

INSTITUTO DE EDUCACIÓN SECUNDARIA JOSÉ PLANES

Departamento de Informática y Comunicaciones  
Técnico Superior en Desarrollo de aplicaciones Web

C/ Maestro Pérez Abadía, 2  
30100 Espinardo – Murcia  
T. 968 834 605  
30010577@murciaeduca.es  
www.iesjoseplanes.es

---

## Memoria del proyecto

### Desarrollo de aplicaciones web

# MyAcademy Manager - Digitalización de la Gestión Académica

#### **Autores/as:**

Antonio Cano Ruiz

Kevin Humberto Castillo Castillo

Pedro de Pedro Fernández

#### **Profesor/a-coordinador/a:**

Javier Andrés Gutiérrez

Murcia, Junio de 2024



Esta obra está bajo una [licencia de Creative Commons  
Reconocimiento-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).

## Contenido

<b>1 Resumen extendido.....</b>	<b>1</b>
<b>2 Palabras clave.....</b>	<b>1</b>
<b>3 Introducción.....</b>	<b>2</b>
<b>4 Estado del arte/trabajos relacionados.....</b>	<b>3</b>
<b>5 Análisis de objetivos y metodología.....</b>	<b>3</b>
<b>6 Diseño y resolución del trabajo realizado.....</b>	<b>5</b>
6.1 Diseño del Logo.....	5
6.2 Diseño de la Base de Datos.....	8
6.3 API REST con Symfony.....	11
6.3.1 Generación de Tokens.....	12
6.3.1.1 Método SignUp.....	12
6.3.1.2 Método CheckToken.....	13
6.3.2 Instrucciones de uso de nuestra API.....	14
6.3.2.1 List Controller.....	14
6.3.2.2 Admin Controller.....	17
6.3.2.3 User Controller.....	18
6.3.2.4 PDF Controller.....	19
6.4 Frontend con Angular.....	19
6.5 Diseño.....	20
6.5.1 Prototipos de Baja Fidelidad.....	20
6.5.2 Prototipos de Alta Fidelidad.....	24
6.5.3 Elección de colores y tipografía.....	28
6.5.4 Accesibilidad.....	29
6.5.5 Usabilidad.....	32
<b>7 Presupuesto.....</b>	<b>33</b>
<b>8 Conclusiones y vías futuras.....</b>	<b>34</b>
<b>9 Manual de uso de la aplicación.....</b>	<b>35</b>
9.1 Inicio de Sesión.....	35
9.2 Home.....	35
9.2.1 Profesor.....	35
9.2.2 Alumno.....	36
9.2.1 Admin.....	36
9.3 Clases.....	37
9.3.1 Profesor y Alumno.....	37
9.3.2 Admin.....	38
9.4 Calificaciones.....	40
9.4.1 Profesor.....	40

9.4.2 Alumno.....	41
9.5 Asistencia.....	41
9.5.1 Profesor.....	41
9.5.2 Alumno.....	42
9.6 Jornadas (Solo profesores).....	42
9.7 Usuarios (Solo admin).....	43
9.8 Mensajes.....	44
<b>10 Licencias.....</b>	<b>45</b>
<b>11 Bibliografía / Webgrafía.....</b>	<b>47</b>



## **1 Resumen extendido**

MyAcademy Manager es una página web de gestión académica desarrollada en Symfony y Angular. Esta aplicación está diseñada para facilitar la digitalización de todas las tareas administrativas requeridas en cualquier tipo de academia.

En MyAcademy Manager, puedes dar de alta asignaturas, aulas, clases y usuarios para tu academia. Estos usuarios pueden tener roles de alumno, profesor o administrador.

Con el rol de alumno, puedes ver tus clases, asignaturas y aulas asignadas para mantener la organización. También puedes revisar tus faltas de asistencia y calificaciones.

Con el rol de profesor, también puedes ver tus clases, asignaturas y aulas asignadas, registrar las faltas de asistencia y calificaciones.

Finalmente, con el rol de administrador, puedes crear clases, asignaturas, aulas y usuarios.

Además, cualquier usuario tendrá acceso a un chat dentro de la aplicación con el que poder comunicarte fácilmente con cualquier usuario de la academia.

## **2 Palabras clave**

Academia, manager, administración, Symfony, angular, php, usuarios, clases, jwt, Tailwind CSS, asignatura, aula, CORS.

### 3 Introducción

El proyecto surgió de la idea de nuestro compañero de proyecto, Antonio, quien es profesor en una academia y notaba la falta de aplicaciones que permitieran organizar sus clases de manera eficiente y que incluyeran todo lo necesario para una academia. Cuando nos presentó la idea, la vimos como una oportunidad para poner en práctica todo lo aprendido en el curso y crear así MyAcademy Manager.

Para comenzar el desarrollo de este proyecto, pensamos en utilizar varias tecnologías, algunas aprendidas durante el curso y otras investigadas por nuestra cuenta para agregar un toque de innovación. Decidimos basar nuestro proyecto en Angular y Symfony, ya que fueron las dos tecnologías con las que más trabajamos a lo largo del curso y pensamos que serían adecuadas para desarrollar nuestra aplicación. Además, son dos tecnologías ampliamente utilizadas en el mundo del desarrollo web.

Para el diseño, optamos por utilizar Tailwind CSS, una nueva herramienta que, aunque ya habíamos oído hablar de ella, no la habíamos utilizado anteriormente ni aprendido durante el curso. Consideramos que era una buena oportunidad para aprender a utilizarla y también para innovar un poco. Utilizamos Canva para el diseño del logo, Figma para realizar un primer boceto de la página web y DIA para el Modelo Entidad-Relación (MER) de la base de datos.

Finalmente, para la organización del equipo, buscamos métodos que nos permitieran mantenernos comunicados y organizados de la mejor manera posible, tanto para asignar tareas como para seguirlas. Utilizamos herramientas convencionales como WhatsApp para la comunicación, Discord para las reuniones y Jira para la gestión de tareas. Elegimos Jira porque, basándonos en nuestra experiencia en las prácticas, vimos que era una herramienta bastante utilizada en el mundo laboral y que nos permitiría organizar nuestras tareas de manera eficiente, especialmente durante las reuniones, además de seguir aprendiendo a utilizarla para un futuro profesional.

## 4 Estado del arte/trabajos relacionados

Como se ha mencionado anteriormente, nuestro proyecto MyAcademy Manager surgió a partir de la idea de nuestro compañero Antonio. Esto, junto con la falta de páginas web en este sector, nos llevó a decantarnos por este proyecto.

MyAcademy Manager es una página web cuya función es la gestión de áreas académicas, con usuarios que pueden ser profesores, alumnos o administradores, con el fin de ofrecer un seguimiento y organización lo más cómoda e intuitiva posible.

## 5 Análisis de objetivos y metodología.

### 5.1 Objetivos

Ya con la idea clara del proyecto, teníamos que decidir cuáles iban a ser los objetivos principales de nuestro proyecto.

- **Utilizar bases de datos relacionales**: Para tener una idea de cómo empezar con nuestro proyecto, tuvimos que hacer una lluvia de ideas para tener una base de datos clara y concisa. Esto tuvo una dificultad al ser tres personas, ya que había muchas opciones y maneras de ver el proyecto para quedarnos solo con una. Finalmente, creemos que la manera en la que la dejamos es la mejor manera que se podría quedar.
- **Tener un diseño característico y único**: Toda página web tiene que tener un diseño y unos colores característicos para ser única. Para ello, gracias a los conceptos dados en diseño, pudimos usar sus técnicas para crear un logo, un primer diseño para la página y una paleta de colores. La mayor dificultad fue decidir entre las muchas opciones buenas, pero gracias a la votación pudimos ponernos de acuerdo.

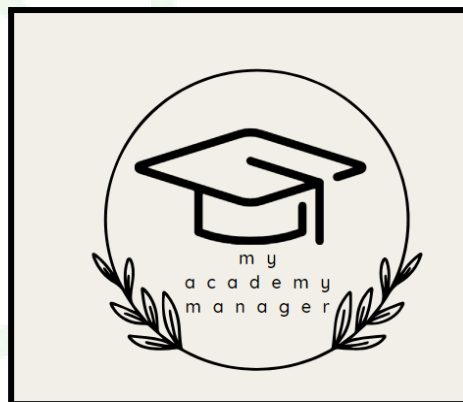
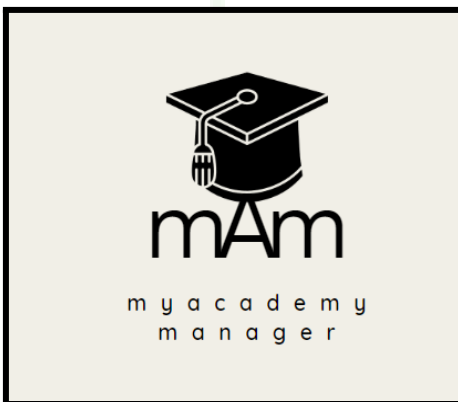
- **Creación de login:** Vimos que es necesario un login para poder tener acceso. Esto fue un reto ya que durante el curso, habíamos visto el proceso del login solamente en Symfony, y además queríamos implementar JWT, por lo que al final tuvimos que destinar gran parte de nuestro tiempo en ver cómo se podía hacer esto.
- **Usuarios:** Obviamente en nuestro proyecto es necesario usuarios, estos los definimos en tres grandes grupos: Profesores, Alumnos y Administradores, cada uno con un rol distinto para tener un control en la página. La mayor dificultad fue decidir la manera en la que depende del tipo de usuario se muestre una cosa u otra en la página.
- **Clases, aulas y asignaturas:** De la misma manera que tenemos usuarios, tenemos clases, aulas y asignaturas y estas van cambiando dependiendo del día y del usuario que entre.
- **Creación de sistema de mensajería entre usuarios:** Vimos necesario un sistema de mensajería para tener un contacto más estrecho entre usuarios, ya que sería más fácil para estos el tema de la comunicación en caso de no poder hacerlo en clase. Esta parte ha sido una de las más difíciles debido a la complejidad que requiere un sistema de mensajería.
- **Creación de sistema de faltas:** Como en cualquier academia el tema de las faltas es algo importante, por lo que también lo tuvimos como objetivo. En la aplicación los usuarios con rol de profesores pueden poner faltas a los alumnos que no estén presentes en las clases.
- **Creación de landing page:** Al principio no estuvo en nuestros planes el objetivo de hacer una landing page, pero esta nos servía como una presentación de proyecto y de la aplicación por lo que decidimos hacerla.
- **Despliegue de aplicación:** El despliegue fue otra de las cosas con las que más problemas tuvimos debido a un tema de que los hostings bloquean el CORS (Cross-Origin Resource Sharing), lo cual impide la transferencia de cabeceras, con las que nosotros hacemos funcionar gran parte de nuestro backend. Al final, hemos encontrado uno con un ajuste que permite el CORS.

## 6 Diseño y resolución del trabajo realizado

### 6.1 Diseño del Logo

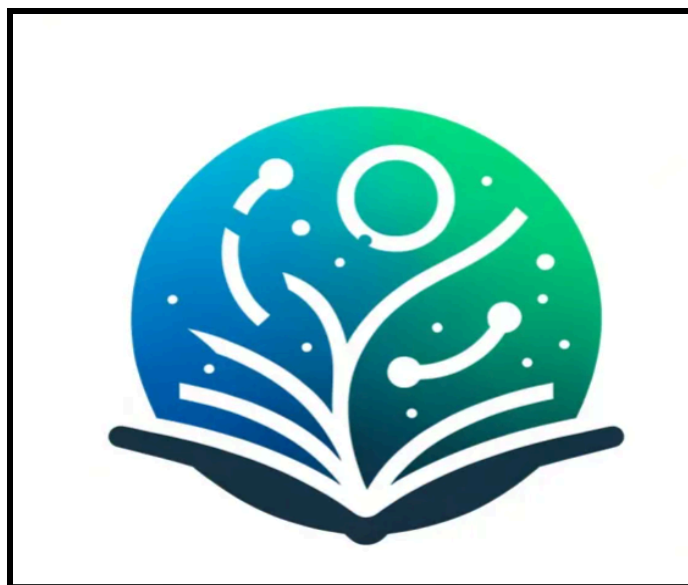
Para la creación del logo, tuvimos varias ideas y para ponerlas en común utilizamos una técnica que aprendimos a lo largo del curso, en la cual utilizamos una herramienta de edición y creación de imágenes, que fue Canva, para hacer una tormenta de ideas.

Estos fueron algunos de los resultados:



Esto creó conflicto porque no nos convencían del todo. Nos dimos cuenta de que buscábamos algo más minimalista y, tras un periodo de búsqueda y reflexión, finalmente decidimos probar con la inteligencia artificial. Esta nos dio varias opciones hasta que, al final, una nos convenció.

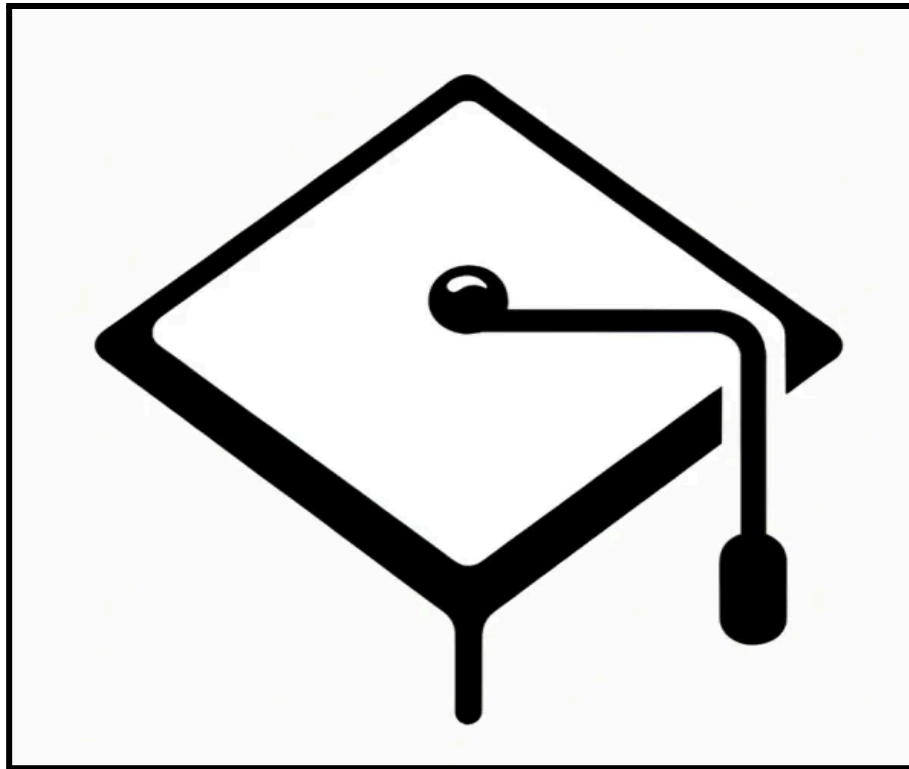




Este primer logo no nos convenció mucho, ya que nos daba la sensación de que se enfocaba demasiado en la naturaleza y las plantas, a pesar de que apareciera el libro.



