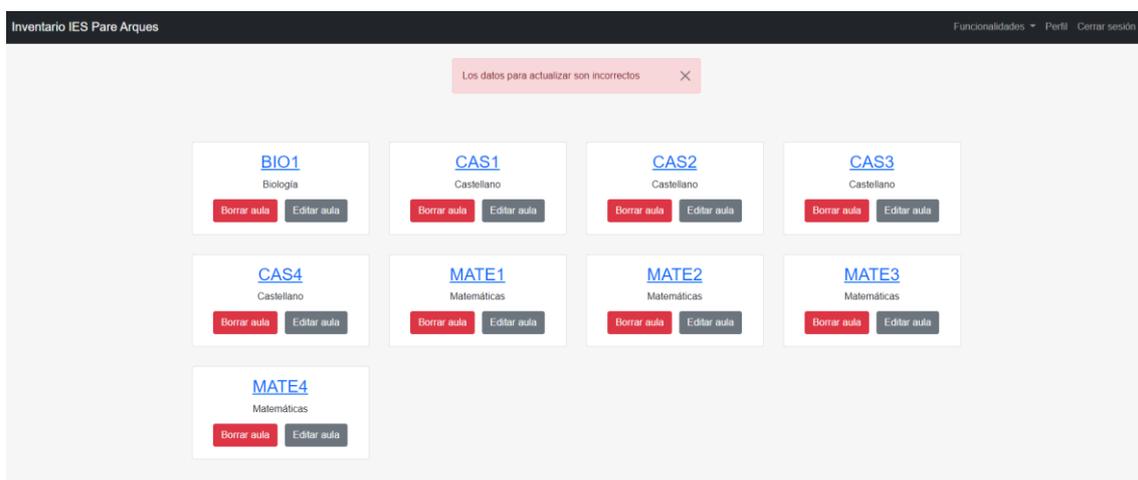


Figura 40. Intento de actualizar datos de (BIO1 a MATE1)

- Eliminación correcta del aula: esta casuística se dará cuando se presione el botón de Borrar aula, en un principio esto solamente debería dar problemas cuando se quiera borrar el aula/departamento y contenga objetos, pero agilizar el proceso de edición de inventario, se ha pensado que si se quiere borrar un aula/departamento, también es lógico que todos los objetos dentro del aula sean borrados también. Una vez se presiona este botón, se redirigirá a la pantalla de visualización de aulas/departamentos y se mostrará un mensaje de éxito referente a este borrado. Para esta prueba se ha borrado el aula BIO1.

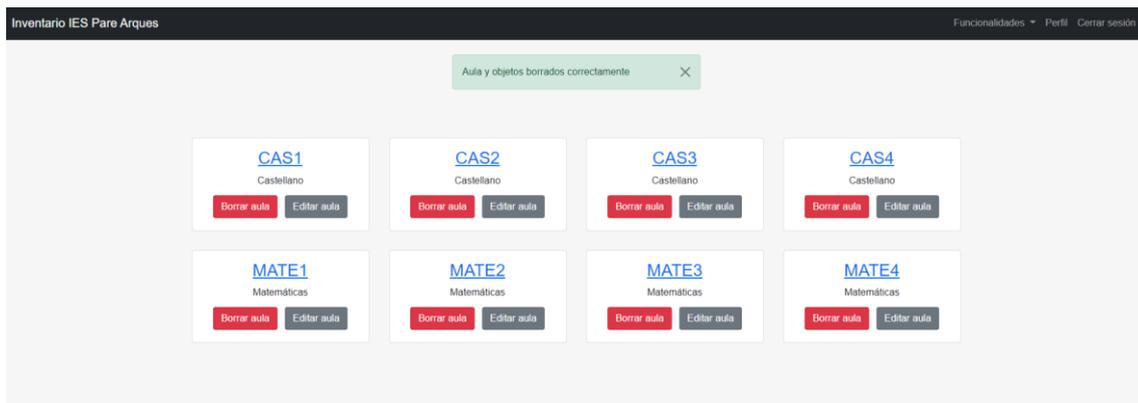


Figura 41. Visualización del mensaje de éxito al borrar BIO1

```
mysql> SELECT * FROM aula;
+-----+-----+
| id_aula | descripcion_aula |
+-----+-----+
| CAS1    | Castellano       |
| CAS2    | Castellano       |
| CAS3    | Castellano       |
| CAS4    | Castellano       |
| MATE1   | Matemáticas     |
| MATE2   | Matemáticas     |
| MATE3   | Matemáticas     |
| MATE4   | Matemáticas     |
+-----+-----+
8 rows in set (0.00 sec)
```

Figura 42. Visualización de las aulas/departamentos en la base de datos al borrar BIO1

- Añadir, modificar o eliminar objetos dentro de las aulas/departamentos.

Para comprobar que este apartado funciona correctamente, vamos a utilizar los formularios creados para los objetos, y ver su comportamiento en distintas casuísticas para ver que todo

funciona correctamente. Así que primero vamos a añadir mediante el formulario que podemos observar en “[Figura 33. Formulario de creación de un objeto nuevo](#)” un objeto nuevo dentro de un aula/departamento cualquiera, para este caso, crearemos una nueva aula como se ha hecho anteriormente donde hacer pruebas, el aula se llamará MUS1, que hará referencia al aula de música. Se pueden esperar distintas casuísticas:

- Creación correcta del objeto (no hay IDs repetidos): esta casuística se dará cuando se introduzca información correcta sobre el objeto, en este caso el único dato sensible es el ID, y dará problemas solamente cuando se encuentre repetido dentro de la base de datos. Una vez creado el objeto se redireccionará al visualizador de objetos con un mensaje de éxito. Ahora añadiremos un objeto al aula MUS1.

