



UNIVERSIDAD DE MÁLAGA



Graduado en Ingeniería del Software

Gestión de la Diversidad

Diversity Management

Realizado por
Peña Díaz, Carlos

Tutorizado por
Pastrana Brincones, José Luis

Departamento
Lenguajes y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

MÁLAGA, julio 2023



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA DEL SOFTWARE

GESTIÓN DE LA DIVERSIDAD

DIVERSITY MANAGEMENT

Realizado por
Peña Díaz, Carlos

Tutorizado por
Pastrana Brincones, José Luis

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JULIO DE 2023

Fecha defensa: septiembre de 2023

Resumen

En el ámbito académico, es común que la mayoría de los docentes no tengan la suficiente información sobre las condiciones especiales de sus alumnos, sobre todo si ni siquiera tienen acceso a documentación o informes que detallen esos datos. Esto puede llevar a acarrear problemas a la hora de enseñar a dichos estudiantes las materias correspondientes, desembocando en la frustración tanto de los alumnos, como de los profesores.

Con este trabajo final de grado, (TFG), se pretende dar una solución al problema en particular, mejorando el conocimiento de los instructores al tener acceso directo a los informes médicos de los estudiantes a su cargo, los cuales serán actualizados por los doctores titulares de dichos alumnos.

Dicho esto, los usuarios capaces de utilizar esta aplicación web podrán ser de cuatro tipos distintos: alumnos, profesores, médicos y administradores, siendo estos últimos los que tengan las funcionalidades más avanzadas para una mayor organización.

En cuanto al resto de los usuarios, podrán acceder a la información básica que tenga que ver con su rol, destacando el usuario médico, los cuales podrán incluso cambiar el informe de un alumno a su cargo.

Palabras clave: Aplicación web, Azure, Educación, .NET Core

Abstract

In the academic field, it is common that most teachers do not have enough information about the special conditions of their students, especially if they do not even have access to documentation or reports that detail this data. This can lead to problems when teaching these students, the corresponding subjects, leading to frustration for both students and teachers.

With this final degree project, (TFG), it is intended to provide a solution to the particular problem, improving the knowledge of the instructors by having direct access to the medical reports of the students under their charge, which will be updated by the regular doctors.

That said, the users capable of using these web applications may be of four different types: Students, teachers, doctors and administrators, the latter being the ones with the most complicated functionalities for greater organization. As for the rest of the users, they will be able to access basic information that has to do with their role, highlighting the medical user, who will even be able to change the report of a student in their charge.

Keywords: Web Application, Azure, Education, .NET Core

Índice

Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Metodología	2
1.4 Estructura de la memoria	4
Tecnologías y herramientas	7
2.1: C#	7
2.2: JavaScript	7
2.3: HTML	8
2.4: CSS	8
2.5: Bootstrap	8
2.6: SQL	9
2.7: LINQ	9
2.8: Visual Studio (2022)	9
2.9: GitHub	10
2.10: GitHub Desktop	10
2.11: SQL Server Management Studio	11
2.12: Draw.io	11
2.13: Lucidchart	11
2.14: SQL Server Express (2022)	12
2.15: Azure: Servicio web y base de datos	12
Análisis	13
3.1 Requisitos funcionales (Todos los Usuarios)	14
3.2 Requisitos funcionales (Alumno)	14
3.3 Requisitos funcionales (Profesor)	14
3.4 Requisitos funcionales (Médico)	15
3.5 Requisitos funcionales (Administrador)	15
3.6 Requisitos no funcionales	16

3.7 Casos de uso: Definición	17
3.8 Casos de uso para todos los usuarios	18
3.9 Casos de uso de alumnos	22
3.10 Casos de uso de profesores	24
3.11 Casos de uso de médicos	27
3.12 Casos de uso de administradores (Navegación)	30
3.13 Casos de uso de administradores (Creación)	36
3.14 Casos de uso de administradores (Borrado)	44
3.15 Casos de uso de administradores (Modificación)	48
Desarrollo e implementación	55
4.1 Preparación y aprendizaje	55
4.2 Iteraciones del back end	56
4.2.1 Iteración de la base de datos (Iteraciones 1 y 2)	56
4.2.1 Iteración del MVC y de las funcionalidades (Iteración 3 y 4)	66
4.3 Iteraciones del front end	71
4.3.1 Iteraciones principales de las vistas (Iteración 1 y 2)	71
4.3.1 Iteraciones avanzadas de las vistas (Iteración 3 y 4)	80
4.4 Diagramas de clase	84
4.4.1 Diagramas de clase UML: Modelos	85
4.4.2 Diagramas de clase UML: UserNavigation	92
4.4.3 Diagramas de clase UML: Constantes	93
4.4.4 Diagramas de clase UML: Interfaces	96
4.4.5 Diagramas de clase UML: ViewModels - Adminsitrador	96
4.4.6 Diagramas de clase UML: ViewModels - Alumnos	98
4.4.7 Diagramas de clase UML: ViewModels - Asignaturas	103
4.4.8 Diagramas de clase UML: ViewModels - Auditorías	105
4.4.9 Diagramas de clase UML: ViewModels - Informes	106
4.4.10 Diagramas de clase UML: ViewModels - Médico	109
4.4.11 Diagramas de clase UML: ViewModels - Pantallas	112
4.4.12 Diagramas de clase UML: ViewModels - Profesores	113
4.4.13 Diagramas de clase UML: ViewModels - Modificación de usuarios	117
4.4.14 Diagramas de clase UML: Controladores	119
4.5 Implementación del web service	137
4.6 Pruebas y testeos	138
Conclusiones y líneas futuras	159
5.1 Conclusiones	159
5.2 Líneas futuras	159
Referencias	163
Manual de despliegue e instalación	165

Manual de usuario	167
<i>B.1 Introducción</i>	<i>167</i>
<i>B.2 Inicio sesión como alumno</i>	<i>167</i>
<i>B.3 Ver las asignaturas de un alumno</i>	<i>168</i>
<i>B.4 Ver los informes de un alumno</i>	<i>169</i>
<i>B.5 Inicio sesión como profesor</i>	<i>170</i>
<i>B.6 Ver las asignaturas de un profesor</i>	<i>171</i>
<i>B.7 Acceder al perfil principal de los alumnos de un profesor</i>	<i>171</i>
<i>B.8 Inicio sesión como médico</i>	<i>172</i>
<i>B.9 Acceder al perfil principal de los alumnos de un médico</i>	<i>173</i>
<i>B.10 Ver los informes de un médico</i>	<i>174</i>
<i>B.11 Modificar el contenido de un informe</i>	<i>174</i>
<i>B.12 Inicio sesión como administrador</i>	<i>175</i>
<i>B.13 Ver la lista de profesores y acceder a su perfil principal</i>	<i>175</i>
<i>B.14 Ver lista de médicos y acceder a su perfil principal</i>	<i>176</i>
<i>B.15 Ver lista de todos los alumnos y acceder a su perfil principal</i>	<i>177</i>
<i>B.16 Ver lista de todas las asignaturas</i>	<i>177</i>
<i>B.17 Ver lista de todas las matrículas</i>	<i>178</i>
<i>B.18 Ver listas de todas las docencias</i>	<i>179</i>
<i>B.19 Ver lista de todos los informes</i>	<i>179</i>
<i>B.20 Lista de la información sobre las pantallas</i>	<i>180</i>
<i>B.21 Lista de todas las auditorías</i>	<i>180</i>
<i>B.22 Crear un nuevo alumno</i>	<i>181</i>
<i>B.23 Borrar un alumno</i>	<i>181</i>
<i>B.24 Modificar un alumno</i>	<i>182</i>
<i>B.25 Crear un nuevo profesor</i>	<i>183</i>
<i>B.26 Borrar un profesor</i>	<i>184</i>
<i>B.27 Modificar un profesor</i>	<i>184</i>
<i>B.28 Crear un nuevo médico</i>	<i>185</i>
<i>B.29 Borrar un médico</i>	<i>186</i>
<i>B.30 Modificar un médico</i>	<i>187</i>
<i>B.31 Crear una nueva asignatura</i>	<i>188</i>
<i>B.32 Borrar una asignatura</i>	<i>188</i>
<i>B.33 Modificar nombre de una asignatura</i>	<i>189</i>
<i>B.34 Crear una nueva matrícula</i>	<i>189</i>

<i>B.35 Borrar una matrícula</i>	<i>190</i>
<i>B.36 Crear una nueva docencia</i>	<i>190</i>
<i>B.37 Borrar una docencia</i>	<i>191</i>
<i>B.38 Crear un nuevo informe</i>	<i>191</i>
<i>B.39 Borrar un informe</i>	<i>192</i>
<i>B.40 Cambiar el médico titular de un informe</i>	<i>192</i>
<i>B.41 Cerrar sesión de un usuario</i>	<i>193</i>
<i>B.42 Volver al perfil principal de un usuario</i>	<i>193</i>

1

Introducción

1.1 Motivación

Debido al gran número de alumnos que los profesores pueden llegar a tener en la actualidad, es normal que no puedan dedicarle el tiempo necesario a profundizar de una mejor manera en la enseñanza de sus alumnos, especialmente si los estudiantes tienen alguna característica extra que les brinde una capa de dificultad adicional para poder seguir el ritmo educativo.

En este proyecto, se quiere poder facilitar la información de dichos estudiantes a los profesores para que puedan encontrar una manera de poder ayudarlos de una forma eficaz y con la mayor rapidez posible.

1.2 Objetivos

Este proyecto consiste en una aplicación web en la que un usuario podrá iniciar sesión para poder acceder a una cantidad específica de información y acciones dependiendo de su rol asignado. Estos cuatro roles son:

1. **Alumno:** El rol de alumno tendrá los permisos necesarios para poder acceder a la información que tenga que ver con sus datos básicos (nombre, apellidos y NIF), las asignaturas a las que esté matriculado y sus propios informes.
2. **Profesor:** El rol de profesor tendrá los permisos necesarios para poder acceder a la información que tenga que ver con sus datos básicos (nombres, apellidos y

NIF) y las asignaturas que imparte.

Además, podrá acceder a la información básica y a todos los informes de los alumnos que estén matriculados en alguna de las asignaturas que imparta el profesor.

Cuando un alumno deja de estar matriculado en alguna de las asignaturas que imparta el profesor, dicho profesor no podrá acceder más a la información del alumno anteriormente mencionada.

3. **Médico:** El rol del médico tendrá los permisos necesarios para poder acceder a la información que tenga que ver con sus datos básicos (nombre, apellidos y NIF), y a la información de los informes que esté a cargo.

También podrá acceder a los datos básicos y a los informes de los alumnos que tengan al menos un informe cuyo titular sea el propio médico.

En adición, será el único rol capaz de modificar el contenido de los informes de los alumnos, pero solo de aquellos informes en los que sea médico titular.

4. **Administrador:** El rol del administrador tendrá los permisos necesarios para poder acceder a toda la información, incluido informes, matrículas, docencias y otra información más especializada como las pantallas o auditorías.

En cuanto a las acciones, será el encargado de crear, borrar y modificar usuarios, además de la creación y borrado de matrículas, docencias, asignaturas e informes. (Aunque no puede modificar un informe como lo haría un médico).

También puede modificar los nombres de usuarios y contraseñas de los demás usuarios: si un usuario quiere cambiar la contraseña y/o usuario por cualquier motivo, tendrá que contactar con el administrador para que le provisione de nuevas credenciales.

1.3 Metodología

Para realizar el proyecto, se optó por elegir la metodología ágil SCRUM debido a la flexibilidad y mejora continua que este método ofrece. Se siguió una serie de iteraciones de forma organizada, realizando tareas específicas en cada una de ellas.

Sin embargo, antes de entrar en todo ese proceso, se tuvo que realizar una preparación

anterior.

En primer lugar, hubo que hacer un aprendizaje básico sobre el lenguaje de programación principal que se iba a usar, el cual sería **C#**, además de aprender por primera vez el entorno de Azure y todos los servicios que este tenía a su disposición. Para ello se realizó varios proyectos de pruebas, que, aunque no eran muy complicados, sí que servían para comprender correctamente el funcionamiento del lenguaje. Esto también se aplica a Azure, pues antes de tener una base de datos y dominio estables, se crearon unos cuantos dominios y bases de datos de prueba para entender mejor cómo funcionaba Azure en general.

Al terminar esta formación, se dispuso a elegir e instalar las herramientas necesarias que se utilizarían en un futuro para el desarrollo de esta aplicación web. Dichas herramientas serán descritas más adelante en este documento.

Finalizada esta fase, se optó por aprender de forma básica algún lenguaje de programación complementario para evitar inconvenientes futuros, como **JavaScript**, a la vez que se iba profuncizando en lenguajes de marcado y estilo como lo son **HTML** y **CSS**.

Después del último punto, se empezó el desarrollo de la aplicación web, realizándose en diferentes iteraciones como se tenía previsto.

Al final de todas las iteraciones, y después de haber consultado y corregido dudas con el cliente, el cual en este caso sería José Luis Pastrana, se realizó pruebas a la aplicación para comprobar el correcto funcionamiento de esta. Si se encontraba algún error o bug, se arreglaba al momento.

Por último, se realizó toda la documentación necesaria para este proyecto.

1.4 Estructura de la memoria

Esta memoria ha sido dividida en distintas secciones para una mejor comprensión de ésta.

Las partes que las componen son:

1. **Introducción:** En este apartado se habla de forma general sobre los objetivos y motivaciones del proyecto. También se comenta acerca de la metodología y arquitectura usada.
2. **Tecnologías y herramientas:** En este apartado se informa sobre las herramientas y las tecnologías empleadas para la realización del proyecto. Esto incluye lenguajes de programación, entornos de desarrollo (IDE), base de datos y editores de texto.
3. **Análisis:** En este apartado se profundiza en el análisis del software del proyecto. Para un mejor entendimiento, se agregarán casos de uso y requisitos funcionales y no funcionales.
4. **Desarrollo e implementación:** En este apartado se explicará las distintas iteraciones que se han estado realizando a lo largo del tiempo, detallando en cada una de ellas el desarrollo que se ha ido efectuando. También se explicará más en profundidad la estructura interna de la aplicación web.
5. **Conclusiones y líneas futuras:** En esta sección se expondrá las distintas conclusiones a las que se ha llegado, y se comentarán algunas ideas sobre las posibles mejoras a realizar sobre el proyecto a futuro.
6. **Referencias:** En este apartado se añadirán enlaces de interés y que han sido de ayuda a la hora de desarrollar esta memoria.
7. **Despliegue e instalación:** En este apartado se informará al usuario de los navegadores en los que se podrá utilizar la aplicación web, además de datos adicionales sobre el despliega de la aplicación.

8. **Manual de usuario:** Este apartado es un apéndice de cómo funciona la aplicación web. Se explicará paso por paso y de manera detallada todas las acciones que un usuario podrá realizar, además de incluir imágenes para facilitar su comprensión.

2

Tecnologías y herramientas

2.1: C#



Figura 1
Logo C#

Es un lenguaje de programación moderno, basado en objetos y con seguridad de tipos. Desarrollado por Microsoft, tiene sus raíces en los lenguajes C y C++, y forma parte de la plataforma .NET [1].

Al tener que desarrollar la aplicación web en Azure, se ha optado por utilizar .NET Core 7, el cual ya tiene integrado herramientas especializadas en C#, haciendo más fácil y cómodo la programación en este lenguaje.

2.2: JavaScript



Figura 2
Logo
JavaScript

JavaScript es un lenguaje de programación que se utiliza en el desarrollo web para crear contenido dinámico e interactivo en una página web. Es un lenguaje de programación de alto nivel, interpretado y con tipado dinámico [2].

Aunque en el trabajo apenas se ha usado activamente JavaScript, ha sido bastante útil a la hora de diseñar elementos dinámicos para la aplicación web, al mismo tiempo que ha sido fundamental para el uso de Bootstrap.

2.3: HTML



Figura 3
Logo
HTML

HTML(*Hyper Text Markup Language*) es el lenguaje de marcado estándar utilizado para crear páginas web. Consiste en una serie de etiquetas que describen el contenido de una página, como encabezados, párrafos, enlaces, imágenes y formularios. Los navegadores web interpretan el código HTML y lo representan visualmente en una página web [3].

HTML ha sido un pilar clave en los aspectos básicos visuales del trabajo, ocupando un buen porcentaje del front end y siendo apoyado por otros elementos como JavaScript o CSS.

2.4: CSS



Figura 4
Logo
CSS

CSS(*Cascading Style Sheets*) es un lenguaje utilizado para dar estilo y formato a los documentos HTML. Permite separar la estructura y el contenido del diseño visual, lo que hace que el código sea más mantenible y escalable [4].

CSS ha sido utilizado para ayudar a los elementos HTML a estar mejor representados en la pantalla.

2.5: Bootstrap



Figura 5
Logo
Bootstrap

Bootstrap es un framework de diseño web de código abierto que se utiliza para crear interfaces de usuario (UI) y experiencias de usuario (UX) modernas y receptivas. Se basa en HTML, CSS y JavaScript, y proporciona una variedad de componentes y herramientas para la creación de sitios web y aplicaciones web [5].

El proyecto creado en .NET Core 7 ya venía integrado con Bootstrap (versión 5.1), por lo que se ha aprovechado esta característica para mejorar el estilo del proyecto.

2.6: SQL



Figura 6
Logo SQL

SQL(*Structured Query Language*) es un lenguaje de programación utilizado para gestionar bases de datos relacionales. Desarrollado por IBM en la década de 1970, SQL se utiliza para realizar tareas como la creación y modificación de base de datos, además de la inserción, eliminación y actualización de datos. Es un lenguaje de consulta que permite especificar qué datos se deben recuperar de una base de datos y cómo se deben ordenar, filtrar y agrupar [6].

Dado que en el proyecto se trabaja con base de datos relacionales, el uso del SQL es la elección por defecto para manejar las bases de datos, además de ya estar familiarizado con este lenguaje.

2.7: LINQ



Figura 7
Logo
LINQ

LINQ(*Language Integrated Query*) es una tecnología de Microsoft que permite consultar y manipular datos de manera declarativa utilizando el lenguaje de programación C#. Introducido por primera vez en la versión .NET Framework 3.5, su sintaxis y uso es similar al de SQL [7].

Aprovechando que el lenguaje de programación principal es C#, y que se utiliza SQL en el proyecto, se ha optado también por utilizar LINQ para aprovechar sus funcionalidades específicas, como una sintaxis más intuitiva.

2.8: Visual Studio (2022)



Figura 8
Logo
Visual
Studio

Visual Studio es un entorno de desarrollo integrado (IDE) creado por Microsoft, que se utiliza para desarrollar aplicaciones de software para Windows, Android, iOS, web, y la nube. Ofrece una amplia variedad de herramientas y funciones para programadores, incluyendo un editor de código, depurador, integración con control de versiones, herramientas de pruebas y diagnósticos entre otros servicios [8].

Se ha elegido utilizar Visual Studio 2022, (en específico la versión Community) por varios motivos. Entre ellos, la familiarización con el entorno, el soporte para múltiples lenguajes de programación, y la gran cantidad de herramientas que ofrece entre muchas otras razones.

2.9: GitHub



Figura 9
Logo
GitHub

GitHub es una plataforma web para alojar y compartir código fuente. Permite a los desarrolladores y equipos de desarrollo trabajar juntos y mantener un registro de los cambios realizados en el código. Además de alojar código, también proporciona herramientas de seguimiento de problemas, revisión de código y colaboración en proyectos, utilizando el control de versiones de Git [9].

Se ha utilizado GitHub principalmente por su utilidad al controlar versiones del código y almacenar este en un repositorio.

2.10: GitHub Desktop



Figura 10
Logo
GitHub
Desktop

GitHub Desktop es una aplicación de escritorio que permite a los desarrolladores trabajar con Git y GitHub sin tener que usar la línea de comandos. La herramienta proporciona una interfaz gráfica de usuario intuitiva para realizar operaciones comunes de Git, como crear ramas o fusionar cambios [10].

GitHub Desktop se ha utilizado como un apoyo para GitHub, provocando que las acciones como subir los cambios al repositorio, o recuperar versiones antiguas del código sea mucho más fácil e intuitivo.

2.11: SQL Server Management Studio



Figura 11
Logo
SSMS

SQL Server Management Studio (SSMS a partir de ahora), es una herramienta de administración y desarrollo de base de datos de Microsoft SQL Server. Proporciona un entorno integrado para administrar y mantener instancias de SQL Server, crear y modificar objetos de base de datos, escribir y depurar scripts, y generar informes [11].

SSMS ha sido una herramienta especialmente útil a la hora de diseñar la base de datos en las primeras fases del proyecto, además de ser más fácil de usar que otras aplicaciones similares.

2.12: Draw.io



Figura 12
Logo
Draw.io

Draw.io es una herramienta de diagramación en línea que permite crear diagramas y visualizaciones de forma rápida y sencilla. Además, al ser una aplicación web, no requiere instalación, y es accesible desde cualquier dispositivo con conexión a Internet [12].

Draw.io se ha utilizado con la principal finalidad de poder realizar los casos de uso, y ha sido elegido por ser intuitivo a la hora de aprender cómo funciona la aplicación.

2.13: Lucidchart



Figura 13
Logo
Lucidchart

Lucidchart es una herramienta en línea para la creación de diagramas y visualización de ideas. Es ampliamente utilizada para diseñar todo tipo de diagramas, facilitando la visualización de ideas y la colaboración de proyectos [13].

Lucidchart se ha empleado principalmente para hacer los diagramas de clase UML por su gran abanico de opciones a la hora de crearlas.

2.14: SQL Server Express (2022)



Figura 14
Logo SQL
Server
Express

SQL Server Express es una edición gratuita y ligera del sistema de gestión de bases de datos relacionales SQL Server. Desarrollado por Microsoft, está diseñado para aplicaciones y proyectos de tamaño pequeño a mediano, y ofrece una amplia gama de características y capacidades para el almacenamiento y procesamiento de datos [14]. SQL Server Express fue utilizado principalmente como primera base de datos en local antes de trasladar la base de datos principal a Azure.

2.15: Azure: Servicio web y base de datos



Figura 15
Logo
Azure

Azure es una plataforma de servicios en la nube ofrecida por Microsoft. Proporciona una amplia gama de servicios y soluciones para el desarrollo, implementación y administración de aplicaciones, almacenamiento de datos, análisis de redes y más. Entre los servicios que incluye, se ha escogido dos de ellos: el de servicio web, y el de base de datos. (Un servicio web es una aplicación basada en la web que permite la comunicación y la interacción entre diferentes sistemas informáticos a través de Internet) [15].

Se escogió Azure, porque estos servicios estarían en la nube en todo momento, ahorrando la parte del despliegue por la parte del cliente, además que de que, al principio del proyecto, ya se especificó que se tendría que usar Azure.

3

Análisis

En este tercer capítulo, se describirá los distintos requisitos funcionales y no funcionales del software, además de los casos de uso. Hay que tener en cuenta, que tanto los requisitos funcionales como los casos de uso, se dividirán en cuatro partes, debido a la posibilidad de poder ser uno de los cuatro tipos de usuarios, los cuales, dependiendo de su rol, tendrán unas funcionalidades disponibles o no.

Sin embargo, hay que aclarar unos pequeños elementos que se irán repitiendo a lo largo del manual:

- **Vista:** Las vistas son las distintas ventanas o pantallas que puede visitar un usuario en la aplicación web.
- **Perfil principal:** Es la primera vista después de iniciar sesión como usuario. Esta vista contiene los datos básicos como el nombre, apellidos, y NIF del usuario. Sin embargo, ciertos usuarios pueden acceder a los perfiles principales de otros, como lo son los administradores con todos los demás usuarios que no sean administradores, o los profesores con sus alumnos.
- **Docencia:** El termino Docencia se refiere a las asignaturas que imparte un profesor. En otras palabras, es la carga docente del profesor.
- **Matrícula:** El termino Matrícula se refiere a las asignaturas en las cuales los alumnos se matriculan.
- **Auditoría:** Cada vez que un usuario realiza una acción que modifique la base de datos, como crear, borrar, o modificar algún elemento, se crea automáticamente

una auditoría. Una auditoría es un registro de todas esas acciones, indicando la operación realizada, autor de la acción, y la fecha en la que se ha realizado.

- **Pantalla:** El termino Pantalla hace referencia a las vistas, aunque en un sentido más técnico. Se utiliza para referirse a una vista cuando se está hablando de permisos o base de datos.
- **Médico_Temporal:** El término Médico_Temporal se utiliza para indicar que el elemento asignado con dicho médico aún no tiene un médico titular que se encargue de él, es decir, no tiene un médico asignado todavía. Aunque técnicamente sea un usuario, no se podrá iniciar sesión con sus credenciales.

3.1 Requisitos funcionales (Todos los Usuarios)

1. Un usuario podrá iniciar sesión.
2. Un usuario podrá cerrar sesión.
3. Un usuario podrá volver a su perfil principal.
4. Un usuario podrá volver a la vista anterior a la que se encontraba.
5. Un usuario podrá filtrar contenidos en las tablas que tengan filtro de búsqueda.

3.2 Requisitos funcionales (Alumno)

1. Un alumno podrá acceder a la lista de asignaturas en las que esté matriculado.
2. Un alumno podrá acceder a su lista de informes.
3. Un alumno podrá acceder a la información básica de sus informes.
 - a. Un alumno podrá abrir en una nueva ventana el contenido del informe en formato PDF.

3.3 Requisitos funcionales (Profesor)

1. Un profesor podrá acceder a la lista de asignaturas que imparte, o en otras palabras, a sus docencias.
2. Un profesor tendrá acceso a la lista de alumnos que estén matriculados en algunas de las asignaturas que imparta.

3. Un profesor podrá acceder al perfil principal de los alumnos que estén matriculados en algunas de las asignaturas que imparta.
4. Un profesor podrá acceder a la lista de informes que tengan que ver con los alumnos que estén matriculados en algunas de las asignaturas que imparta.
5. Un profesor tendrá acceso a la información básica del informe.
 - a. Un profesor podrá abrir en una nueva ventana el contenido del informe en formato PDF.

3.4 Requisitos funcionales (Médico)

1. Un médico podrá acceder a la lista de informes de los cuales sea médico titular.
2. Un médico podrá acceder a la lista de alumnos formados por aquellos alumnos que tengan al menos uno de sus informes asignado a dicho médico.
3. Un médico podrá acceder al perfil principal de los alumnos descritos en el punto 2, pero no podrá acceder a la información de las matrículas de dichos alumnos.
4. Un médico podrá acceder a la información básica del informe.
 - a. Un médico podrá abrir en una nueva ventana el contenido del informe en formato PDF.
5. Un médico podrá cambiar el contenido de los informes en los que sea médico titular por otro nuevo, siendo este contenido en formato PDF obligatoriamente.