Este segundo no nos disgustaba, pero tampoco nos convencía del todo, sobre todo por los colores.

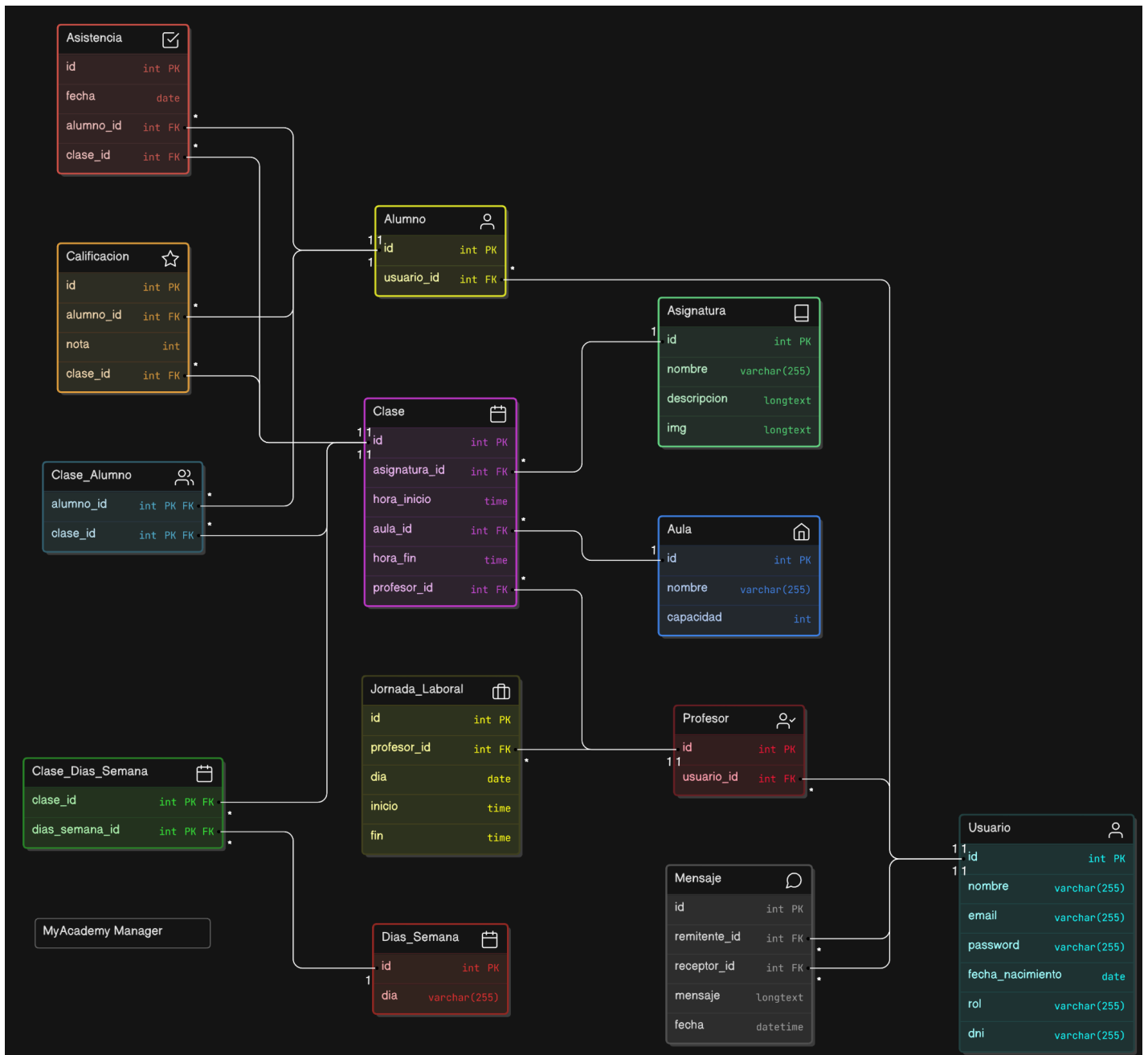


Finalmente, la última opción nos encantó por las siguientes razones:

1. **Simplicidad y Claridad:** La simplicidad del diseño garantiza que sea fácilmente reconocible para nuestros usuarios.
2. **Significado Simbólico:** El birrete, como símbolo de logro académico, comunica de manera inmediata la naturaleza educativa de nuestra plataforma. Esta representación simbólica establece una conexión instantánea con nuestro público objetivo, transmitiendo nuestro propósito y función de manera clara y efectiva.
3. **Elegancia y Modernidad:** El diseño minimalista y la línea artística del logo le aportan una apariencia elegante y profesional. Esta estética moderna es crucial para atraer a un público joven y tecnológicamente consciente.

En conclusión, este logo combina elementos clave como simplicidad, significado simbólico y modernidad, lo que lo convierte en la mejor opción y la más apropiada para representar nuestro gestor académico y comunicar nuestra identidad y valores de manera efectiva.

## 6.2 Diseño de la Base de Datos



1. Alumno:

- id (PK), integer
- usuario\_id (FK de Usuario), integer

2. Asignatura:

- id (PK), integer
- nombre, varchar (255)
- descripcion longtext
- img, longtext

3. Asistencia:

- id (PK), integer
- clase\_id (FK de Clase), integer
- alumno\_id (FK de Alumno), integer
- fecha, date

4. Aula:

- id (PK), integer
- nombre, varchar (255)
- capacidad, integer

5. Calificacion:

- id (PK), integer
- clase\_id (FK de Clase), integer
- alumno\_id (FK de Alumno), integer
- nota, integer

6. Clase:

- id (PK), integer
- asignatura\_id (FK de Asignatura), integer
- profesor\_id (FK de Profesor), integer
- aula\_id (FK de Aula), integer
- hora\_inicio, time
- hora\_fin, time

7. Clase\_Alumno:

- clase\_id (FK de Clase, PK), integer
- alumno\_id (FK de Alumno, PK), integer

8. Clase\_Dias\_Semana:

- clase\_id (FK de Clase, PK), integer
- dias\_semana\_id (FK de Dias\_Semana, PK), integer

9. Dias\_Semana:

- id (PK), integer
- dia, varchar(255)

10. Jornada\_Laboral:

- id (PK), integer
- profesor\_id (FK de Profesor), integer
- dia, date
- inicio, time
- fin, time

11. Mensaje:

- id (PK), integer
- remitente\_id (FK de Usuario), integer
- receptor\_id (FK de Usuario), integer
- mensaje, longtext
- fecha, datetime

12. Profesor:

- id (PK), integer
- usuario\_id (FK de Usuario), integer

13. Usuario:

- id (PK), integer
- nombre, varchar (255)
- email, varchar (255)
- password, varchar (255)
- fecha\_nacimiento, date
- rol, varchar (255)
- dni, varchar (255)

### 6.3 API REST con Symfony

Para nuestro backend, hemos decidido desarrollar una API en Symfony, ya que es una tecnología que hemos trabajado durante este curso y que conocemos.

La experiencia inicial en el desarrollo de nuestra primera API ha resultado ser un desafío considerable. En las etapas iniciales, nos enfrentamos a diversos obstáculos, particularmente en la comprensión y configuración del JSON para su presentación conforme a nuestras necesidades y la implementación de los JWT para cualquier acción en nuestro backend. No obstante, una vez asimilado, el proceso de desarrollo adquirió una fluidez notable.

Para estructurar la API, optamos por dividirla en tres controladores, cada uno encargado de las operaciones relacionadas con listar información, funciones de administrador y gestión de usuarios, respectivamente. Esta subdivisión facilitó la gestión del código, proporcionando una organización más clara y modular que favorece tanto el mantenimiento como la escalabilidad del sistema.

Además, implementamos un riguroso control de errores mediante el uso de excepciones directamente en la lógica de la API. Esta medida se adoptó con el propósito de salvaguardar la integridad de la base de datos frente a posibles fallos o vulnerabilidades en el frontend.

### 6.3.1 Generación de Tokens

Para la generación de tokens para el acceso a todas las funciones, hemos creado un servicio , basado en la librería de Firebase, [JWT](#). Cada funcionalidad requiere la autenticación mediante este servicio, por lo que, es una de las funcionalidades más importantes de nuestro proyecto.

#### 6.3.1.1 Método SignUp

El primer método de nuestro servicio, lo utilizamos para loguearnos y dependiendo del parámetro gettoken, devolveremos el token o no. En nuestro proyecto, cuando hacemos la petición de login, ponemos el gettoken en true, para que nos envíe al frontend el token para guardarlo en LocalStorage para poder usarlo más adelante en las siguientes peticiones.

```
public function singup($email, $password, $gettoken=null){
    //Comprobar si el usuario existe
    $user=$this->manager->getRepository(Usuario::class)->findOneBy([
        'email'=>$email,
        'password'=>$password
    ]);

    $signup=false;

    if(is_object($user)){
        $signup=true;
    }

    //Si existe, devolver token jwt
    if($signup){
        $token = [
            'sub' => $user->getId(),
            'email' => $user->getEmail(),
            'nombre' => $user->getNombre(),
            'apellidos' => $user->getApellidos(),
            'rol' => $user->getRol(),
            'id_profesor' => ($user->getRol() == 'profesor') ? $user->getProfesors()->first()->getId() : null,
            'id_alumno' => ($user->getRol() == 'alumno') ? $user->getAlumnos()->first()->getId() : null,
            'fecha_nacimiento' => $user->getFechaNacimiento(),
            'iat' => time(),
            'exp' => time() + (7 * 24 * 60 * 60)
        ];

        $jwt = JWT::encode($token, $this->key, 'HS256');
        $decoded = JWT::decode($jwt, $this->key, ['HS256']);

        //Devolver datos decodificados o token en funcion de un parametro
        if($gettoken){
            $data = $jwt;
        }else{
            $data = $decoded;
        }

        //Comprobar si las credenciales son correctas

        //Devolver token jwt
    }
    else{
        $data = [
            'status' => 'error',
            'message' => 'Login incorrecto'
        ];
    }
    return $data;
}
```

### 6.3.1.2 Método CheckToken

El segundo método de nuestro servicio de tokens, se utiliza en cada uno de los demás métodos de nuestra API. En este se comprueba que el token es legítimo, y si es así, nos devuelve la información del token, para usarlo en nuestros controladores.

Un ejemplo es para sacar una lista de tus notas, en la API, solo recibimos el token, que nos llega por la cabecera 'Authorization'. Comprobamos su legitimidad y guardamos la identidad en una variable para su uso posterior.

```
public function checkToken($jwt, $identity=false){
    $auth=false;

    try{
        $decoded = JWT::decode($jwt, $this->key, ['HS256']);
    }catch(\UnexpectedValueException $e){
        $auth=false;
    }catch(\DomainException $e){
        $auth=false;
    }

    if(!empty($decoded) && is_object($decoded) && isset($decoded->sub)){
        $auth=true;
    }else{
        $auth=false;
    }

    if($identity){
        return $decoded;
    }

    return $auth;
}
```



## 6.3.2 Instrucciones de uso de nuestra API

### 6.3.2.1 List Controller

#### 6.3.2.1.1 Listar Asignaturas

- Method = GET
- url= <http://localhost:8000/api/list/asignaturas>
- Función= Nos devuelve, sólo si somos admin, todas las asignaturas.

#### 6.3.2.1.2 Listar Usuarios

- Method = GET
- url= <http://localhost:8000/api/list/usuarios>
- Función= Nos devuelve, sólo si somos admin, todos los usuarios.

#### 6.3.2.1.3 Listar Clases

- Method = GET
- url= <http://localhost:8000/api/list/clases>
- Función= Nos devuelve, sólo si somos admin, todas las clases.

#### 6.3.2.1.4 Listar Aulas

- Method = GET
- url= <http://localhost:8000/api/list/aulas>
- Función= Nos devuelve, sólo si somos admin, todas las aulas.

#### 6.3.2.1.5 Listar Profesores

- Method = GET
- url= <http://localhost:8000/api/list/profesores>
- Función= Nos devuelve, sólo si somos admin, todos los profesores.

#### 6.3.2.1.6 Listar Días

- Method = GET
- url= <http://localhost:8000/api/list/dias>
- Función= Nos devuelve los días de la semana.

#### 6.3.2.1.7 Listar Mis Clases

- Method = GET
- url= <http://localhost:8000/api/list/misclases>
- Función= Nos devuelve, si somos profesores, las clases en las que somos profesores; y si somos alumnos, las clases en las que estamos inscritos como alumnos.

#### 6.3.2.1.8 Listar Mis Clases de Hoy

- Method = GET
- url= <http://localhost:8000/api/list/misclaseshoy>
- Función= Hace lo mismo que la función Mis Clases, pero solo nos devuelve las que tenemos ese día.

#### 6.3.2.1.9 Listar Detalles de una Clase

- Method = GET
- url= <http://localhost:8000/api/list/clase/{id}>
- Función= Nos devuelve todos los detalles de la clase(alumno, profesor, hora,días que se imparte,etc.).

#### 6.3.2.1.10 Registrar Faltas de Asistencia

- Method = POST
- url= <http://localhost:8000/api/list/asistencia>
- Función= Registramos, sólo si somos profesores, las faltas del día y la asignatura.

#### 6.3.2.1.11 Listar mis faltas de asistencia

- Method = GET
- url= <http://localhost:8000/api/list/misfaltas>
- Función= Nos devuelve, sólo si somos alumno, todas las faltas de asistencia.

#### **6.3.2.1.12 Listar mis calificaciones**

- Method = GET
- url= <http://localhost:8000/api/list/misnotas>
- Función= Nos devuelve, sólo si somos alumno, todas nuestras calificaciones.

#### **6.3.2.1.13 Registrar calificaciones**

- Method = POST
- url= <http://localhost:8000/api/list/calificaciones>
- Función= Registramos, sólo si somos profesores, las calificaciones de nuestros alumnos en las clases

#### **6.3.2.1.14 Listar chats de un usuario**

- Method = GET
- url= <http://localhost:8000/api/list/mismensajes>
- Función= Nos devuelve una lista de nuestros chats con otros usuarios.

#### **6.3.2.1.15 Listar mensajes de un usuario con otro**

- Method = GET
- url= <http://localhost:8000/api/list/mismensajes/{id}>
- Función= Nos devuelve todo el historial de una conversación con otro usuario

#### **6.3.2.1.16 Registrar mensajes**

- Method = POST
- url= <http://localhost:8000/api/list/mismensajes/crear>
- Función= Registramos un mensaje, con toda su información (remitente, receptor, mensaje y fecha).