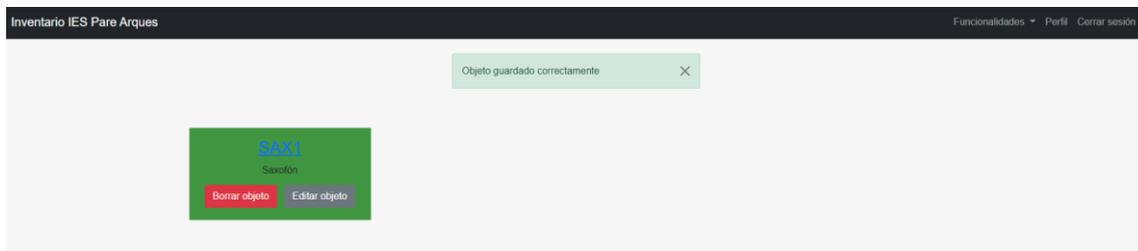


Figura 43. Objeto añadido al aula MUS1

```
mysql> SELECT * FROM objeto WHERE ubicacion = 'MUS1';
+-----+-----+-----+-----+
| id_objeto | descripcion_objeto | ubicacion | estado_objeto |
+-----+-----+-----+-----+
| SAX1      | Saxofón            | MUS1      | Buen estado   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Figura 44. Visualización del objeto añadido al aula MUS1

- Creación errónea del objeto: esta casuística ocurrirá cuando se quiera o bien editar un objeto, o añadir un nuevo objeto con un ID repetido, una vez hayamos confirmado que queremos guardar los cambios, la aplicación no lo va a permitir y nos redireccionará otra vez a la página de visualización de objetos con un mensaje de error. Como ya tenemos el objeto SAX1 creado, vamos a intentar duplicarlo para ver si esto es funciona correctamente.

El estado de la base de datos quedará igual que en “[Figura 44. Visualización del objeto añadido al aula MUS1](#)” ya que el objeto no se ha añadido a la base de datos.

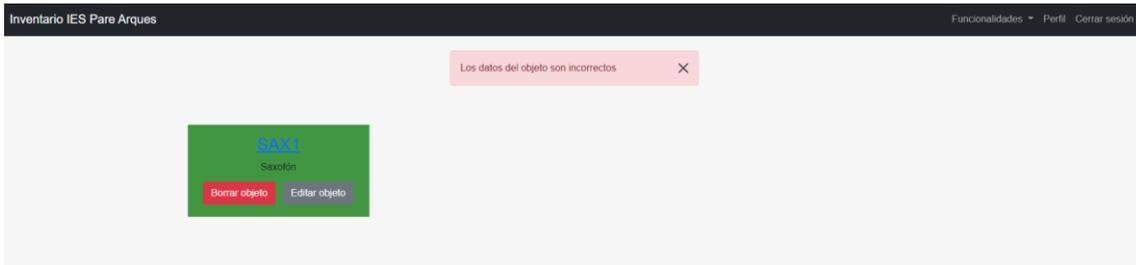


Figura 45. Intento de duplicar el objeto con ID SAX1 en la base de datos

- Eliminación correcta del objeto: en este caso, siempre se podrá eliminar un objeto, hay dos formas distintas de eliminar el objeto, una mediante la eliminación del objeto dentro del aula, o de la eliminación del aula/departamento. Una vez se elimine el objeto, se redirigirá a la página de visualización de objetos con un mensaje de éxito. En el caso de las aulas/departamentos, cuando se borre uno de los dos, se borrarán todos los objetos que se encuentren dentro. En este caso se redirigirá a la página de visualización de aulas/departamentos con un mensaje de éxito de borrado de tanto aula/departamento como objetos.

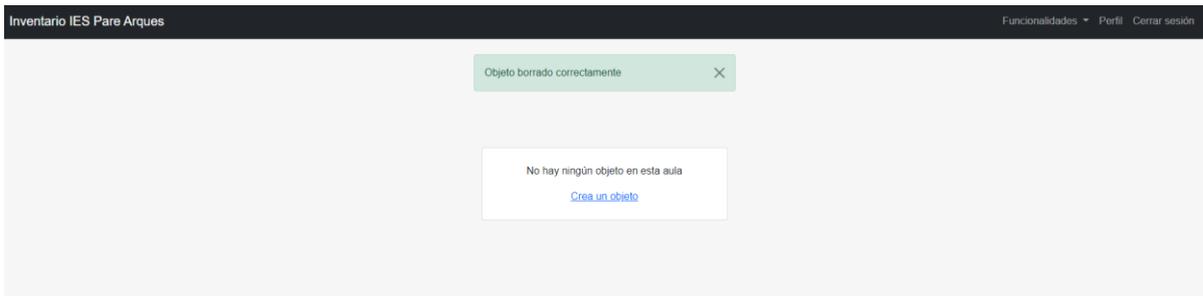


Figura 46. Visualización de la eliminación del objeto SAX1

```
mysql> SELECT * FROM objeto WHERE ubicacion = 'MUS1';  
Empty set (0.00 sec)
```

Figura 47. Visualización de la eliminación del objeto SAX1 en la base de datos

- Registrar nuevos usuarios.

Para comprobar que este apartado funciona correctamente, vamos a utilizar los formularios creados para nuevos usuarios, y ver su comportamiento en distintas casuísticas para ver que todo funciona correctamente. Así que primero vamos a añadir mediante el formulario que podemos observar en “[Figura 17. Formulario para registrar nuevos usuarios](#)” un usuario nuevo. En la

creación de un nuevo usuario, tampoco se espera que haya ningún problema, principalmente queremos ver que se añade el usuario correctamente a la base de datos. Así que crearemos un nuevo usuario, se espera que al crear un nuevo usuario se cree con un id interno, además de que en la base de datos se encripte la contraseña utilizando el módulo bcrypt que se menciona en “[2.2. Tecnologías actuales](#)”, cuando se hablan de los módulos utilizados en NodeJS. Creamos un nuevo usuario mediante el formulario con nombre completo Toni Sirvent, y nombre de usuario HZTONI.