3.5 Requisitos funcionales (Administrador)

1. Un administrador tendrá acceso a la lista de todas las asignaturas, incluyendo listas específicas de los demás usuarios.
2. Un administrador podrá crear nuevas asignaturas.
3. Un administrador podrá acceder a la lista de todos los informes, incluyendo listas específicas que pertenezcan a los demás usuarios.
4. Un administrador podrá crear nuevos informes.
5. Un administrador podrá cambiar el médico titular encargado de un informe. En caso de no haber ninguno disponible, se podrá asignar un médico temporal.
6. Un administrador podrá acceder a la lista de pantallas.
7. Un administrador podrá acceder a la lista de auditorías.

8. Un administrador podrá acceder a la lista de matrículas de todos los alumnos.
9. Un administrador podrá acceder a la lista de docencias de todos los profesores.
10. Un administrador podrá acceder a la lista de todos los usuarios, dividiéndose en listas de alumnos, lista de profesores, y lista de médicos.
11. Un administrador podrá crear nuevos usuarios con roles a elegir entre alumno, profesor o médico.
12. Un administrador podrá crear nuevas matrículas.
13. Un administrador podrá crear nuevas docencias.
14. Un administrador podrá borrar cualquier usuario a excepción de otros administradores.
15. Un administrador podrá borrar matrículas.
16. Un administrador podrá borrar asignaturas.
17. Un administrador podrá borrar docencias.
18. Un administrador podrá borrar informes, a no ser que el alumno posea únicamente dicho informe, en cuyo caso no podrá eliminar el informe.
19. Un administrador podrá modificar el nombre de las asignaturas.
20. Un administrador podrá modificar los campos de los usuarios a excepción de los administradores, entre los cuales se encuentra:
 - a. Nombre.
 - b. Primer apellido.
 - c. Segundo apellido.
 - d. Contraseña.
 - e. Nombre de usuario.

3.6 Requisitos no funcionales

1. El sistema funcionará correctamente en ordenadores cuyo sistema operativo sea Microsoft Windows.
2. El sistema responderá a cualquier petición en menos de 30 segundos.
3. El sistema guardará la sesión de un usuario hasta un máximo de dos horas de inactividad.

3.7 Casos de uso: Definición

Los casos de uso son una técnica utilizada en el análisis y diseño de sistemas de software para capturar los requisitos funcionales de un sistema desde la perspectiva de los usuarios. Un caso de uso describe una interacción específica entre el sistema y los actores (usuarios, sistemas externos...) que interactúan con él.

En este manual, se ha decidido dividir los casos de uso según el tipo de rol, y en el caso del administrador, subdividirlo según sus actividades más importantes debido a la gran cantidad de acciones posibles que puede realizar en la aplicación web.

Para un mejor entendimiento de los elementos de las imágenes, se procede a explicar algunos componentes que podrían resultar confusos a la hora de distinguirlos [16] :

- **<<extend>>**: Las flechas invertidas formadas por líneas discontinuas que contienen la expresión <<extend>>, hacen referencia a una acción que podría suceder o no. Es decir, no siempre ocurre, y normalmente para que se dé el caso, tiene que suceder algún evento en concreto.
A este tipo de flecha se le puede llamar “**Extensión**”.
- **<<include>>**: Las flechas formadas por líneas discontinuas que contienen la expresión <<include>>, hacen referencia a una acción que obligatoriamente va a suceder después de realizar la actividad de donde nace la flecha. Esto suele suceder en las acciones que tengan que ver con borrar elementos.
A este tipo de flecha se le puede llamar “**Inclusión**”.
- **—▶** : Este tipo de flechas llamadas “Generalización” indican algún tipo de herencia. Por ejemplo, más adelante en un caso de uso, podemos ver que de *Acceder Informe* nacen otros dos casos de uso. Estos en cuestión también son informes, pero mientras que uno hace referencia a los informes de los médicos, el otro lo hace a los de los alumnos.

3.8 Casos de uso para todos los usuarios

En este apartado se mostrará las acciones que podrán realizar todos los usuarios independientemente de sus roles y permisos.

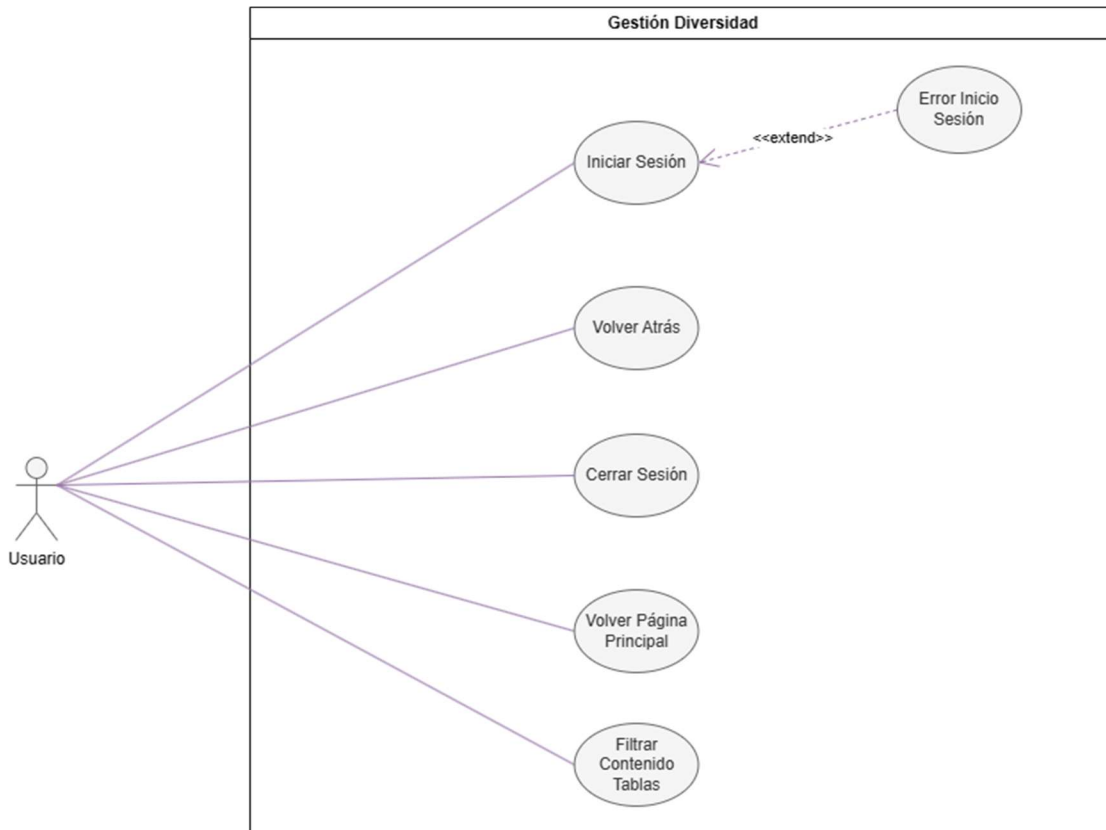


Imagen 1: Casos de uso en común de todos los usuarios.

Identificador	UC-U-1
Nombre	Iniciar Sesión.
Actores	Usuarios.
Precondiciones	La aplicación debe estar ejecutándose, mostrando la pantalla de iniciar sesión.
Postcondiciones	Se accede a la pantalla principal del usuario, o salta un error en la pantalla.
Descripción	El usuario accede a su pantalla principal, la cual tendrá información básica del usuario que haya iniciado sesión.

Escenario principal

1. El usuario rellena el campo *Usuario* con un nombre de usuario.
2. El usuario rellena el campo *Password* con una contraseña.
3. El usuario selecciona el botón *Iniciar Sesión*.
4. El sistema verifica que tanto el nombre de usuario como la contraseña introducidos sean correctas.
5. El sistema redirige al usuario a su perfil principal.

Escenario alternativo

Campos vacíos

1. El usuario deja vacío uno, o los dos campos al iniciar sesión.
2. El usuario selecciona el botón *Iniciar Sesión*.
3. El sistema muestra una notificación indicando que los campos están vacíos.

Usuario o contraseña no válidos:

1. El usuario rellena el campo *Usuario* con un nombre de usuario.
2. El usuario rellena el campo *Password* con una contraseña.
3. El usuario selecciona el botón *Iniciar Sesión*.
4. El sistema detecta que algunos de los campos son incorrectos.
5. El sistema muestra un mensaje por pantalla indicando que alguno de los campos es incorrecto.

Iniciar sesión como *Médico_Temporal*:

1. El usuario rellena los campos *Usuario* y *Password* con las credenciales de un *Médico_Temporal*.
2. El usuario selecciona el botón *Iniciar Sesión*.
3. El sistema detecta que se está iniciando como un *Médico_Temporal*.
4. El sistema muestra un mensaje por pantalla indicando que no se puede iniciar sesión con esas credenciales.

Tabla 1: UC-U-1: Iniciar Sesión.

Identificador	UC-U-2
Nombre	Volver Atrás.
Actores	Usuarios.
Precondiciones	El Usuario debe haber iniciado sesión. El Usuario debe encontrarse en una pantalla que no sea su perfil principal.
Postcondiciones	Se accede a la pantalla anterior la cual el usuario se encontraba antes de estar en la pantalla actual.
Descripción	El usuario vuelve una página atrás de donde se encuentra respectivamente.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario selecciona el botón <i>Volver Atrás</i>. 2. El sistema redirige el usuario a la vista anterior respecto a donde se encontraba en ese momento. 	

Tabla 2: UC-U-2: Volver atrás.

Identificador	UC-U-3
Nombre	Cerrar Sesión.
Actores	Usuarios.
Precondiciones	El usuario debe haber iniciado sesión.
Postcondiciones	El usuario habrá vuelto a la pantalla de inicio de sesión. El sistema habrá vaciado los campos <i>Usuario</i> y <i>Password</i> de la pantalla de inicio de sesión.
Descripción	El usuario vuelve a la pantalla de inicio de sesión.
Escenario principal	

1. El usuario selecciona el botón *Cerrar Sesión* en la barra de navegación.
2. El sistema redirige al usuario a la pantalla de inicio de sesión.

Tabla 3: UC-U-3: Cerrar Sesión.

Identificador	UC-U-4
Nombre	Volver Página Principal.
Actores	Usuarios.
Precondiciones	El usuario debe haber iniciado sesión.
Postcondiciones	El usuario habrá vuelto a su perfil principal.
Descripción	El usuario puede volver a su perfil principal en cualquier momento seleccionando el botón <i>Perfil Principal</i> en la barra de navegación.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario selecciona el botón <i>Perfil Principal</i> en la barra de navegación. 2. El sistema redirige al usuario a su perfil principal. 	

Tabla 4: UC-U-4: Volver Página Principal.

Identificador	UC-U-5
Nombre	Filtrar Contenido Tablas.
Actores	Usuarios.
Precondiciones	El usuario debe haber iniciado sesión. El usuario debe encontrarse en una pantalla que contenga una lista de elementos en una tabla.
Postcondiciones	El sistema filtrará los resultados según los caracteres escritos por el usuario.
Descripción	El usuario puede filtrar los contenidos de las tablas que se

	encuentren en las pantallas que sean listas de algún elemento.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario selecciona caja de texto <i>Buscar</i>. 2. El usuario escribe los caracteres con los cuales quiere filtrar algún dato en la tabla. 3. El sistema comprueba si hay coincidencias. 4. El sistema actualiza la tabla según los criterios anteriormente escritos. 	

Tabla 5: UC-U-5: Filtrar Contenido Tablas.

3.9 Casos de uso de alumnos

En este apartado se mostrará las acciones que podrán realizar todos los alumnos.

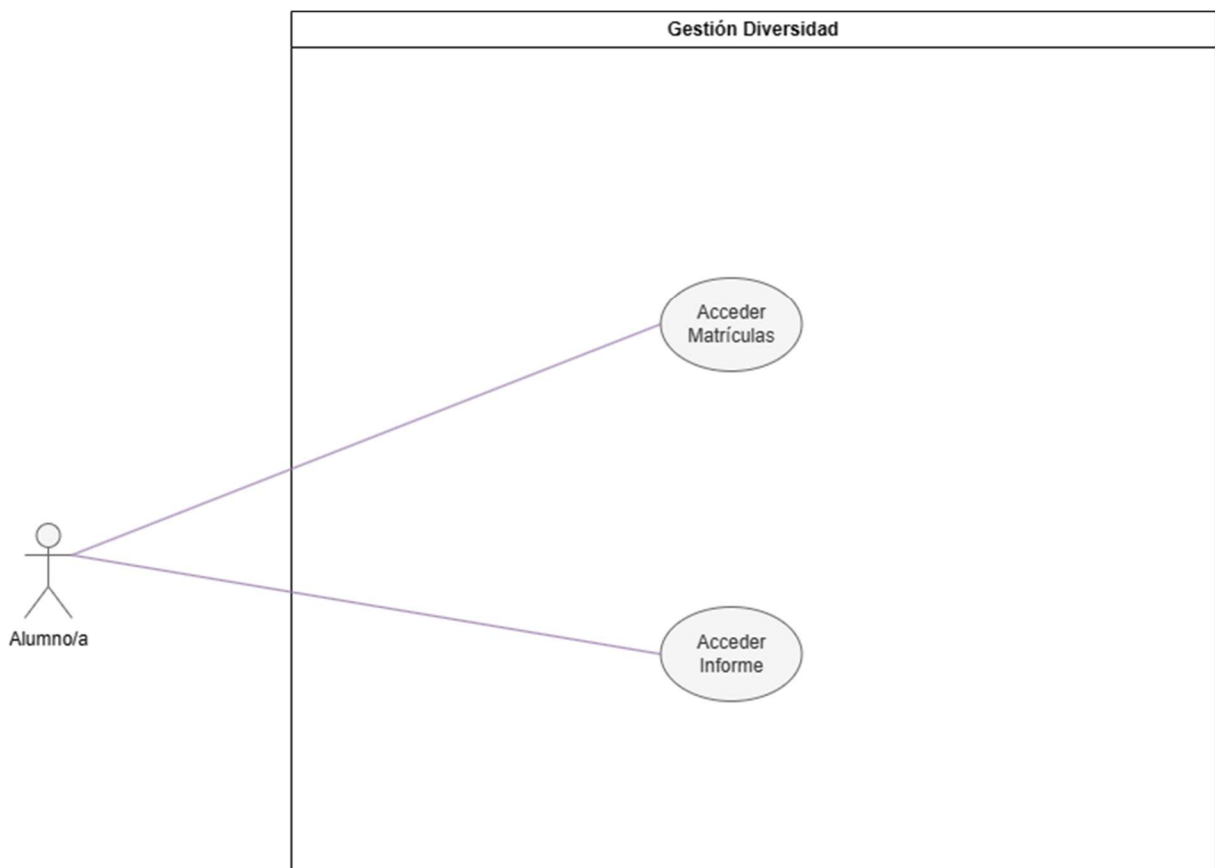


Imagen 2: Casos de uso de los alumnos.

Identificador	UC-A-1
Nombre	Acceder Matrículas.
Actores	Alumnos.
Precondiciones	El alumno debe haber iniciado sesión. El alumno debe encontrarse en su perfil principal.
Postcondiciones	El Sistema redirigirá al alumno a la pantalla de listas de sus matrículas.
Descripción	El alumno puede dirigirse a una pantalla en la cual se encuentren todas las asignaturas que cursa, es decir, en las que esté matriculado.
Escenario principal	
<ol style="list-style-type: none"> 1. El alumno selecciona <i>Asignaturas</i>. 2. El sistema redirige el alumno a una pantalla con la lista de asignaturas en las que está matriculado. 	

Tabla 6: UC-A-1: Acceder Matrículas.

Identificador	UC-A-2
Nombre	Acceder Informe.
Actores	Alumnos.
Precondiciones	El alumno debe haber iniciado sesión. El alumno debe encontrarse en la pantalla de listas de informes.
Postcondiciones	El Sistema redirigirá al alumno a la pantalla principal del informe, además de abrir en otra pestaña el contenido del informe en PDF.
Descripción	El alumno puede acceder a la información básica de sus informes, a la vez que ver el contenido de ellos.

Escenario principal

1. El alumno selecciona *Detalles*.
2. El sistema redirige el alumno a la pantalla principal del informe.
3. El alumno selecciona *Ver informe*.
4. El sistema muestra el contenido del informe con formato PDF en una nueva pestaña.

Tabla 7: UC-A-2: Acceder Informe.

3.10 Casos de uso de profesores

En este apartado se mostrará las acciones que podrán realizar todos los profesores.

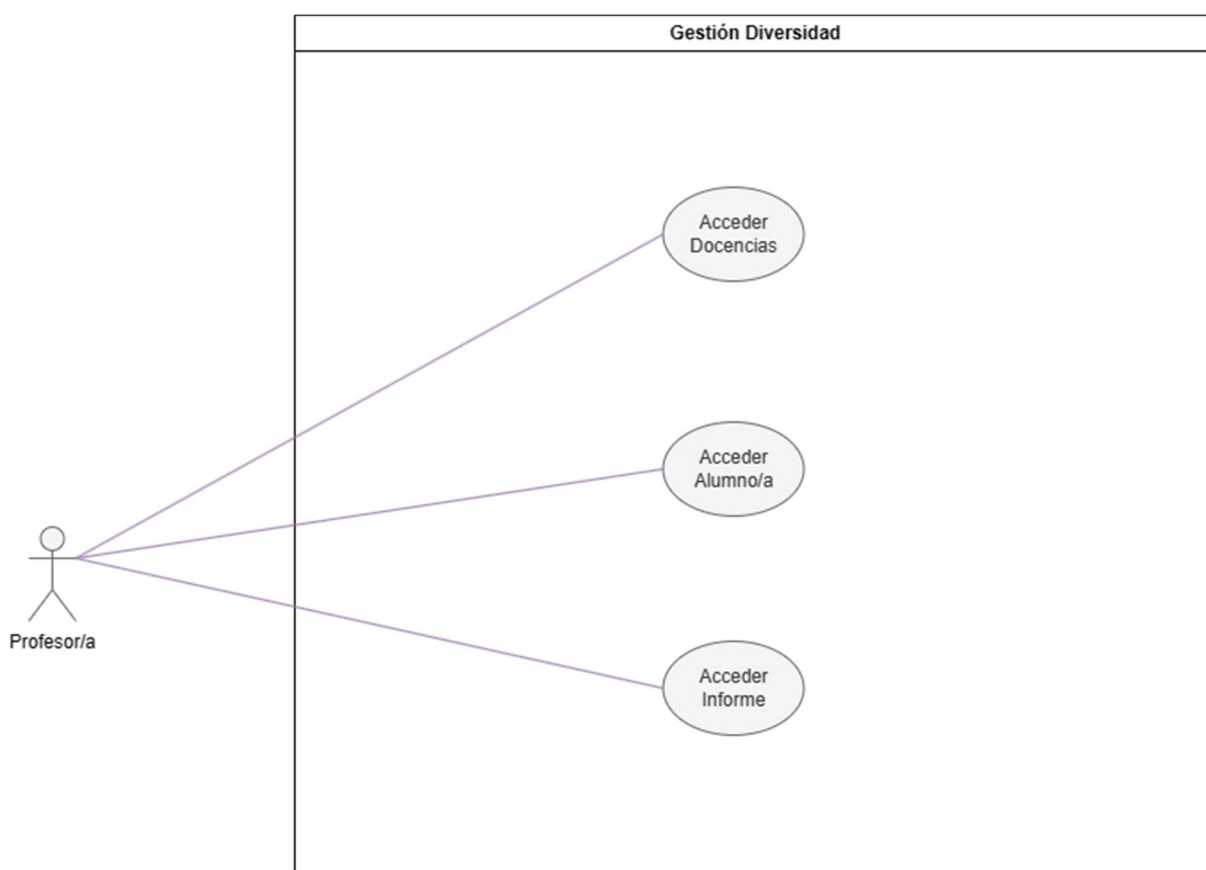


Imagen 3: Casos de uso de los profesores.

Identificador	UC-P-1
Nombre	Acceder Docencias.

Actores	Profesores.
Precondiciones	El profesor debe haber iniciado sesión. El profesor debe encontrarse en su perfil principal.
Postcondiciones	El Sistema redirigirá al profesor a la pantalla de listas de sus docencias.
Descripción	El profesor puede dirigirse a una pantalla en la cual se encuentren todas las docencias que cursa, es decir, las asignaturas que imparte.
Escenario principal	
<ol style="list-style-type: none"> 1. El profesor selecciona <i>Asignaturas</i>. 2. El sistema redirige el profesor a una pantalla con la lista de asignaturas que está impartiendo. 	

Tabla 8: UC-P-1: Acceder Docencias.

Identificador	UC-P-2
Nombre	Acceder Alumnos.
Actores	Profesores.
Precondiciones	El profesor debe haber iniciado sesión. El profesor debe encontrarse en la pantalla de lista de alumnos.
Postcondiciones	El Sistema redirigirá al profesor al perfil principal de uno de sus alumnos.
Descripción	El profesor puede dirigirse a una pantalla en la cual se encuentren todos los alumnos que estén matriculados en alguna asignatura que imparta el profesor.
Escenario principal	

1. El profesor selecciona *Alumnos*.
2. El sistema redirige el profesor a una pantalla con la lista de alumnos matriculados en alguna asignatura que imparta el profesor.
3. El profesor selecciona *Detalles* de un alumno.
4. El sistema redirige al profesor al perfil principal del alumno.

Tabla 9: UC-P-2: Acceder Alumnos.

Identificador	UC-P-3
Nombre	Acceder Informe.
Actores	Profesores.
Precondiciones	El profesor debe encontrarse a en el perfil principal de un alumno.
Postcondiciones	El sistema redirigirá al profesor a la pantalla principal del informe, además de abrir en otra pestaña el contenido del informe en PDF.
Descripción	El profesor puede acceder a la información básica de los informes de sus alumnos, a la vez que ver el contenido de ellos.
Escenario principal	
<ol style="list-style-type: none"> 1. El profesor selecciona <i>Informes</i>. 2. El sistema redirige al profesor a la pantalla de listas de informes. 3. El profesor selecciona <i>Detalles</i>. 4. El sistema redirige al profesor a la página principal del informe. 5. El profesor selecciona <i>Ver informe</i>. 6. El sistema muestra el contenido del informe con formato PDF en una nueva pestaña del navegador. 	

Tabla 10: UC-P-3: Acceder Informe.

3.11 Casos de uso de médicos

En este apartado se mostrará las acciones que podrán realizar todos los médicos.

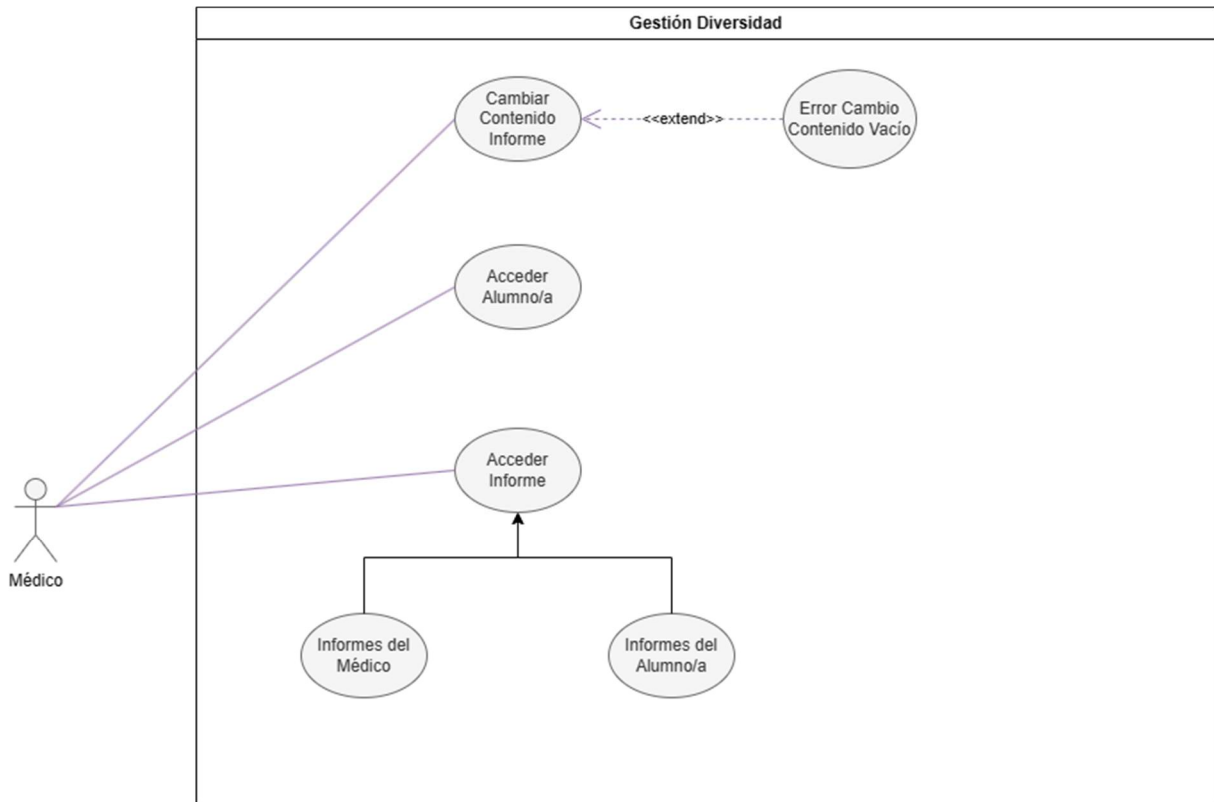


Imagen 4: Casos de uso de los médicos.

Identificador	UC-M-1
Nombre	Acceder Alumno.
Actores	Médicos.
Precondiciones	El médico debe haber iniciado sesión.
Postcondiciones	El sistema redirigirá al médico al perfil principal del alumno.
Descripción	El médico puede acceder al perfil principal de los alumnos que tengan al menos un informe cuyo médico titular sea dicho médico. Sin embargo, no tendrá acceso a las asignaturas que cursa el alumno, es decir, a sus matrículas.

Escenario principal
<ol style="list-style-type: none"> 1. El médico selecciona <i>Alumnos</i>. 2. El sistema redirige al médico a la pantalla de listas de alumnos. 3. El médico selecciona <i>Detalles</i> de un alumno. 4. El sistema redirige al médico al perfil principal del alumno.

Tabla 11: UC-M-1: Acceder Alumno.