#### **6.3.2.1.17 Listar Jornada Laboral de hoy**

- Method = GET
- url= <http://localhost:8000/api/list/mijornadalaboral>
- Función= Nos devuelve los datos de la jornada laboral de hoy.

#### **6.3.2.1.18 Iniciar Jornada Laboral de hoy**

- Method = GET
- url= <http://localhost:8000/api/list/iniciojornada>
- Función= Comienza la jornada laboral de hoy con la hora actual.

#### **6.3.2.1.19 Finalizar Jornada Laboral de hoy**

- Method = GET
- url= <http://localhost:8000/api/list/finjornada>
- Función= Finaliza la jornada laboral de hoy con la hora actual.

#### **6.3.2.1.20 Listar Alumnos que no pertenecen a una clase**

- Method = GET
- url= <http://localhost:8000/api/list/noalumnosclase/{id}>
- Función= Nos devuelve, si somos admin, los alumnos que no pertenecen a una clase.

### **6.3.2.2 Admin Controller**

#### **6.3.2.2.1 Crear Asignaturas**

- Method = POST
- url= <http://localhost:8000/api/admin/asignaturas/crear>
- Función= Registramos, sólo si somos admin, una nueva asignatura.

#### **6.3.2.2.2 Crear Aulas**

- Method = POST
- url= <http://localhost:8000/api/admin/aulas/crear>
- Función= Registramos, sólo si somos admin, una nueva aula.

#### **6.3.2.2.3 Crear Clases**

- Method = POST
- url= <http://localhost:8000/api/admin/clase/crear>
- Función= Registramos, sólo si somos admin, una nueva clase.

#### 6.3.2.2.4 Eliminar Clases

- Method = GET
- url= <http://localhost:8000/api/admin/eliminar/clase/{id}>
- Función= Eliminamos, sólo si somos admin, la clase que tiene el id que le pasamos.

#### 6.3.2.2.5 Eliminar Alumnos de Clases

- Method = GET
- url= <http://localhost:8000/api/admin/eliminar/alumno/{clase}/{id}>
- Función= Eliminamos, sólo si somos admin, el alumno con el id, que le pasamos de la clase que le pasamos.

#### 6.3.2.2.5 Añadir Alumnos a Clases

- Method = POST
- url= <http://localhost:8000/api/admin/add/alumno/{clase}>
- Función= Añadimos, sólo si somos admin, los alumnos seleccionados a la clase seleccionada.

### 6.3.2.3 User Controller

#### 6.3.2.3.1 Crear Usuarios

- Method = POST
- url= <http://localhost:8000/api/usuarios/crear>
- Función= Registramos, sólo si somos admin, un nuevo usuario.

#### 6.3.2.3.2 Login

- Method = POST
- url= <http://localhost:8000/api/usuarios/login>
- Función= Con esta función nos logueamos haciendo uso de JWT.

#### 6.3.2.3.3 Editar Usuario

- Method = PUT
- url= <http://localhost:8000/api/usuarios/editar>
- Función= Editamos los datos de nuestro usuario.

#### 6.3.2.3.4 EliminarUsuario

- Method = PUT
- url= <http://localhost:8000/api/usuarios/borrar/{id}>
- Función= Borramos, solo si somos admin, el usuario de la id que le pasamos.

#### 6.3.2.4 PDF Controller

##### 6.3.2.4.1 PDF con notas del alumno

- Method = GET
- url= <http://localhost:8000/api/pdf/misnotas>
- Función= Nos genera un pdf con las notas del alumno.

##### 6.3.2.4.2 PDF con jornadas laborales

- Method = GET
- url= <http://localhost:8000/api/pdf/misjornadas>
- Función= Nos genera un pdf con las jornadas laborales del profesor.

### 6.4 Frontend con Angular

Para el frontend de nuestra aplicación, hemos optado por desarrollar en Angular debido a su robustez y capacidad para crear aplicaciones web dinámicas y de alta respuesta. Su arquitectura basada en componentes facilita la reutilización de código y mejora la mantenibilidad.

Inicialmente, enfrentamos desafíos en la integración con nuestra API RESTful en Symfony, especialmente en la configuración de servicios HTTP y la gestión de la autenticación mediante JWT. Superar estos obstáculos estableció una base sólida para el desarrollo continuo.

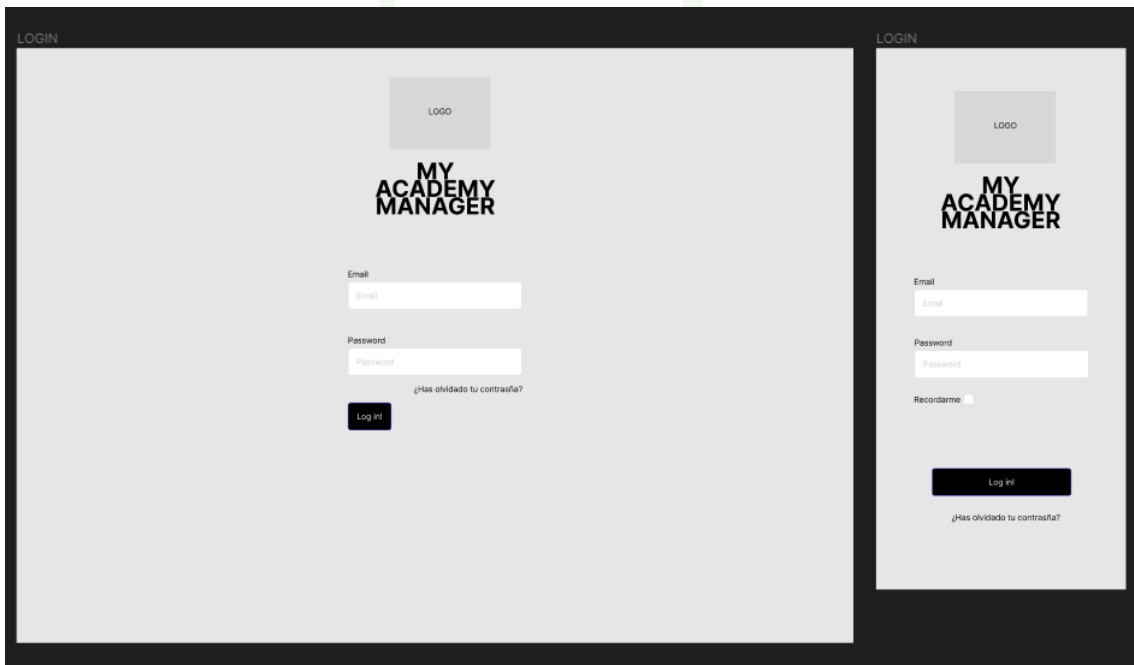
Hemos dividido nuestra aplicación en módulos y componentes específicos para una mejor organización y eficiencia del código. Implementamos servicios dedicados para la comunicación con la API, mejorando así la experiencia del usuario.

En resumen, Angular nos ha permitido desarrollar un frontend moderno, escalable y de alto rendimiento, superando desafíos iniciales y asegurando una aplicación funcional.