```
mysql> SELECT * FROM users;
+-----+-----+-----+-----+
| id | username | password | fullname |
+-----+-----+-----+-----+
| 6 | IESPA | $2a$10$10YcF8g2NzgBSI6qxsbmJebq.HFmwm daPpEvwErkVwWFDt22GyTAu | IES Pare Arques |
| 7 | HZTONI | $2a$10$aJQLo41Y1kNRyG1gB21VcesKkY588Eesny/xiSqMvBUwbVNMP2F5q | Toni Sirvent |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Figura 48. Visualización de los usuarios en la base de datos

5.3. Conclusiones de las pruebas realizadas

Como hemos podido ver en los apartados anteriores “[5.1. Seguridad de la aplicación](#)” y “[5.2. Guardado de datos en la base de datos](#)” se han hecho todas las comprobaciones referentes a seguridad y guardado de datos en la base de datos.

Concluimos así la parte de pruebas realizadas, y podemos ver que el desarrollo de tanto las funcionalidades requeridas por la parte de cliente, como las funcionalidades que se han sugerido desde la parte de la empresa para dotar a la aplicación de una parte de seguridad han sido satisfactorias, ya que todo funciona como se esperaba en todas las casuísticas posibles.

Esta parte es muy importante cuando se tiene que presentar un desarrollo de una aplicación al cliente, ya que mediante esto comprobamos el funcionamiento de la aplicación en los distintos casos donde el código de la aplicación puede fallar y puede llegar a dar errores, además nos aseguramos de que una vez se haya entregado la aplicación al cliente todos los errores estén controlados y no existan problemas a la hora de utilizar la aplicación.

Si bien es cierto, en muchos casos de desarrollo, una vez ya la aplicación está funcionando, pueden dar lugar distintos errores menores. Esta es la razón por la que la mayoría de desarrollos incluyen distintas versiones, en la que el número de las versiones suele incluir el número de “releases”, o veces que se ha subido una nueva versión de la misma aplicación que soluciona problemas ocurrían en la anterior versión.

6. Conclusiones generales

Al finalizar el desarrollo de esta aplicación web, que principalmente es para gestionar el inventario dentro de una entidad educativa, en este caso el instituto IES Pare Arques, podemos sacar distintas conclusiones. Principalmente mencionar todo el proceso de aprendizaje que se ha obtenido en el uso de toda la tecnología que se menciona en "[2. Estado del arte](#)", y como ha sido necesario todo esto para alcanzar el desarrollo completo de las funcionalidades que ha pedido el cliente y las sugerencias por parte de la empresa.

Se ha trabajado en la aplicación principalmente durante el segundo cuatrimestre del curso, y así se ha aprendido a desarrollar paso a paso una aplicación, como hemos podido ver en las "[1.4. Fases del Desarrollo](#)", en las que se ha ido montando toda la aplicación desde el inicio.

Esta aplicación se ha desarrollado esencialmente para el instituto IES Pare Arques, ofreciendo al cliente la posibilidad de realizar el inventario de una forma digital, de forma fácil y sin tener que introducir datos innecesarios.

Es bastante importante mencionar la parte de "[5. Pruebas](#)" en la conclusión, ya que mediante las distintas pruebas que se han hecho, podemos observar que tanto la parte de visualización de datos por parte del cliente, tanto como el guardado de datos en la parte del back-end funciona perfectamente y tiene en cuenta las posibles inconsistencias de datos que se puedan dar en la aplicación.

A pesar de que esta aplicación está diseñada exclusivamente para el instituto IES Pare Arques, se ha pensado que, para cualquier otra entidad que necesite disponer de un inventario simple para administrar sus departamentos y las cosas que se guarden en ellos, la aplicación también sea funcional, pues solamente habría que cambiar los logos y alguna que otra nomenclatura, pero esto son cambios más superficiales y no afectan para nada a la funcionalidad de la aplicación, así que no habría ningún problema en adaptarla a otras entidades.

En conclusión, se ha desarrollado una aplicación que ha proporcionado experiencia al alumno en tanto el ámbito de desarrollo de aplicaciones, como en el uso de distinta tecnología que era nueva para el alumno.

6.1. Problemas encontrados

A la hora de desarrollar la aplicación se han ido encontrando distintos problemas, tanto relacionados con la parte del back-end, en el cual nos han surgido distintos problemas a la hora de intentar hacer la conexión con NodeJS.

También por otra parte también se han tenido distintos problemas a la hora de utilizar parámetros en nuestras direcciones URL para enviar datos de una vista a otra y tener una estructura de rutas más descriptiva para el usuario.

Por último, mencionar que se tuvieron bastantes problemas también para conseguir que cuando se borrara un aula/departamento, se borrarán también los objetos que esta aula/departamento contenía, pues esto genera muchos problemas de consistencia de datos.

6.2. Solución de los problemas encontrados

Para la solución de los distintos problemas, al ser problemas más de permisos de bases de datos, versiones de programas, y de incompatibilidad más que problemas de código, se ha intentado buscar las distintas soluciones en internet, como, por ejemplo:

- Problemas para ver el servidor MySQL en la herramienta MySQL Workbench [14]
- Poder acceder a MySQL mediante terminal en VSCode [15]
- Problemas para actualizar datos de dos tablas que se encuentren unidas mediante una clave ajena [16]

Estos son los casos que más problemas han dado a la hora de utilizar la aplicación, la solución de ellos ha venido por foros experimentados [14][15][16] donde se ha conseguido la información necesaria para solventar estos problemas. Para temas más orientado a código, se han recurrido a páginas más de documentación que de foros experimentados en este caso [4][6][8].