Identificador	UC-M-2
Nombre	Acceder Informe.
Actores	Médicos.
Precondiciones	<p>El médico debe haber iniciado sesión.</p> <p>El médico debe encontrarse en una pantalla de listas de informes.</p>
Postcondiciones	El Sistema redirigirá al médico a la pantalla principal del informe.
Descripción	El médico puede acceder a la página principal de los informes. Las pantallas de listas de informes pueden variar, siendo la primera la lista de todos los informes de los cuáles es titular, y la segunda siendo la lista de informes de un alumno en concreto.
Escenario principal	
<ol style="list-style-type: none"> 1. El médico selecciona <i>Informes</i>. 2. El sistema redirige al médico a la pantalla de listas de informes. 3. El médico selecciona <i>Detalles</i> de un informe. 4. El sistema redirige al médico a la página principal del informe. 5. El médico selecciona <i>Ver informe</i>. 6. El sistema muestra el contenido del informe con formato PDF en una nueva pestaña del navegador. 	

Tabla 12: UC-M-2: Acceder Informe.

Identificador	UC-M-3
Nombre	Cambiar Contenido Informe.
Actores	Médicos.
Precondiciones	El médico debe haber iniciado sesión. El médico debe encontrarse en la página principal del informe. El médico debe ser titular del informe.
Postcondiciones	El sistema habrá cambiado el contenido del informe.
Descripción	El médico puede cambiar el contenido de los informes, pero solo de aquellos en los que sea médico titular.
Escenario principal	
<ol style="list-style-type: none"> 1. EL médico selecciona el botón <i>Elegir archivo</i>. 2. El médico selecciona el nuevo contenido en formato PDF. 3. El médico selecciona el botón <i>abrir</i>. 4. El médico selecciona el botón <i>Modificar archivo</i>. 5. El sistema cambia el contenido anterior del informe por el nuevo proporcionado por el médico. 6. El sistema guarda la acción creando una nueva auditoría. 7. El sistema redirige al médico a la página principal del informe con el contenido actualizado. 	
Escenario alternativo	
Contenido nuevo vacío:	
<ol style="list-style-type: none"> 1. El usuario deja vacío el nuevo contenido a cambiar. 2. El usuario selecciona el botón <i>Modificar archivo</i>. 3. El sistema muestra una notificación indicando que el contenido está vacío. 	

Tabla 13: UC-M-3: Cambiar Contenido Informe.

3.12 Casos de uso de administradores (Navegación)

En este apartado se mostrará las acciones que podrán realizar todos los administradores, en específico, las acciones que tengan que ver con su navegación.

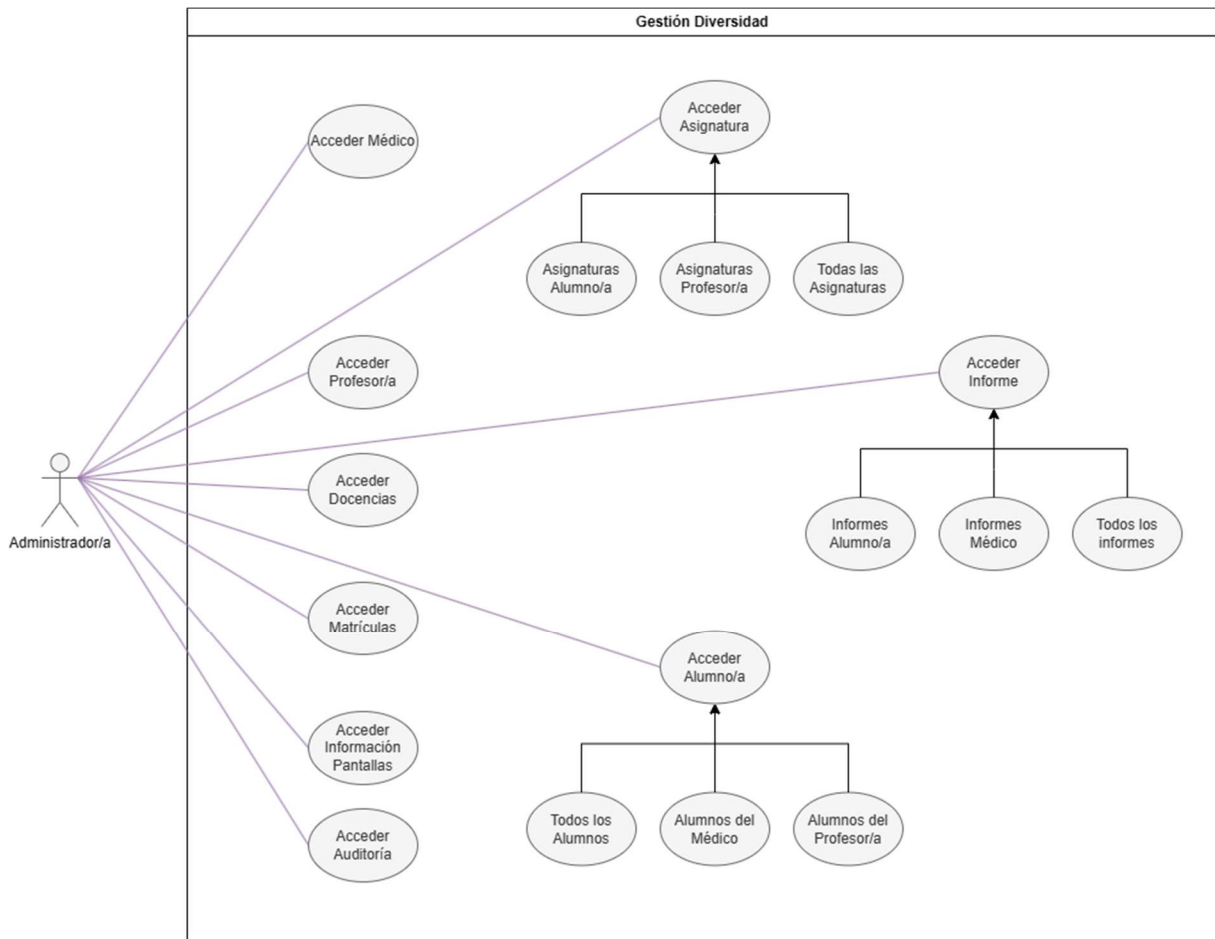


Imagen 5: Casos de uso de la navegación de los administradores.

Identificador	UC-AD-N-1
Nombre	Acceder Médico.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión.
Postcondiciones	El Sistema redirigirá al administrador al perfil principal del médico.
Descripción	El administrador puede acceder a la página principal de

	todos los médicos.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona <i>Médicos</i>. 2. El sistema redirige el administrador a la pantalla de listas de médicos. 3. El administrador selecciona <i>Detalles</i> de un médico. 4. El sistema redirige el administrador al perfil principal del médico. 	

Tabla 14: UC-AD-N-1: Acceder Médico.

Identificador	UC-AD-N-2
Nombre	Acceder Profesor.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión.
Postcondiciones	El Sistema redirigirá el administrador a la pantalla principal del profesor.
Descripción	El administrador puede acceder a la página principal de todos los profesores.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona <i>Profesores</i>. 2. El sistema redirige el administrador a la pantalla de listas de profesores. 3. El administrador selecciona <i>Detalles</i> de un profesor. 4. El sistema redirige el administrador al perfil principal del profesor. 	

Tabla 15: UC-AD-N-2: Acceder Profesor.

Identificador	UC-AD-N-3
Nombre	Acceder Docencias.

Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión.
Postcondiciones	El sistema redirigirá al administrador a la pantalla de listas de las docencias.
Descripción	El administrador puede acceder a la pantalla donde se encuentra la lista de todas las docencias.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona <i>Docencias</i>. 2. El sistema redirige el administrador a la pantalla de listas de docencias. 	

Tabla 16: UC-AD-N-3: Acceder Docencias.

Identificador	UC-AD-N-4
Nombre	Acceder Matrículas.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión.
Postcondiciones	El Sistema redirigirá al administrador a la pantalla de listas de las matrículas.
Descripción	El administrador puede acceder a la pantalla donde se encuentra la lista de todas las matrículas.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona <i>Matrículas</i>. 2. El sistema redirige el administrador a la pantalla de listas de matrículas. 	

Tabla 17: UC-AD-N-4: Acceder Matrículas.

Identificador	UC-AD-N-5
Nombre	Acceder Información Pantallas.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión.
Postcondiciones	El Sistema redirigirá al administrador a la pantalla de listas de las pantallas.
Descripción	El administrador puede acceder a la pantalla donde se encuentra la lista de las explicaciones de todas las pantallas.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona <i>Pantallas</i>. 2. El sistema redirige el administrador a la pantalla de listas de pantallas. 	

Tabla 18: UC-AD-N-5: Acceder Información Pantallas.

Identificador	UC-AD-N-6
Nombre	Acceder Auditoría.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión.
Postcondiciones	El sistema redirigirá al administrador a la pantalla de listas de auditorías.
Descripción	El administrador puede acceder a la pantalla donde se encuentra la lista de auditorías.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona <i>Auditorías</i>. 2. El sistema redirige el administrador a la pantalla de listas de auditorías. 	

Tabla 19: UC-AD-N-6: Acceder Auditoría.

Identificador	UC-AD-N-7
Nombre	Acceder Asignaturas.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en una pantalla de listas de asignaturas. Estas listas pueden ser tres: todas las asignaturas, las asignaturas en las que un alumno esté matriculado, o las asignaturas que un profesor imparte.
Postcondiciones	El Sistema redirigirá el administrador a una pantalla de listas de asignaturas.
Descripción	El administrador puede acceder a la página donde se encuentran las listas de asignaturas.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona el botón <i>Asignaturas</i>. 2. El sistema redirige el administrador a la página de las listas de asignaturas. 	

Tabla 20: UC-AD-N-7: Acceder Asignaturas.

Identificador	UC-AD-N-8
Nombre	Acceder Informe.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en una pantalla de listas de informes. Estas listas pueden ser tres: todas las listas de informes, los informes específicos de un alumno,

	o los informes de un médico titular.
Postcondiciones	El Sistema redirigirá el administrador a la página principal del informe.
Descripción	El administrador puede acceder a la página principal de cualquier informe.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona <i>Detalles</i>. 2. El sistema redirige el administrador a la página principal del informe. 3. El administrador selecciona <i>Ver informe</i>. 4. El sistema muestra el contenido del informe en formato PDF en una nueva pestaña. 	

Tabla 21: UC-AD-N-8: Acceder Informe.

Identificador	UC-AD-N-9
Nombre	Acceder Alumno.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en una pantalla de listas de alumnos. Estas listas pueden ser tres: todas las listas de alumnos, los alumnos específicos de un profesor; o los alumnos específicos de un médico titular.
Postcondiciones	El Sistema redirigirá el administrador al perfil principal del alumno.
Descripción	El administrador puede acceder al perfil principal de cualquier alumno.
Escenario principal	

1. El administrador selecciona *Detalles*.
2. El sistema redirige el administrador al perfil principal del alumno.

Tabla 22: UC-AD-N-9: Acceder Alumno.

3.13 Casos de uso de administradores (Creación)

En este apartado se mostrará las acciones que podrán realizar todos los administradores, en específico, las acciones que tengan que ver con la creación de elementos.

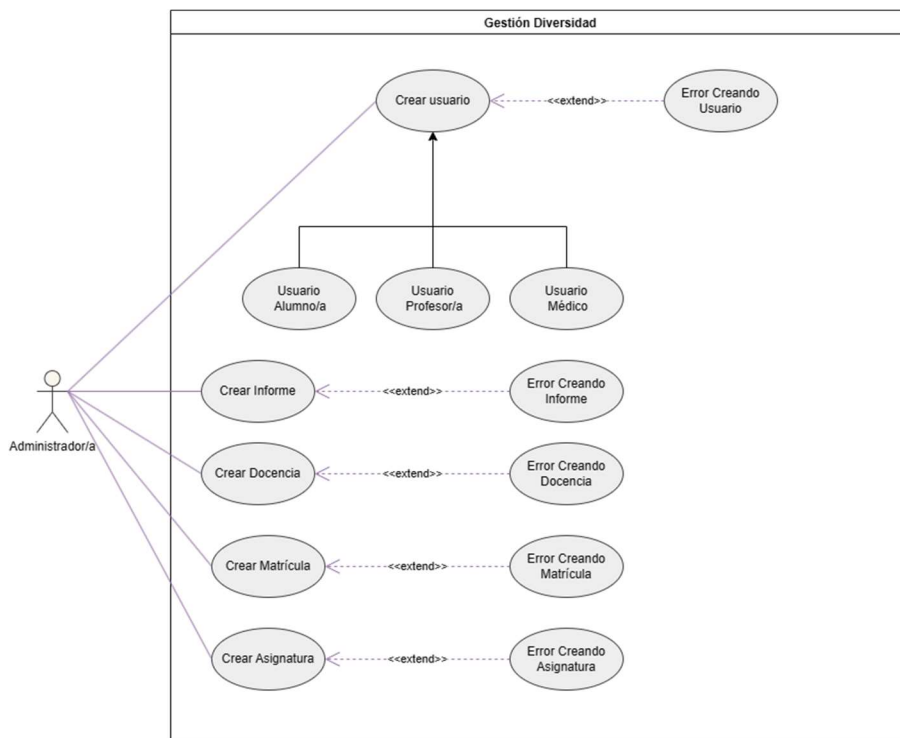


Imagen 6: Casos de uso de la creación de elementos de los administradores.

Identificador	UC-AD-C-1
Nombre	Crear usuario.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en una pantalla de lista de usuarios.

Postcondiciones	El sistema crea y almacena un nuevo usuario en la base de datos.
Descripción	<p>El administrador puede crear nuevos usuarios.</p> <p>Dependiendo de la lista en la que se encuentre, podrá crear un tipo de usuario u otro: Si se encuentra en una lista de alumnos, podrá crear un alumno, si se encuentra en una lista de profesores, podrá crear un profesor, y si se encuentra en una lista de médicos, podrá crear un médico.</p>
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario selecciona el botón <i>Crear (tipo usuario)</i>. 2. El sistema redirige al administrador a la pantalla de creación del nuevo usuario. 3. El administrador completa todos los campos necesarios para crear el usuario. 4. El administrador seleccionar el botón <i>Crear (tipo usuario)</i>. 5. El sistema confirmar los campos y crea correctamente el nuevo usuario. 6. El sistema guarda la acción creando una nueva auditoría. 7. El sistema redirige al administrador a la lista de los usuarios donde se encontraba anteriormente. 	
Escenario alternativo	
<p>Campos vacíos:</p> <ol style="list-style-type: none"> 3. El administrador deja vacío alguno de los campos necesarios para crear un usuario 4. El administrador pulsa el botón <i>Crear (tipo usuario)</i>. 5. El sistema detecta que hay algún campo vacío. 6. El sistema redirige al administrador a la página de creación de usuarios, mostrando por pantalla un mensaje indicando una falta de los campos necesarios para la creación del usuario. <p>Usuario no válido:</p>	

3. El administrador rellena el campo *Usuario* con un nombre de usuario ya existente.
4. El administrador pulsa el botón *Crear (tipo de usuario)*.
5. El sistema detecta que el nombre del campo *Usuario* ya está en uso.
6. El sistema redirige al administrador a la pantalla de creación de usuarios, mostrando por pantalla una notificación informando sobre el uso del nombre del usuario.

Contraseñas no coincidentes:

3. El administrador rellena el campo *Password* y *Confirmación Password*.
4. El administrador pulsa el botón *Crear (tipo de usuario)*.
5. El sistema detecta que el campo *Password* y *Confirmación Password* no coinciden.
6. El sistema redirige el administrador a la pantalla de creación de usuario, mostrando por pantalla una notificación informando sobre la diferencia entre dichos campos.

Tabla 23: UC-AD-C-1: Crear Usuario.

Identificador	UC-AD-C-2
Nombre	Crear Informe.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en una pantalla de lista de informes.
Postcondiciones	El sistema crea y almacena un nuevo informe en la base de datos.
Descripción	El administrador puede crear nuevos informes.
Escenario principal	
1. El administrador selecciona el botón <i>Crear nuevo Informe</i> .	

<ol style="list-style-type: none"> 2. El sistema redirige el administrador a la pantalla de creación de informes. 3. El administrador rellene los campos necesarios para crear el informe. 4. El administrador pulsa el botón <i>Crear informe</i>. 5. El sistema confirma los datos y guarda el nuevo informe en la base de datos. 6. El sistema guarda la acción creando una nueva auditoría. 7. El sistema redirige el administrador a la pantalla de listas de informes anterior a la creación del informe.
Escenario alternativo
<p>Campos vacíos:</p> <ol style="list-style-type: none"> 3. El administrador deja vacío alguno de los campos. 4. El administrador pulsa el botón <i>Crear informe</i>. 5. El sistema detecta que algún campo está vacío. 6. El sistema muestra por pantalla una notificación informando sobre los campos que están vacíos. <p>Alumno o médico incorrectos:</p> <ol style="list-style-type: none"> 3. El administrador intenta insertar un médico o alumno que no está en la base de datos. 4. El sistema detecta que alguno de los datos es erróneo. 5. El sistema muestra por pantalla un mensaje informando sobre el error.

Tabla 24: UC-AD-C-2: Crear Informe.

Identificador	UC-AD-C-3
Nombre	Crear Docencia.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en la pantalla de lista de docencias.

Postcondiciones	El sistema crea y almacena una nueva docencia en la base de datos.
Descripción	El administrador puede crear nuevas docencias que aún no existan.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona el botón <i>Crear nueva Docencia</i>. 2. El sistema redirige el administrador a la pantalla de creación de docencias. 3. El administrador rellena los campos necesarios para crear una docencia. 4. El administrador pulsa el botón <i>Crear docencia</i>. 5. El sistema confirma los datos y guarda la nueva docencia en la base de datos. 6. El sistema guarda la acción creando una nueva auditoría. 7. El sistema redirige el administrador a la pantalla de listas de docencias anterior a la creación de la nueva docencia. 	
Escenario alternativo	
<p>Campos vacíos:</p> <ol style="list-style-type: none"> 3. El administrador deja vacío alguno de los campos. 4. El administrador pulsa el botón <i>Crear docencia</i>. 5. El sistema detecta que algún campo está vacío. 6. El sistema muestra por pantalla una notificación informando sobre los campos que están vacíos. <p>Existe Docencia:</p> <ol style="list-style-type: none"> 3. El administrador rellena los campos con una combinación de profesor y asignatura que ya existe. 4. El administrador pulsa el botón <i>Crear docencia</i>. 5. El sistema detecta que ya existe esa docencia. 6. EL sistema muestra por pantalla una notificación informando sobre la 	

existencia de esa combinación.

Asignatura o profesor incorrectos:

3. El administrador inserta una asignatura o un profesor que no se encuentra en la base de datos.
4. El sistema detecta que alguno de los datos es erróneo.
5. El sistema muestra por pantalla un mensaje informando sobre el error.

Tabla 25: UC-AD-C-3: Crear Docencia.

Identificador	UC-AD-C-4
Nombre	Crear Matrícula.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en la pantalla de lista de matrículas.
Postcondiciones	El sistema crea y almacena una nueva matrícula en la base de datos.
Descripción	El administrador puede crear nuevas matrículas que aún no existan.
Escenario principal	
<ol style="list-style-type: none">1. El administrador selecciona el botón <i>Crear Nueva Matrícula</i>.2. El sistema redirige el administrador a la pantalla de creación de matrículas.3. El administrador rellena los campos necesarios para crear una matrícula.4. El administrador pulsa el botón <i>Crear matrícula</i>.5. El sistema confirma los datos y guarda la nueva matrícula en la base de datos.6. El sistema guarda la acción creando una nueva auditoría.	

7. El sistema redirige el administrador a la lista de matrículas anterior a la creación de la nueva matrícula.
Escenario alternativo
<p>Campos vacíos:</p> <ol style="list-style-type: none"> 3. El administrador pulsa el botón <i>Crea matrícula</i>. 4. El sistema detecta que algún campo está vacío. 5. El sistema muestra por pantalla una notificación informando sobre los campos que están vacíos. <p>Existe Matrícula:</p> <ol style="list-style-type: none"> 3. El administrador rellena los campos con una combinación de alumno y asignaturas que ya existe. 4. El administrador pulsa el botón <i>Crear matrícula</i>. 5. El sistema detecta si ya existe esa matrícula. 6. El sistema muestra por pantalla una notificación informando sobre la existencia de esa matrícula. <p>Asignatura o alumno erróneos:</p> <ol style="list-style-type: none"> 3. El administrador inserta un alumno o una asignatura que no está en la base de datos. 4. El sistema detecta que alguno de los datos es erróneo. 5. El sistema muestra un mensaje por pantalla informando del error.

Tabla 26: UC-AD-C-4: Crear Matrícula.

Identificador	UC-AD-C-5
Nombre	Crear Asignatura.
Actores	Usuarios.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en alguna pantalla de lista de asignaturas.

Postcondiciones	El sistema crea y almacena una nueva asignatura en la base de datos.
Descripción	El administrador puede crear nuevas asignaturas que aún no existan.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona el botón <i>Crear Nueva Asignatura</i>. 2. El sistema redirige el administrador a la pantalla de creación de asignaturas. 3. El administrador rellena los campos necesarios para crear una asignatura. 4. El administrador pulsa el botón <i>Crear asignatura</i>. 5. El sistema confirma los datos y guarda la nueva asignatura en la base de datos. 6. El sistema guarda la acción creando una nueva auditoría. 7. El sistema redirige el administrador a la lista de asignaturas anterior a la creación de la nueva asignatura. 	
Escenario alternativo	
<p>Campos vacíos:</p> <ol style="list-style-type: none"> 3. El administrador pulsa el botón <i>Crear asignatura</i>. 4. El sistema detecta que algún campo está vacío. 5. El sistema muestra por pantalla una notificación informando sobre los campos que están vacíos. <p>Existe Asignatura:</p> <ol style="list-style-type: none"> 3. El administrador rellena el campo <i>Nombre</i> con un nombre que ya está asignado a una asignatura. 4. El administrador pulsa el botón <i>Crear asignatura</i>. 5. El sistema detecta que ya hay una asignatura con ese nombre. 6. El sistema muestra por pantalla una notificación sobre el uso de ese nombre. 	

Tabla 27: UC-AD-C-5: Crear Asignatura.

3.14 Casos de uso de administradores (Borrado)

En este apartado se mostrará las acciones que podrán realizar todos los administradores, en específico, las acciones que tengan que ver con la eliminación de elementos.

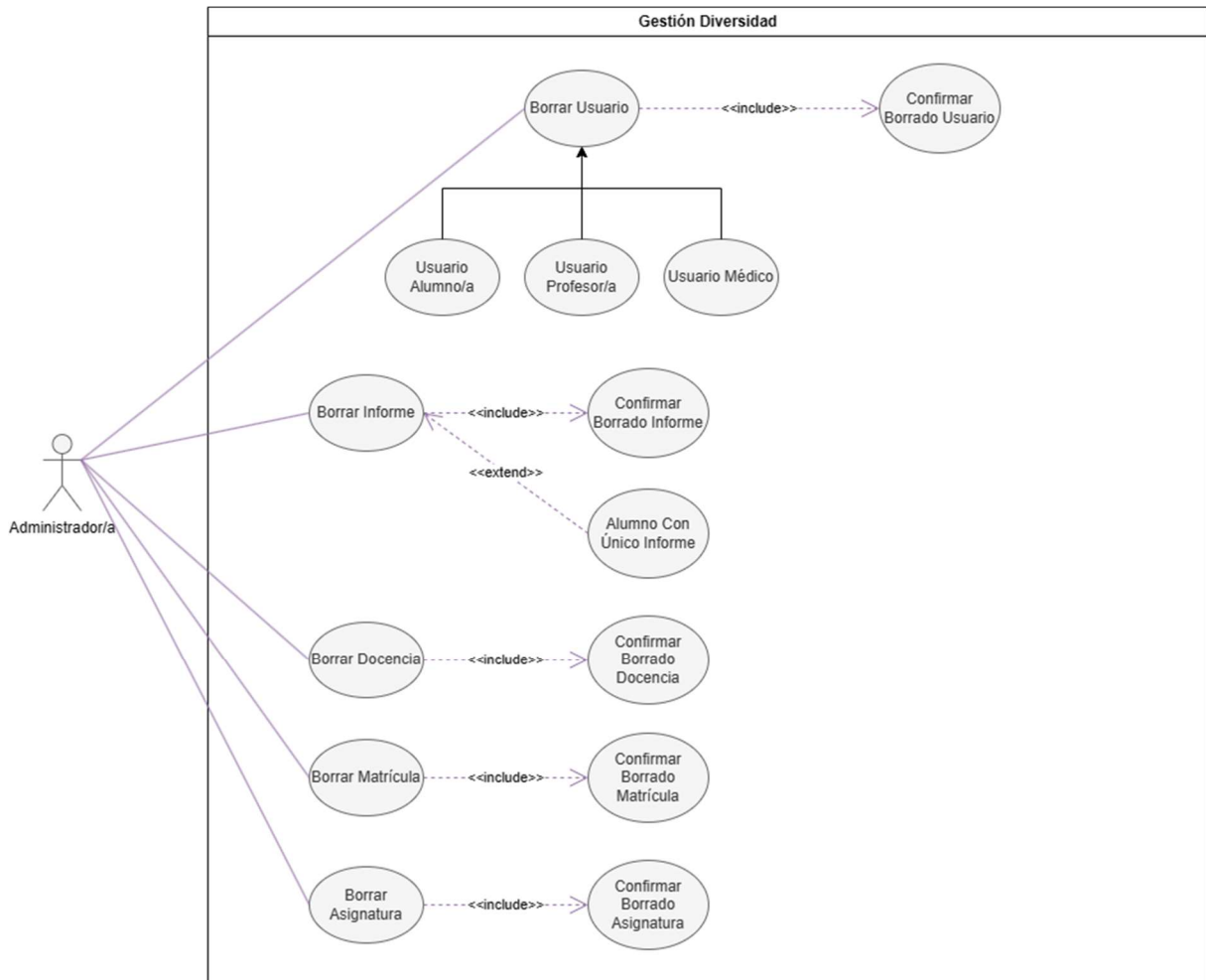


Imagen 7: Casos de uso de la eliminación de elementos de los administradores.

Identificador	UC-AD-B-1
Nombre	Borrar Usuario.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión.

	El administrador debe encontrarse en una pantalla de lista de usuarios.
Postcondiciones	El sistema borra un usuario de la base de datos.
Descripción	El administrador puede borrar usuarios. Dependiendo de la lista en la que se encuentre, podrá borrar un tipo de usuario u otro. Si se encuentra en una lista de alumnos, podrá borrar un alumno, si se encuentra en una lista de profesores, podrá borrar un profesor, y si se encuentra en una lista de médicos, podrá borrar un médico.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario selecciona el botón <i>Borrar</i>. 2. El sistema redirige al administrador a la vista de confirmación de borrado. 3. El administrador selecciona el botón <i>Borrar (tipo de usuario)</i>. 4. El sistema borra el usuario, junto a todos los elementos que estén enlazados con este. En el caso de los informes cuyo titular sea un médico, se les asignará un médico temporal. 5. El sistema guarda la acción creando una nueva auditoría. 6. El sistema redirige al administrador a la lista de los usuarios donde se encontraba antes. 	