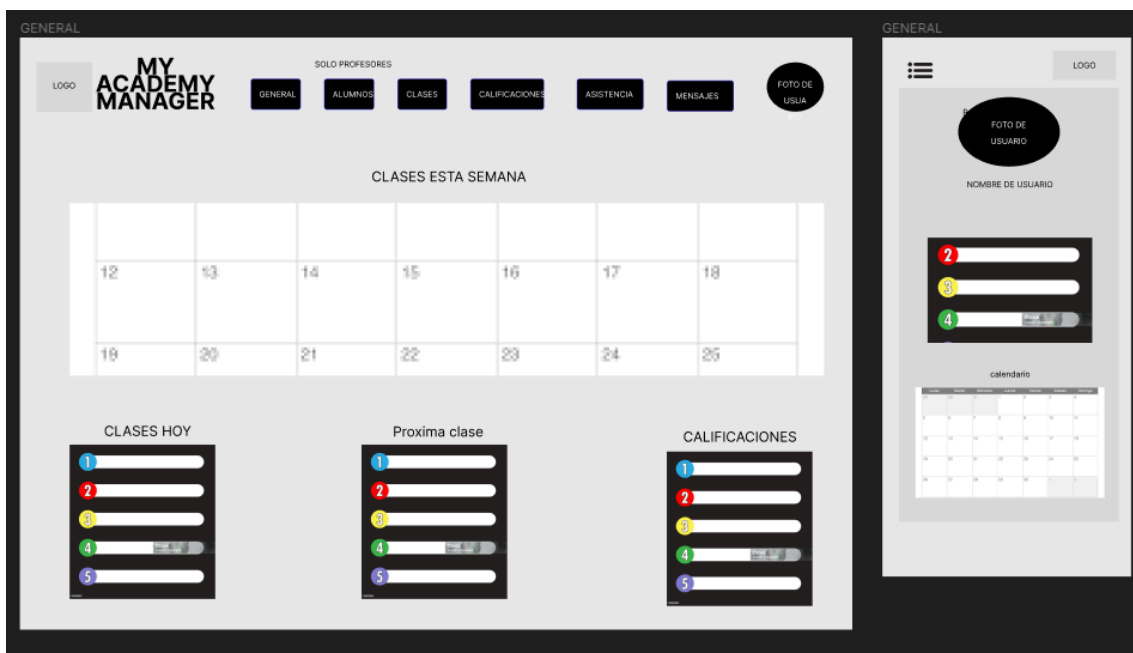
## 6.5 Diseño

### 6.5.1 Prototipos de Baja Fidelidad

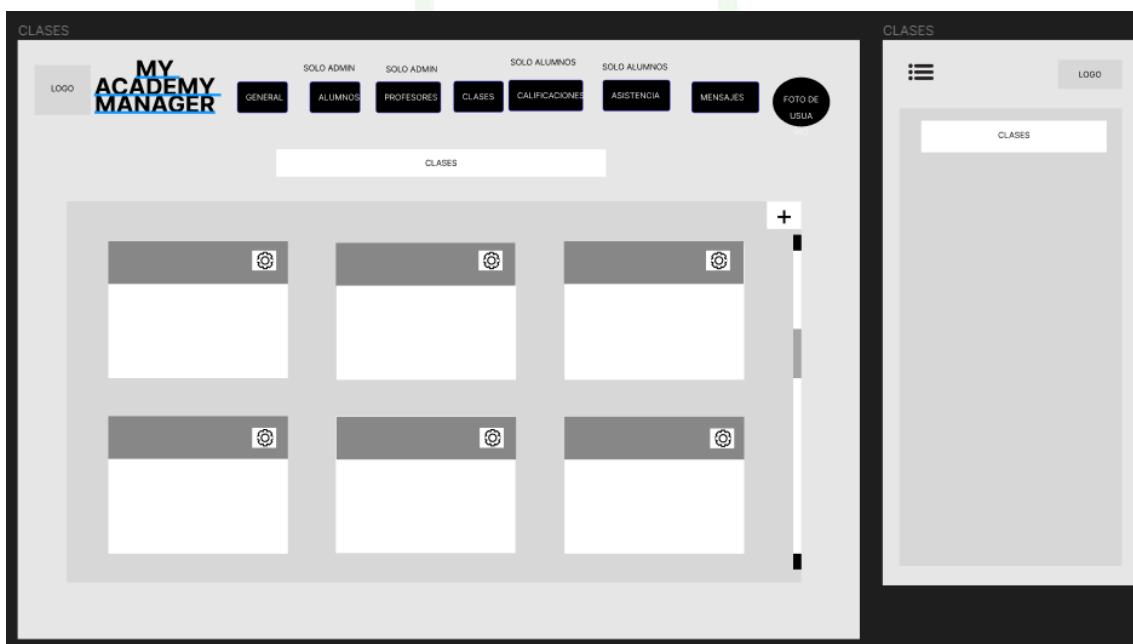
#### 6.5.1.1 Login



### 6.5.1.2 Home

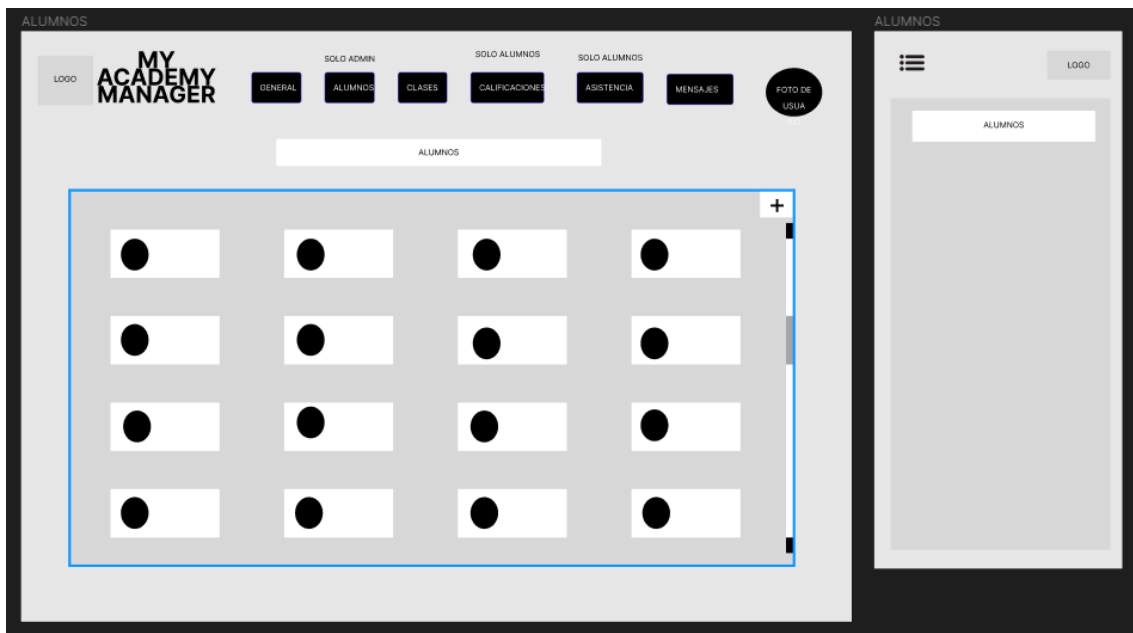


### 6.5.1.3 Clases

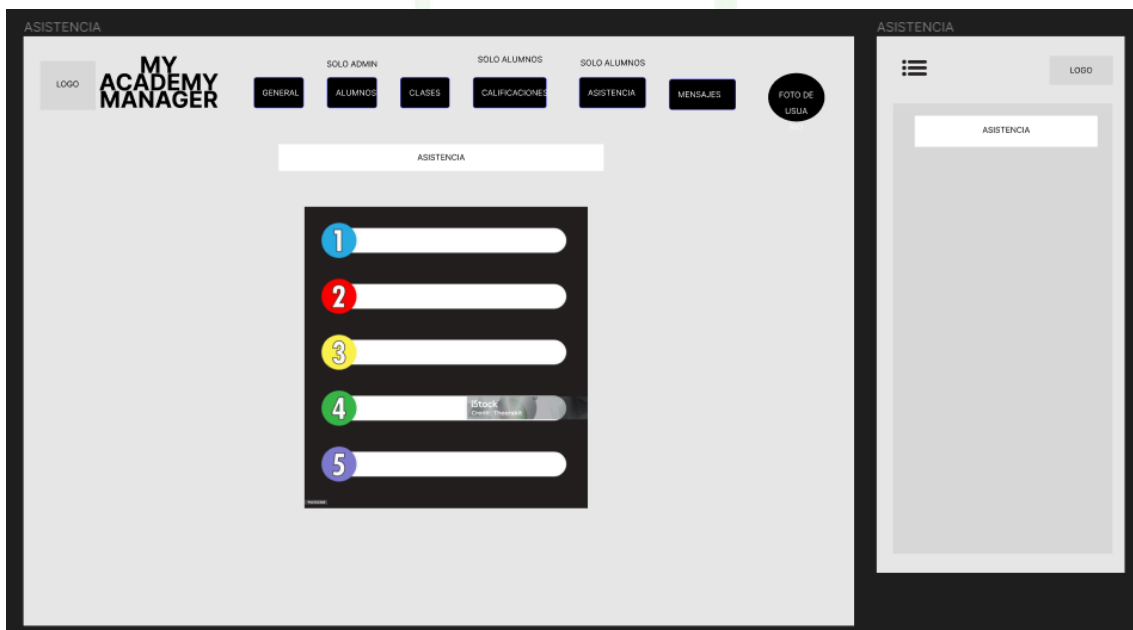




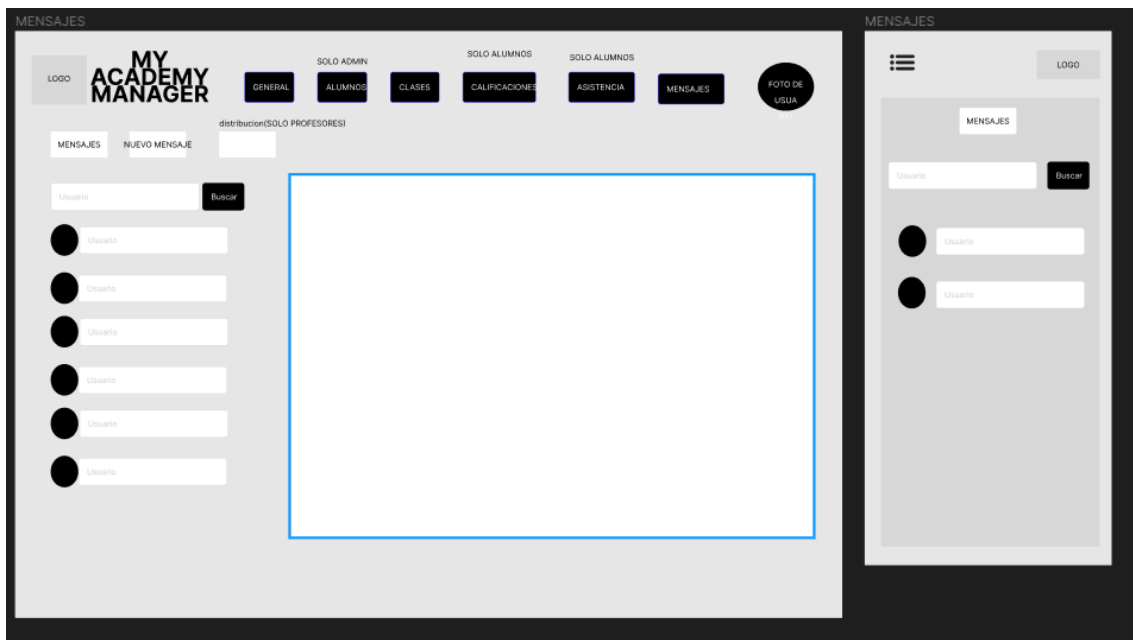
### 6.5.1.3 Usuarios



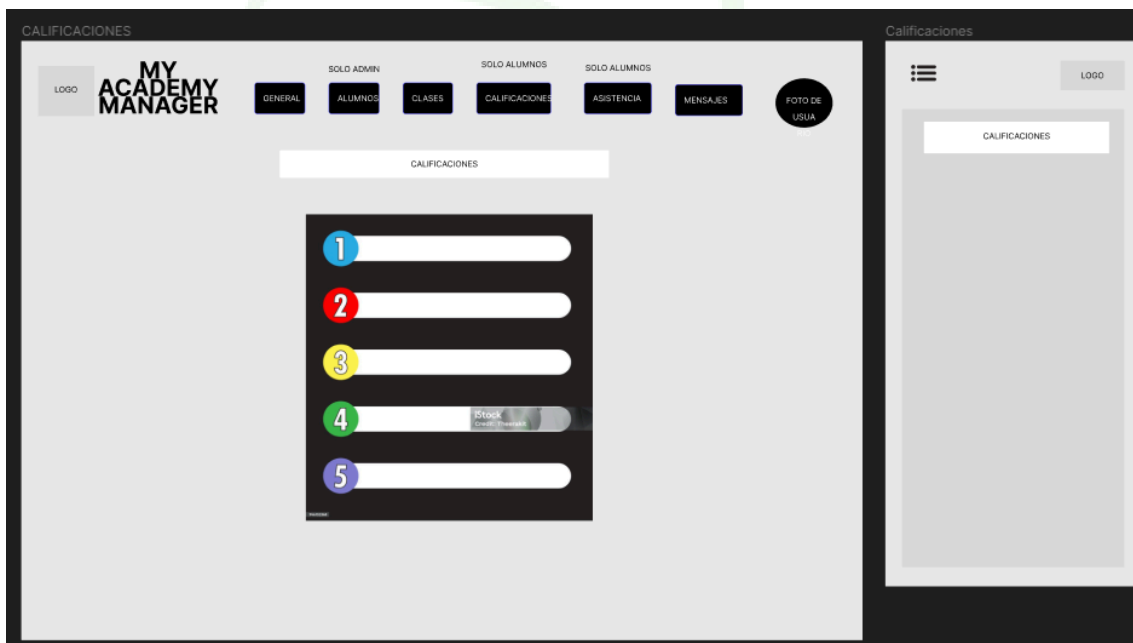
### 6.5.1.4 Asistencia



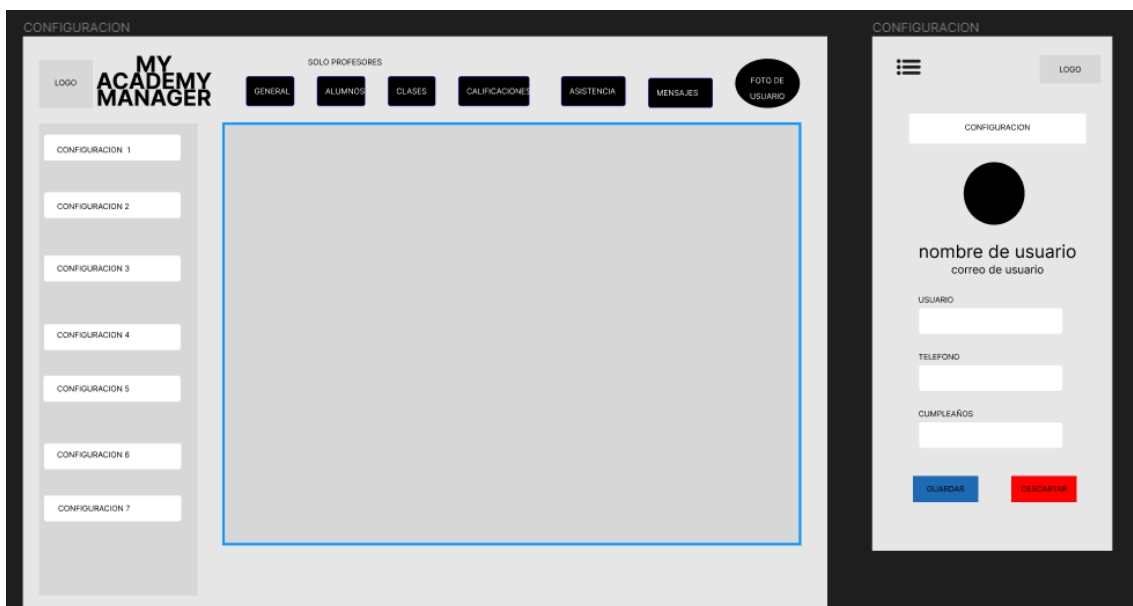
### 6.5.1.4 Mensajes



### 6.5.1.4 Calificaciones

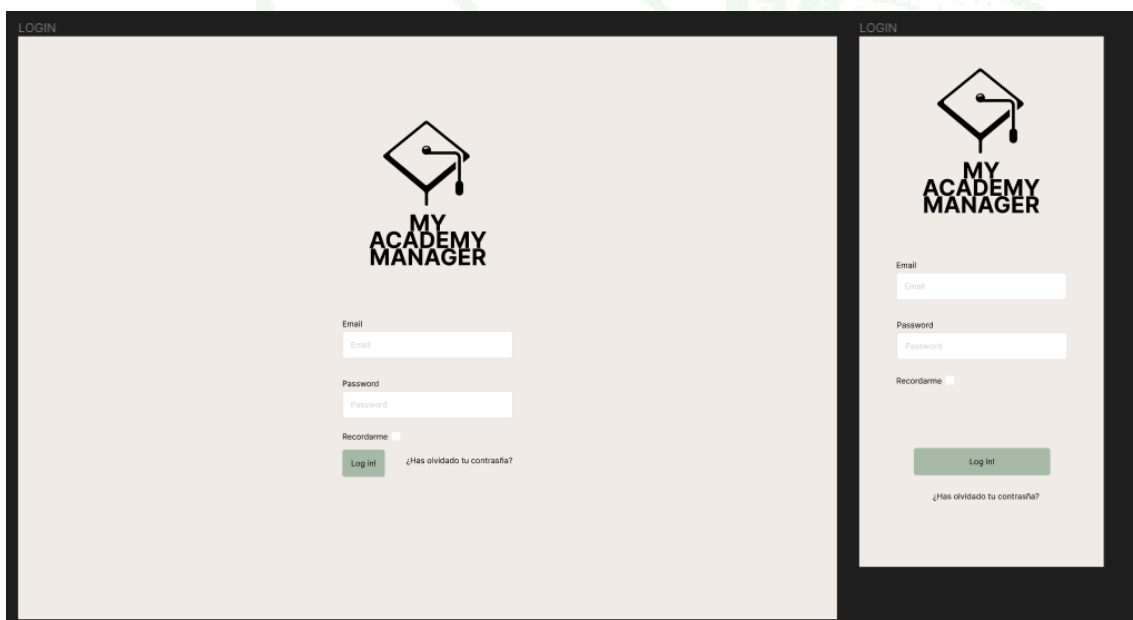


### 6.5.1.4 Perfil

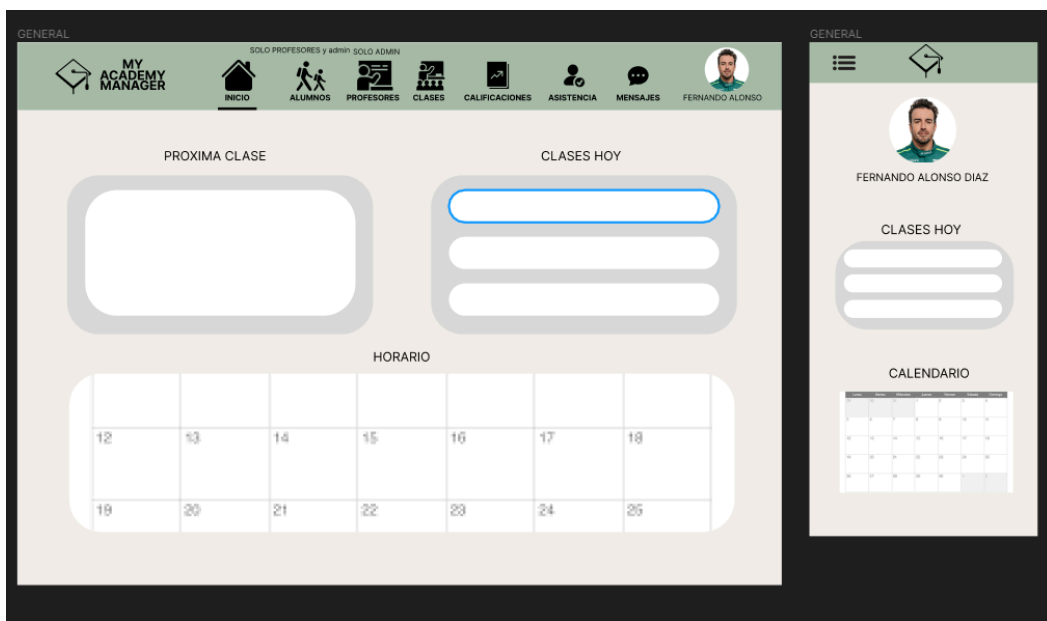


## 6.5.2 Prototipos de Alta Fidelidad

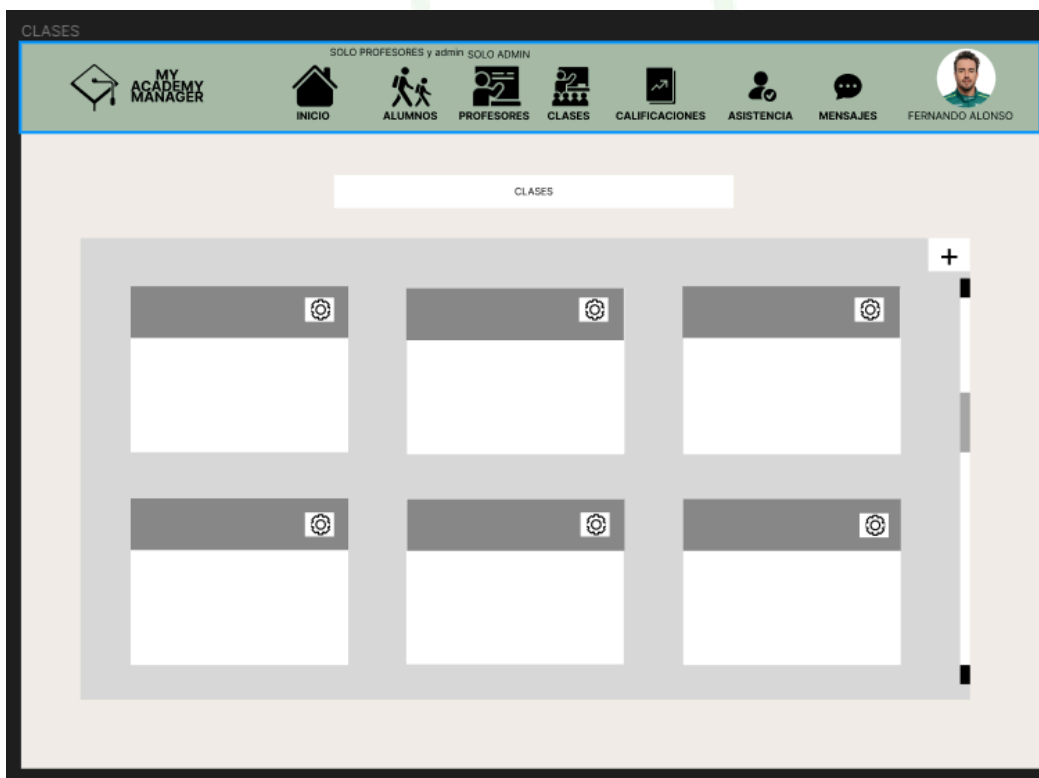
### 6.5.2.1 Login



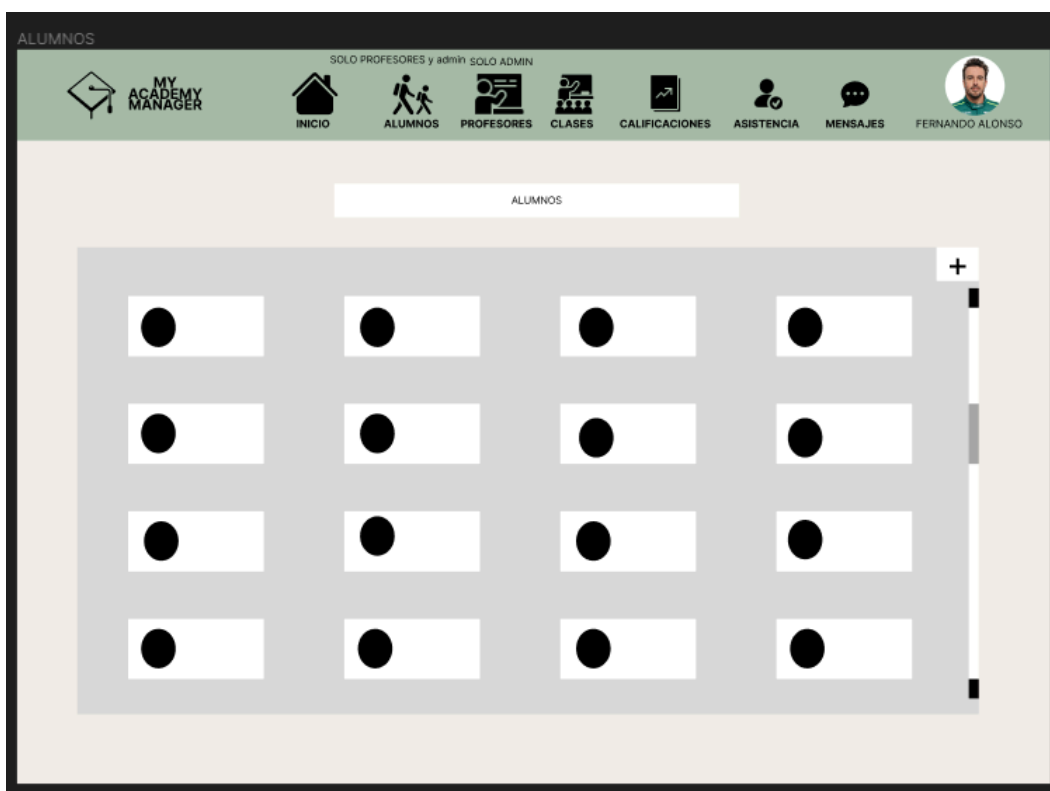
### 6.5.2.2 Home



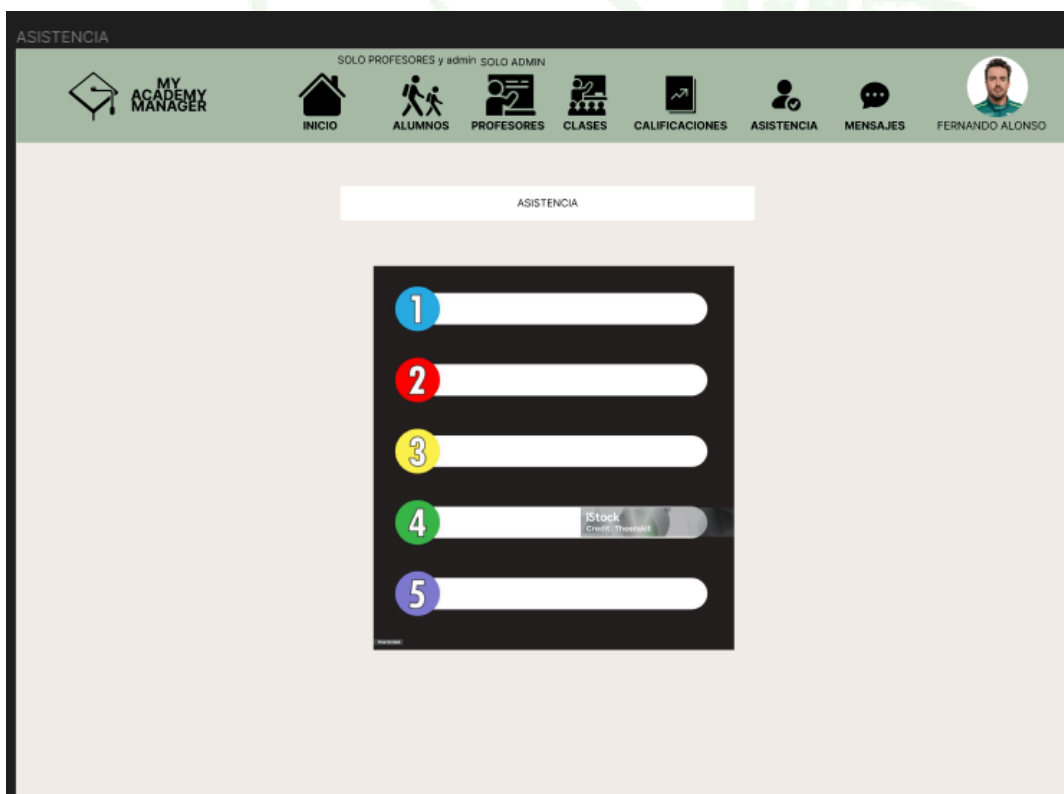
### 6.5.2.3 Clases



### 6.5.2.4 Usuarios



### 6.5.2.5 Asistencia



### 6.5.2.6 Mensajes

MENSAJES

SOLO PROFESORES y admin SOLO ADMIN

MY ACADEMY MANAGER

INICIO ALUMNOS PROFESORES CLASES CALIFICACIONES ASISTENCIA MENSAJES FERNANDO ALONSO

distribucion(SOLO PROFESOP

NUEVO MENSAJE

Usuario  Buscar

Usuario

Usuario

Usuario

Usuario

Usuario

Usuario

### 6.5.2.7 Calificaciones

CALIFICACIONES

SOLO PROFESORES y admin SOLO ADMIN

MY ACADEMY MANAGER

INICIO ALUMNOS PROFESORES CLASES CALIFICACIONES ASISTENCIA MENSAJES FERNANDO ALONSO

CALIFICACIONES

1

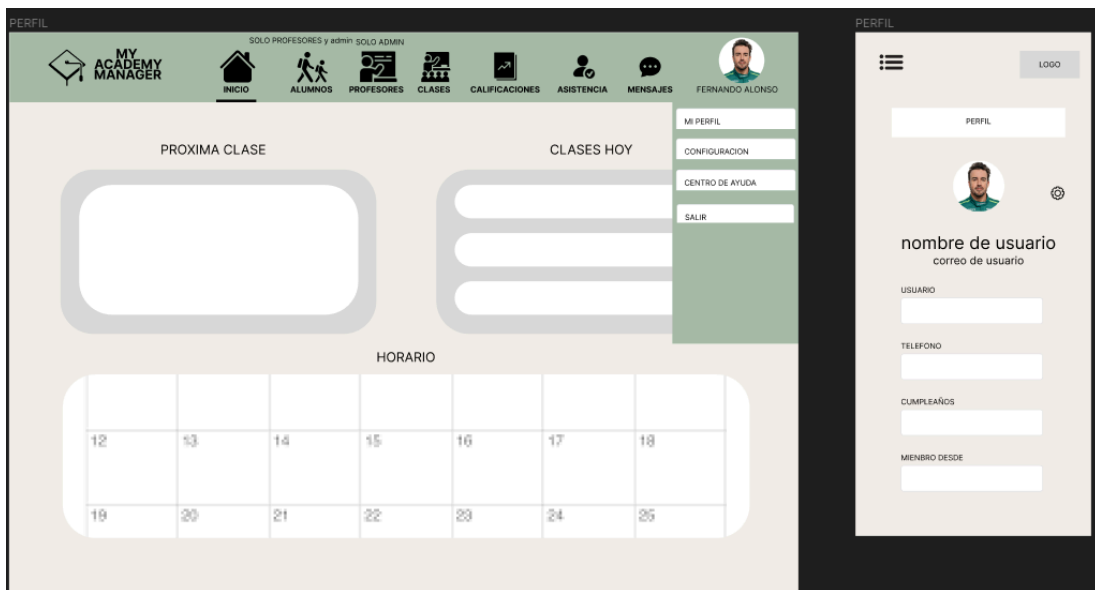
2

3

4  iStock Credit-Therapist

5

### 6.5.2.8 Perfil



### 6.5.3 Elección de colores y tipografía


Queríamos que nuestra aplicación fuera sencilla, sin mucha carga visual, para poder ubicarte rápidamente en las diferentes secciones de nuestra aplicación. Para ello, elegimos el color #F1EFE7 como nuestro color de fondo.

|||  HEX #F1EFE7 RGB 241, 239, 231 HSL 48, 4%, 93%

Como color primario, nos hemos decidido por un verde azulado suave y calmado (#A6BDA9). Este color evoca sensaciones de tranquilidad, frescura y naturaleza. Nos ha gustado especialmente por su aspecto relajante y armonioso.

|||  HEX #A6BDA9 RGB 166, 189, 169 HSL 128, 12%, 70%

Como color secundario, el tono verde grisáceo #4D6E51 complementa muy bien al verde azulado suave y calmado (#A6BDA9). Este color terroso y profundo añade una sensación de estabilidad y conexión con la naturaleza a la paleta. Además, crea un contraste visual interesante que puede resaltar elementos importantes del diseño, como texto o iconos.

 HEX #4D6E51 RGB 77, 110, 81 HSL 127, 30%, 37%

Juntos, estos tres colores forman una paleta armoniosa que evoca una sensación de tranquilidad y frescura, mientras que también añade profundidad y equilibrio al diseño. Creemos que esta combinación puede ser muy efectiva para transmitir la imagen deseada y crear una experiencia visual agradable para los usuarios.

#### 6.5.4 Accesibilidad

A continuación explicamos los criterios del primer principio del W3C que más se han tenido en cuenta haciendo MyAcademyManager.