6.3. Tecnologías aprendidas

Este proyecto se he considerado posible desde la parte del alumno, ya que se poseen conocimientos en bases de datos, y en alguno de los lenguajes que se utilizan. Así pues, el alumno tenía conocimiento de:

- MySQL, y consultas a la base de datos mediante queries.
- HTML5 y CSS.

En este caso, se ha tenido que aprender acerca de toda la tecnología que no se ha incluido en los dos puntos anteriores:

- JavaScript.
- NodeJS.
- Handlebars.

Así pues, se ha destinado una buena parte del tiempo del desarrollo del proyecto a ver qué posibilidades, y aprender estas tecnologías para que fueran satisfactorias una vez utilizadas en el desarrollo de la aplicación. También se ha adquirido conocimientos en varios módulos que nos ofrece NodeJS mediante su herramienta NPM (Node Package Manager).

6.4. Relación del trabajo desarrollado con los estudios cursados

Si bien es cierto que en la carrera no se suele dar JavaScript, si no asignaturas más orientadas a programación en Java, se ha pensado que el proyecto sería igual de viable.

En un principio este proyecto estaba destinado a desarrollarse en Java y mediante HTML5, y MySQL teniendo en principal objetivo una de las asignaturas de 3° del grado de ingeniería informática: **Tecnología de sistemas de información en la red**, ya que en esta asignatura se suelen ver temas más relacionados con el desarrollo web. No obstante, las posibilidades que ofrece la combinación de Java y HTML5, son bastante “estáticas”, así que se optó por ir un poco más allá de lo aprendido en las asignaturas de la carrera y dar el paso a programar con JavaScript la parte funcional, para así poder dotar a la aplicación de una flexibilidad superior a que si hubiese estado desarrollada con Java.

También hay que destacar la relación del desarrollo con todas las asignaturas relacionadas con bases de datos, como pueden ser **Bases de datos y sistemas de información**, y la asignatura optativa de **Diseño y gestión de bases de datos**.

7. Futuras líneas de Desarrollo

En este punto, se expondrán distintas mejoras que pueden ayudar mucho a las distintas funcionalidades ya existentes, y puede mejorar la aplicación, tanto en diseño, como en ampliación de funcionalidad.

7.1. Ingesta y obtención de aulas/departamentos mediante protocolos de ETL.

Antes de empezar con la explicación del porqué de esta futura línea de desarrollo, hay que mencionar qué es un protocolo ETL.

Un protocolo ETL, como su nombre indica, es un protocolo orientado a Extraer, Transformar y Cargar. Con esto se quiere hacer referencia a las tres fases que se pueden dar en este protocolo.

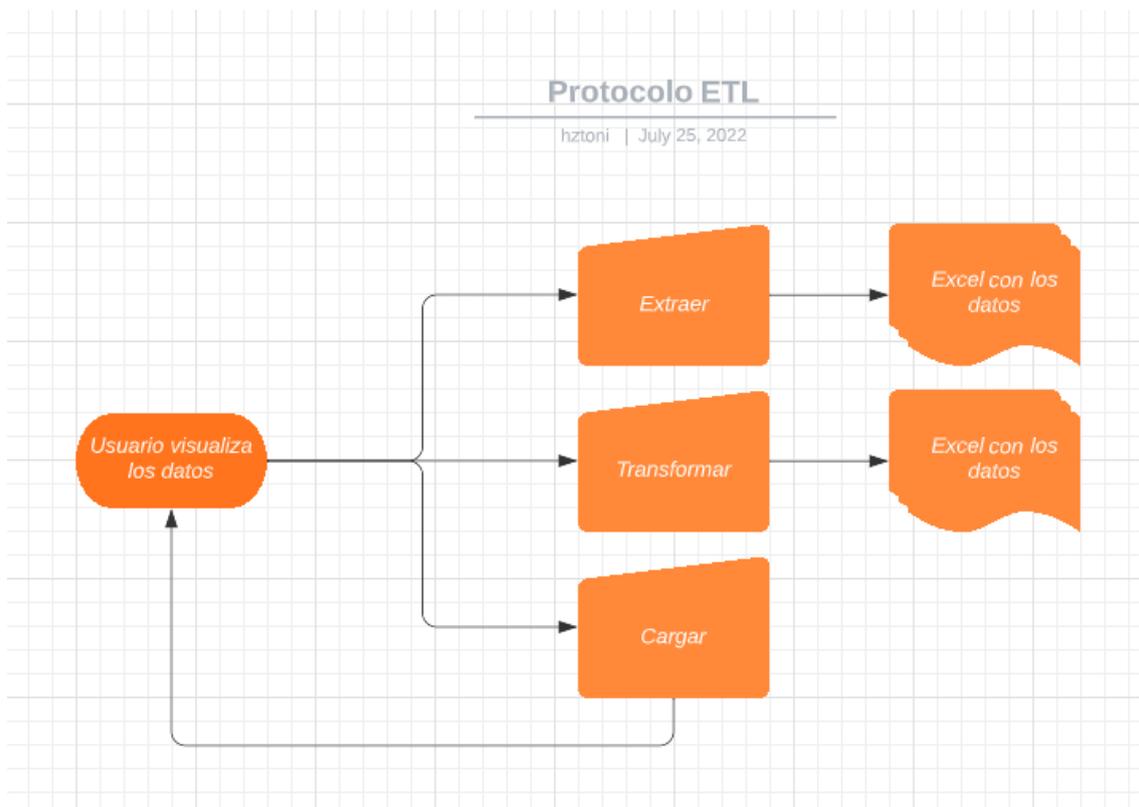


Figura 49. Diagrama de flujo de como funcionaría el protocolo ETL

Además, aparte de ser un proceso que engloba la carga, la transformación y la extracción de los datos, también tiene en cuenta que se tienen que comprobar que los datos que se están tanto como introduciendo como saliendo del sistema son correctos y válidos.