Tabla 28: UC-AD-B-1: Borrar Usuario.

Identificador	UC-AD-B-2
Nombre	Borrar Informe.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en alguna pantalla de lista de informes.
Postcondiciones	El sistema borra un nuevo informe, o un alumno solo tiene

	un informe y lo muestra por pantalla.
Descripción	El administrador puede borrar cualquier informe, a no ser que un alumno solo tenga un solo informe, en tal caso no podría hacerlo, ya que un alumno debe tener mínimo un informe para estar en la base de datos.
Escenario principal	
	<ol style="list-style-type: none"> 1. El usuario selecciona el botón <i>Borrar</i>. 2. El sistema redirige al administrador a la vista de confirmación de borrado. 3. El administrador selecciona el botón <i>Borrar Informe</i>. 4. El sistema borra el informe. 5. El sistema guarda la acción creando una nueva auditoría. 6. El sistema redirige al administrador a la lista de los informes donde se encontraba antes.
Escenario alternativo	
	<p>Alumnos con un solo informe:</p> <ol style="list-style-type: none"> 1. El usuario selecciona el botón <i>Borrar</i>. 2. El sistema detecta que ese alumno solo tiene un informe asignado. 3. El sistema muestra por una notificación por pantalla informando sobre la situación del alumno.

Tabla 29: UC-AD-B-2: Borrar Informe.

Identificador	UC-AD-B-3
Nombre	Borrar Docencia.
Actores	Administradores.
Precondiciones	<p>El administrador debe haber iniciado sesión.</p> <p>El administrador debe encontrarse en la pantalla de lista de docencias.</p>

Postcondiciones	El sistema borra una docencia de la base de datos.
Descripción	El administrador puede borrar docencias.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario selecciona el botón <i>Borrar</i>. 2. El sistema redirige al administrador a la vista de confirmación de borrado. 3. El administrador selecciona el botón <i>Borrar docencia</i>. 4. El sistema borra la docencia. 5. El sistema guarda la acción creando una nueva auditoría. 6. El sistema redirige al administrador a la lista de docencias donde se encontraba antes. 	

Tabla 30: UC-AD-B-3: Borrar Docencia.

Identificador	UC-AD-B-4
Nombre	Borrar Matrícula.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en la pantalla de lista de matrículas.
Postcondiciones	El sistema borra una matrícula de la base de datos.
Descripción	El administrador puede borrar matrículas.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario selecciona el botón <i>Borrar</i>. 2. El sistema redirige al administrador a la pantalla de confirmación de borrado. 3. El administrador selecciona el botón <i>Borrar matrícula</i>. 4. El sistema borra la matrícula. 5. El sistema guarda la acción creando una nueva auditoría. 	

6. El sistema redirige al administrador a la lista de matrículas donde se encontraba antes.

Tabla 31: UC-AD-B-4: Borrar Matrícula.

Identificador	UC-AD-B-5
Nombre	Borrar Asignatura.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en alguna pantalla de lista de asignaturas.
Postcondiciones	El sistema borra una asignatura de la base de datos.
Descripción	El administrador puede borrar asignaturas.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario selecciona el botón <i>Borrar</i>. 2. El sistema redirige al administrador a la pantalla de confirmación de borrado. 3. El administrador selecciona el botón <i>Borrar Asignatura</i>. 4. El sistema borra la asignatura junto con todos sus elementos relacionados. En este caso, matrículas y docencias que contengan dicha asignatura. 5. El sistema guarda la acción creando una nueva auditoría. 6. El sistema redirige al administrador a la lista de asignaturas donde se encontraba antes. 	

Tabla 32: UC-AD-B-5: Borrar Asignatura.

3.15 Casos de uso de administradores (Modificación)

En este apartado se mostrará las acciones que podrán realizar todos los administradores, en específico, las acciones que tengan que ver con la modificación de elementos.

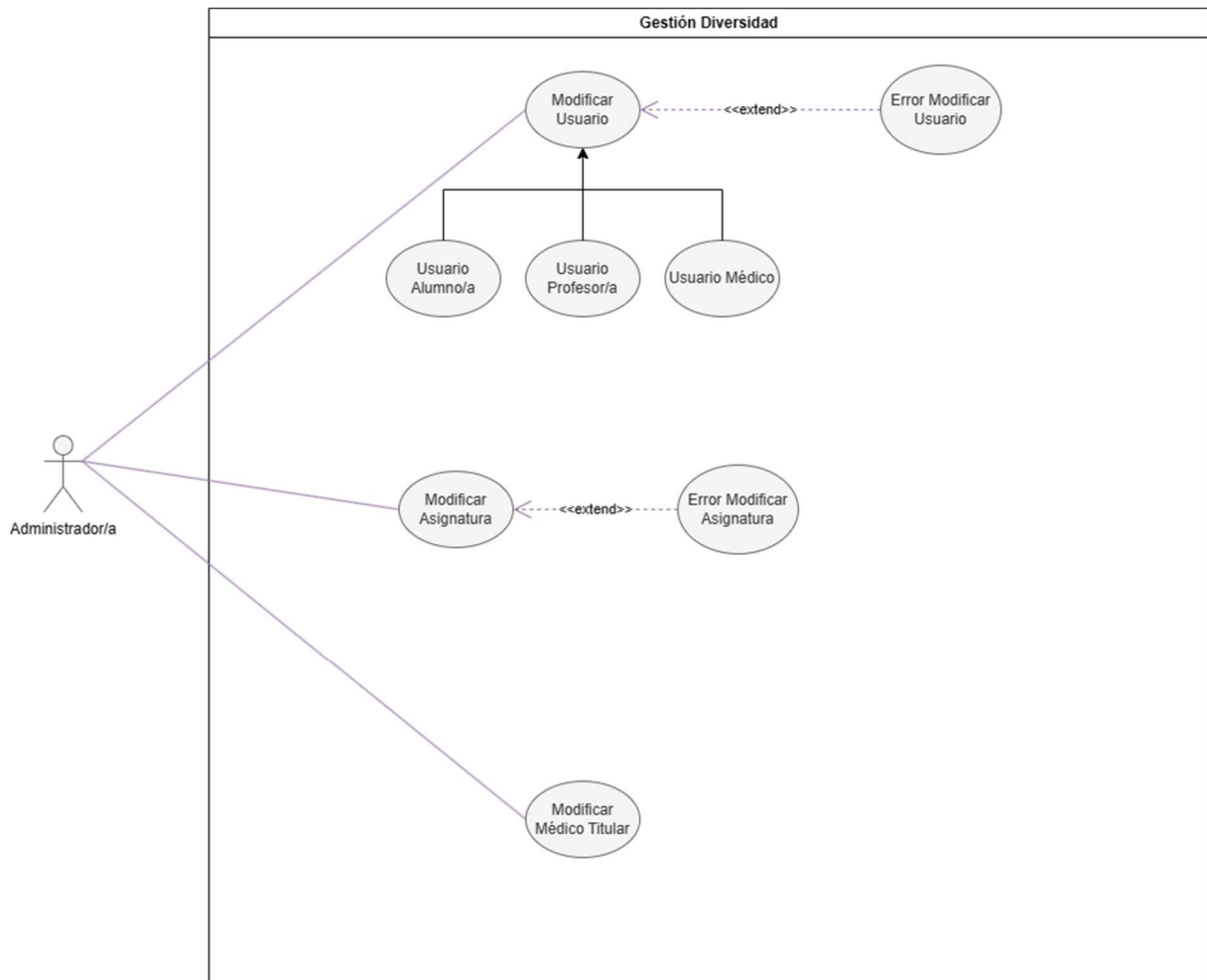


Imagen 8: Casos de uso de la modificación de elementos de los administradores.

Identificador	UC-AD-M-1
Nombre	Modificar Usuario.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en el perfil principal de un usuario.
Postcondiciones	El sistema actualiza los campos del usuario modificado.
Descripción	El administrador puede modificar los campos de cualquier usuario que sea alumno, profesor o médico.

Escenario principal

1. El administrador selecciona el botón *Modificar (tipo de usuario)*.
2. El sistema redirige el administrador a la pantalla de modificación del usuario.
3. El administrador modifica los campos del usuario.
4. El administrador pulsa el botón *Modificar*.
5. El sistema actualiza los campos del usuario.
6. El sistema guarda la acción creando una nueva auditoría.
7. El sistema redirige el administrador a la página principal del usuario modificado.

Escenario alternativo

Campo vacío:

3. El administrador deja vacío un campo.
4. El sistema detecta que hay un campo vacío.
5. El sistema muestra por pantalla una notificación indicando el campo que está vacío.

Nombre de usuario en uso:

3. El administrador modificar el campo *Usuario* con el nombre de un usuario que ya existe.
4. El sistema reconoce el nombre del usuario.
5. El sistema muestra una notificación por pantalla informando sobre como el nombre escrito en el campo *Usuario* ya está en uso.

Contraseñas no coinciden:

3. El administrador modifica los campos *Password* y *Confirmar Password* de forma que no coincidan.
4. El sistema detecta que los dos campos no coinciden.
5. El sistema muestra por pantalla una notificación informando sobre como los dos campos no son iguales.

Tabla 32: UC-AD-M-1: Modificar Usuario.

Identificador	UC-AD-M-2
Nombre	Cambiar Médico Informe.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en una pantalla de listas de informes.
Postcondiciones	El sistema cambia el médico titular de un informe.
Descripción	El administrador puede modificar el médico titular de un informe, incluyendo la asignación de Médico_Temporal.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona el botón <i>Cambiar médico</i>. 2. El sistema redirige el administrador a la pantalla de modificación de médico de un informe. 3. El administrador elige su nuevo médico en la lista de médicos disponibles. 4. El administrador selecciona el botón <i>Cambiar médico</i>. 5. El sistema confirma los datos. 6. El sistema cambia el médico titular del informe por el seleccionado por el administrador. 7. El sistema registra la acción creando una nueva auditoría. 8. El sistema redirige el administrador a la anterior lista de informes. 	
Escenario alternativo	
<p>Campo vacío:</p> <ol style="list-style-type: none"> 2. El administrador deja vacío un campo. 3. El sistema detecta que hay un campo vacío. 4. El sistema muestra por pantalla una notificación indicando el campo que está vacío. <p>Médico no encontrado:</p> <ol style="list-style-type: none"> 2. El administrador inserta un médico que no se encuentra en la lista de 	

<p>médicos.</p> <p>3. El sistema detecta que el médico insertado no está en la lista proporcionada.</p> <p>4. El sistema muestra una notificación por pantalla informando sobre la ausencia de dicho médico.</p>

Tabla 33: UC-AD-M-2: Cambiar Médico Informe.

Identificador	UC-AD-M-3
Nombre	Modificar Asignatura.
Actores	Administradores.
Precondiciones	El administrador debe haber iniciado sesión. El administrador debe encontrarse en una pantalla de listas asignaturas.
Postcondiciones	El sistema modifica el nombre de la asignatura.
Descripción	El administrador puede modificar el nombre de una asignatura.
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador selecciona el botón <i>Modificar</i>. 2. El sistema redirige el administrador a la pantalla de modificación de la asignatura. 3. El administrador modifica el nombre de la asignatura. 4. El administrador selecciona el botón <i>Modificar</i>. 5. El sistema confirma los datos. 6. El sistema cambia el nombre de la asignatura. 7. El sistema guarda la acción creando una nueva auditoría. 8. El sistema redirige el administrador a la lista de asignaturas. 	
Escenario alternativo	
Campo vacío:	

1. El administrador deja vacío un campo.
2. El sistema detecta que hay un campo vacío.
3. El sistema muestra por pantalla una notificación indicando el campo que está vacío.

Nombre en uso:

1. El administrador inserta un nombre que ya está en uso por otra asignatura.
2. El sistema detecta el problema.
3. El sistema muestra una notificación por pantalla informando sobre la duplicidad del nombre de la asignatura.

Tabla 33: UC-AD-M-3: Modificar Asignatura.

4

Desarrollo e implementación

En este cuarto capítulo, se describirá el funcionamiento interno de la aplicación web, además de las distintas iteraciones realizadas y la preparación que se ha tenido que realizar antes de comenzar con el proyecto.

4.1 Preparación y aprendizaje

Antes de empezar a programar la aplicación web, se debió aprender las distintas herramientas y programas que se iban a utilizar para la realización de este proyecto.

Una vez con los conocimientos básicos, se planeó seguir una metodología SCRUM, dividiendo las fases del proyecto en distintas iteraciones.

Dichas iteraciones se pueden agrupar en dos grupos: las iteraciones del back end, y las iteraciones del front end.

El primer tipo de iteración ha sido el que más tiempo ha consumido, pues había que asegurarse que el funcionamiento interno de la aplicación web fuese correcto. El segundo tipo de iteración, aunque no se invirtió tanto tiempo como en el primero, sí es cierto que hubo que aprender a usar las herramientas necesarias para manipular la parte del front end, encontrando una dificultad extra en ello.

Una vez dicho esto, se procederá a describir los distintos tipos de iteraciones.

4.2 Iteraciones del back end

4.2.1 Iteración de la base de datos (Iteraciones 1 y 2)

En esta iteración se decidió diseñar e implementar la base de datos. Fue escogida la primera porque sin una buena organización de la base de datos, la aplicación web podría sufrir de grandes carencias en un futuro, por lo que se le dio bastante prioridad. Aunque es cierto que al principio los datos estaban situados en una base de datos local, esa situación solo se dio en las primeras fases de estas iteraciones, pues después se migraría a una base de datos en la nube para que los clientes pudiesen acceder a ella sin problemas.

Se ha usado una base de datos relacional, organizando la información en tablas relacionadas entre sí. La primera base de datos se creó con SQL Server Express en local, y después se pasó a utilizar una base de datos relacional en la nube que proporciona Azure como uno de sus servicios. Estas bases de datos utilizan SQL como lenguaje principal para modificarlas, aunque también se ha utilizado LINQ a la hora de alterar los datos de la base de datos.

El esquema de la base de datos con sus tablas y relaciones es el siguiente:

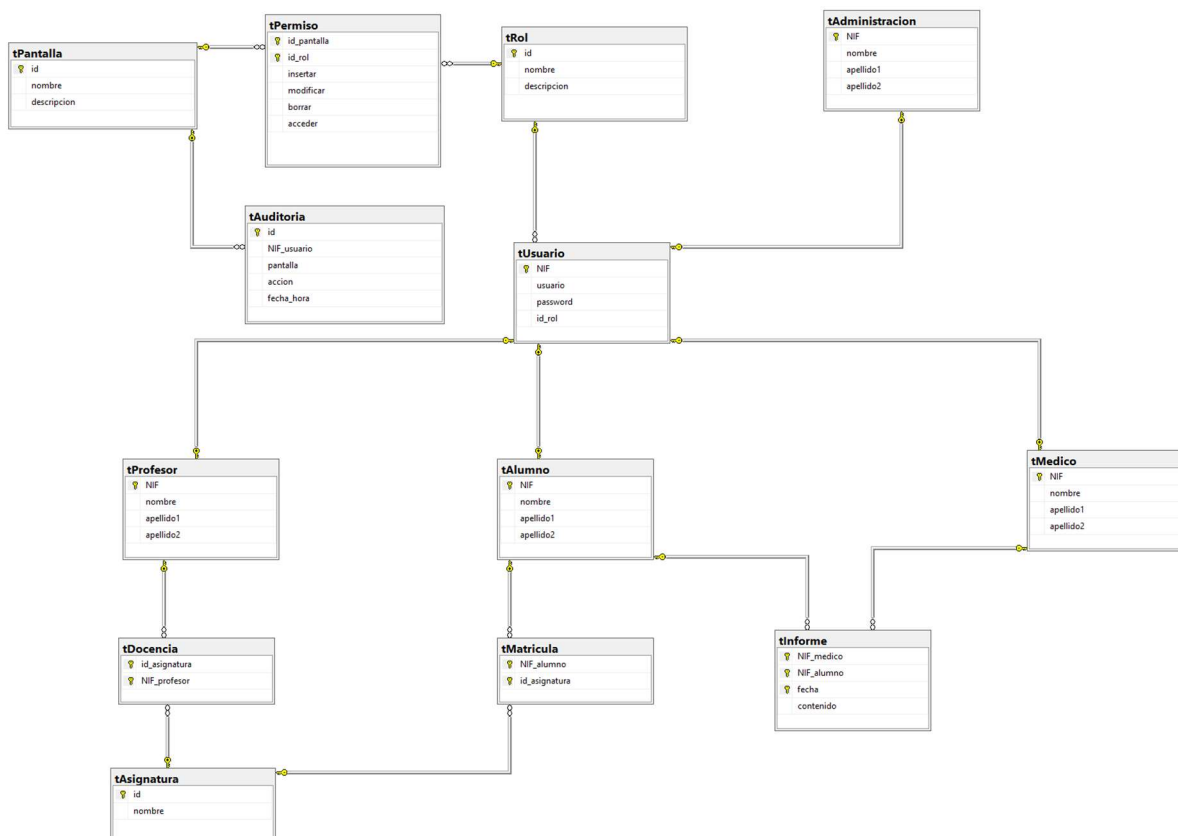


Imagen 9: Diagrama de la base de datos.

Se explicará las tablas y atributos que la conforman para mayor claridad, además de dar significado a los encabezados de las tablas:

- **Parámetro:** Nombre del atributo de la tabla.
- **Tipo:** Tipo del atributo de la tabla.
- **PK:** Siglas pertenecientes a “*Primary Key*”. Indica la clave primaria de un atributo. Esta propiedad es única, y no podrá existir más de una a la vez con la misma PK en dicha tabla.
- **FK:** Siglas pertenecientes a “*Foreign Key*”. Indica si el atributo es clave foránea de otra tabla. Si así se da el caso, en la descripción se detallará la tabla originaria del atributo.
- **NN:** Siglas pertenecientes a “*Not Null*”. Indica si el atributo puede estar vacío o no. En esta base de datos en particular, la mayoría de los atributos serán marcados como NN.
- **II:** Siglas pertenecientes a “*Identity Increment*”. Ligada a la sigla PK, indica si la clave primaria ha sido generada de forma incremental y automática, es decir, se crea ella sola tomando en cuenta el número de datos que contiene la tabla. Por ejemplo, si se crea un nuevo elemento en una tabla con una clave primaria numérica, y ya hubo 3 elementos creados con anterioridad, la nueva clave primaria tendrá como valor 4.

A continuación, las tablas que forman la base de datos y sus parámetros: (Las casillas marcadas con un **X**, significan que el atributo de la tabla en cuestión posee esas características. Esto también se aplica a la tabla que muestra los permisos pertenecientes a cada tipo de usuario).

- **tPantalla:** Esta tabla se encarga de describir las pantallas principales que existen en la aplicación web. Las pantallas son vistas importantes en las que un usuario puede o no acceder y realizar acciones según su rol y permisos. Hay un total de 14 pantallas principales:
 - **alumno:** Pantalla que contiene el perfil principal de un alumno.
 - **profesor:** Pantalla que contiene el perfil principal de un profesor.
 - **medico:** Pantalla que contiene el perfil principal de un médico.

- **administracion:** Pantalla que contiene el perfil principal de un administrador.
- **informe:** Pantalla que contiene la página principal de un informe.
- **auditoria:** Pantalla que contiene la lista de las auditorías.
- **lista_alumnos:** Pantalla que contiene una lista de alumnos.
- **lista_profesores:** Pantalla que contiene una lista de profesores.
- **lista_medicos:** Pantalla que contiene una lista de médicos.
- **lista_asignaturas:** Pantalla que contiene una lista de asignaturas.
- **lista_pantallas:** Pantalla que contiene la lista de pantallas junto a sus descripciones.
- **lista_informes:** Pantalla que contiene una lista de informes.
- **lista_matriculas:** Pantalla que contiene la lista de todas las matrículas.
- **lista_docencias:** Pantalla que contiene la lista de todas las docencias.

Parámetro	Tipo	PK	FK	NN	II	Descripción
id	int	X		X	X	Identificador único de la pantalla.
nombre	varchar(50)			X		Nombre de la pantalla.
descripción	varchar(500)			X		Explicación medianamente breve sobre la pantalla en cuestión.

Tabla 34: tPantalla.

- **tRol:** Esta tabla se encarga de almacenar los cuatro posibles roles que un usuario puede adoptar, los cuales sería: alumno, profesor, médico o administrador, cada uno con sus respectivos permisos.

Parámetro	Tipo	PK	FK	NN	II	Descripción
id	int	X		X	X	Identificador único del rol.
nombre	varchar(50)			X		Nombre del rol.
descripción	varchar(500)			X		Explicación medianamente breve sobre el rol y sus permisos.

Tabla 35: tRol.

- **tPermiso:** Esta tabla se encarga de guardar los permisos de cada rol. Los permisos se utilizan para comprobar si un usuario con un rol determinado puede realizar una determinada acción en una pantalla, o si quiera acceder a ella. Hay cuatro tipos de permisos: insertar, modificar, borrar y acceder:
 - **Insertar:** Permite a un usuario insertar elementos en una pantalla determinada.
(Ejemplo: Un usuario administrador puede insertar un nuevo usuario alumno en la pantalla de lista de alumnos).
 - **Modificar:** Permite a un usuario modificar algún atributo de un elemento existente.
(Ejemplo: Un usuario médico puede modificar el contenido de un informe del que esté a cargo).
 - **Borrar:** Permite a un usuario borrar elementos en una determinada pantalla. Normalmente se necesita confirmación adicional para ello.
(Ejemplo: Un usuario administrador puede borrar el usuario de un alumno en la pantalla de listas de alumnos, junto a todos los elementos que tengan que ver con dicho alumno, como sus matrículas o informes).
 - **Acceder:** Permite a un usuario acceder a una pantalla determinada. Sin este permiso, ni siquiera se podría acceder a dicha pantalla.
(Ejemplo: Un usuario administrador puede viajar a la pantalla lista de alumnos, pero un usuario alumno no, por lo que nunca podrá acceder a esa pantalla en específico).

A continuación, se mostrará la tabla que representa a **tPermiso**, junto a otra adicional que muestra los permisos que contiene cada usuario:

Parámetro	Tipo	PK	FK	NN	II	Descripción
Id_pantalla	int	X	X	X		Identificador único de la pantalla. El origen de la clave foránea es la tabla tPantalla .
Id_rol	int	X	X	X		Identificador único del rol. El origen de la clave foránea es la tabla tRol .
insertar	bit			X		Atributo que puede contener dos posibles valores: 0 o 1. En este caso,

						1 representa la presencia del permiso insertar , mientras que el 0 indica su ausencia.
modificar	bit			X		Atributo que puede contener dos posibles valores: 0 o 1. En este caso, 1 representa la presencia del permiso modificar , mientras que el 0 indica su ausencia.
borrar	bit			X		Atributo que puede contener dos posibles valores: 0 o 1. En este caso, 1 representa la presencia del permiso borrar , mientras que el 0 indica su ausencia.
acceder	bit			X		Atributo que puede contener dos posibles valores: 0 o 1. En este caso, 1 representa la presencia del permiso acceder , mientras que el 0 indica su ausencia.

Tabla 36: tPermiso

Pantalla	Usuario	Insertar	Modificar	Borrar	Acceder
alumno	alumno				X
alumno	profesor				X
alumno	médico				
alumno	administrador		X	X	X
profesor	alumno				
profesor	profesor				X
profesor	médico				
profesor	administrador		X	X	X
medico	alumno				
medico	profesor				
medico	médico				X
medico	administrador		X	X	X
administracion	alumno				
administracion	profesor				
administracion	médico				
administracion	administrador		X		X

informe	alumno				X
informe	profesor				X
informe	médico		X		X
informe	administrador			X	X
auditoria	alumno				
auditoria	profesor				
auditoria	médico				
auditoria	administrador				X
lista_alumnos	alumno				
lista_alumnos	profesor				X
lista_alumnos	médico				X
lista_alumnos	administrador	X		X	X
lista_profesores	alumno				
lista_profesores	profesor				
lista_profesores	médico				
lista_profesores	administrador	X		X	X
lista_medicos	alumno				
lista_medicos	profesor				
lista_medicos	médico				
lista_medicos	administrador	X		X	X
lista_asignaturas	alumno				X
lista_asignaturas	profesor				X
lista_asignaturas	médico				
lista_asignaturas	administrador	X	X	X	X
lista_pantallas	alumno				
lista_pantallas	profesor				
lista_pantallas	médico				
lista_pantallas	administrador				X
lista_informes	alumno				X
lista_informes	profesor				X
lista_informes	médico				X
lista_informes	administrador	X	X	X	X
lista_matriculas	alumno				
lista_matriculas	profesor				
lista_matriculas	médico				
lista_matriculas	administrador	X		X	X
lista_docencias	alumno				
lista_docencias	profesor				
lista_docencias	médico				
lista_docencias	administrador	X		X	X

Tabla 37: Permisos según el tipo de usuario y la pantalla en donde se encuentren.

- **tUsuario:** Esta tabla se encarga de almacenar la contraseña, nombre de usuario y NIF de cada usuario creado, además de señalar su rol. Cuando se crea un usuario de cualquier tipo, primero siempre se creará un registro de tipo **tUsuario**, y después un tipo **tAlumno**, **tProfesor**, o **tMedico** según su rol.

Parámetro	Tipo	PK	FK	NN	II	Descripción
NIF	varchar(50)	X		X		Identificador único del usuario. Corresponde al NIF o DNI real de la persona.
usuario	varchar(50)			X		Nombre de usuario del nuevo elemento registrado.
password	varchar(50)			X		Contraseña del usuario.
Id_rol	int		X	X		Identificador único del rol. El origen de la clave foránea es la tabla tRol .

Tabla 38: tUsuario.

- **tAlumno:** Esta tabla se encarga de guardar toda la información necesaria y relativa a los usuarios cuyo rol sea el de alumno.

Parámetro	Tipo	PK	FK	NN	II	Descripción
NIF	varchar(50)	X	X	X		Identificador único del alumno. El origen de la clave foránea es la tabla tUsuario .
nombre	varchar(50)			X		Nombre del alumno.
apellido1	varchar(50)			X		Primer apellido del alumno.
apellido2	varchar(50)			X		Segundo apellido del alumno.

Tabla 39: tAlumno.

- **tProfesor:** Esta tabla se encarga de guardar toda la información necesaria y relativa a los usuarios cuyo rol sea el de profesor.

Parámetro	Tipo	PK	FK	NN	II	Descripción
NIF	varchar(50)	X	X	X		Identificador único del profesor. El origen de la clave foránea es la tabla tUsuario .
nombre	varchar(50)			X		Nombre del profesor.
apellido1	varchar(50)			X		Primer apellido del profesor.
apellido2	varchar(50)			X		Segundo apellido del profesor.