##### 6.5.4.1 Contenido no textual (1.1.1 – nivel A)

El criterio 1.1.1 sobre contenido no textual establece que se deben proporcionar alternativas de texto para todo el contenido no textual.

```
<a routerlink="/inicio" class="flex items-center flex-shrink-0" *ngIf="identity">  
    
  <span id="nombre" class="text-center" style="vertical-align: middle; margin-bottom: 10px;">My<br>Academy<br>Manager</span>  
</a>
```

##### 6.5.4.2 Secuencia significativa (1.3.2 – nivel A)

El criterio 1.3.2 establece que, independientemente del método de interacción con la web, la presentación de la información en la pantalla debe seguir siempre una secuencia lógica.



#### 6.5.4.3 Características sensoriales (1.3.3 – nivel A)

El criterio 1.3.3 establece que cualquier tipo de instrucción o indicación no debe depender de un formato específico, ubicación espacial u otra característica sensorial.

#### 6.5.4.4 Identificar objetivo de entrada (1.3.5 – nivel AA)

El criterio 1.3.5 establece que los usuarios deben poder identificar claramente qué información deben proporcionar en los campos de un formulario.



A screenshot of a user registration form. The form is light beige with rounded corners. It contains the following fields, each with a label above it: 'Nombre' (Name), 'Apellidos' (Last names), 'DNI', 'Email', 'Contraseña' (Password), and 'Fecha de Nacimiento' (Date of birth). The 'Fecha de Nacimiento' field has a date picker icon on the right. The labels are in a small, dark font.

En cada uno de los formularios utilizados en MyAcademy Manager, esta claridad se logra mediante el uso de etiquetas (label) que indican explícitamente qué tipo de información debe introducirse en cada campo.

#### 6.5.4.5 Cambiar tamaño de texto (1.4.4 – nivel AA) y reflujo (1.4.10 – nivel AA)

El criterio 1.4.4 establece que al hacer zoom en la pantalla hasta un 200%, los textos deben mantenerse legibles y fáciles de leer, sin que se altere la presentación de la información.

Por otro lado, el criterio 1.4.10 indica que al aumentar el zoom hasta un 400%, la información debe adaptarse para evitar el desplazamiento horizontal.

Hemos logrado cumplir con estos criterios haciendo que todo el sitio web sea completamente responsive.

Para comprobar la accesibilidad hemos hecho un test en Lighthouse, que viene dentro de las herramientas de desarrollador de Google Chrome. obteniendo una puntuación de 93.



## Accesibilidad

Estas comprobaciones permiten identificar oportunidades para [mejorar la accesibilidad de tu aplicación web](#). La detección automática solo puede detectar un subconjunto de problemas y no garantiza la accesibilidad de tu aplicación web, por lo que también te recomendamos que hagas [pruebas manuales](#).

También hemos hecho pruebas de contraste entre los colores de los botones.



Contrast Finder

Contrast-Finder encuentra los contrastes adecuados de color para crear webs accesibles

Los criterios [WCAG](#) de éxito 1.4.3 requieren que el texto tenga un contraste mínimo de 4.5 (3 para textos largos).