El desarrollo de la aplicación, solamente contempla la carga y visualización de datos desde la base de datos, hasta la aplicación, y viceversa, que, a pesar de que cumpla con todas las funcionalidades pedidas para el cliente, si que es verdad que para entidades muy grandes esta

funcionalidad puede ser bastante tediosa, comentar que se ha tenido en cuenta que en el instituto IES Pare Arques no se tienen que guardar grandes cantidades de aulas/departamentos. Entonces se ha contemplado que en futuras líneas de desarrollo se pueda tener en cuenta esta mejora, aunque compleja, bastante importante en la aplicación.

7.2. Ingesta y obtención de objetos mediante archivos protocolos de ETL.

De igual modo que hemos visto en “[7.1.Ingesta y obtención de aulas/departamentos mediante protocolos de ETL.](#)”, se ha estado pensando aplicar el mismo protocolo de ETL a los objetos, ya que funcionan de una manera similar a las aulas/departamentos.

La aplicación igual que se ha comentado anteriormente, también tiene un proceso de añadir objetos que si se pretende utilizar en inventarios bastante grandes puede dar lugar a una tarea tediosa, entonces se ha pensado que también sería adecuado todo este proceso para los objetos también.

7.3. Creación de elementos buscadores de aulas/departamentos y/o objetos

Se ha pensado también, que sería bastante adecuado para la aplicación, generar distintos elementos buscadores, tanto en la parte de los objetos como en la parte de las aulas/departamentos, por las mismas razones anteriores, en inventarios grandes igual buscar el aula/departamento que se necesita es bastante tedioso.

En el diseño original, como hemos comentado anteriormente, el diseño de la aplicación se ha tenido en cuenta para un instituto, entonces la cantidad de datos que se pretende manipular no es tan grande la de almacenes, y también, dentro de la base de datos, los datos se almacenan alfabéticamente, y así es como se muestran en el visor de aulas/departamentos y objetos.

No obstante, en caso de tener la opción de escalar la aplicación a otras entidades/empresas que vayan a manejar grandes cantidades de datos, se podría añadir elementos buscadores de las distintas aulas/departamentos y objetos, por eso se ha considerado que debería estar futuras líneas de desarrollo.

7.4. Mejorar la parte visual de la aplicación.

En último punto como mejora se ha pensado que, al utilizar los elementos visuales de Bootstrap, quizás la aplicación pueda llegar a verse un poco sencilla en lo que respecta a la visión general de la aplicación. No obstante, se pueden explorar distintos diseños y formas de mostrar la información con el fin de mejorar la aplicación visualmente.

8. Bibliografía

- [1]. Frankier Flores, qué es Visual Studio Code y sus Ventajas. (Julio, 2022). [Online]
<https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- [2]. Documentación sobre Visual Studio Code. (Julio, 2022). [Online]
<https://code.visualstudio.com/docs>
- [3]. Juan Diego Pérez Jiménez, qué es HTML5. (Enero, 2019). [Online]
<https://openwebinars.net/blog/que-es-html5/>
- [4]. ManzDev (usuario en redes sociales), Documentación sobre HTML5. (Junio, 2022).
[Online] <https://lenguajehtml.com>
- [5]. ManzDev, (usuario en redes sociales), qué es CSS. (Junio, 2022). [Online]
<https://lenguajecss.com/css/introduccion/que-es-css/>
- [6]. ManzDev, (usuario en redes sociales), Documentación sobre CSS. (Junio, 2022).
[Online] <https://lenguajecss.com/css/>
- [7]. ManzDev, (usuario en redes sociales), qué es JavaScript. (Junio, 2022). [Online]
<https://lenguajejs.com/javascript/introduccion/que-es-javascript/>
- [8]. ManzDev, (usuario en redes sociales), Documentación sobre JavaScript. (Junio, 2022).
[Online] <https://lenguajejs.com/javascript/>
- [9]. Handlebars, Documentación sobre Handlebars. (Junio, 2022). [Online]
<https://handlebarsjs.com/>
- [10]. Jesús Lucas, qué es NodeJS (Septiembre, 2019). [Online]
<https://openwebinars.net/blog/que-es-nodejs/>
- [11]. NodeJS, Documentación NodeJS (Junio, 2022). [Online]
<https://nodejs.org/es/docs/>
- [12]. Angel Robledano, qué es MySQL. (Septiembre, 2019). [Online]
<https://openwebinars.net/blog/que-es-mysql/>

- [13]. MySQL, Documentación MySQL. (Junio, 2022). [Online]
<https://dev.mysql.com/doc/>
- [14]. MySQL, bugs MySQL. (Febrero, 2021). [Online]
<https://bugs.mysql.com/bug.php?id=102690>
- [15]. MySQL, problema con reconocimiento en VSCode. (Mayo, 2020). [Online]
<https://www.weblonia.com/foro/tema/209-mysql-no-se-reconoce-como-un-comando-interno-o-externo-programa-o-archivo-por-lotes-ejecutable>
- [16]. MySQL, problema con actualización de tablas, (Julio, 2021). [Online]
<https://stackoverflow.com/questions/5005388/cannot-add-or-update-a-child-row-a-foreign-key-constraint-fails>
- [17]. NPM, Node Package Manager: express, (Mayo, 2022). [Online]
<https://www.npmjs.com/package/express>
- [18]. NPM, Node Package Manager: express-handlebars, (Mayo, 2022). [Online]
<https://www.npmjs.com/package/express-handlebars>
- [19]. NPM, Node Package Manager: express-session, (Mayo, 2022). [Online]
<https://www.npmjs.com/package/express-session>
- [20]. NPM, Node Package Manager: mysql, (Mayo, 2022). [Online]
<https://www.npmjs.com/package/mysql>
- [21]. NPM, Node Package Manager: express-mysql-session, (Mayo, 2022). [Online]
<https://www.npmjs.com/package/express-mysql-session>
- [22]. NPM, Node Package Manager: bcryptjs, (Mayo, 2022). [Online]
<https://www.npmjs.com/package/bcryptjs>
- [23]. NPM, Node Package Manager: passport, (Mayo, 2022). [Online]
<https://www.npmjs.com/package/passport>