Tabla 40: tProfesor.

- **tMedico:** Esta tabla se encarga de guardar toda la información necesaria y relativa a los usuarios cuyo rol sea el de médico.

Parámetro	Tipo	PK	FK	NN	II	Descripción
NIF	varchar(50)	X	X	X		Identificador único del médico. El origen de la clave foránea es la tabla tUsuario .
nombre	varchar(50)			X		Nombre del médico.
apellido1	varchar(50)			X		Primer apellido del médico.
apellido2	varchar(50)			X		Segundo apellido del médico.

Tabla 41: tMedico.

- **tAdministracion:** Esta tabla se encarga de guardar toda la información necesaria y relativa a los usuarios cuyo rol sea el de administrador.

Parámetro	Tipo	PK	FK	NN	II	Descripción
NIF	varchar(50)	X	X	X		Identificador único del administrador. El origen de la clave foránea es la tabla tUsuario .
nombre	varchar(50)			X		Nombre del administrador.
apellido1	varchar(50)			X		Primer apellido del administrador.
apellido2	varchar(50)			X		Segundo apellido del administrador.

Tabla 42: tAdministracion.

- **tAsignatura:** Esta tabla se encarga de almacenar los datos que tengan que ver con las asignaturas.

Parámetro	Tipo	PK	FK	NN	II	Descripción
id	int	X		X	X	Identificador único de la asignatura.
nombre	varchar(50)			X		Nombre de la asignatura.

Tabla 43: tAsignatura.

- **tInforme:** Esta tabla se encarga de almacenar los datos que tengan que ver con los informes.

Parámetro	Tipo	PK	FK	NN	II	Descripción
NIF_medico	varchar(50)	X	X	X		Identificador único del médico titular. El origen de la clave foránea es la tabla tMedico .
NIF_alumno	varchar(50)	X	X	X		Identificador único del alumno al que pertenece el informe. El origen de la clave foránea es la tabla tAlumno .
fecha	datetime	X		X		Fecha de la creación del informe. El formato está en: (año/mes/día hora:minutos:segundos).
contenido	varbinary(MAX)			X		Contenido del informe en PDF.

Tabla 43: tInforme.

- **tDocencia:** Esta tabla se encarga de almacenar los datos que tengan que ver con las asignaturas que los profesores imparten.

Parámetro	Tipo	PK	FK	NN	II	Descripción
id_asignatura	int	X	X	X		Identificador único de la asignatura. El origen de la clave foránea es la tabla tAsignatura .
NIF_profesor	varchar(50)	X	X	X		Identificador único del profesor. El origen de la clave foránea es la tabla tProfesor .

Tabla 44: tDocencia.

- **tMatricula:** Esta tabla se encarga de almacenar los datos que estén relacionados con las asignaturas en las que los alumnos estén matriculados.

Parámetro	Tipo	PK	FK	NN	II	Descripción
NIF_alumno	varchar(50)	X	X	X		Identificador único del alumno. El origen de la clave foránea es la tabla tAlumno .
id_asignatura	int	X	X	X		Identificador único de la asignatura. El origen de la clave foránea es la tabla tAsignatura .

Tabla 45: tMatricula.

- **tAuditoria:** Esta tabla se encarga de almacenar los datos que tengan que ver con la modificación, borrado o inserción de algún elemento en la aplicación web mediante el uso de las vistas. En otras palabras, lleva un registro de todas las acciones que hayan tenido que ver con la modificación de algún elemento en la base de datos.

Parámetro	Tipo	PK	FK	NN	II	Descripción
id	int	X		X	X	Identificador único de la auditoría.
NIF_usuario	varchar(50)			X		Identificador único del usuario que ha realizado la acción.
pantalla	int		X	X		Identificador único de la pantalla. El origen de la clave foránea es la tabla tPantalla .
accion	varchar(500)			X		Descripción de la acción cometida.
fecha_hora	datetime			X		Fecha de la realización de la acción. El formato está en: (año/mes/día hora:minutos:segundos).

Tabla 46: tAuditoria.

4.2.1 Iteración del MVC y de las funcionalidades (Iteración 3 y 4)

Una vez acabado el diseño de la estructura de la base de datos, y haber incluido algunos datos de prueba para probar distintas funcionalidades, se pasó a crear el nuevo proyecto en .NET Core 7.

El patrón de diseño elegido para el proyecto sería el de modelo vista controlador (**MVC**), debido a que se trata de una estructura clara y modular para el desarrollo de aplicaciones, facilitando la separación de responsabilidades, la reutilización de código, el mantenimiento y la escalabilidad.

Terminada la elección del patrón de diseño, y la conexión a la base de datos, se dispuso a crear los elementos de un MVC en el siguiente orden:

- 1) **Modelo:** En un patrón MVC, los modelos son los responsables de la representación de datos, el acceso a la capa de datos, la implementación de la lógica de negocio, y la notificación de cambios en los datos.

Para crear los modelos de forma fidedigna a la base de datos, se utilizó el concepto de andamiaje (o *scaffolding* en inglés), que consiste en la generación automática de código base y estructura inicial de un proyecto. Esta técnica se aplicó a la base de datos para trasladar la estructura de las tablas al proyecto con precisión. Aquí una imagen de los modelos del proyecto:

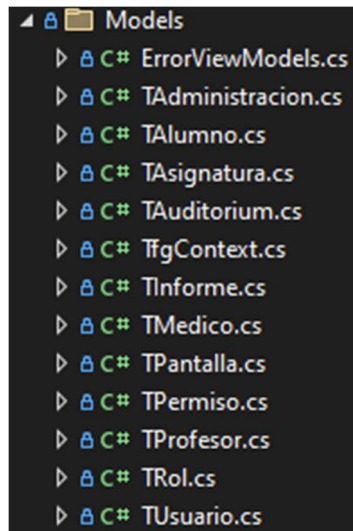


Imagen 10: Carpeta con las clases que representan a los modelos.

- 2) **Controlador:** En un MVC, los controladores son los componentes responsables de manejar las solicitudes del usuario, procesar la lógica de negocio y coordinar la comunicación entre los modelos y las vistas. Los nombres son muy intuitivos, y cada controlador pertenece a la tabla del principio de su nombre. (*Ejemplo: **TAlumnosController** pertenecería a la tabla **TAlumnos***). Aquí una imagen de los controladores:

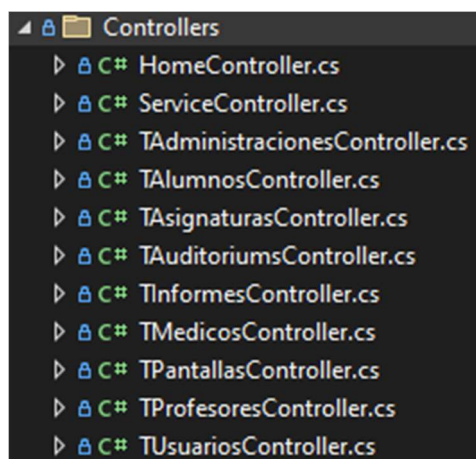


Imagen 11: Carpeta con las clases que representan a los controladores.

3) **Vista:** En un MVC, las vistas son los componentes encargados de presentar la información al usuario y manejar la interfaz de usuario.

En el caso de las vistas en estas dos iteraciones, no se les dio mucha prioridad, pues el objetivo principal es que se completara de forma correcta toda la funcionalidad correspondiente al back end. Por ende, las primeras vistas que se crearon se hicieron con un HTML básico, siendo mejoradas y modificadas en las iteraciones del front end. Aquí está una captura de las vistas creadas:

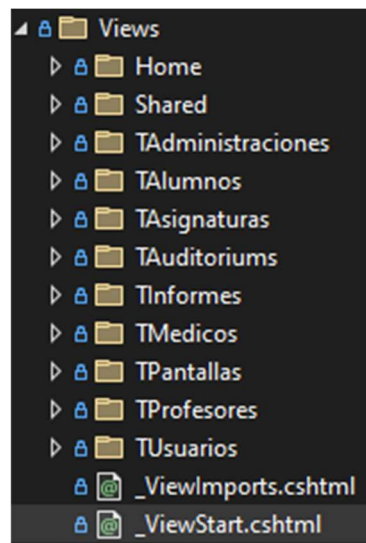


Imagen 12: Carpeta con las clases que representan a las vistas.

Paralelamente se fue tratando y arreglando distintos problemas que tenía la aplicación. Por dicho motivo, se tuvo que crear clases adicionales, y expandir otras para que cumpliesen con las funcionalidades requeridas. Se pueden dividir en cinco tipos:

1) **Constantes:** Para no repetir valores, se creó una clase donde se almacenaría todos los valores que tuviesen un uso frecuente a lo largo del proyecto. Así, no sería necesario escribirlo más de una vez. La carpeta y la clase serían los siguientes:

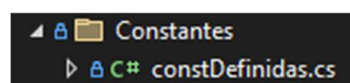


Imagen 13: Carpeta con la clase que representan a las constantes.

- 2) **Interfaz:** Cuando se creó las funciones que iban a utilizarse de forma global en el proyecto, se tuvo que diseñar a la vez una interfaz para poder incluirlo en toda la aplicación web. La carpeta y la interfaz serían los siguientes:

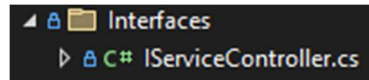


Imagen 14: Carpeta con la clase que representan a la interfaz de la clase ServiceController.cs.

- 3) **ViewModels:** El pasar más de un dato a una vista supuso un problema, por lo que se creó una serie de clases diseñadas para almacenar distintos atributos y pasarlo a una vista en concreto como si fuera un paquete. La carpeta y las clases serían los siguientes:

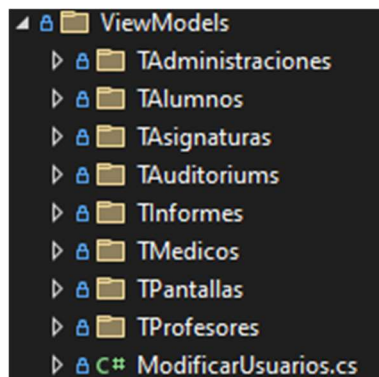


Imagen 15: Carpeta que representa a las vistas del proyecto.

- 4) **Program.cs:** Esta clase en particular es fundamental para iniciar la ejecución de la aplicación web, ya que contiene métodos tan importantes como el que se encarga de cargar el host. Se tuvo que modificar esta clase para añadirle funcionalidades extra, como la existencia de una sesión a lo largo del programa para poder guardar datos en ella, la inyección de la clase ServiceController.cs para que otros métodos puedan usarla, la conexión a la base de datos, y otros servicios importantes.

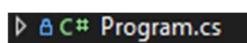


Imagen 16: Program.cs.

5) **Navigation:** La navegación en la aplicación web se podría comparar a la de una lista enlazada: cada vez que se cambia de vista, la lista apunta a esa nueva vista, o vuelve a la que estaba apuntando antes según corresponda. Por este motivo se creó una clase que simule este comportamiento, ya que la sesión no era suficiente para llevar un registro de donde se encontraba el usuario. Esta clase contiene 3 atributos, 2 de ellos para almacenar el NIF y el rol del usuario en el que se encuentra, (de ahora en adelante llamado como **usuario actual**), y otro atributo especial que apuntará a su “padre”, es decir, al anterior usuario actual. El funcionamiento de esta clase es el siguiente: Cuando se inicia sesión, se crea una instancia de la clase UserNavigation con el “atributo” padre a **null**, ya que no hay donde volver. El usuario actual sería el usuario con el que se haya iniciado sesión. Ahora bien, cada vez que accedamos a la vista del perfil principal de otro usuario, se creará otra instancia de UserNavigation. En ese caso, el usuario actual cambiaría al del usuario al que le pertenezca esa página principal, mientras que el atributo “padre” se convertiría en el usuario anterior antes de entrar en dicho perfil principal, formando en esencia la lista enlazada mencionada al principio. Aquí una imagen de la carpeta y la clase, más unos ejemplos gráficos de la operatividad de la clase:

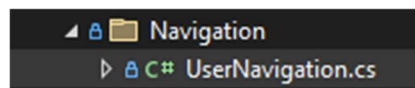


Imagen 17: Carpeta que representa a la clase que controla la navegación en la aplicación web.

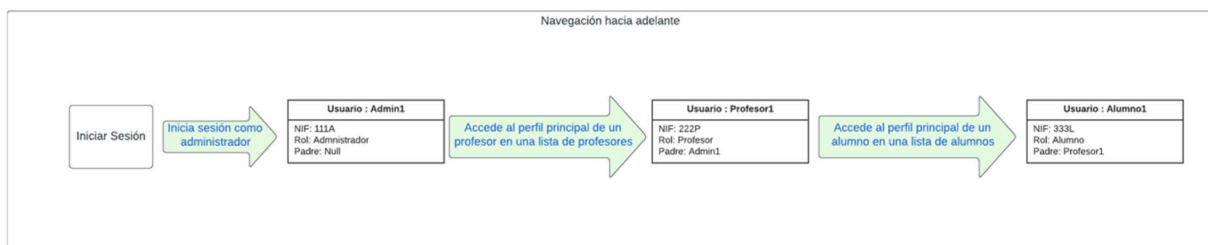


Imagen 18: Representación de la navegación de un administrador a través de varios usuarios con una navegación “progresiva”.

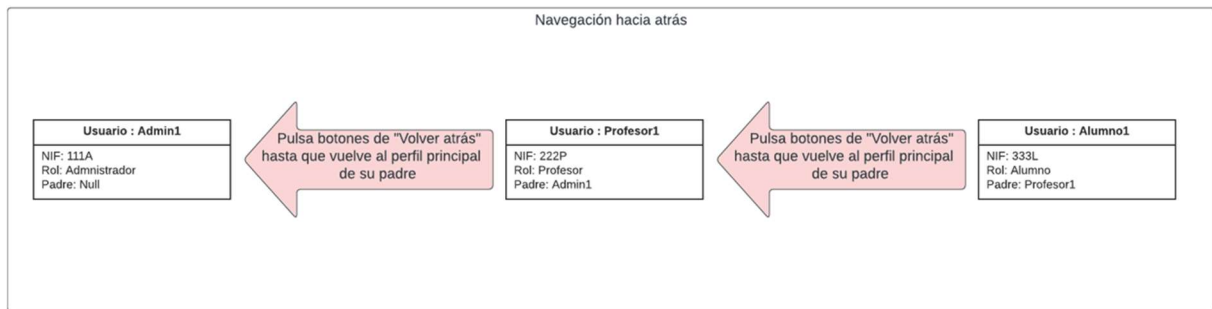


Imagen 19: Representación de la navegación de un administrador a través de varios usuarios con una navegación “regresiva”.

4.3 Iteraciones del front end

4.3.1 Iteraciones principales de las vistas (Iteración 1 y 2)

Una vez terminado el back end, se decidió pulir y mejorar todos los aspectos que tuviesen que ver con el front end. Decidiendo dejar los elementos interactivos para lo último, se empezó a diseñar las vistas comunes más básicas y simples. Con esto en mente, se creó una pequeña serie de patrones para que hubiese distinción y orden a la hora de diseñar las vistas:

- **Botones:** El diseño de los botones sería rectangular con bordes redondeados, y cada color tendría un significado:
 - **Azul primario** : Se utiliza para representar la creación de un elemento nuevo, la modificación de un elemento existente, u otra acción especial como volver al perfil principal o iniciar sesión.
 - **Rojo peligro** : Se utiliza para representar la confirmación de la eliminación permanente de un elemento, o para indicar el cierre de sesión de un usuario.
 - **Azul info** : Se utiliza para representar los botones de navegación entre las vistas o para ver los informes.
 - **Negro oscuro** : Se utiliza para representar los botones que pueden volver a vistas anteriores, es decir, volver atrás en la navegación.
 - **Amarillo advertencia** : Se utiliza para representar botones que tengan que ver con la eliminación de un elemento. Sin embargo, estos

botones solo se limitarán a llevar al usuario a una vista para confirmar el borrado, por lo que son distintos a los de color rojo peligro.

- **Logo:** La aplicación web cuenta con un logo tanto en la barra de navegación, como en el *favicon*, (el icono pequeño de las ventanas de un navegador). No tiene ninguna funcionalidad extra aparte de diseño, por lo que, si se intenta interactuar con él pulsándolo con el puntero del ratón, no hará nada.



Imagen 20: Logo de la aplicación web.

- **Barra de navegación:** En la aplicación web, habrá una barra de menú en todo momento que según si un usuario ha iniciado sesión o no, mostrará cierta información, como su rol y nombre de usuario, además de activarse los botones para poder cerrar sesión y volver al perfil principal del usuario.



Gestión Diversidad

Imagen 21: Barra de navegación al inicio de la aplicación web.



Gestión Diversidad

Cerrar Sesión

Perfil Principal

Usuario: admin1 Rol: Administrador/a

Imagen 22: Barra de navegación después de que un usuario haya iniciado sesión.

- **Advertencias:** En cuanto a las advertencias, (o *warnings* en inglés), normalmente son de un color *rojo peligro*, siendo más comunes en la creación o búsqueda de

algún elemento. También existen advertencias de color azul *info* claro, informando sobre algún aspecto que no es incorrecto, pero que debería revisarse.

Gestión Diversidad Cerrar Sesión Perfil Principal Usuario: admin1 Rol: Administrador/a

Lista de informes: Administrador/a leafeon (NIF: 111AD)

Mostrar: 10 informes

Buscar:

Alumno(NIF)	Médico(NIF)	Fecha	Acción: Detalles del informe	Acción: Borrar informe	Acción: Cambiar médico titular
Juan (11111111A)	Ricardo (11111111M)	2023/05/31 23:06:54	Detalles	Borrar	Cambiar médico
Juan (11111111A)	Carlos (33333333M)	2023/06/06 23:00:51	Detalles	Borrar	Cambiar médico
Marta (22222222A)	Javier (22222222M)	2023/05/31 23:08:34	Detalles	Borrar	Cambiar médico
Nacho (33333333A)	Carlos (33333333M)	2023/05/31 23:09:51	Detalles	Borrar	Cambiar médico

Mostrando 1 a 4 de 4 informes Anterior 1 Siguiente

[Crear nuevo Informe](#) [Volver atrás](#)

A ese alumno solo le queda un informe. Como es obligatorio que tenga al menos uno, por favor, borre el alumno completamente o añada otro informe

Imagen 23: Error al intentar borrar el único informe que le queda a un alumno.

Gestión Diversidad Cerrar Sesión Perfil Principal Usuario: admin1 Rol: Administrador/a

Crear nuevo/a profesor/a

Nif:
El Nif no puede estar vacío

Usuario:
El nombre del usuario ya está en uso

Password:
La contraseña no puede estar vacía

Confirmar Password:
La confirmación de la contraseña no puede estar vacía

Nombre:
El nombre no puede estar vacío

Primer apellido:
El primer apellido no puede estar vacío

Segundo apellido:
El segundo apellido no puede estar vacío

[Crear profesor](#) [Volver atrás](#)

© 2023 - Gestión Diversidad

Imagen 24: Errores en los atributos al intentar crear un nuevo profesor.

Modificar médico

Nombre:

Primer Apellido:

Segundo Apellido:

Usuario:

Password:

Confirmar Password:

[Modificar](#)

[Volver atrás](#)

No se ha modificado ningún parámetro del usuario

Imagen 25: Advertencia de cómo no se ha modificado ningún atributo de un usuario.

- **Fondo de pantalla:** El fondo de pantalla de la aplicación es una imagen con diseño simple y minimalista, pero a la vez elegante y ligero.



Imagen 26: Imagen del fondo de pantalla sin ningún elemento.

Una vez planteada esa estructura, se diseñó primero aquellas páginas más estáticas, o, en otras palabras, aquellas vistas en las que no se necesitaba demasiada funcionalidad extra, como por ejemplo las listas de elementos, o cualquier elemento que necesite ser

modificado de forma dinámica. Las páginas que fueron diseñadas primeros fueron:

- **Iniciar sesión:** Esta vista es la primera que verá cualquier usuario, ya que es la encargada de iniciar sesión.



Gestión Diversidad

Inicio sesión

Usuario:

Password:

Iniciar Sesión

© 2023 - Gestión Diversidad

Imagen 27: Inicio de sesión.

- **Perfil principal:** Esta vista representa la página principal de un usuario, en la cual se mostrará datos básicos como su nombre, apellidos y NIF. También estarán los botones correspondientes al rol y permisos del usuario que haya iniciado sesión. Esta vista es a la que redirigirá el botón "*Perfil Principal*" siempre que se pulse, independientemente de donde se encuentre el usuario.



Información del/a administrador/a

Nombre:
leafeon

Primer apellido:
leafeon1

Segundo apellido:
leafeon2

Nif:
111AD

Alumnos Profesores Médicos

Asignaturas Matrículas Docencias

Informes Pantallas Auditorías

© 2023 - Gestión Diversidad

Imagen 28: Perfil principal de un administrador.



Información del/a alumno/a

Nombre:
Juan

Primer apellido:
García

Segundo apellido:
Martínez

Nif:
11111111A

Asignaturas

Informes

© 2023 - Gestión Diversidad

Imagen 29: Perfil principal de un alumno.



Información del profesor

Nombre
Alejandro

Primer apellido:
Gómez

Segundo apellido:
López

Nif del profesor:
11111111P

Asignaturas

Alumnos

© 2023 - Gestión Diversidad

Imagen 30: Perfil principal de un profesor.



Información de médico

Nombre:
Carlos

Primer apellido:
Ramírez

Segundo apellido:
Navarro

Nif:
33333333M

Alumnos

Informes

© 2023 - Gestión Diversidad

Imagen 31: Perfil principal de un médico.

- **Página del informe:** Esta vista es el equivalente al “Perfil Principal” de un usuario, pero para un informe. Esta vista mostrará el nombre del alumno al que corresponde, a la vez que el nombre de su médico titular. En adición, la vista indicará la fecha de creación del informe.

Hay un botón en esta vista para poder ver el contenido del informe. Dicho contenido se abrirá en una ventana nueva, y podrá guardarse en formato PDF como cualquier documento.

Si el usuario tiene el rol de Médico en esta vista, obtiene la capacidad de modificar el contenido de dicho informe por otro PDF que el usuario elija, siempre y cuando sea el médico titular.



Imagen 32: “Perfil Principal” de un informe habiendo iniciado sesión como administrador.



Imagen 33: “Perfil Principal” de un informe habiendo iniciado sesión como su médico titular.

- **Página para confirmar borrado:** Siempre que se quiera eliminar de forma permanente un elemento, ya sea usuario, matrícula, informe... se tendrá que pasar por esta vista. Esta página preguntará al usuario si realmente quiere borrar el elemento en cuestión. En ella, se mostrarán principalmente dos botones: Uno para volver hacia atrás, y otro para confirmar de forma definitiva la eliminación del elemento.

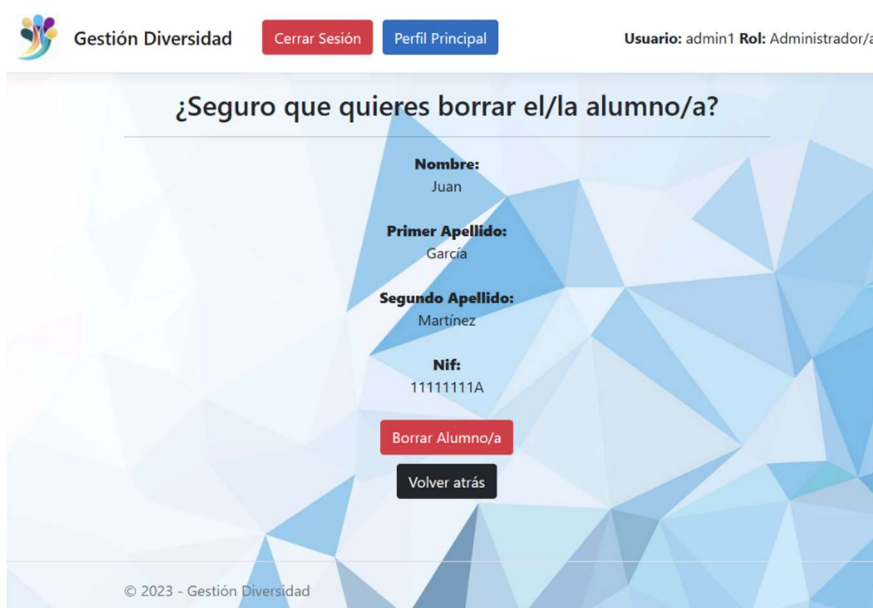


Imagen 34: Confirmación de la eliminación de un alumno.

Terminado de diseñar las primeras vistas, las dos primeras iteraciones del front end finalizaron.

4.3.1 Iteraciones avanzadas de las vistas (Iteración 3 y 4)

Ahora que las vistas más simples estaban terminadas, se procedería a diseñar aquellas vistas que necesitasen de más funcionalidad para hacer que funcionen de forma correcta. Entre estas vistas se incluyen:

- **Listas de elementos:** Cuando se accede a una pantalla que tenga información sobre una lista de elementos, estos estarán organizados en una tabla. La tabla mostrará información básica de los elementos, además de los botones correspondientes a las acciones que se podrá realizar según el usuario que haya iniciado sesión. Para explicar las opciones que brinda este tipo de página, se dividirá en subapartados:
 - **Filtro de búsqueda:** Arriba de la tabla, se puede encontrar una caja de texto en la que pone “Buscar”. Si se escribe ahí, la tabla en el momento filtrará los elementos que contengan dichos caracteres. Esta función no distingue entre mayúsculas y minúsculas, pero sí las letras que contengan tildes o caracteres especiales.
 - **Cantidad por mostrar:** Se puede configurar la tabla para que muestre un número predefinido de elementos. Esta cantidad será 10, 25, 50 y 100. Si los elementos superan la cantidad numérica, la tabla organizará los demás datos sobrantes en páginas. Tanto el número de los elementos mostrados, como el número total de ellos, se muestran abajo a la izquierda. Además, se puede navegar entre las páginas de la tabla con los botones “Anterior” o “Siguiente”, o simplemente pulsando en los números que se muestran.
 - **Orden alfabético y numérico:** Si se pulsa en las cabeceras de las columnas de las tablas, se ordenarán los elementos alfabéticamente según la columna que se haya pulsado. Lo mismo ocurre con los números, ordenándolos de mayor a menor y viceversa. Sin embargo, hay un par de excepciones. En cuanto a las letras, caracteres especiales como letras con tilde, las considerará después de las letras del alfabeto, y en las fechas, solo tendrá en cuenta los números como tal.