Color del texto:  

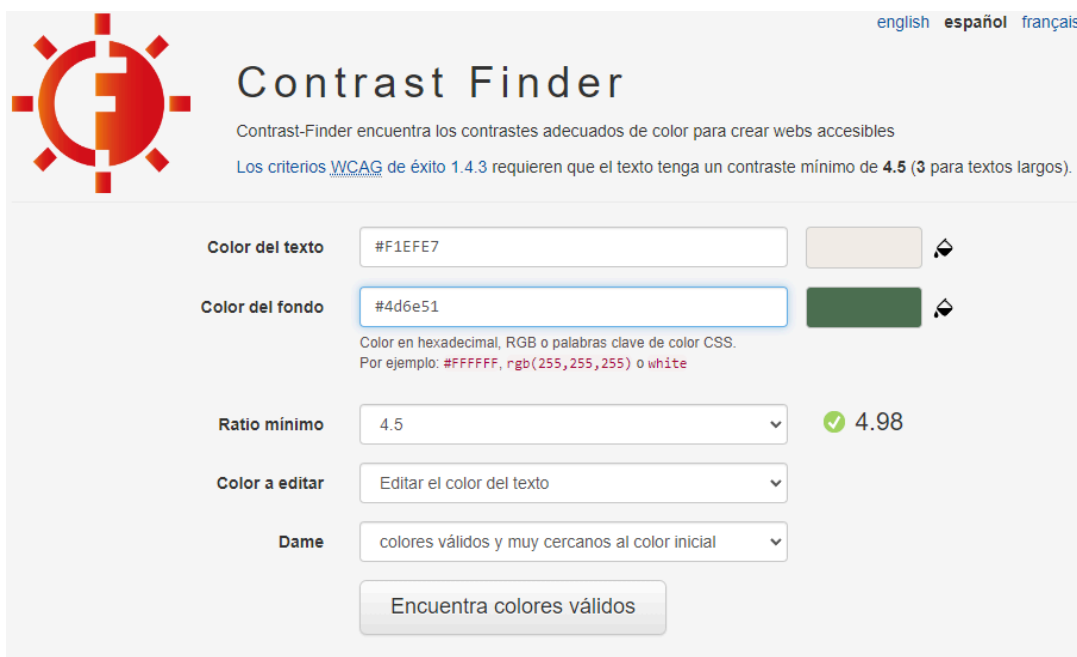
Color del fondo:  

Color en hexadecimal, RGB o palabras clave de color CSS.  
Por ejemplo: #FFFFFF, rgb(255,255,255) o white

Ratio mínimo:   9.32

Color a editar:

Dame:



The image shows the 'Contrast Finder' web application interface. At the top left is a red logo consisting of a stylized 'G' with radiating lines. To the right of the logo are language links: 'english', 'español', and 'français'. The main title 'Contrast Finder' is in a large, dark font. Below the title is a descriptive sentence in Spanish: 'Contrast-Finder encuentra los contrastes adecuados de color para crear webs accesibles'. Below that is a link to WCAG criteria: 'Los criterios [WCAG de éxito 1.4.3](#) requieren que el texto tenga un contraste mínimo de 4.5 (3 para textos largos)'. The interface contains several input fields and a button. The 'Color del texto' field has the value '#F1EFE7' and a corresponding color swatch. The 'Color del fondo' field has the value '#4d6e51' and a corresponding color swatch. Below the background color field is a small text explanation: 'Color en hexadecimal, RGB o palabras clave de color CSS. Por ejemplo: #FFFFFF, rgb(255,255,255) o white'. The 'Ratio mínimo' field has a dropdown menu set to '4.5' and a green checkmark icon next to the value '4.98'. The 'Color a editar' field has a dropdown menu set to 'Editar el color del texto'. The 'Dame' field has a dropdown menu set to 'colores válidos y muy cercanos al color inicial'. At the bottom is a button labeled 'Encuentra colores válidos'.

Contrast Finder

Contrast-Finder encuentra los contrastes adecuados de color para crear webs accesibles

Los criterios [WCAG de éxito 1.4.3](#) requieren que el texto tenga un contraste mínimo de 4.5 (3 para textos largos).

Color del texto: #F1EFE7

Color del fondo: #4d6e51

Color en hexadecimal, RGB o palabras clave de color CSS.  
Por ejemplo: #FFFFFF, rgb(255,255,255) o white

Ratio mínimo: 4.5 ✓ 4.98

Color a editar: Editar el color del texto

Dame: colores válidos y muy cercanos al color inicial

Encuentra colores válidos

### 6.5.5 Usabilidad

En cuanto a los formularios del sitio web, optamos por un diseño que utiliza etiquetas (label). Cuando el usuario coloca el cursor en el campo de texto o lo rellena, la etiqueta se posiciona sobre el campo para indicar claramente qué información se espera en cada momento. Además, todos los campos de los formularios se validan simultáneamente, mostrando los errores en color rojo, un estándar asociado comúnmente con alertas y errores, para que el usuario identifique fácilmente dónde se cometió un error.

Reconociendo la importancia de mejorar la experiencia del usuario, implementamos filtros que permiten al usuario organizar las clases, usuarios, alumnos, etc, según su prioridad de manera simple y efectiva. Todos los botones también están diseñados de manera consistente en forma y color, facilitando que el usuario los identifique rápidamente como botones interactivos.

Estas decisiones de diseño fueron tomadas con el objetivo de mejorar la usabilidad y la experiencia del usuario en MyAcademy Manager, asegurando que cada elemento visual y funcional esté diseñado para ser intuitivo, claro y efectivo.

## 7 Presupuesto

TAREAS	HORAS	PRECIO
Diseño de logotipo	2	24,00 €
Nombre de la aplicación	1	12,00 €
Prototipo baja fiabilidad	5	60,00 €
Prototipo alta fiabilidad	8	96,00 €
Paleta de colores	1	12,00 €
Diseño de la base de datos	4	48,00 €
Login y autenticación JWT	50	600,00 €
Maquetación en Angular	150	1.800,00 €
Landing Page	4	48,00 €
API REST	250	3.000,00 €
Chat	10	120,00 €
Despliegue	10	120,00 €
Documentación	20	240,00 €
Precio Hora	Total Horas	Total Presupuesto
12,00 €	515	6.180,00 €

## 8 Conclusiones y vías futuras

En conclusión, estamos conformes con los objetivos que planteamos al inicio del proyecto, habiendo conseguido la gran mayoría de ellos. El desarrollo del sitio ha logrado proporcionar una plataforma funcional y eficiente para la gestión de procesos académicos, facilitando la administración de información y mejorando la comunicación entre estudiantes y profesores.

Aunque somos conscientes de que hay aspectos a mejorar, hemos priorizado perfeccionar los elementos básicos y principales que nos planteamos al principio y, posteriormente, agregar algún elemento complementario fuera de la programación. Sin embargo, debido a la escasez de tiempo, no ha sido posible. Aun así, los tenemos en cuenta para un futuro.

Unas mejoras opcionales son:

- **Mejora de Hosting:** Conseguir una solución para el hosting y conseguir una mejora en el despliegue de la aplicación.
- **Personalización de foto de perfil:** Tener la capacidad de poner foto de perfil al usuario en vez de su inicial.
- **Mejora de iconos:** Tener unos mejores iconos más adaptados al diseño.
- **Mejora en Mensajes:** Tener la capacidad de poder bloquear usuarios para no recibir mensajes o hacer grupos de personas para mandar mensajes, de poder establecer los mensajes que ha leído el usuario para que cuando entre ponerlo directamente en los mensajes nuevos.
- Tener una aplicación un poco más accesible para móviles.
- **Interfaz multilingüe:** Ofrecer la interfaz de usuario en varios idiomas.
- **Ajuste de tema de la aplicación:** Permitir a los usuarios elegir diferentes temas y colores para sus perfiles.
- **Mejorar accesibilidad:** Mejora de accesibilidad en la página.

## 9 Manual de uso de la aplicación

En este manual de uso, revisaremos la aplicación y sus diferentes versiones dependiendo del tipo de usuario. Contamos con tres tipos de usuarios: profesor, alumno y administrador.

### 9.1 Inicio de Sesión

Iniciar sesión con correo electrónico y contraseña: este apartado es igual independientemente del tipo de usuario que seas.

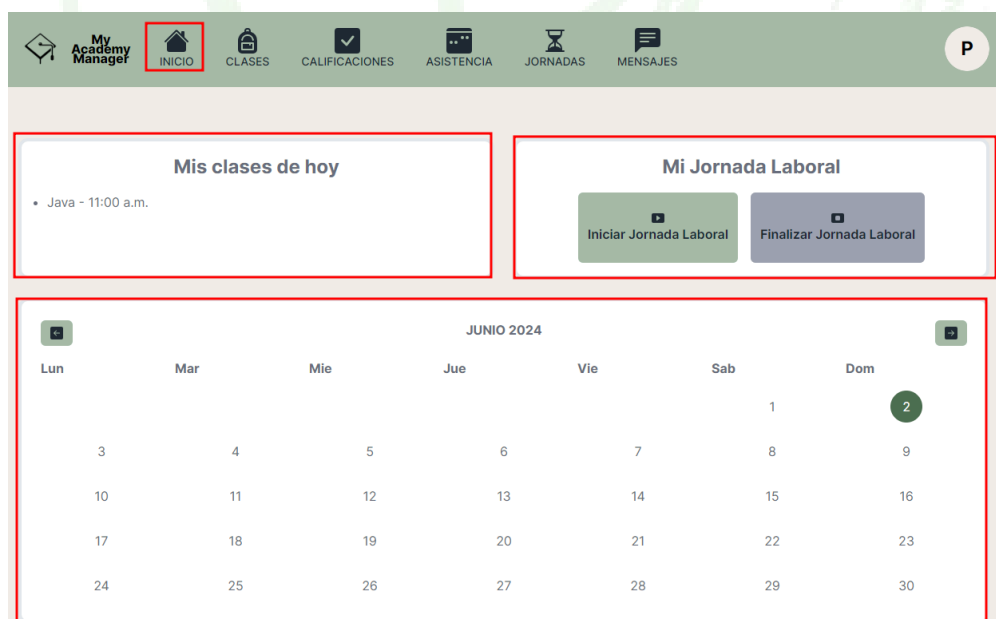


The login form is titled "My Academy Manager" and features a "Iniciar Sesión" button in the top right corner. The main form area contains two input fields: "Correo Electrónico" and "Contraseña", both with placeholder text. Below these fields is a green "Iniciar Sesión" button.

### 9.2 Home

#### 9.2.1 Profesor

Como podemos observar, en el home tendremos varios apartados. En el caso del usuario profesor, tenemos un calendario, las clases que se tienen ese día y un apartado con dos botones para poder iniciar y terminar la jornada laboral.

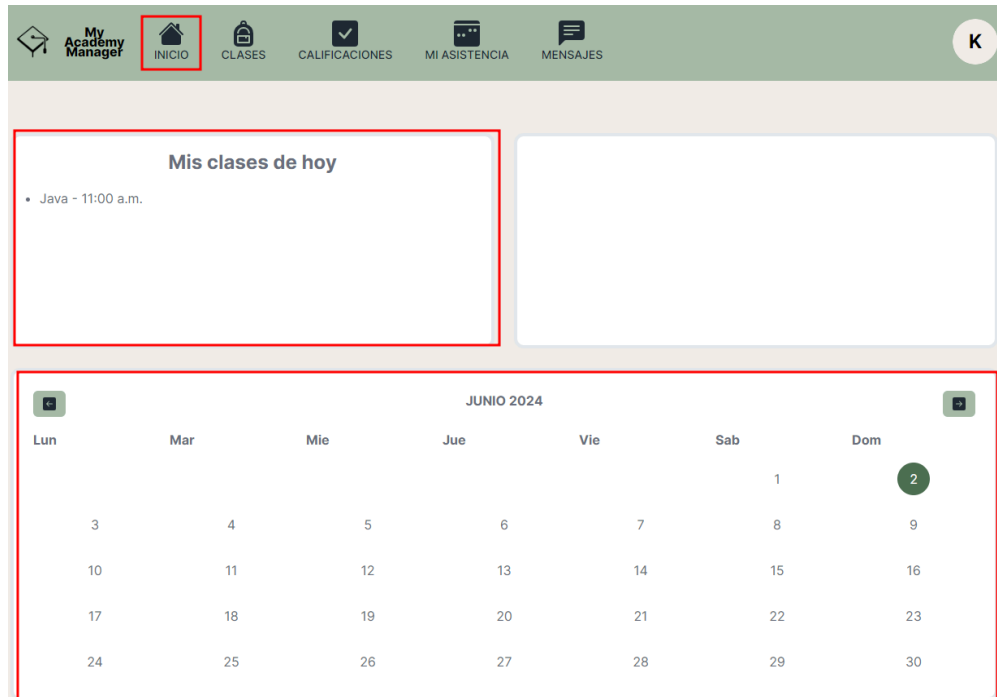


The home dashboard for a professor includes a navigation bar with icons for "INICIO", "CLASES", "CALIFICACIONES", "ASISTENCIA", "JORNADAS", and "MENSAJES". The main content area is divided into three sections: "Mis clases de hoy" (listing "Java - 11:00 a.m."), "Mi Jornada Laboral" (with "Iniciar Jornada Laboral" and "Finalizar Jornada Laboral" buttons), and a calendar for "JUNIO 2024". The calendar shows the days of the week and dates, with the 2nd of June highlighted.

Lun	Mar	Mie	Jue	Vie	Sab	Dom
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

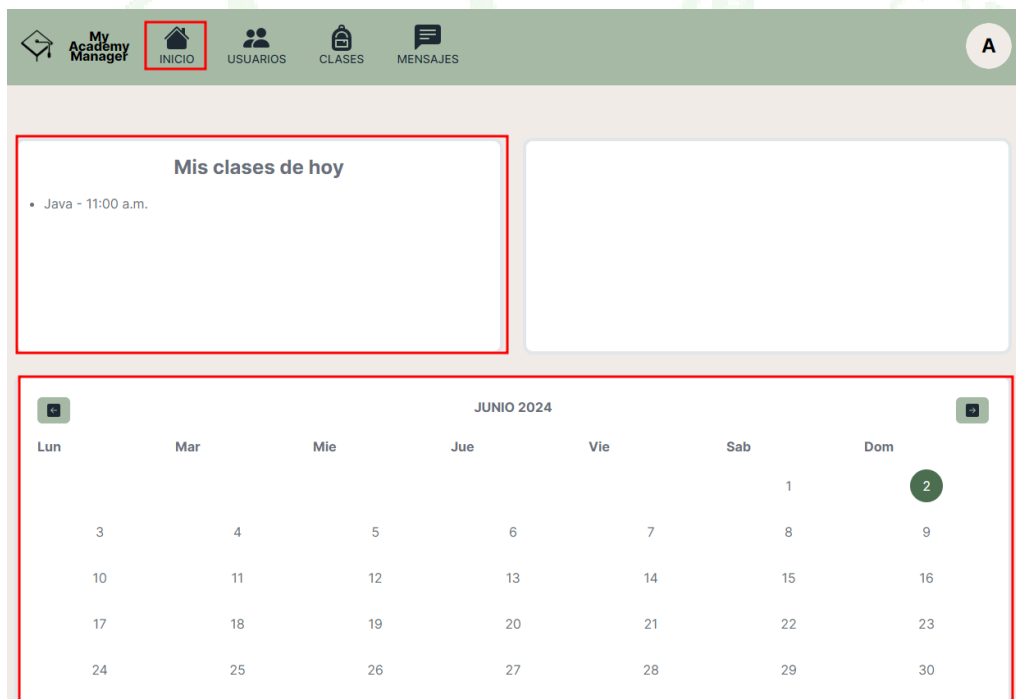
### 9.2.2 Alumno

Para el home del usuario alumno, podemos observar que hay dos apartados: el calendario y las clases que tiene hoy como alumno.