[24]. NPM, Node Package Manager: passport-local, (Mayo, 2022). [Online]

<https://www.npmjs.com/package/passport-local>

[25]. NPM, Node Package Manager: connect-flash, (Mayo, 2022). [Online]

<https://www.npmjs.com/package/connect-flash>

[26]. NPM, Node Package Manager: express-validator, (Mayo, 2022). [Online]

<https://www.npmjs.com/package/express-validator>

[27]. Bootstrap, Bootstrap, (Julio, 2022). [Online]

<https://getbootstrap.com/docs/4.1/getting-started/introduction/>

A. Anexo

A.1. Manual de usuario

En esta parte del anexo, se va a mostrar un pequeño manual de usuario para que, en caso de existir cualquier problema con la aplicación, que se pueda orientar fácilmente en ella con este manual.

A.1.1. Pantalla de inicio

Para acceder a la aplicación, lo haremos mediante la URL en donde esté alojada nuestra aplicación, introduciendo el link, en este caso como la estamos utilizando en la máquina sin subir a ningún servidor, nosotros somos el propio host de la aplicación, así que accederemos mediante localhost:4000. Esta será la vista que tendremos al acceder a esta dirección.



Figura 50. Página de inicio de la aplicación (no autenticado)

En caso de ser la primera vez que accedamos a la aplicación, nos tendremos que identificar, y en caso de ya estar previamente identificados, y de no haber perdido la sesión activa, es decir, ni se cambie de navegador ni se cierre la sesión, podremos ver la vista de abajo, en la que tendremos opciones de cierre de sesión, perfil y de acceso directo a las funcionalidades, como podemos ver a continuación:



Figura 51. Página de inicio de la aplicación (usuario autenticado)

A.1.2. Inicio de sesión

Una vez entremos por primera vez, o sin tener una sesión activa, tanto el botón de Identificarse de “[Figura 44. Página de inicio de la aplicación \(no autenticado\)](#)”, como la parte de Sign in nos servirán para identificarnos, una vez hagamos click en cualquiera de ambos enlaces, nos redirigirá a la página que podemos ver a continuación:

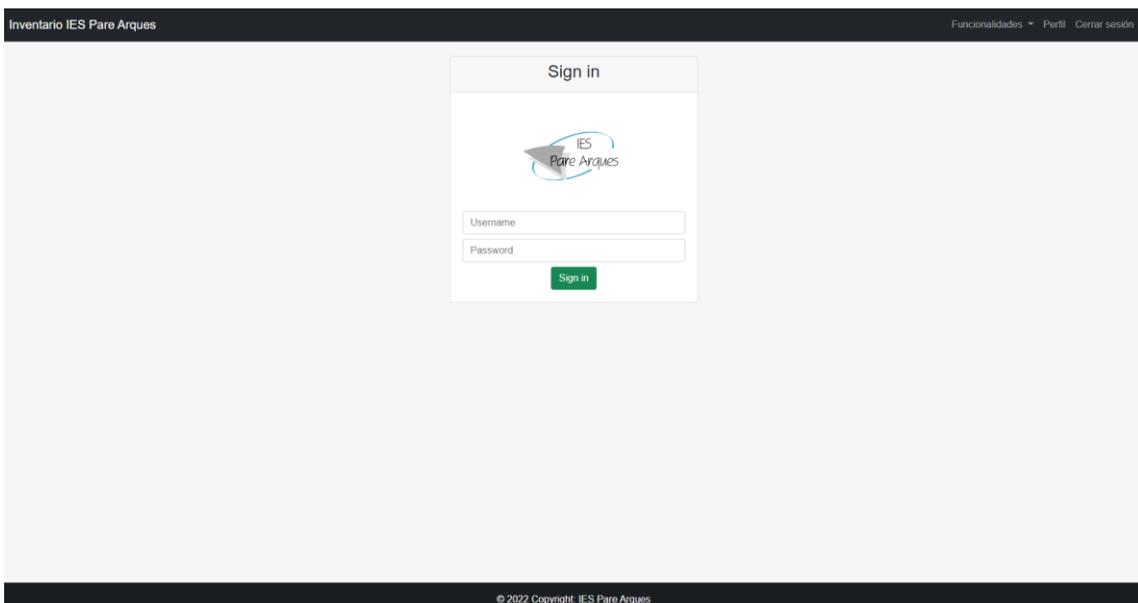


Figura 52. Página de inicio de sesión

A.1.3. Visualización de perfil

Una vez el usuario se haya identificado correctamente, al pulsar en el botón de Sign in, la aplicación nos redireccionará al perfil del usuario, en este caso la dirección (localhost:4000/profile), y podremos ver tanto el acceso directo a las aulas, como la opción de registrar usuarios nuevos:

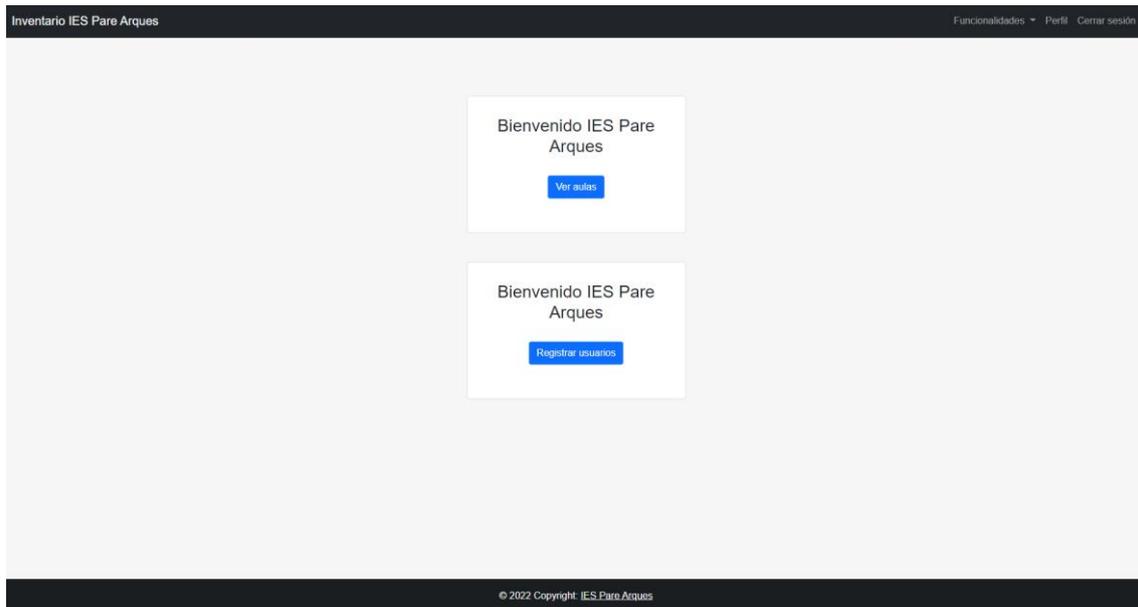


Figura 53. Página de visualización de perfil

A.1.4. Visualización de aulas/departamentos

Si lo que se desea es visualizar las aulas, para esto, se accederá a ellas mediante el botón de Ver aulas, y así nos redirigirá la aplicación a la siguiente página donde podremos ver todas las aulas que existan en la base de datos. En caso de no existir ninguna, aparecerá un cuadro de texto explicando que no existe ningún aula, y un enlace para poder crear un aula directamente.

No obstante, se ha añadido un botón en la parte inferior derecha que funciona como acceso directo para la creación de nuevas aulas. Cada aula tendrá un enlace a su contenido, que en este caso, será el visualizador de objetos.

Comentar también que existe la posibilidad de editar aulas y borrarlas también, en caso de que un aula haya quedado desactualizada, o simplemente ya no exista en el sistema.

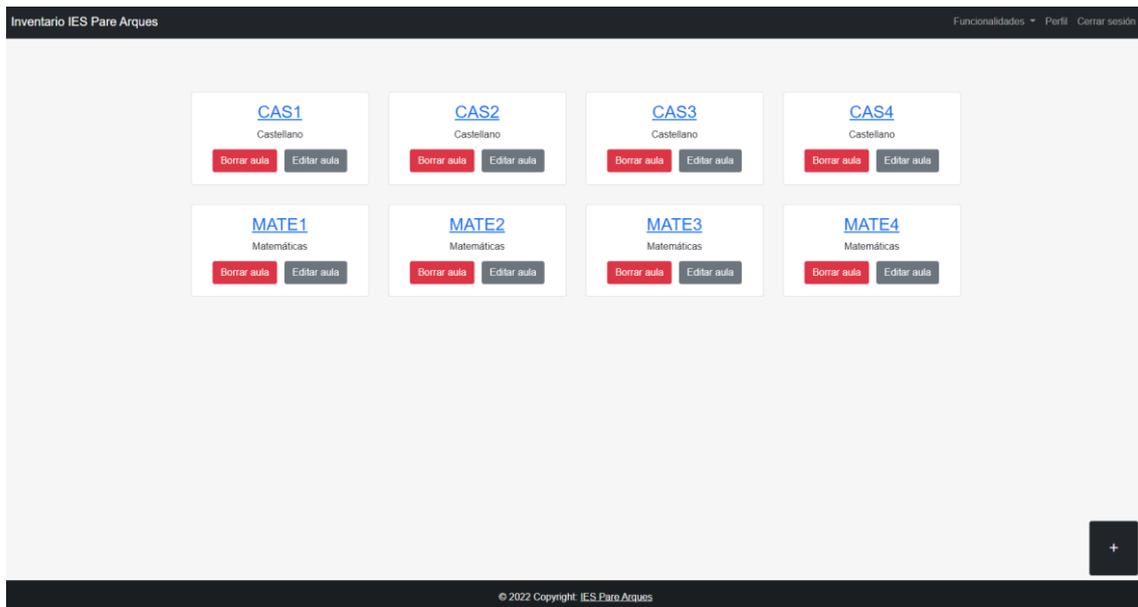


Figura 54. Página de visualización de aulas/departamentos

A.1.5. Visualización de objetos

Desde los enlaces que podemos ver en “[Figura 48. Página de visualización de aulas/departamentos](#)” tendremos acceso directo a cada una de las aulas, para poder observar su contenido, en este caso por ejemplo, podemos ver los objetos correspondientes al aula CAS1.