Lista matrículas de alumnos

Mostrar matriculas

Buscar:

Alumno	Nif	Asignatura	Acción: Borrar matrícula
Marta	22222222A	Cálculo	Borrar
Nacho	33333333A	Lengua	Borrar
Nacho	33333333A	Álgebra	Borrar

Mostrando 1 a 3 de 3 matriculas

[Anterior](#) [1](#) [Siguiete](#)

[Crear nueva matrícula](#)

[Volver atrás](#)

© 2023 - Gestión Diversidad

Imagen 35: Lista de las matrículas de los alumnos.

- **Creación de un nuevo elemento:** Las vistas para crear nuevos elementos como un nuevo usuario, o un nuevo informe, están compuestas de varios componentes en los que se puede rellenar los atributos necesarios para la creación de la nueva instancia. Estos pueden ser:
 - **Cajas de texto:** Usadas normalmente para rellenar atributos como el nombre o apellidos de un usuario. En casos especiales, como el nombre de una asignatura, o el NIF de un usuario, el sistema detectará si ya existe alguna asignatura o usuario con esos elementos, indicándolo en un mensaje en rojo debajo del campo.
 - **Listas de elementos con buscadores:** Para elegir un elemento de una lista, se ha proporcionado una lista de donde elegir las posibles opciones. También se puede escribir en ella para filtrar los posibles resultados. Hay que tener en cuenta, que aparte de no considerar los espacios en blanco al inicio y al final, también tiene una estructura predefinida (nombre | NIF), por lo que es aconsejable elegir los elementos de la lista en vez de escribirlo uno mismo a mano. Si se intenta simular la estructura con datos inventados, el sistema lo detectará e informará de que no se encuentra los elementos insertados.

- **Botón de selección de Archivo:** Este botón es exclusivo para insertar el contenido de un informe, como cuando se va a crear un alumno o un informe a parte. Solo aceptará archivos en PDF, ignorando los demás formatos.

Gestión Diversidad Cerrar Sesión Perfil Principal Usuario: admin1 Rol: Administrador/a

Crear nuevo/a alumno/a

Un/a alumno/a deberá tener un informe asignado como mínimo

Nif:

Usuario:

Password:

Confirmar Password:

Nombre:

Primer apellido:

Segundo apellido:

Seleccione el archivo PDF:

No se ha seleccionado ningún archivo

Médico titular:

© 2023 - Gestión Diversidad

Imagen 36: Creación de un nuevo alumno.

- **Modificación de los elementos:** Los atributos a modificar pueden variar según el elemento, aunque por lo general son todos aquellos atributos presentes cuando se creó el elemento, a excepción de algunos como el NIF de un usuario o el id de una asignatura. También tiene las mismas advertencias que las vistas anteriores, sumando ahora una advertencia adicional por si se intenta modificar los campo con exactamente los mismos datos. En dicha situación, saltará una alarma informativa advirtiendo sobre eso. También hay una vista donde se puede cambiar el médico titular de un informe.

Modificar alumno/a

Nombre:

Primer Apellido:

Segundo Apellido:

Usuario:

Password:

Confirmar Password:

[Modificar](#)

[Volver atrás](#)

© 2023 - Gestión Diversidad

Imagen 37: Modificación de un alumno.

Cambiar el médico de un informe

Alumno:
Juan (11111111A)

Médico actual:
Ricardo (11111111M)

Nuevo médico:

[Cambiar médico](#)

[Volver atrás](#)

© 2023 - Gestión Diversidad

Imagen 38: Modificación del médico titular de un informe.

4.4 Diagramas de clase

Para una visión más completa del proyecto, se mostrarán las diferentes clases que lo forman junto a sus atributos. Sin embargo, debido a la gran cantidad de clases, se ha decidido representar estas entidades a un alto nivel y en distintas clasificaciones.

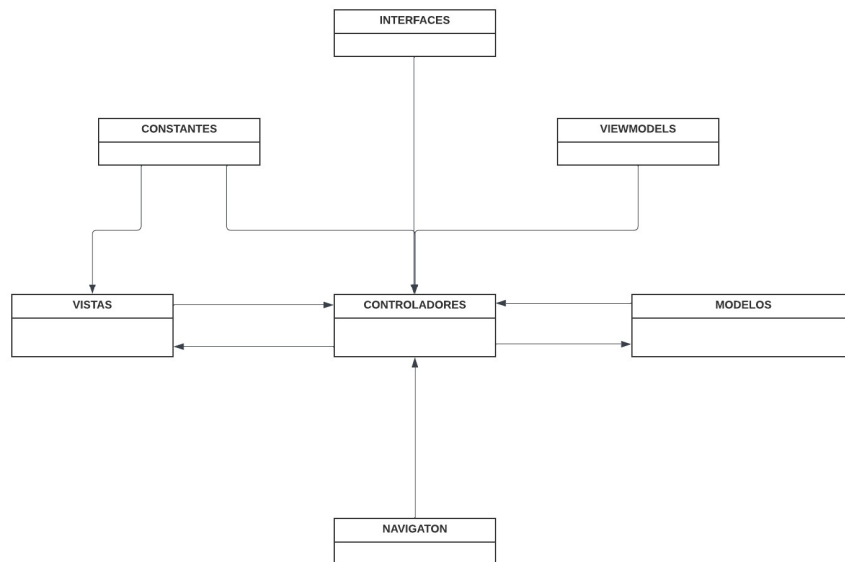


Imagen 39: Diagrama a alto nivel de las clases de la aplicación web.

Las tablas que se van a utilizar para representar las clases se dividen en dos tipos:

- **Tablas de color verde:** Se encargan de describir los atributos de una clase.
- **Tablas de color naranja:** Se encargan de describir los métodos de una clase.

Antes de seguir, se explicará algunos conceptos algo difíciles de entender para una mejor comprensión de este apartado:

- **Task:** Es una clase genérica que representa a una tarea asíncrona. Esta tarea puede devolver distintos tipos de resultados como lo haría un método normal, (Ejemplo: `Task<int>`), y normalmente se utiliza `<Task>` en los métodos que accedan a la base de datos por cualquier motivo.
- **ICollection:** Es una interfaz que define un conjunto de miembros a trabajar con colecciones de elementos. Normalmente es utilizada como si fuera una lista de elementos.
- **DateTime:** Es una estructura que representa una fecha y hora específicas. En este caso, el formato sería: *año/mes/día hora:minutos:segundos*.

Ahora se va a explicar parte por parte los métodos y los atributos:

4.4.1 Diagramas de clase UML: Modelos



Imagen 40: Diagrama a alto nivel de las clases que forman los modelos.

- **TRol**: Clase que representa a los roles que pueden tomar los usuarios.

Atributo	Tipo	Descripción
Id	int	Atributo que guarda el identificador único del rol.
Nombre	string	Atributo que guarda el nombre del rol.
Descripcion	string	Atributo que guarda una descripción del rol.
TPermisos	ICollection<TPermiso>	Atributo que conecta un rol con los permisos. Guarda los permisos de dichos roles.

TUsuarios	ICollection<TUsuario>	Atributo que conecta un rol con un usuario. Guarda los usuarios que tienen roles.
------------------	-----------------------	--------------------------------------------------------------------------------------

Tabla 47: Atributos de tRol

- **TPermiso:** Clase que representa a los permisos que tiene cada rol.

Atributo	Tipo	Descripción
IdPantalla	int	Atributo que guarda el identificador único de la pantalla a la que se aplica el permiso.
idRol	int	Atributo que guarda el identificador único del rol al que se le aplica el permiso.
Insertar	bool	Atributo que guarda si el permiso tiene acceso a la acción de crear nuevos elementos.
Modificar	bool	Atributo que guarda si el permiso tiene acceso a la acción de modificar elementos ya existentes.
Borrar	bool	Atributo que guarda si el permiso tiene acceso a la acción de borrar elementos ya existentes.
Acceder	bool	Atributo que guarda si el permiso tiene acceso a la acción de acceder a una pantalla en concreto.
IdPantallaNavigation	TPantalla	Atributo que permite navegar hacia la instancia de la pantalla que guarda.
IdRolNavigation	TRol	Atributo que permite navegar hacia la instancia del rol que guarda.

Tabla 48: Atributos de tPermiso

- **TInforme:** Clase que representa a los informes.

Atributo	Tipo	Descripción
NifMedico	string	Atributo que guarda el identificador único del médico titular del informe.
NifAlumno	string	Atributo que guarda el identificador único del alumno al que pertenece el informe.

Fecha	DateTime	Atributo que guarda la fecha de la creación del informe.
Contenido	byte[]	Atributo que guarda el contenido del informe en formato PDF.
NifAlumnoNavigation	TAlumno	Atributo que permite navegar hacia la instancia del alumno que guarda.
NifMedicoNavigation	TMedico	Atributo que permite navegar hacia la instancia del médico que guarda.

Tabla 49: Atributos de tInforme.

- **ErrorViewModel:** Clase que se encarga de revisar los errores generales de la aplicación web.

Atributo	Tipo	Descripción
RequestId	string	Atributo que se utiliza para registrar y rastrear de manera única cada solicitud que se produce en la aplicación.
ShowRequestId	bool	Atributo que determina si se debe mostrar o no el identificador de solicitud asociado con un error.

Tabla 50: Atributos de ErrorViewModel.

- **TUsuario:** Clase que representa a los usuarios.

Atributo	Tipo	Descripción
Nif	string	Atributo que guarda el identificador único de un usuario.
Usuario	string	Atributo que guarda el nombre de usuario de un usuario.
Password	string	Atributo que guarda la contraseña de un usuario.
IdRol	int	Atributo que guarda el identificador único del rol que tiene un usuario.

IdRolNavigation	TRol	Atributo que permite navegar hacia la instancia del rol que guarda.
TAdministracion	TAdministracion	Atributo que conecta a un usuario con su instancia de administrador si posee dicho rol.
TAlumno	TAlumno	Atributo que conecta a un usuario con su instancia de alumno si posee dicho rol.
TMedico	TMedico	Atributo que conecta a un usuario con su instancia de médico si posee dicho rol.
TProfesor	TProfesor	Atributo que conecta a un usuario con su instancia de profesor si posee dicho rol.

Tabla 51: Atributos de tUsuario.

- **TAlumno**: Clase que representa a los alumnos.

Atributo	Tipo	Descripción
Nif	string	Atributo que guarda el identificador único de un alumno.
Nombre	string	Atributo que guarda el nombre de un alumno.
Apellido1	string	Atributo que guarda el primer apellido de un alumno.
Apellido2	string	Atributo que guarda el segundo apellido de un alumno.
NifNavigation	TUsuario	Atributo que permite navegar hacia la instancia del usuario que guarda.
TInformes	ICollection<TInforme>	Atributo que conecta a un alumno con su conjunto de informes.
IdAsignaturas	ICollection<TAsignatura>	Atributo que conecta a un alumno con su conjunto de asignaturas.

Tabla 52: Atributos de tAlumno.

- **TProfesor:** Clase que representa a los profesores.

Atributo	Tipo	Descripción
Nif	string	Atributo que guarda el identificador único de un profesor.
Nombre	string	Atributo que guarda el nombre de un profesor.
Apellido1	string	Atributo que guarda el primer apellido de un profesor.
Apellido2	string	Atributo que guarda el segundo apellido de un profesor.
NifNavigation	TUsuario	Atributo que permite navegar hacia la instancia del usuario que guarda.
IdAsignaturas	ICollection<TAsignatura>	Atributo que conecta a un profesor con el conjunto de sus asignaturas.

Tabla 53: Atributos de tProfesor.

- **TMedico:** Clase que representa a los médicos.

Atributo	Tipo	Descripción
Nif	string	Atributo que guarda el identificador único de un médico.
Nombre	string	Atributo que guarda el nombre de un médico.
Apellido1	string	Atributo que guarda el primer apellido de un médico.
Apellido2	string	Atributo que guarda el segundo apellido de un médico.
NifNavigation	TUsuario	Atributo que permite navegar hacia la instancia del usuario que guarda.
TInformes	ICollection<TInforme>	Atributo que conecta a un médico con un conjunto de informes.

Tabla 54: Atributos de tMedico.

- **TAdministracion:** Clase que representa a los administradores.

Atributo	Tipo	Descripción
Nif	string	Atributo que guarda el identificador único de un administrador.
Nombre	string	Atributo que guarda el nombre de un administrador.
Apellido1	string	Atributo que guarda el primer apellido de un administrador.
Apellido2	string	Atributo que guarda el segundo apellido de un administrador.
NifNavigation	TUsuario	Atributo que permite navegar hacia la instancia del usuario que guarda.

Tabla 55: Atributos de tAdministracion.

- **TAsignatura:** Clase que representa a las asignaturas.

Atributo	Tipo	Descripción
Id	int	Atributo que guarda el identificador único de una asignatura.
Nombre	string	Atributo que guarda el nombre de una asignatura.
NifAlumnos	ICollection<TAlumno>	Atributo que conecta una asignatura con un conjunto de alumnos.
NifProfesores	ICollection<TProfesor>	Atributo que conecta una asignatura con un conjunto de profesores.

Tabla 56: Atributos de tAsignatura.

- **TPantalla:** Clase que representa a las pantallas.

Atributo	Tipo	Descripción
Id	int	Atributo que guarda el identificador único de una pantalla.

Nombre	string	Atributo que guarda el nombre de una pantalla.
Descripcion	string	Atributo que guarda una descripción sobre la pantalla.
TAuditoria	ICollection<TAuditorium>	Atributo que conecta una pantalla con un conjunto de auditorías.
TPermiso	ICollection<TPermiso>	Atributo que conecta una pantalla con un conjunto de permisos.

Tabla 57: Atributos de tPantalla.

- **TAuditorium:** Clase que representa a las auditorías.

Atributo	Tipo	Descripción
Id	int	Atributo que guarda el identificador único de una auditoría.
NifUsuario	string	Atributo que guarda el identificador único de un usuario.
Pantalla	int	Atributo que guarda el identificador único de una pantalla.
Accion	string	Atributo que guarda la acción que ha realizado un usuario. Aparte de la acción, contiene una descripción detallada de dicho acto.
FechaHora	DateTime	Atributo que guarda la fecha de realización de la acción.
PantallaNavigation	TPantalla	Atributo que permite navegar hacia la instancia de la pantalla que guarda.

Tabla 58: Atributos de tAuditorium.

4.4.2 Diagramas de clase UML: UserNavigation

Clase que representa a Navigation (la navegación).

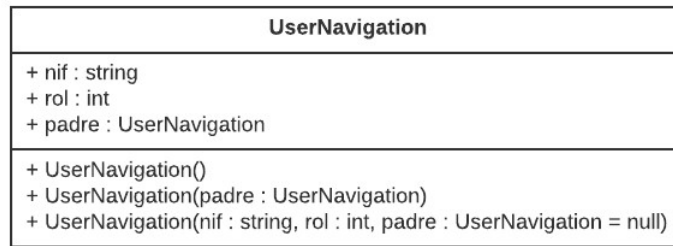


Imagen 41: Diagrama a alto nivel de la clase UserNavigation.cs.

Atributo	Tipo	Descripción
nif	string	Atributo que guarda el identificador único del usuario actual. Este atributo se irá actualizando según se vaya navegando entre los perfiles principales de los usuarios.
rol	int	Atributo que guarda el identificador único del rol del usuario actual. Este atributo se irá actualizando según se vaya navegando entre los perfiles principales de los usuarios.
padre	UserNavigation	Atributo que guarda el anterior “usuario actual”. Al iniciar sesión, este campo estará a null . Este atributo se irá actualizando según se vaya navegando entre los perfiles principales de los usuarios.

Tabla 59: Atributos de UserNavigation.

Método	Tipo	Descripción
UserNavigation()	Constructor	Constructor vacío de la clase. Es el constructor que utiliza la aplicación por defecto.
UserNavigation (padre)	Constructor	Constructor con un parámetro de la clase que lo único que hace es sustituir el atributo padre por el que se le pasa al método.

UserNavigation (nif, rol, padre)	Constructor	Constructor con tres parámetros que crea una nueva instancia completa de la clase UserNavigation.
---------------------------------------------	-------------	---------------------------------------------------------------------------------------------------

Tabla 60: Métodos de UserNavigation.

4.4.3 Diagramas de clase UML: Constantes

Clase que representa a las constantes.



Imagen 42: Diagrama a alto nivel de la clase constDefinidas.cs.

Atributo	Tipo	Descripción
rolAlumno	int	Constante que contiene el número que representa al rol de alumno. En este caso, el número sería 1.
rolProfesor	int	Constante que contiene el número que representa al rol de profesor. En este caso, el número sería 2.

rolMedico	int	Constante que contiene el número que representa al rol de médico. En este caso, el número sería 3.
rolAdmin	string	Constante que contiene el número que representa al rol de administrador. En este caso sería el 4.
keyRol	string	Constante que guarda una palabra usada para recuperar de la sesión el rol del usuario que ha iniciado sesión. Guarda la palabra “_rol”.
keyNif	string	Constante que guarda una palabra usada para recuperar de la sesión el NIF del usuario que ha iniciado sesión. Guarda la palabra “_nif”.
keyUser	string	Constante que guarda una palabra usada para recuperar de la sesión el nombre de usuario del usuario que ha iniciado sesión. Guarda la palabra “_user”.
keyActualUser	string	Constante que guarda una palabra usada para recuperar un string que representa a una instancia de la clase UserNavigation. Esta instancia es el usuario en el que nos encontramos según la lógica de UserNavigation. Guarda la palabra “_actualUser”.
keyInformePDF	string	(Deprecated)
keyMedico Temporal	string	Constante que guarda una palabra usada para recuperar el NIF del médico temporal. Guarda la palabra “Medico_Temporal”.
accionCrear	int	(Deprecated)
accionBorrar	int	(Deprecated)
accionModificar	int	(Deprecated)
accionCrear Elemento	int	Constante que contiene un número que representa la acción de crear un elemento. Guarda el número 1.
accionBorrar Elemento	int	Constante que contiene un número que representa la acción de borrar un elemento. Guarda el número 2.
screenAlumno	int	Constante que contiene un número que representa el perfil principal de un alumno. Guarda el número 1.
screenProfesor	int	Constante que contiene un número que representa el perfil principal de un profesor. Guarda el número 2.

screenMedico	int	Constante que contiene un número que representa el perfil principal de un médico. Guarda el número 3.
screen Administracion	int	Constante que contiene un número que representa el perfil principal de un administrador. Guarda el número 4.
screenInforme	int	Constante que contiene un número que representa la pantalla principal de un informe. Guarda el número 5.
screenAuditoria	int	Constante que contiene un número que representa la pantalla de las listas de auditorías. Guarda el número 6.
screenLista Alumnos	int	Constante que contiene un número que representa la pantalla de las listas de alumnos. Guarda el número 7.
screenLista Profesores	int	Constante que contiene un número que representa la pantalla de las listas de profesores. Guarda el número 8.
screenLista Medicos	int	Constante que contiene un número que representa la pantalla de las listas de médicos. Guarda el número 9.
screenLista Asignaturas	int	Constante que contiene un número que representa la pantalla de las listas de asignaturas. Guarda el número 10.
screenLista Pantallas	int	Constante que contiene un número que representa la pantalla de las listas de pantallas. Guarda el número 11.
screenLista Informes	int	Constante que contiene un número que representa la pantalla de las listas de informes. Guarda el número 12.
screenLista Matriculas	int	Constante que contiene un número que representa la pantalla de las listas de matrículas. Guarda el número 13.
screenLista Docencia	int	Constante que contiene un número que representa la pantalla de las listas de docencias. Guarda el número 15.

Tabla 61: Atributos de constDefinidas.cs.

4.4.4 Diagramas de clase UML: Interfaces

Esta interfaz es necesaria para que los métodos puedan ser utilizados por las demás clases de la aplicación web. Contiene los mismos métodos que la clase ServiceController, por lo que se desarrollará más en detalle en su apartado.

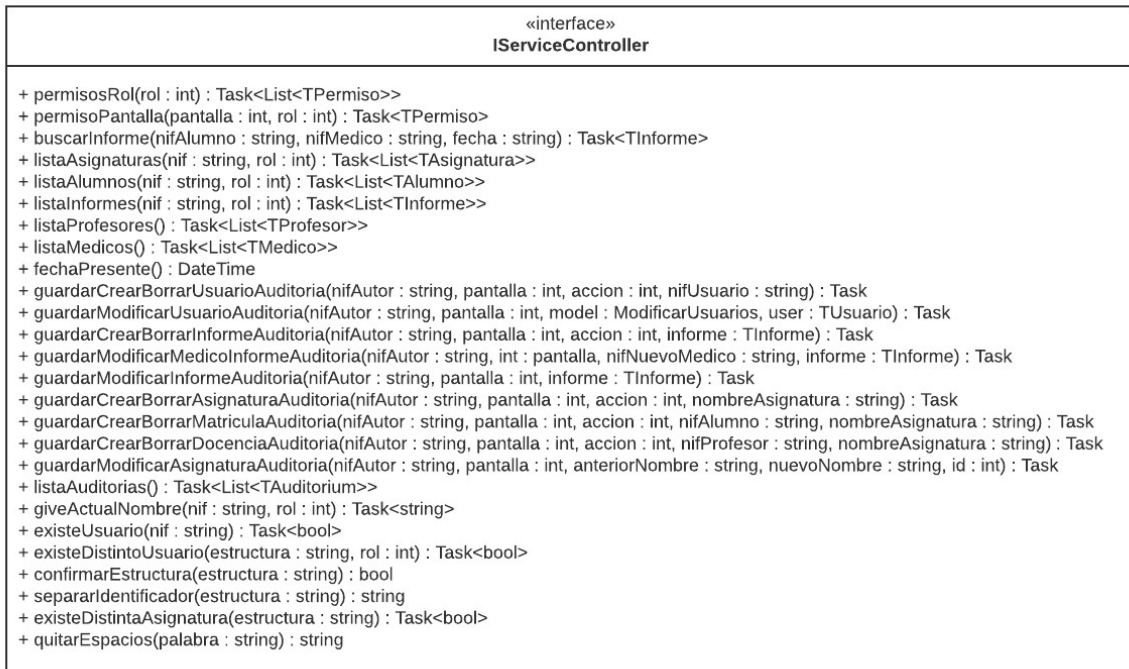


Imagen 43: Diagrama a alto nivel de la clase IServiceController.cs.

4.4.5 Diagramas de clase UML: ViewModels - Adminsitrador

Clase que representa los atributos que se les pasa a las vistas del perfil principal de un administrador.



Imagen 44: Diagrama a alto nivel de la clase AdminView.cs.

Atributo	Tipo	Descripción
Admin	TAdministracion	Atributo que guarda un usuario de tipo administrador.
Permiso	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en el perfil principal de un administrador.
LAsignaturas	TPermiso	Atributo que guarda los permisos del usuario que haya iniciado sesión sobre la pantalla de las listas de asignaturas.
LMatriculas	TPermiso	Atributo que guarda los permisos del usuario que haya iniciado sesión sobre la pantalla de las listas de matrículas.
LMedicos	TPermiso	Atributo que guarda los permisos del usuario que haya iniciado sesión sobre la pantalla de las listas de médicos.
LProfesores	TPermiso	Atributo que guarda los permisos del usuario que haya iniciado sesión sobre la pantalla de las listas de profesores.
LAlumnos	TPermiso	Atributo que guarda los permisos del usuario que haya iniciado sesión sobre la pantalla de las listas de alumnos.
LInformes	TPermiso	Atributo que guarda los permisos del usuario que haya iniciado sesión sobre la pantalla de las listas de informes.
LDocencias	TPermiso	Atributo que guarda los permisos del usuario que haya iniciado sesión sobre la pantalla de las listas de docencias.
LPantallas	TPermiso	Atributo que guarda los permisos del usuario que haya iniciado sesión sobre la pantalla de las listas de pantallas.
LAuditorías	TPermiso	Atributo que guarda los permisos del usuario que haya iniciado sesión sobre la pantalla de las listas de auditorías.
Rol	int	Atributo que guarda el rol del usuario actual.
Nif	string	Atributo que guarda el NIF del usuario actual.

SesionRol	int	Atributo que guarda el rol del usuario que haya iniciado sesión.
SesionNif	string	Atributo que guarda el NIF del usuario que haya iniciado sesión.

Tabla 62: Atributos de AdminView.cs.

4.4.6 Diagramas de clase UML: ViewModels - Alumnos

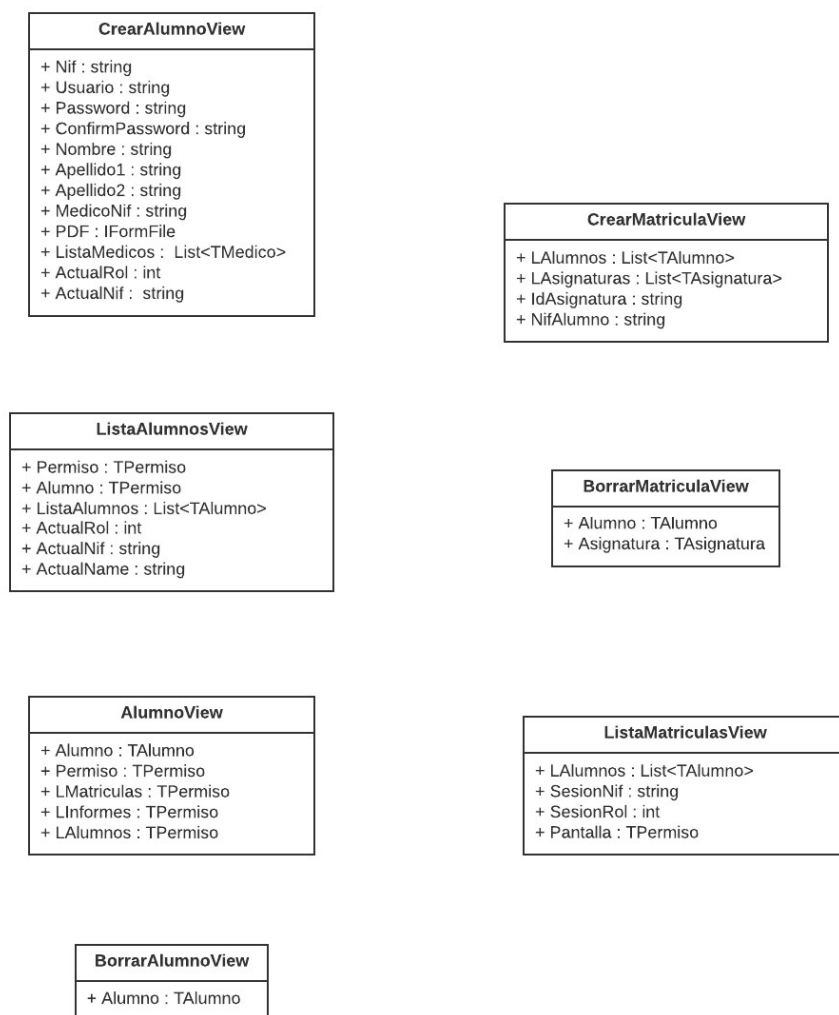


Imagen 45: Diagrama a alto nivel de las clases que forman la carpeta TAlumnos dentro de ViewModels. Corresponden a los ViewModels de los alumnos.