### 9.2.1 Admin

Al igual que el usuario alumno, podemos ver que tienen dos apartados: el calendario y todas las clases que hay en la academia ese día.



## 9.3 Clases

### 9.3.1 Profesor y Alumno

En el apartado de clases, que es igual para los usuarios alumno y profesor, se mostrarán todas las clases en las que está inscrito el alumno o las clases que imparte el profesor. Además, habrá un apartado de más información donde se podrá ver información sobre qué alumnos están en la clase y quién las imparte.

My Academy Manager

INICIO CLASES CALIFICACIONES MI ASISTENCIA MENSAJES K

Buscar clase...

**Química**  
Comienzo clases Química  
[Más información →](#)

**Análisis Musical**  
Introducción al análisis Musical. Curso 1  
[Más información →](#)

**Java**  
Programación en Java. POO. Clases y Estructuras de datos  
[Más información →](#)

My Academy Manager

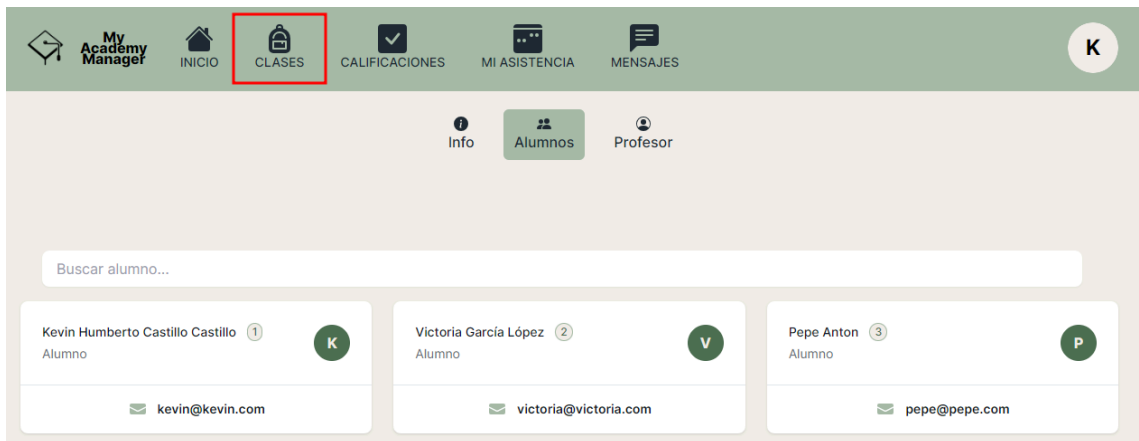
INICIO USUARIOS CLASES MENSAJES A

Volver

**Diseño**  
Información de la clase:

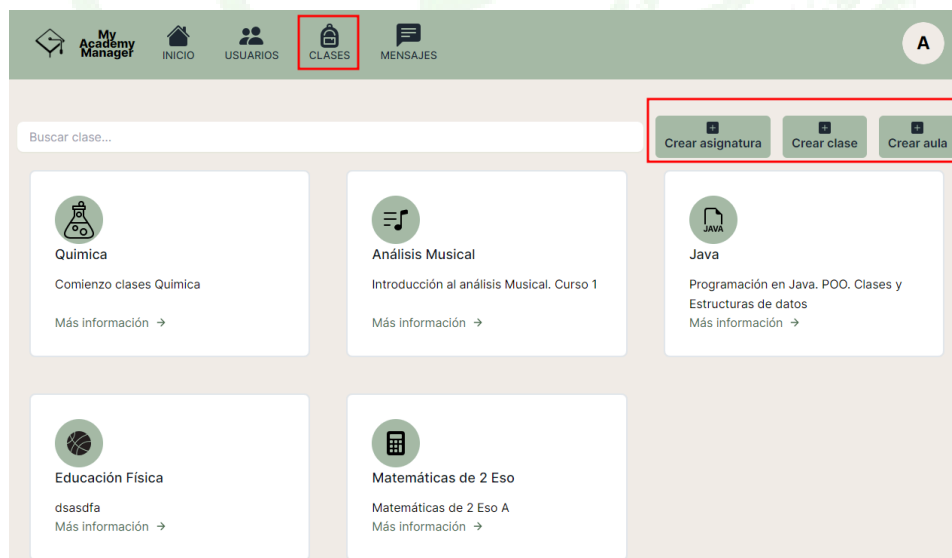
Descripción:	sd
Días:	Lunes, Miércoles, Viernes
Hora de inicio:	10:00
Hora de fin:	13:00
Aula:	Aula de Informática
Capacidad:	28





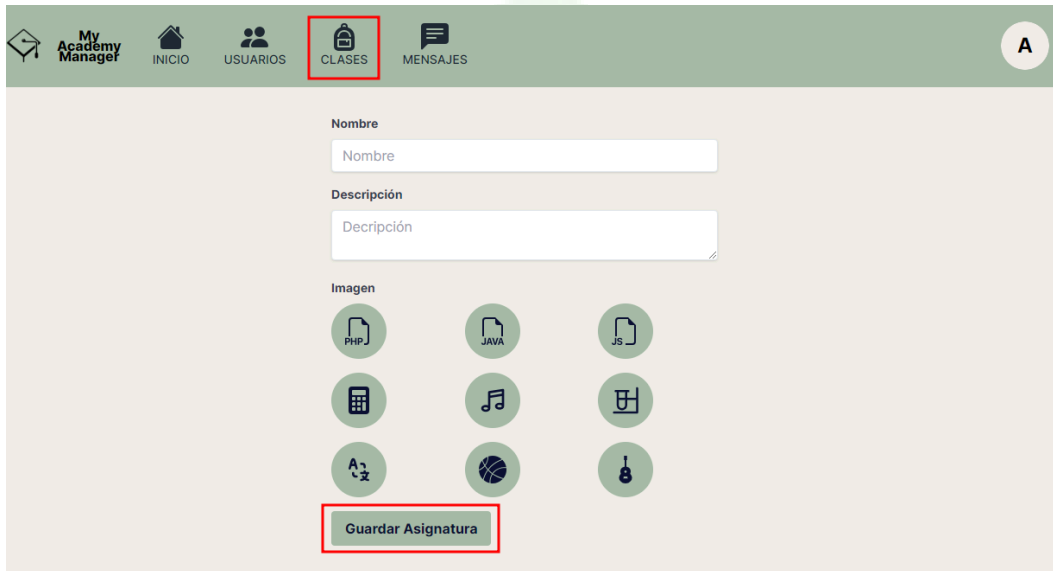
### 9.3.2 Admin

Para el usuario administrador, el apartado de clases cambia bastante, ya que tiene la capacidad de crear asignaturas, clases y aulas para los demás usuarios.



### 9.3.2.1 Crear asignatura

Al crear la asignatura, tendrá que introducir un nombre, una descripción y seleccionar una imagen para poder crearla.



My Academy Manager

INICIO USUARIOS CLASES MENSAJES A

Nombre  
Nombre

Descripción  
Descripción

Imagen

PHP JAVA JS

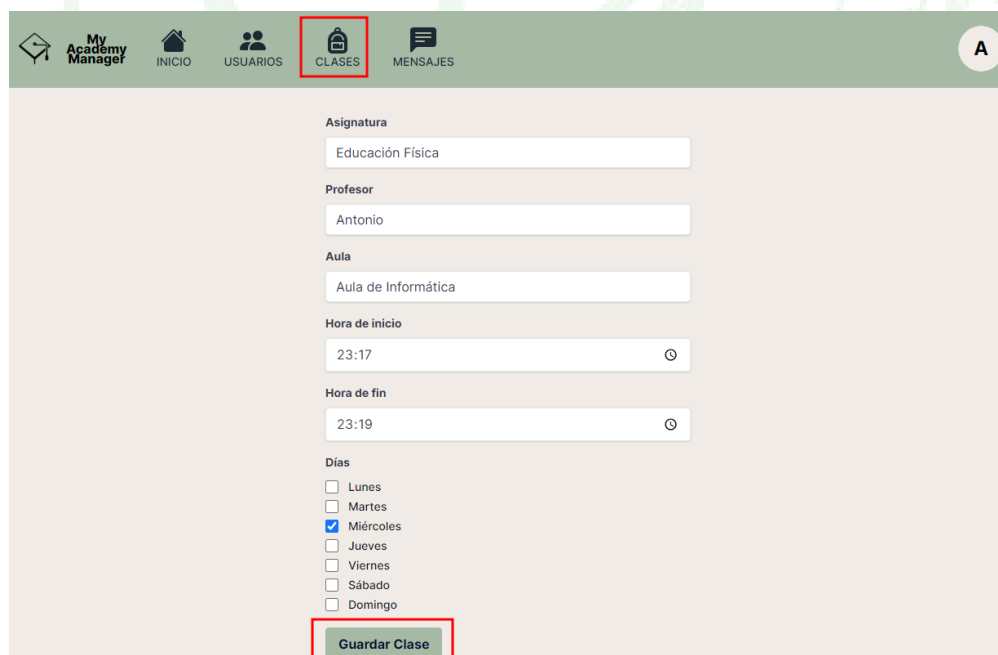
Calculator Musical Note Bar Chart

Person with Star Globe Guitar

Guardar Asignatura

### 9.3.2.2 Crear clase

En clase, tendrá que especificar a qué asignatura pertenece, qué profesor la impartirá, qué aula se utilizará, su hora de inicio y de fin, y los días en que se impartirá.



My Academy Manager

INICIO USUARIOS CLASES MENSAJES A

Asignatura  
Educación Física

Profesor  
Antonio

Aula  
Aula de Informática

Hora de inicio  
23:17

Hora de fin  
23:19

Días

☐ Lunes

☐ Martes

☒ Miércoles

☐ Jueves

☐ Viernes

☐ Sábado

☐ Domingo

Guardar Clase

### 9.3.2.3 Crear aula

Finalmente, para crear las aulas, únicamente tendrá que introducir el nombre y la capacidad que tendrán.

The screenshot shows the 'My Academy Manager' interface with the 'CLASES' (Classes) menu item highlighted. The form contains two input fields: 'Nombre' (Name) with the value 'aula1' and 'Capacidad' (Capacity) with the value '14'. A 'Crear Aula' (Create Class) button is at the bottom, highlighted with a red box.

## 9.4 Calificaciones

### 9.4.1 Profesor

En el apartado de calificaciones, tendrá que seleccionar la clase a la que quiere evaluar. A continuación, le aparecerá un listado con los alumnos que están en esa asignatura y podrá ingresar las notas y guardar los cambios.

The screenshot shows the 'My Academy Manager' interface with the 'CALIFICACIONES' (Grades) menu item highlighted. A dropdown menu is open, showing three options: 'Química - 06:00 p.m.', 'Java - 11:00 a.m.', and 'Educación Física - 11:11 a.m.'.

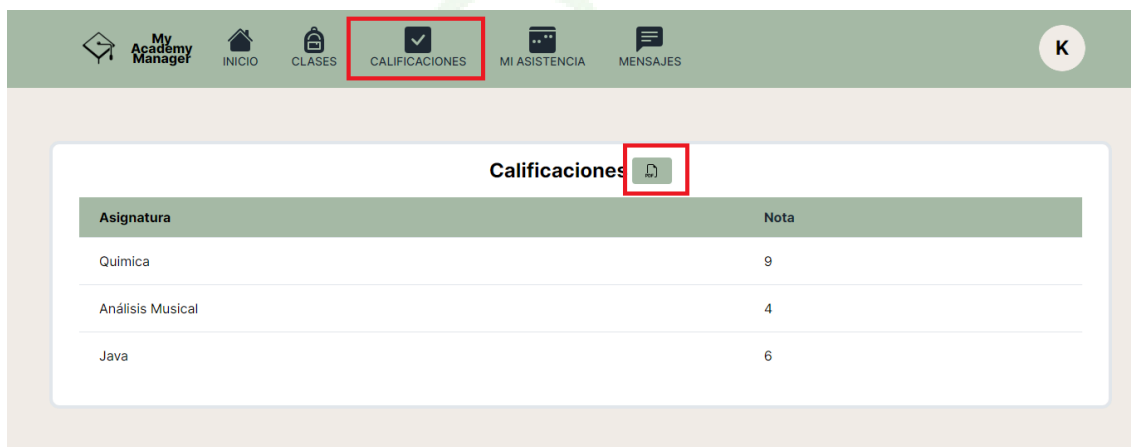
The screenshot shows the 'My Academy Manager' interface with the 'CALIFICACIONES' (Grades) menu item highlighted. The 'Clase' dropdown is set to 'Química - 06:00 p.m.'. Below it is a table with columns 'ALUMNO' and 'NOTA'.

ALUMNO	NOTA
Kevin Humberto Castillo Castillo	2
Victoria García López	1
Pepe Anton	1

A 'Guardar Calificaciones' (Save Grades) button is at the bottom, highlighted with a red box.

### 9.4.2 Alumno

Para las calificaciones, contamos con un resumen de las notas que tendremos en cada asignatura, además de la posibilidad de obtener las calificaciones en formato PDF.