Los objetos tienen distintos colores dependiendo del estado en el que se encuentren. En el visor de los objetos del aula, existen opciones para borrar el objeto o editarlo, en caso de que no existan o bien su estado haya cambiado:

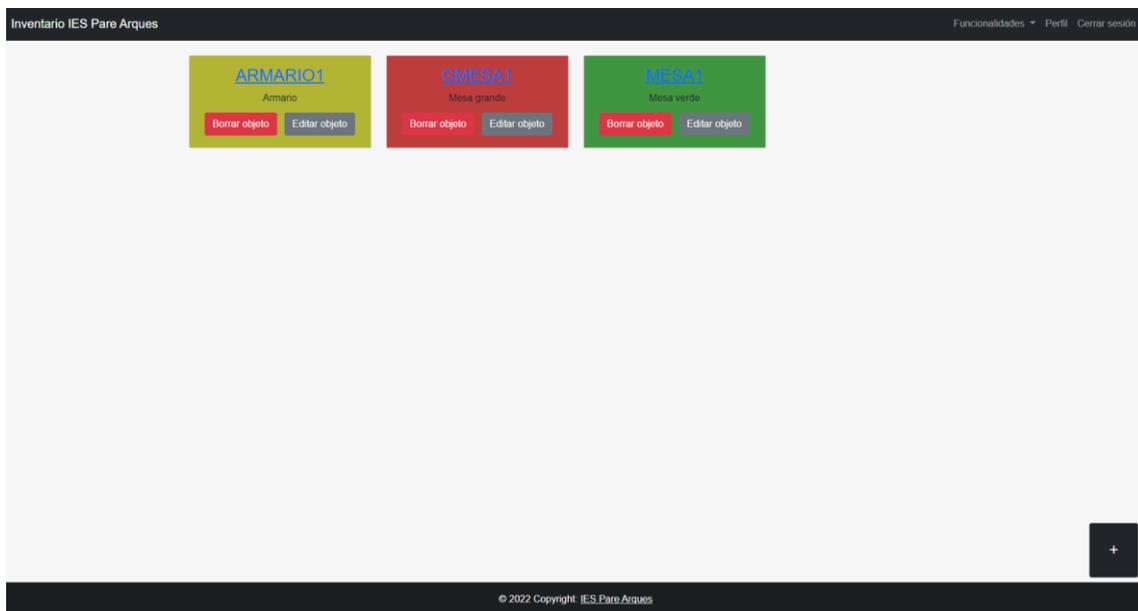


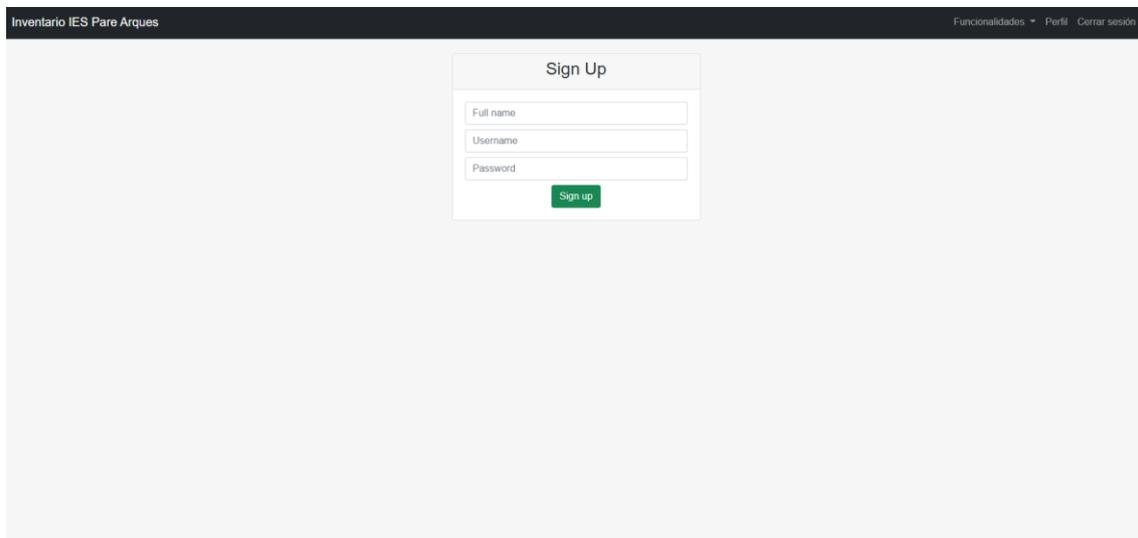
Figura 55. Página de visualización de objetos en aula CASI

A.1.6. Visualización de los formularios

Por último, tendremos las distintas visualizaciones de cuando queramos registrar un nuevo usuario, añadir nuevas aulas/departamentos, o añadir nuevos objetos, como hemos podido ver anteriormente, cada vez que editemos o vayamos a añadir objetos en la base de datos podremos observar los siguientes formularios para rellenar los datos correctamente.

Añadir también que la redirección en cada uno de los casos lleva a la pantalla de visualización (nuevo usuario – perfil, nueva aula – visor de aulas, nuevo objeto – visor de objetos).

- Usuarios:



The image shows a web browser window with a dark header. On the left of the header, it says 'Inventario IES Pare Arques'. On the right, there are links for 'Funcionalidades', 'Perfil', and 'Cerrar sesión'. The main content area is light gray and contains a white 'Sign Up' form. The form has a title 'Sign Up' and three input fields: 'Full name', 'Username', and 'Password'. Below the fields is a green button with the text 'Sign up'.

Figura 56. Página con formulario para añadir un usuario

- Aulas/departamentos:

The screenshot shows a web application interface for adding a classroom or department. At the top, there is a dark header with the text 'Inventario IES Pare Arques' on the left and 'Funcionalidades', 'Perfil', and 'Cerrar sesión' on the right. The main content area is light gray and contains a white form box. The form has two text input fields: 'Nombre del aula' and 'Descripcion'. Below these fields is a green 'Save' button.

Figura 57. Página con formulario para añadir un aula/departamento

- Objetos:

The screenshot shows a web application interface for adding an object to CASI. At the top, there is a dark header with the text 'Inventario IES Pare Arques' on the left and 'Funcionalidades', 'Perfil', and 'Cerrar sesión' on the right. The main content area is light gray and contains a white form box. The form has four input fields: 'Nombre del objeto', 'Descripcion del objeto', 'Buen estado', and 'CASI'. Below these fields is a green 'Save' button.

Figura 58. Página con formulario para añadir un objeto a CASI