- **CrearAlumnoView**: Clase que representa los atributos que se les pasa a las vistas de creación de un alumno.

Atributo	Tipo	Descripción
Nif	string	Atributo que guarda el NIF del nuevo alumno a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco, ni tampoco puede usarse un NIF que ya esté ocupado por otro usuario.
Usuario	string	Atributo que guarda el nombre de usuario del nuevo alumno a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco, ni tampoco puede usarse un nombre de usuario que ya esté ocupado por otro usuario.
Password	string	Atributo que guarda la contraseña del nuevo alumno a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
ConfirmPassword	string	Atributo que sirve para confirmar la contraseña. Necesita ser igual al atributo Password para que esté correcto.
Nombre	string	Atributo que guarda el nombre del nuevo alumno a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
Apellido1	string	Atributo que guarda el primer apellido del nuevo alumno a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
Apellido2	string	Atributo que guarda el segundo apellido del nuevo alumno a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
MedicoNif	string	Atributo que guarda el NIF del médico titular del informe del nuevo alumno a crear. Este atributo no puede estar vacío, y debe elegirse desde la lista proporcionada en la vista.

PDF	IFormFile	Atributo que guarda el contenido del informe del nuevo alumno a crear. Este atributo no puede estar vacío, y solo se podrán elegir archivos en PDF.
ListaMedicos	List<TMedico>	Atributo que guarda la lista de todos los médicos disponibles.
ActualRol	int	Atributo que guarda el rol del usuario actual.
ActualNif	string	Atributo que guarda el NIF del usuario actual.

Tabla 63: Atributos de CrearAlumnoView.

- **ListaAlumnosView:** Clase que representa los atributos que se les pasa a las vistas que contienen las listas de alumnos.

Atributo	Tipo	Descripción
Permiso	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de alumnos.
Alumno	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de los perfiles principales de los alumnos.
ListaAlumnos	List<TAlumno>	Atributo que guarda una lista de alumnos. Según el usuario actual en el que nos encontremos, la lista de alumnos puede variar.
ActualRol	int	Atributo que guarda el rol del usuario actual.
ActualNif	string	Atributo que guarda el NIF del usuario actual.
ActualName	string	Atributo que guarda el nombre (no confundir con el nombre de usuario) del usuario actual.

Tabla 64: Atributos de ListaAlumnosView.

- **AlumnoView:** Clase que representa los atributos que se les pasa a las vistas del perfil principal de un alumno.

Atributo	Tipo	Descripción
Alumno	TAlumno	Atributo que guarda el alumno poseedor del perfil principal.

Permiso	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de los perfiles principales de los alumnos.
LMatriculas	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de matrículas.
LInformes	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de informes.
LAlumnos	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de alumnos.

Tabla 65: Atributos de AlumnoView.

- **BorrarAlumnoView**: Clase que representa los atributos que se les pasa a las vistas de borrado de alumnos.

Atributo	Tipo	Descripción
Alumno	TAlumno	Atributo que guarda una instancia de un usuario tipo alumno. Esta instancia es la que se va a eliminar de la base de datos si se confirma el borrado.

Tabla 66: Atributos de BorrarAlumnoView.

- **CrearMatriculaView**: Clase que representa los atributos que se les pasa a las vistas para crear nuevas matrículas.

Atributo	Tipo	Descripción
LAlumnos	List<TAlumno>	Atributo que guarda una lista con todos los alumnos.
LAsignaturas	List<TAsignatura>	Atributo que guarda una lista con todas las asignaturas.

IdAsignatura	string	Atributo que guarda la asignatura elegida en la lista de asignaturas. Este atributo no puede estar ni vacío ni formado por espacios en blanco, además debe ser seleccionada de la lista de asignaturas proporcionada en la vista.
NifAlumno	string	Atributo que guarda el alumno elegido en la lista de alumnos. Este atributo no puede estar ni vacío ni formado por espacios en blanco, además debe ser seleccionada de la lista de alumnos proporcionada en la vista.

Tabla 67: Atributos de CrearMatriculaView.

- **BorrarMatriculaView**: Clase que representa los atributos que se les pasa a las vistas para crear borrar matrículas.

Atributo	Tipo	Descripción
Alumno	TAlumno	Atributo que guarda el alumno al que le pertenece la matricula a borrar.
Asignaturas	TAsignatura	Atributo que guarda la asignatura correspondiente a la matricula a borrar.

Tabla 68: Atributos de BorrarMatriculaView.

- **ListaMatriculasView**: Clase que representa los atributos que se les pasa a las vistas que contienen las listas de las matrículas.

Atributo	Tipo	Descripción
LAlumnos	List<TAlumno>	Atributo que guarda la lista de alumnos con matrículas.
SesionNif	string	Atributo que guarda el NIF del usuario que haya iniciado sesión.
SesionRol	int	Atributo que guarda el rol del usuario que haya iniciado sesión.

Pantalla	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de matrículas.
-----------------	----------	----------------------------------------------------------------------------------------------------------------------------

Tabla 69: Atributos de ListaMatriculasView.

4.4.7 Diagramas de clase UML: ViewModels - Asignaturas

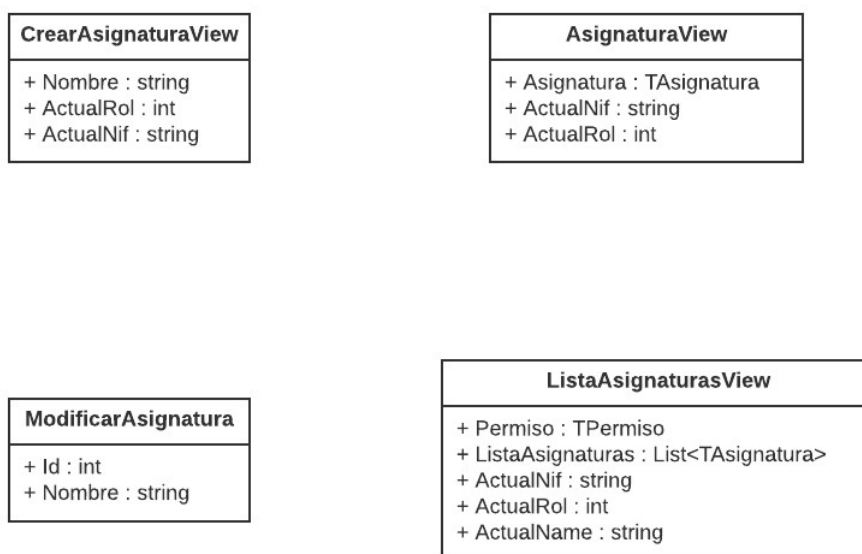


Imagen 46: Diagrama a alto nivel de las clases que forman la carpeta TAsignaturas dentro de ViewModels. Corresponden a los ViewModels de las asignaturas.

- **CrearAsignaturaView:** Clase que representa los atributos que se les pasa a las vistas para crear nuevas asignaturas.

Atributo	Tipo	Descripción
Nombre	string	Atributo que guarda el nombre de la asignatura crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco. Además, tampoco acepta nombres que ya estén usados por otras asignaturas.
ActualRol	int	Atributo que guarda el rol del usuario actual.
ActualNif	string	Atributo que guarda el NIF del usuario actual.

Tabla 70: Atributos de CrearAsignaturaView.

- **ModificarAsignatura:** Clase que representa los atributos que se les pasa a las vistas de modificación de asignaturas.

Atributo	Tipo	Descripción
Id	int	Atributo que guarda el identificador único de la asignatura que se va a modificar.
Nombre	string	Atributo que guarda el nombre de la asignatura que se va a modificar.

Tabla 71: Atributos de ModificarAsignatura.

- **AsignaturaView:** Clase que representa los atributos que se les pasa a las vistas para borrar asignaturas.

Atributo	Tipo	Descripción
Asignatura	string	Atributo que guarda el nombre de la asignatura que va a ser borrada.
ActualNif	int	Atributo que guarda el NIF del usuario actual.
ActualRol	string	Atributo que guarda el rol del usuario actual.

Tabla 72: Atributos de AsignaturaView.

- **ListaAsignaturasView:** Clase que representa los atributos que se les pasa a las vistas que contienen las listas de asignaturas.

Atributo	Tipo	Descripción
Permiso	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la vista de las listas de asignaturas.
ListaAsignaturas	List<TAsignatura>	Atributo que guarda una lista de asignaturas. Según el usuario actual en el que nos encontremos, la lista de asignaturas puede variar.
ActualNif	string	Atributo que guarda el NIF del usuario actual.
ActualRol	int	Atributo que guarda el rol del usuario actual.

ActualName	string	Atributo que guarda el nombre del usuario actual.
-------------------	--------	---------------------------------------------------

Tabla 73: Atributos de ListaAsignaturasView.

4.4.8 Diagramas de clase UML: ViewModels - Auditorías

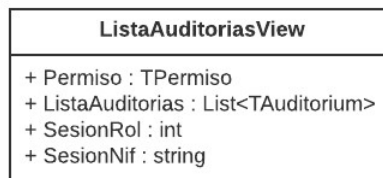


Imagen 47: Diagrama a alto nivel de la clase que forma la carpeta TAuditoriums dentro de ViewModels. Corresponde al ViewModel de las auditorías.

Atributo	Tipo	Descripción
Permiso	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de auditorías.
ListaAuditorias	List<TAuditorium>	Atributo que guarda una lista de todas las auditorías.
SesionRol	int	Atributo que guarda el rol del usuario que ha iniciado sesión.
SesionNif	string	Atributo que guarda el NIF del usuario que ha iniciado sesión.

Tabla 74: Atributos de ListaAuditoriasView.

4.4.9 Diagramas de clase UML: ViewModels - Informes

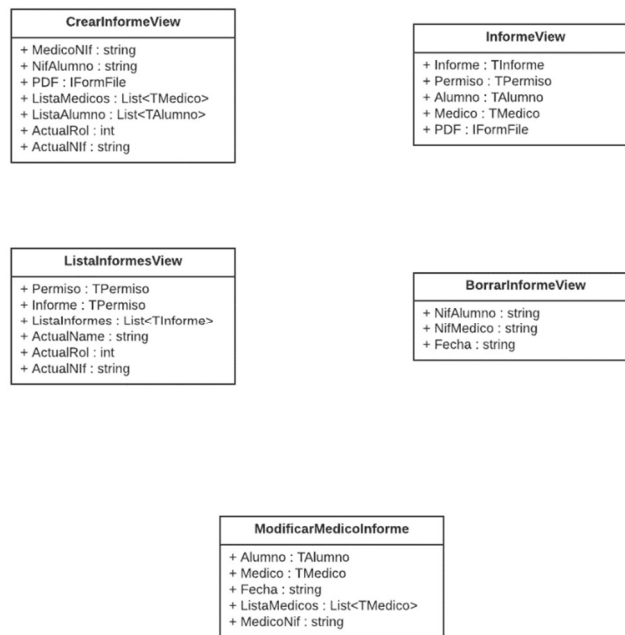


Imagen 48: Diagrama a alto nivel de las clases que forman la carpeta TInformes dentro de ViewModels. Corresponde a los ViewModels de los informes.

- **InformeView:** Clase que representa los atributos que se les pasa a las vistas que representan a las páginas principales de los informes.

Atributo	Tipo	Descripción
Informe	TInforme	Atributo que guarda un informe.
Permiso	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de la página principal del informe.
Alumno	TAlumno	Atributo que guarda el usuario de tipo alumno al cual le pertenece el informe actual.
Medico	TMedico	Atributo que guarda usuario médico el cual es médico titular del informe.
PDF	IFormFile	Atributo que guarda el contenido del informe en formato PDF.

Tabla 75: Atributos de InformeView.

- **ListaInformesView:** Clase que representa los atributos que se les pasa a las vistas que contienen las listas de informes.

Atributo	Tipo	Descripción
Permiso	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de informe.
Informe	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla principal de un informe.
ListaInformes	List<TInforme>	Atributo que guarda una lista de informes. Según el usuario actual en el que nos encontremos, la lista de informes puede variar.
ActualName	string	Atributo que guarda el nombre del usuario actual.
ActualRol	int	Atributo que guarda el rol del usuario actual.
ActualNif	string	Atributo que guarda el NIF del usuario actual.

Tabla 76: Atributos de ListaInformesView.

- **CrearInformeView:** Clase que representa los atributos que se les pasa a las vistas utilizadas para crear nuevos informes.

Atributo	Tipo	Descripción
MedicoNif	string	Atributo que guarda el médico elegido en la lista de médicos. Este atributo no puede estar ni vacío ni formado por espacios en blanco, además debe ser seleccionada de la lista de médicos proporcionada en la vista.
NifAlumno	string	Atributo que guarda el alumno elegido en la lista de alumnos. Este atributo no puede estar ni vacío ni formado por espacios en blanco, además debe ser seleccionada de la lista de alumnos proporcionada en la vista.

PDF	IFormFile	Atributo que guarda un PDF seleccionado por el usuario desde su equipo. Este atributo no puede estar vacío.
ListaMedicos	List<TMedico>	Atributo que guarda una lista de todos los médicos.
ListaAlumno	List<TAlumno>	Atributo que guarda una lista de todos los alumnos.
ActualRol	int	Atributo que guarda el rol del usuario actual.
ActualNif	string	Atributo que guarda el NIF del usuario actual.

Tabla 77: Atributos de CrearInformeView.

- **BorrarInformeView**: Clase que representa los atributos que se les pasa a las vistas utilizadas para borrar informes.

Atributo	Tipo	Descripción
NifAlumno	string	Atributo que guarda el identificador único del alumno al que le pertenece del informe.
NifMedico	string	Atributo que guarda el identificador único del médico titular del informe.
Fecha	string	Atributo que guarda la fecha de creación del informe.

Tabla 78: Atributos de BorrarInformeView.

- **ModificarMedicoInforme**: Clase que representa los atributos que se les pasa a las vistas utilizadas para modificar el médico titular de un informe.

Atributo	Tipo	Descripción
Alumno	TAlumno	Atributo que guarda el alumno al que le pertenece el informe a modificar.
Medico	TMedico	Atributo que guarda el médico titular del informe a modificar.
Fecha	string	Atributo que guarda la fecha de la creación del informe.
ListaMedicos	ListaMedicos	Atributo que guarda una lista de todos los médicos.
MedicoNif	string	Atributo que guarda el médico elegido en la lista de médicos. Este atributo no puede estar ni vacío ni

		formado por espacios en blanco. Además debe ser seleccionado de la lista de médicos proporcionada en la vista.
--	--	----------------------------------------------------------------------------------------------------------------

Tabla 78: Atributos de BorrarInformeView.

4.4.10 Diagramas de clase UML: ViewModels - Médico

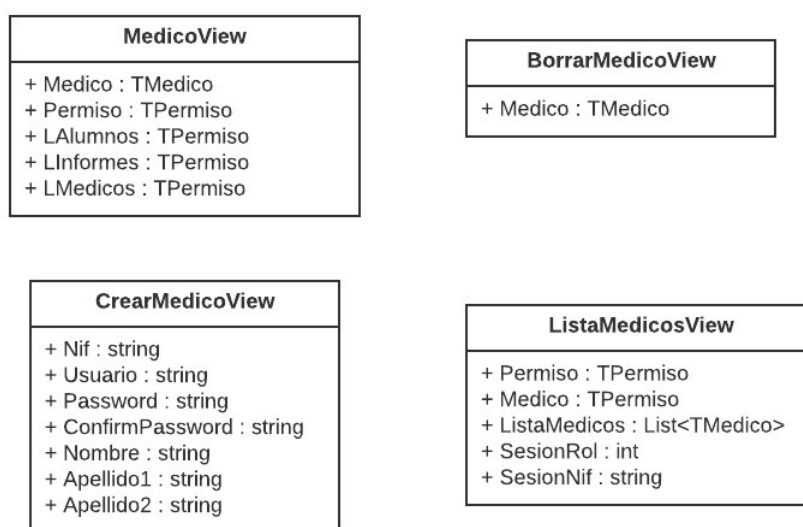


Imagen 49: Diagrama a alto nivel de las clases que forman la carpeta TMedicos dentro de ViewModels. Corresponde a los ViewModels de los médicos.

- **MedicoView:** Clase que representa los atributos que se les pasa a las vistas del perfil principal de un médico.

Atributo	Tipo	Descripción
Medico	TMedico	Atributo que guarda un médico.
Permiso	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla del perfil principal de un médico.
LAlumnos	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de alumnos.

LInformes	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de informes.
LMedicos	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de médicos.

Tabla 79: Atributos de MedicoView.

- **CrearMedicoView**: Clase que representa los atributos que se les pasa a las vistas de creación de un médico.

Atributo	Tipo	Descripción
Nif	string	Atributo que guarda el NIF del nuevo médico a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco, ni tampoco puede usarse un NIF que ya esté ocupado por otro usuario.
Usuario	string	Atributo que guarda el nombre de usuario del nuevo médico a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco, ni tampoco puede usarse un nombre de usuario que ya esté ocupado por otro usuario.
Password	string	Atributo que guarda la contraseña del nuevo médico a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
ConfirmPassword	string	Atributo que sirve para confirmar la contraseña. Necesita ser igual al atributo Password para que esté correcto.
Nombre	string	Atributo que guarda el nombre del nuevo médico a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.

Apellido1	string	Atributo que guarda el primer apellido del nuevo médico a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
Apellido2	string	Atributo que guarda el segundo apellido del nuevo médico a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.

Tabla 80: Atributos de CrearMedicoView.

- **BorrarMedicoView:** Clase que representa los atributos que se les pasa a las vistas utilizadas para borrar médicos.

Atributo	Tipo	Descripción
Medico	TMedico	Médico que va a ser borrado.

Tabla 81: Atributos de BorrarMedicoView.

- **ListaMedicoView:** Clase que representa los atributos que se les pasa a las vistas que contienen a las listas de médicos.

Atributo	Tipo	Descripción
Permiso	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de médicos.
Medico	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla del perfil principal de un médico.
ListaMedicos	List<TMedico>	Atributo que guarda una lista de médicos. Según el usuario actual en el que nos encontremos, la lista de médicos puede variar.
SesionRol	string	Atributo que guarda el rol del usuario que ha iniciado sesión.
SesionNif	int	Atributo que guarda el NIF del usuario que ha iniciado sesión.

Tabla 82: Atributos de ListaMedicoView.

4.4.11 Diagramas de clase UML: ViewModels - Pantallas



Imagen 50: Diagrama a alto nivel de la clase que forma la carpeta TPantallas dentro de ViewModels. Corresponde al ViewModel de las pantallas.

Atributo	Tipo	Descripción
ListaPantallas	List<TPantalla>	Atributo que guarda la lista de todas las pantallas.
SesionRol	int	Atributo que guarda el rol del usuario que ha iniciado sesión.
SesionNif	string	Atributo que guarda el NIF del usuario que ha iniciado sesión.

Tabla 83: Atributos de ListaPantallasView.

4.4.12 Diagramas de clase UML: ViewModels - Profesores

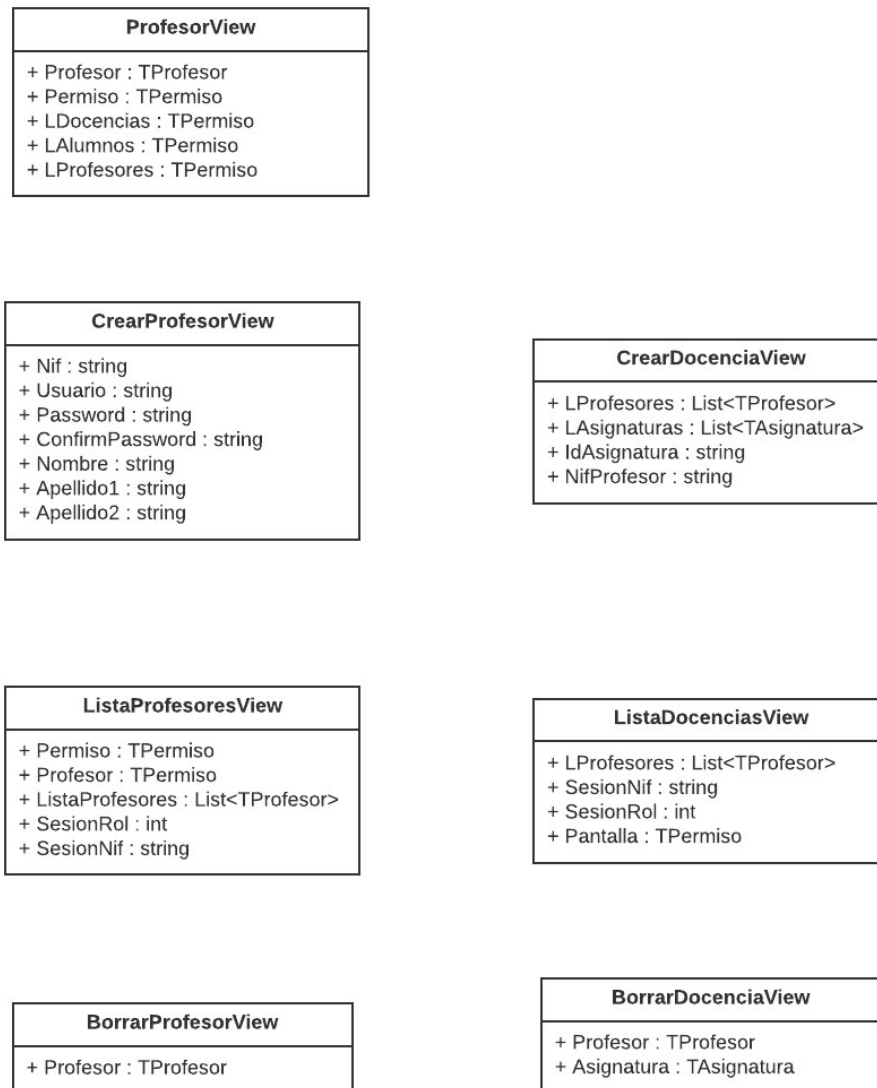


Imagen 51: Diagrama a alto nivel de las clases que forman la carpeta TProfesores dentro de ViewModels. Corresponde a los ViewModels de los profesores.

- **ProfesorView:** Clase que representa los atributos que se les pasa a las vistas del perfil principal de un profesor.

Atributo	Tipo	Descripción
Profesor	TProfesor	Atributo que guarda el profesor en cuyo perfil principal nos encontramos

Permiso	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla del perfil principal de un profesor.
LDocencias	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de docencias.
LAlumnos	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de alumnos.
LProfesores	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de profesores.

Tabla 84: Atributos de ProfesorView.

- **CrearProfesorView**: Clase que representa los atributos que se les pasa a las vistas de creación de un profesor.

Atributo	Tipo	Descripción
Nif	string	Atributo que guarda el NIF del nuevo profesor a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco, ni tampoco puede usarse un NIF que ya esté ocupado por otro usuario.
Usuario	string	Atributo que guarda el nombre de usuario del nuevo profesor a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco, ni tampoco puede usarse un nombre de usuario que ya esté ocupado por otro usuario.
Password	string	Atributo que guarda la contraseña del nuevo profesor a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.

ConfirmPassword	string	Atributo que sirve para confirmar la contraseña. Necesita ser igual al atributo Password para que esté correcto.
Nombre	string	Atributo que guarda el nombre del nuevo profesor a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
Apellido1	string	Atributo que guarda el primer apellido del nuevo profesor a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
Apellido2	string	Atributo que guarda el segundo apellido del nuevo profesor a crear. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.

Tabla 85: Atributos de CrearProfesorView.

- **ListaProfesoresView**: Clase que representa los atributos que se les pasa a las vistas que contienen a las listas de profesores.

Atributo	Tipo	Descripción
Permiso	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de profesores.
Profesor	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de los perfiles principales de los profesores.
ListaProfesores	List<TProfesor>	Atributo que guarda una lista de todos los profesores.
SesionRol	int	Atributo que guarda el rol del usuario que ha iniciado sesión.
SesionNif	string	Atributo que guarda el NIF del usuario que ha iniciado sesión.

Tabla 86: Atributos de ListaProfesoresView.

- **BorrarProfesorView:** Clase que representa los atributos que se les pasa a las vistas utilizadas para borrar profesores.

Atributo	Tipo	Descripción
Profesor	TProfesor	Atributo que guarda el profesor a borrar.

Tabla 87: Atributos de BorrarProfesorView.

- **CrearDocenciaView:** Clase que representa los atributos que se les pasa a las vistas encargadas de la creación de las docencias.

Atributo	Tipo	Descripción
LProfesores	List<TProfesor>	Atributo que guarda una lista con todos los profesores.
LAsignaturas	List<TAsignatura>	Atributo que guarda una lista con todas las asignaturas.
IdAsignatura	string	Atributo que guarda la asignatura elegida en la lista de asignaturas. Este atributo no puede estar ni vacío ni formado por espacios en blanco. Además, debe ser seleccionada de la lista de asignaturas proporcionada en la vista.
NifProfesor	string	Atributo que guarda el profesor elegido en la lista de profesores. Este atributo no puede estar ni vacío ni formado por espacios en blanco. Además, debe ser seleccionada de la lista de profesores proporcionada en la vista.

Tabla 88: Atributos de CrearDocenciaView.

- **ListaDocenciasView:** Clase que representa los atributos que se les pasa a las vistas que contienen a las listas de docencias.

Atributo	Tipo	Descripción
LProfesores	List<TProfesor>	Atributo que guarda una lista con todos los profesores.

SesionNif	string	Atributo que guarda el NIF del usuario que ha iniciado sesión.
SesionRol	int	Atributo que guarda el rol del usuario que ha iniciado sesión.
Pantalla	TPermiso	Atributo que guarda los permisos que tiene el usuario que haya iniciado sesión en la pantalla de las listas de docencias.

Tabla 89: Atributos de ListaDocenciaView.

- **BorrarDocenciaView:** Clase que representa los atributos que se les pasa a las vistas utilizadas para borrar docencias.

Atributo	Tipo	Descripción
Profesor	TProfesor	Atributo que guarda un profesor que cursa esa asignatura.
Asignatura	TAsignatura	Atributo que guarda una asignatura perteneciente a esa docencia.

Tabla 90: Atributos de BorrarDocenciaView.

4.4.13 Diagramas de clase UML: ViewModels – Modificación de usuarios

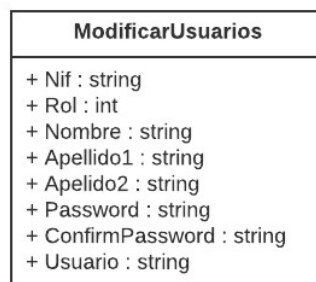


Imagen 52: Diagrama a alto nivel de la clase que forma la carpeta que corresponde a la clase encargada de pasar los atributos a las vistas encargadas de la modificación de los usuarios.