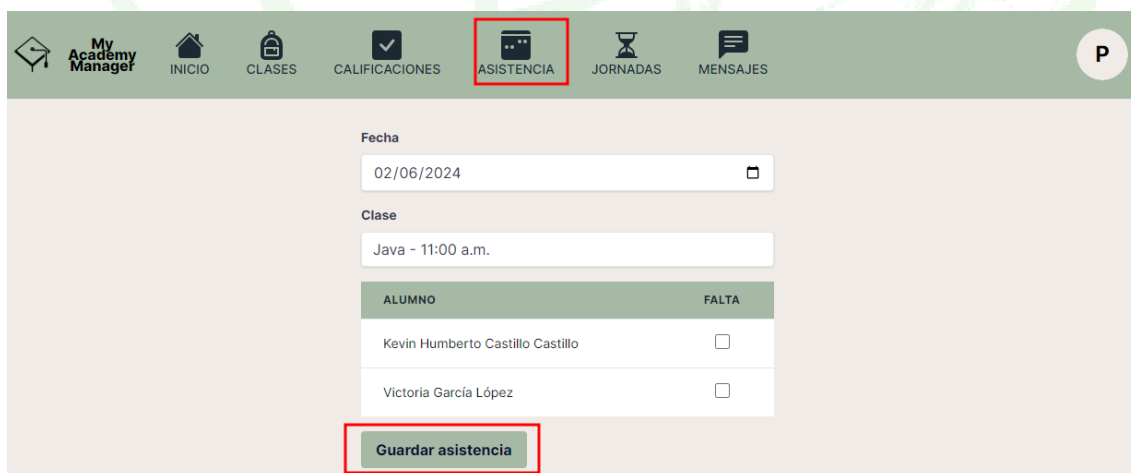


Asignatura	Nota
Química	9
Análisis Musical	4
Java	6

## 9.5 Asistencia

### 9.5.1 Profesor

En la sección de asistencia, deberá seleccionar el día y la clase en la que desee registrar las faltas y marcar con una marca de verificación al alumno que haya faltado ese día. Luego, pulse el botón para guardar las faltas de asistencia.




ALUMNO	FALTA
Kevin Humberto Castillo Castillo	<input type="checkbox"/>
Victoria García López	<input type="checkbox"/>

**Guardar asistencia**

### 9.5.2 Alumno

El alumno podrá ver todas las faltas de asistencia que tiene, además de un recuento de las faltas por clase.



**MI ASISTENCIA**

#### Faltas de Asistencia

**Miércoles, 17 de abril de 2024**  
Química - 12:00:00 AM  
Análisis Musical - 12:00:00 AM

**Martes, 16 de abril de 2024**  
Química - 12:00:00 AM

**Jueves, 18 de abril de 2024**  
Química - 12:00:00 AM

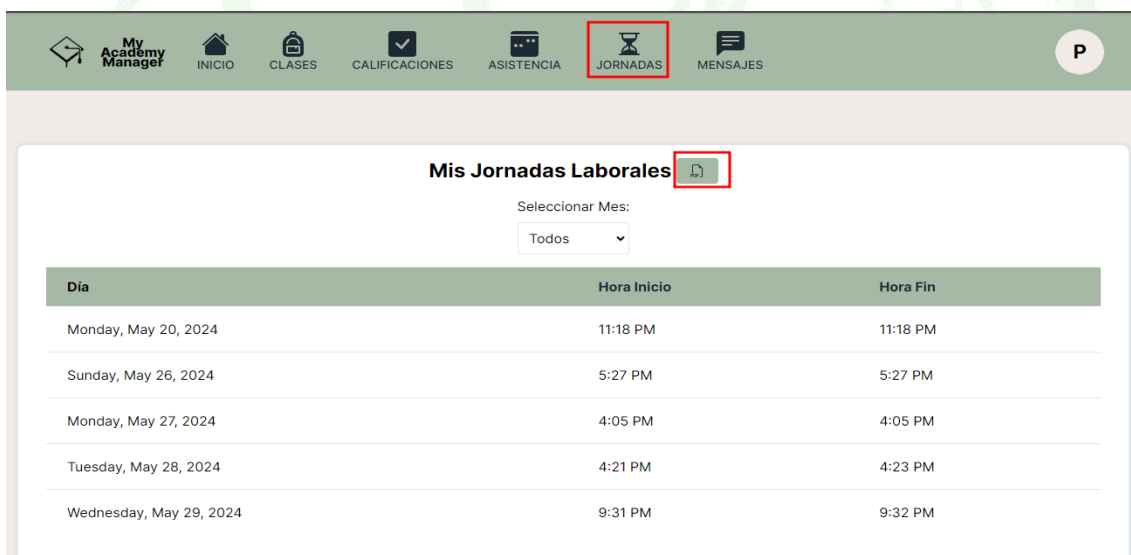
**Domingo, 12 de mayo de 2024**  
Química - 12:00:00 AM

**Miércoles, 29 de mayo de 2024**  
Química - 12:00:00 AM

Química - Faltas: 5  
Análisis Musical - Faltas: 1  
Java - Faltas: 0

### 9.6 Jornadas (Solo profesores)

Este apartado únicamente lo tienen los profesores. Aquí tendrán un apartado con sus jornadas trabajadas, pudiendo seleccionar el mes que desean visualizar. También cuentan con un botón para obtener sus jornadas en formato PDF.



**JORNADAS**

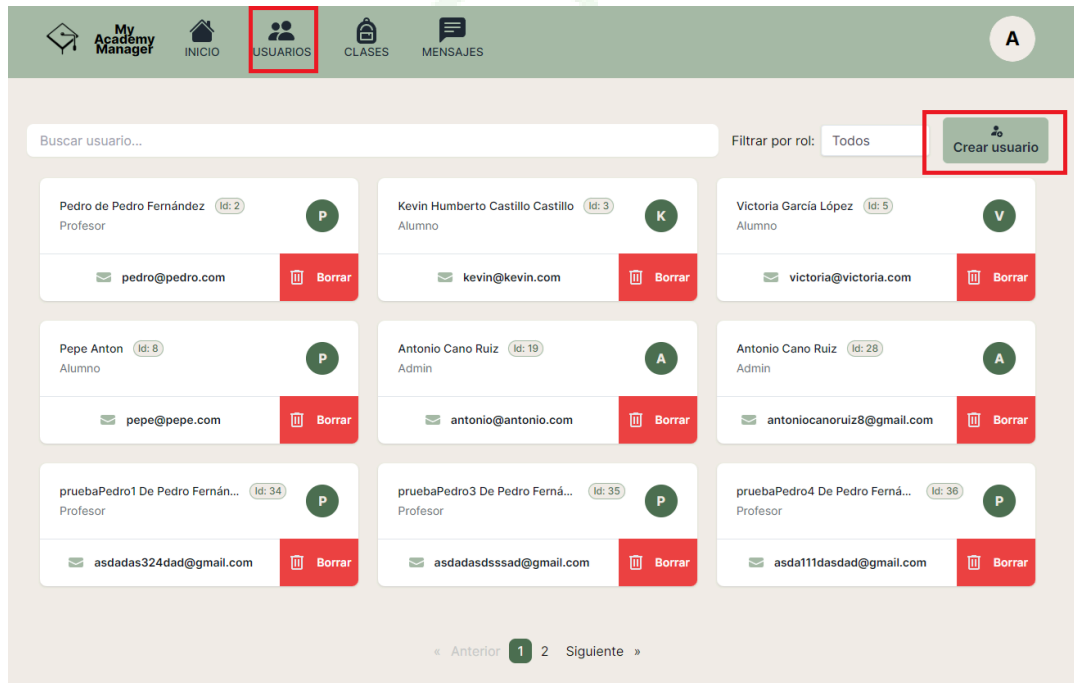
#### Mis Jornadas Laborales

Seleccionar Mes:  
Todos

Día	Hora Inicio	Hora Fin
Monday, May 20, 2024	11:18 PM	11:18 PM
Sunday, May 26, 2024	5:27 PM	5:27 PM
Monday, May 27, 2024	4:05 PM	4:05 PM
Tuesday, May 28, 2024	4:21 PM	4:23 PM
Wednesday, May 29, 2024	9:31 PM	9:32 PM

## 9.7 Usuarios (Solo admin)

Este apartado solo lo tienen los usuarios administradores. En él, se pueden crear usuarios. Únicamente tendrán que pulsar en el botón de "Crear usuario".



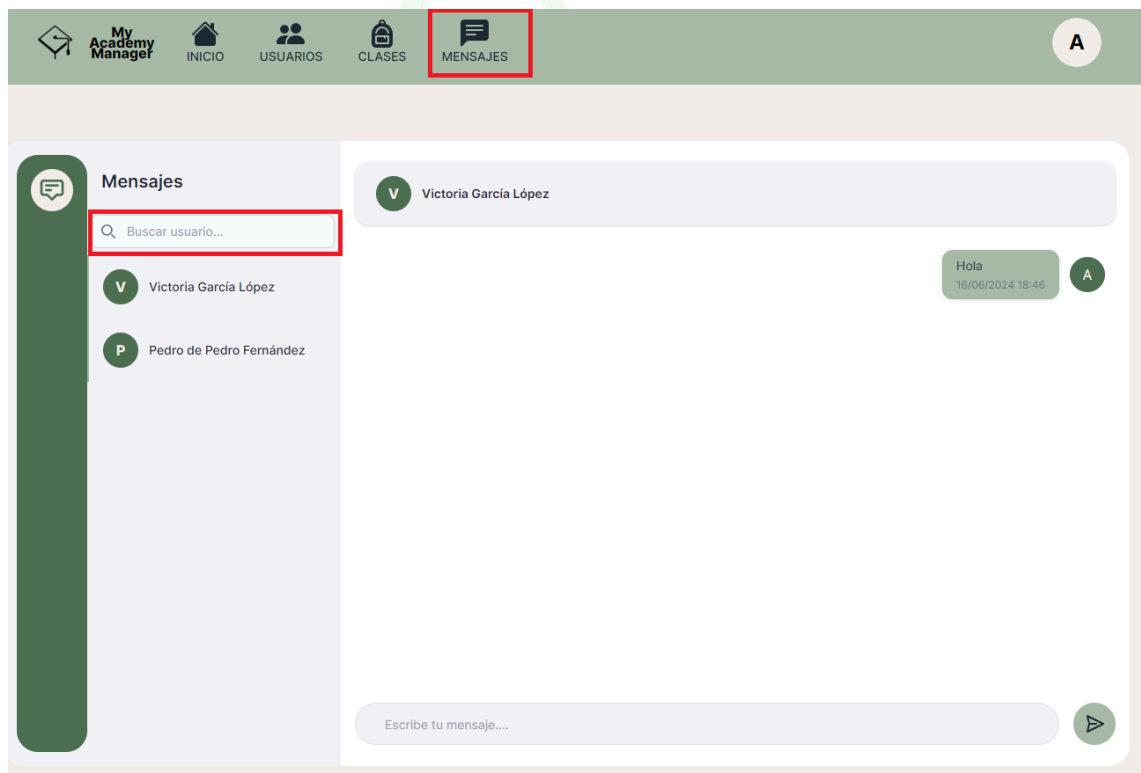
Una vez dentro, introduzca los datos para el usuario y pulse en el botón "Registrar Usuario".

The screenshot shows the 'Registrar Usuario' (Register User) form. At the top, there is a navigation bar with icons for 'My Academy Manager', 'INICIO', 'USUARIOS' (highlighted with a red box), 'CLASES', and 'MENSAJES'. A user profile icon 'A' is in the top right. Below the navigation bar, there is a green button labeled 'Volver' (Return). The form contains the following fields: 'Nombre' (Name) with value 'prueba', 'Apellidos' (Surnames) with value 'prueba', 'DNI' (ID Number) with value 'dniprueba', 'Email' with value 'prueba@gmail.com', 'Contraseña' (Password) with masked characters '.....', 'Fecha de Nacimiento' (Date of Birth) with value '11/11/1111' and a calendar icon, and 'Rol' (Role) with value 'Administrador'. A green button labeled 'Registrar Usuario' (highlighted with a red box) is at the bottom of the form.

## 9.8 Mensajes

"Mensajes" es uno de los apartados que no cambia, independientemente del usuario. Aquí podrás enviar y recibir mensajes de cualquier usuario que tenga cuenta en MyAcademy Manager. Su funcionalidad es similar a la de cualquier otro sistema de mensajería.

Para buscar a cualquier usuario y comenzar una conversación, deberás pulsar el símbolo buscarlo con el buscador que se encuentra en la parte superior.



## 10 Licencias

En MyAcademy Manager, hemos incluido los siguientes iconos:

“[House Fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Backpack Fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Check Square Fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Calendar Week Fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Hour Glass Split](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Chat left text fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Person Fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Person Fill Add](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Plus Square Fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Box arrow in left](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Record btn fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Arrow left square fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Arrow right square fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Filetype js](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Filetype java](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Filetype php](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Filetype pdf](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Calculator Fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[English To Chinese SVG Vector](#)” by [DazzleUI](#) licenciado bajo [CC Attribution License via SVG Repo](#)

“[Music SVG Vector](#)” by [DazzleUI](#) licenciado bajo [CC Attribution License via SVG Repo](#)



“[Test Tube](#)” by [SVG Repo](#) licenciado bajo [CC0 1.0 UNIVERSAL](#)

“[Arrow right square fill](#)” by [Bootstrap](#) licenciado bajo [MIT LICENSE](#)

“[Basketball SVG Vector](#)” by [Pixelbazaar](#) licenciado bajo [CC Attribution License via SVG Repo](#)

“[Guitar SVG Vector](#)” by [Keith Sample](#) licenciado bajo [CC Attribution License via SVG Repo](#)



## 11 Bibliografía / Webgrafía

- *Angular*. (2024). Angular.dev; Angular. <https://angular.dev/overview>
- *ChatGPT*. (2024). Chatgpt.com. <https://chatgpt.com/?oai-dm=1>
- *Master en webs Full Stack: Angular, Node, Laravel, Symfony +.* (2024). Udemy; <https://www.udemy.com/course/master-en-desarrollo-web-full-stack-angular-node-laravel-symfony>
- *Messages UI Layout | Pages*. (2024). Tailwindcomponents.com. <https://tailwindcomponents.com/component/messages-ui-layout>
- Miguel, L. (2020, January 17). *Qué es Json Web Token y cómo funciona*. OpenWebinars.net. <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>
- *Symfony*. (2024). Symfony.com; <https://symfony.com/projects/apiplatform>
- *Tailwind CSS - Rapidly build modern websites without ever leaving your HTML*. (2020, November 15). Tailwindcss.com; Tailwind CSS. <https://tailwindcss.com/>
- *Tailwind CSS Grid with profile image | Cards, Grid, List*. (2024). Tailwindcomponents.com. <https://tailwindcomponents.com/component/grid-with-profile-image-1>
- *Tailwind UI - Official Tailwind CSS Components & Templates*. (2024). Tailwindui.com. <https://tailwindui.com/?ref=top>
- Vergara, S. (2019, December 26). *Primeros pasos con Symfony 5 como API REST*. Blog ITDO - Agencia de Desarrollo Web, APPs Y Marketing En Barcelona; Blog ITDO - Agencia de desarrollo Web, APPs y Marketing en Barcelona. <https://www.itdo.com/blog/primeros-pasos-con-symfony-5-como-api-rest/>
- *DiagramGPT*. (2024). eraser.io. <https://www.eraser.io/diagramgpt>