Atributo	Tipo	Descripción
Nif	string	Atributo que guarda el NIF del usuario a modificar. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco, ni tampoco puede usarse un NIF que ya esté ocupado por otro usuario.
Rol	string	Atributo que guarda el rol del usuario a modificar.
Nombre	string	Atributo que guarda el nombre del usuario a modificar. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
Apellido1	string	Atributo que guarda el primer apellido del usuario a modificar. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
Apellido2	string	Atributo que guarda el segundo apellido del usuario a modificar. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
Password	string	Atributo que guarda la contraseña del usuario a modificar. Este atributo no puede estar ni vacío ni formado solo por espacios en blanco.
ConfirmPassword	string	Atributo que sirve para confirmar la contraseña. Necesita ser igual al atributo Password para que esté correcto.
Usuario	string	Atributo que guarda el nombre de usuario del usuario a modificar. Este atributo no puede estar vacío, ni formado por espacios en blanco.

Tabla 91: Atributos de ModificarUsuarios.

4.4.14 Diagramas de clase UML: Controladores



Imagen 53: Diagrama a alto nivel de las clases que forman la carpeta Controllers, los cuales representan a los controladores de la aplicación web.

- **ServiceController:** Clase encargada de albergar métodos útiles utilizados por las demás clases.

Atributo	Tipo	Descripción
<code>_context</code>	TfgContext	Atributo que guarda una instancia de la base de datos a la que se conecta la aplicación web.

Tabla 92: Atributos de ServiceController.

Método	Tipo	Descripción
permisosRol	Task<List<TPermiso>>	Método que devuelve una lista de permisos según el rol que se le pase a la función.
permisoPantalla	Task<TPermiso>	Método que devuelve un permiso de un rol en una pantalla en concreto.
buscarInforme	Task<TInforme>	Método que busca un informe según el médico, alumno y fecha pasados.
listaAsignaturas	Task<List<TAsignatura>>	Método que devuelve una lista de asignaturas según el rol y el NIF del usuario.
listaAlumnos	Task<List<TAlumno>>	Método que devuelve una lista de alumnos según el rol y el NIF del usuario.
listaInformes	Task<List<TInforme>>	Método que devuelve una lista de informes según el rol y el NIF del usuario.
listaProfesores	Task<List<TProfesor>>	Método que devuelve una lista de todos los profesores
listaMedicos	Task<List<TMedico>>	Método que devuelve una lista de todos los médicos.
fechaPresente	DateTime	Método que devuelve la fecha y hora del momento en el que se llama a la función.
guardarCrearBorrar UsuarioAuditoria	Task	Método que guarda la creación o borrado de un usuario como nueva auditoría.
guardarModificar UsuarioAuditoria	Task	Método que guarda la modificación de un usuario como auditoría.

guardarCrearBorrar InformeAuditoria	Task	Método que guarda la creación o borrado de un informe como nueva auditoría.
guardarModificar MedicoInforme Auditoria	Task	Método que guarda el cambio del médico titular de un informe como nueva auditoría.
guardarModificar InformeAuditoria	Task	Método que guarda la modificación de un informe como auditoría.
guardarCrearBorrar Asignatura Auditoria	Task	Método que guarda la creación o borrado de una asignatura como nueva auditoría.
guardarCrearBorrar MatriculaAuditoria	Task	Método que guarda la creación o borrado de una matrícula como nueva auditoría.
guardarCrearBorrar DocenciaAuditoria	Task	Método que guarda la creación o borrado de una docencia como nueva auditoría.
guardarModificar Asignatura Auditoria	Task	Método que guarda la modificación del nombre de una asignatura como nueva auditoría.
listaAuditorias	Task<List<TAuditorium>>	Método que devuelve una lista de todas las auditorías.
giveActualNombre	Task<string>	Método que devuelve el nombre del usuario cuyo NIF y rol se correspondan con los pasados al método.
existeUsuario	Task<bool>	Método que confirma la existencia de un usuario con el NIF que se le pasa.
existeDistinto Usuario	Task<bool>	Método que confirma la existencia de un usuario con el rol y la estructura que se le pasa. Este método se usa

		normalmente para la selección de un usuario de una lista desplegable.
confirmar Estructura	bool	Método que confirma que el string que se le pasa tiene la estructura deseada. En este caso sería: (nombre) (identificador único).
separar Identificador	string	Método que devuelve el identificador único de la estructura del método “ confirmarEstructura ”.
existeDistinta Asignatura	Task<bool>	Método que confirma si existe o no una asignatura a partir de la estructura pasada al método. Normalmente se utiliza en la selección de una asignatura en una lista desplegable.
quitarEspacios	string	Método que quita todos los espacios en blanco al principio y al final del string que se le pasa.

Tabla 93: Métodos de ServiceController.

- **TUsuariosController:** Clase que representa al controlador que maneja los métodos que tengan que ver con los usuarios, es decir, las instancias del tipo TUsuario.

Atributo	Tipo	Descripción
_context	TfgContext	Atributo que guarda una instancia de la base de datos a la que se ha conectado la aplicación web.
_serviceController	IServiceController	Atributo que guarda una instancia de la clase IServiceController , que es la encargada de almacenar funciones para que se utilicen de forma global en todo el proyecto.

Tabla 94: Atributos de TUsuariosController.

Método	Tipo	Descripción
InicioSesion	IActionResult	Método encargado de que al iniciar la aplicación web, la primera vista en mostrarse sea la de iniciar sesión.
login	IActionResult	Método encargado del inicio de sesión. Si el usuario y contraseña son correctos, se redirigirá al perfil principal del usuario con esas credenciales. En otro caso, se mostrará alguna advertencia señalando un problema en el inicio de sesión.
giveSesionNif	string	Método que devuelve el NIF del usuario que ha iniciado sesión.
giveActualUser	UserNavigation	Método que devuelve una instancia de la clase UserNavigation . Se utiliza sobre todo para comprobar en qué perfil principal nos encontramos.
volverPerfil	IActionResult	Método que redirige al usuario a su perfil principal. Utilizado sobre todo para el botón " <i>Perfil Principal</i> " en la barra de navegación.
verficarNif	Task<IActionResult>	Método que verifica que no se repita un NIF en un usuario. Se utiliza sobre todo en la creación de usuarios para asegurarnos la singularidad del NIF.
verificarModificarNombreUsuario	Task<IActionResult>	Método que verifica que no se repita un nombre de usuario en un usuario. Se utiliza sobre todo en la modificación de usuarios para

		asegurarnos la singularidad del nombre de usuario.
verificarCrearNombreUsuario	Task<IActionResult>	Método que verifica que no se repita un nombre de usuario en un usuario. Se utiliza sobre todo en la creación de usuarios para asegurarnos la singularidad del nombre de usuario.
verificarAlumno	Task<IActionResult>	Método que verifica tanto la estructura como la existencia de un alumno elegido en una lista desplegable.
verificarProfesor	Task<IActionResult>	Método que verifica tanto la estructura como la existencia de un profesor elegido en una lista desplegable.
verificarMedico	Task<IActionResult>	Método que verifica tanto la estructura como la existencia de un médico elegido en una lista desplegable.
verificarEEBNombre	IActionResult	Método que verifica que el campo Nombre de un usuario no esté conformado solo de espacios en blanco.
verificarEBBApellido1	IActionResult	Método que verifica que el campo Apellido1 de un usuario no esté conformado solo de espacios en blanco.
verificarEBBApellido2	IActionResult	Método que verifica que el campo Apellido2 de un usuario no esté conformado solo de espacios en blanco.

verificarEBB Password	IActionResult	Método que verifica que el campo Password de un usuario no esté conformado solo de espacios en blanco.
crearUsuario Profesor	Task<IActionResult>	Método que se encarga de crear un nuevo usuario para un nuevo profesor.
crearUsuario Alumno	Task<IActionResult>	Método que se encarga de crear un nuevo usuario para un nuevo alumno.
crearUsuario Medico	Task<IActionResult>	Método que se encarga de crear un nuevo usuario para un nuevo médico.
verificarCambios	Task<bool>	Método que verifica si se ha cambiado algún elemento en la vista de modificación de un elemento.
confirmarCambios	Task<IActionResult>	Método que guarda las modificaciones realizadas a un usuario.
cerrarSesion	IActionResult	Método encargado de cerrar sesión. Redirigirá al usuario a la vista donde se inicia sesión. Tanto el UserNavigation como la Sesión se vaciarán.
volverPerfil Principal	IActionResult	Método encargado de devolver al usuario a su perfil principal. Utiliza el método " <i>volverPerfil</i> " para ello, y es el utilizado por el botón "Perfil Principal" en la barra de navegación.

Tabla 95: Métodos de TUsuariosController.

- **TInformesController:** Clase que representa al controlador que maneja los métodos que tengan que ver con los informes, es decir, las instancias del tipo TInforme.

Atributo	Tipo	Descripción
_context	TfgContext	Atributo que guarda una instancia de la base de datos a la que se ha conectado la aplicación web.
_serviceController	IServiceController	Atributo que guarda una instancia de la clase IServiceController , que es la encargada de almacenar funciones que se utilicen de forma global en todo el proyecto.

Tabla 96: Atributos de TInformesController.

Método	Tipo	Descripción
giveSesionRol	int	Método que devuelve el rol del usuario que ha iniciado sesión.
giveSesionNif	string	Método que devuelve el NIF del usuario que ha iniciado sesión.
giveActualUser	UserNavigation	Método que devuelve una instancia de la clase UserNavigation . Se utiliza sobre todo para comprobar en qué perfil se encuentra el usuario.
listaInformes	Task<IActionResult>	Método que redirige a un usuario hacia la vista que contiene las listas de informes. Según el usuario actual, las listas de informes podrían variar.
infoBasica	Task<IActionResult>	Método que redirige al usuario hacia la vista que contiene los datos básicos de un informe.
verInforme	Task<IActionResult>	Método encargado de abrir el contenido de un informe en formato PDF en una nueva pestaña del navegador.
ActualizarPDF	Task<IActionResult>	Método encargado de modificar el contenido de un informe. Solo permitirá archivos en formato PDF, y

		si no se ha seleccionado ningún archivo, mostrará una advertencia sobre ello.
crearInformeNuevo	Task<IActionResult>	Método que guarda un nuevo informe creado.
insertarInforme	Task<IActionResult>	Método que redirige a un usuario a la vista encargada de crear un nuevo informe.
borrarInforme	Task<IActionResult>	Método que redirige a un usuario a la vista encargada de confirmar el borrado de un informe.
confirmar BorradoInforme	Task<IActionResult>	Método que borra un informe de la base de datos de forma definitiva.
elegirMedico Informe	Task<IActionResult>	Método encargado de redirigir al usuario a la vista para cambiar el médico titular de un informe. Si no existen médicos registrados a parte del "Médico_Temporal", se notificará un mensaje advirtiendo la situación.
confirmarCambio MedicoInforme	Task<IActionResult>	Método que guarda los cambios en el médico titular de un informe en la base de datos.

Tabla 97: Métodos de TInformesController.

- **TAsignaturasController**: Clase que representa al controlador que maneja los métodos que tengan que ver con las asignaturas, es decir, las instancias del tipo TAsignatura.

Atributo	Tipo	Descripción
_context	TfgContext	Atributo que guarda una instancia de la base de datos a la que se ha conectado la aplicación web.

_serviceController	IServiceController	Atributo que guarda una instancia de la clase IServiceController , que es la encargada de almacenar funciones que se utilicen de forma global en todo el proyecto.
---------------------------	--------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 98: Atributos de TAsignaturasController.

Método	Tipo	Descripción
giveSesionRol	int	Método que devuelve el rol del usuario que ha iniciado sesión.
giveSesionNif	string	Método que devuelve el NIF del usuario que ha iniciado sesión.
giveActualUser	UserNavigation	Método que devuelve una instancia de la clase UserNavigation . Se utiliza sobre todo para comprobar en que perfil se encuentra el usuario.
listaAsignaturas	Task<IActionResult>	Método que redirige a un usuario hacia la vista que contiene las listas de asignaturas. Según el usuario actual, las listas de asignaturas podrían variar.
modificar Asignatura	Task<IActionResult>	Método que redirige al usuario hacia a vista encargada de modificar una asignatura.
nombres Asignatura	Task<List<string>>	Método que devuelve una lista con los nombres de todas las asignaturas.
insertarAsignatura	IActionResult	Método que redirige al usuario hacia la vista encargada de crear una nueva asignatura.
crearAsignatura	Task<IActionResult>	Método que guarda una nueva asignatura en la base de datos.
verificarAsignatura	Task<IActionResult>	Método que verifica tanto la existencia como la estructura de una

		asignatura elegida desde una lista desplegable.
verificarNombre Asignatura	Task<IActionResult>	Método que verifica si el nombre para crear una nueva asignatura ya está siendo utilizado por otra.
verificarModificar Asignatura	Task<IActionResult>	Método que verifica si el nombre para modificar una asignatura ya está siendo utilizado por otra.
confirmarCambio	Task<IActionResult>	Método que guarda las modificaciones realizadas a una asignatura en la base de datos.
borrarAsignatura	Task<IActionResult>	Método que redirige a un usuario a la vista encargada de confirmar el borrado de una asignatura.
confirmarBorrado	Task<IActionResult>	Método que borra una asignatura de la base de datos de forma permanente.

Tabla 99: Métodos de TAsignaturasController.

- **TAdministracionesController:** Clase que representa al controlador que maneja los métodos que tengan que ver con los administradores, es decir, las instancias del tipo TAdministracion.

Atributo	Tipo	Descripción
_context	TfgContext	Atributo que guarda una instancia de la base de datos a la que se ha conectado la aplicación web.
_serviceController	IServiceController	Atributo que guarda una instancia de la clase IServiceController , que es la encargada de almacenar funciones que se utilicen de forma global en todo el proyecto.

Tabla 100: Atributos de TAdministracionesController.

Método	Tipo	Descripción
giveSesionRol	int	Método que devuelve el rol del usuario que ha iniciado sesión.
giveSesionNif	string	Método que devuelve el NIF del usuario que ha iniciado sesión
giveActualUser	UserNavigation	Método que devuelve una instancia de la clase UserNavigation . Se utiliza sobre todo para comprobar en qué perfil se encuentra el usuario
infoBasica	Task<IActionResult>	Método que redirige a un usuario hacia el perfil principal de un administrador.

Tabla 101: Métodos de TAdministracionesController.

- **TAuditoriumsController**: Clase que representa al controlador que maneja los métodos que tengan que ver con las auditorías, es decir, las instancias del tipo TAuditorium.

Atributo	Tipo	Descripción
_context	TfgContext	Atributo que guarda una instancia de la base de datos a la que se ha conectado la aplicación web.
_serviceController	IServiceController	Atributo que guarda una instancia de la clase IServiceController , que es la encargada de almacenar funciones que se utilicen de forma global en todo el proyecto.

Tabla 102: Atributos de TAuditoriumsController.

Método	Tipo	Descripción
giveSesionRol	int	Método que devuelve el rol del usuario que ha iniciado sesión.
giveSesionNif	string	Método que devuelve el NIF del usuario que ha iniciado sesión.

listaAuditorias	Task<IActionResult>	Método que redirige a un administrador a la vista que contiene la lista de todas las auditorías.
------------------------	---------------------	--------------------------------------------------------------------------------------------------

Tabla 103: Métodos de TAuditoriumsController.

- **TPantallasController:** Clase que representa al controlador que maneja los métodos que tengan que ver con las pantallas, es decir, las instancias del tipo TPantalla.

Atributo	Tipo	Descripción
_context	TfgContext	Atributo que guarda una instancia de la base de datos a la que se ha conectado la aplicación web.

Tabla 104: Atributos de TPantallasController.

Método	Tipo	Descripción
giveSesionRol	int	Método que devuelve el rol del usuario que ha iniciado sesión.
giveSesionNif	string	Método que devuelve el NIF del usuario que ha iniciado sesión.
listaPantalla	Task<IActionResult>	Método que redirige a un administrador a la vista que contiene la lista de todas las pantallas.

Tabla 105: Atributos de TPantallasController.

- **TAlumnosController:** Clase que representa al controlador que maneja los métodos que tengan que ver con los alumnos, es decir, las instancias del tipo TAlumno.

Atributo	Tipo	Descripción
_context	TfgContext	Atributo que guarda una instancia de la base de datos a la que se ha conectado la aplicación web.

_serviceController	IServiceController	Atributo que guarda una instancia de la clase IServiceController , que es la encargada de almacenar funciones que se utilicen de forma global en todo el proyecto.
---------------------------	--------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 106: Atributos de TAlumnosController.

Método	Tipo	Descripción
giveSesionRol	int	Método que devuelve el rol del usuario que ha iniciado sesión.
giveSesionNif	string	Método que devuelve el NIF del usuario que ha iniciado sesión.
giveActualUser	UserNavigation	Método que devuelve una instancia de la clase UserNavigation . Se utiliza sobre todo para comprobar en qué perfil se encuentra el usuario.
infoBasica	Task<IActionResult>	Método encargado de redirigir a un usuario al perfil principal de un alumno.
listaAlumnos	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de mostrar las listas de alumnos. Según el usuario actual, la lista de alumnos podría variar.
insertarAlumno	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de crear un nuevo alumno.
listaMatriculas	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de mostrar las listas de matrículas. Dependiendo del usuario actual, la lista de matrículas podría variar.

insertarMatricula	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de crear una nueva matrícula.
crearMatricula	Task<IActionResult>	Método que guarda una nueva matrícula en la base de datos. Si la matrícula en cuestión ya existe, se notificará un mensaje informando de ello y se cancelará la creación de la matrícula.
modificarAlumno	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de modificar un alumno.
borrarMatricula	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de confirmar el borrado de una matrícula.
confirmarBorrado Matricula	Task<IActionResult>	Método encargado de borrar una matrícula de forma definitiva de la base de datos.
borrarAlumno	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de confirmar el borrado de un alumno.
confirmarBorrado Alumno	Task<IActionResult>	Método encargado de borrar un alumno de forma definitiva de la base de datos.

Tabla 107: Métodos de TAlumnosController.

- **TProfesoresController:** Clase que representa al controlador que maneja los métodos que tengan que ver con los profesores, es decir, las instancias del tipo TProfesor.

Atributo	Tipo	Descripción
_context	TfgContext	Atributo que guarda una instancia de la base de datos a la que se ha conectado la aplicación web.
_serviceController	IServiceController	Atributo que guarda una instancia de la clase IServiceController , que es la encargada de almacenar funciones que se utilicen de forma global en todo el proyecto.

Tabla 108: Atributos de TProfesoresController.

Método	Tipo	Descripción
giveSesionRol	int	Método que devuelve el rol del usuario que ha iniciado sesión.
giveSesionNif	string	Método que devuelve el NIF del usuario que ha iniciado sesión.
giveActualUser	UserNavigation	Método que devuelve una instancia de la clase UserNavigation . Se utiliza sobre todo para comprobar en qué perfil se encuentra el usuario.
infoBasica	Task<IActionResult>	Método encargado de redirigir a un usuario al perfil principal de un profesor.
listaProfesores	Task<IActionResult>	Método encargado de redirigir a un administrador a la vista encargada de mostrar la lista de todos los profesores.
listaDocencias	Task<IActionResult>	Método encargado de redirigir a un administrador a la vista encargada de mostrar la lista de todas las docencias.
insertarDocencia	Task<IActionResult>	Método encargado de redirigir a un administrador a la vista encargada de crear nuevas docencias.

crearDocencia	Task<IActionResult>	Método que guarda una nueva docencia creada. Si la docencia en cuestión ya existe, se notificará con un mensaje, cancelando la creación de la docencia.
modificarProfesor	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de modificar un profesor.
insertarProfesor	IActionResult	Método encargado de redirigir a un usuario a la vista encargada de crear nuevos profesores.
crearProfesor	Task<IActionResult>	Método que guarda la creación de un nuevo profesor en la base de datos.
borrarDocencia	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de borrar una docencia.
confirmarBorrado Docencia	Task<IActionResult>	Método que borra de forma definitiva una docencia de la base de datos.
borrarProfesor	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de borrar un profesor.
confirmarBorrado Profesor	Task<IActionResult>	Método que borra de forma definitiva un profesor de la base de datos.

Tabla 109: Métodos de TProfesoresController.

- **TMedicosController**

Atributo	Tipo	Descripción
_context	TfgContext	Atributo que guarda una instancia de la base de datos a la que se ha conectado la aplicación web.
_serviceController	IServiceController	Atributo que guarda una instancia de la clase IServiceController, que es la encargada de

		almacenar funciones que se utilicen de forma global en todo el proyecto.
--	--	--------------------------------------------------------------------------

Tabla 110: Atributos de TMedicosController.

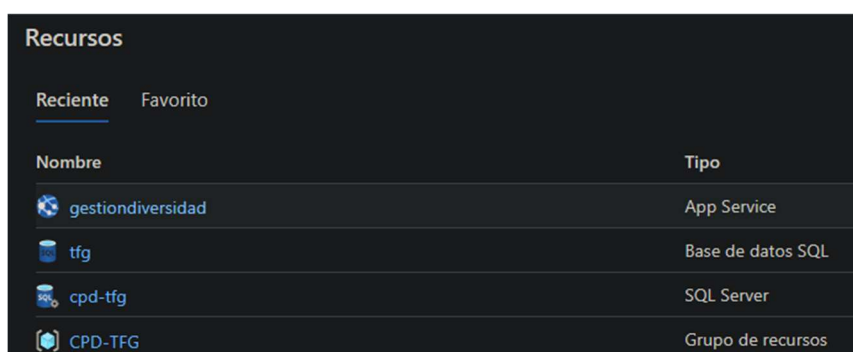
Método	Tipo	Descripción
giveSesionRol	int	Método que devuelve el rol del usuario que ha iniciado sesión.
giveSesionNif	string	Método que devuelve el NIF del usuario que ha iniciado sesión.
giveActualUser	UserNavigation	Método que devuelve una instancia de la clase UserNavigation . Se utiliza sobre todo para comprobar en qué perfil se encuentra el usuario.
infoBasica	Task<IActionResult>	Método encargado de redirigir a un usuario al perfil principal de un médico.
listaMedicos	Task<IActionResult>	Método encargado de redirigir a un administrador a la vista encargada de mostrar la lista de todos los médicos.
insertarMedico	IActionResult	Método encargado de redirigir a un usuario a la vista encargada de crear nuevos médicos.
crearMedico	Task<IActionResult>	Método que guarda la creación de un nuevo médico en la base de datos.
modificarMedico	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de modificar un médico.
borrarMedico	Task<IActionResult>	Método encargado de redirigir a un usuario a la vista encargada de borrar un médico.
confirmarBorrado Medico	Task<IActionResult>	Método que borra de forma definitiva un médico de la base de datos.

Tabla 111: Métodos de TMedicosController.

4.5 Implementación del web service

Se ha empleado los servicios de Azure para realizar la implementación web de la aplicación web. Para mayor detalle, se explicarán las partes utilizadas:

1. **Grupo de recursos:** Los grupos de recursos en Azure, son contenedores lógicos que almacenan los recursos relacionados con una solución o proyecto de Azure. Estos grupos permiten administrar, asignar y aplicar recursos, como las bases de datos o el servicio web.
2. **Servidor SQL y Base de datos SQL:** Azure SQL es una familia de productos que usan el motor de SQL Server. Esto permite un uso más sencillo de la base de datos de forma organizada y sencilla: Se crea un servidor para base de datos SQL (el servidor SQL), y en ese servidor se podrán almacenar distintas bases de datos SQL.
3. **Servicio web:** Un servicio web es un sistema de software diseñado para soportar la interacción interoperativa de máquina a máquina a través de una red. Gracias a este servicio de Azure, es posible hacer que la aplicación web esté en el Internet, y es posible acceder a él gracias al enlace creado para acceder a su dominio: gestiondiversidad.azurewebsites.net



Recursos	
Nombre	Tipo
gestiondiversidad	App Service
tfg	Base de datos SQL
cpd-tfg	SQL Server
CPD-TFG	Grupo de recursos

Imagen 54: Imagen de los servicios utilizados para la aplicación web.

Una vez conseguido los servicios necesarios para tener la aplicación web en la nube, solo hace falta publicarlo. Para ello, lo único que hizo falta fue dirigirse al proyecto y realizar dicha acción con los recursos creados con anterioridad.

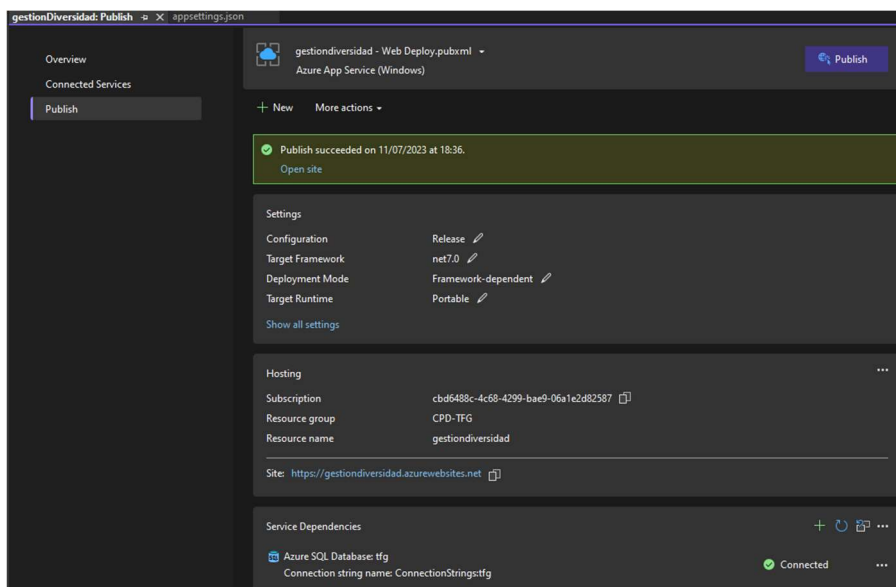


Imagen 55: Imagen de la publicación de la aplicación web mediante Visual Studio.

4.6 Pruebas y testeos

Aunque las pruebas y testeos han tenido su propia fase, lo cierto es que se han estado corrigiendo los errores que surgían al final de cada iteración. Esto se ha realizado así por la metodología ágil [17] que sigue el proyecto.

Sin embargo, esto no quiere decir que no se hayan hecho las respectivas pruebas para poder comprobar y verificar el correcto funcionamiento de la aplicación web. Aquí están las pruebas realizadas:

Identificador	T1-U
Acción	Iniciar sesión correctamente.
Resultado esperado	
	El usuario accederá a su perfil principal.
Resultado obtenido	
	El usuario accede al perfil principal.

Tabla 112: Prueba de inicio de sesión.

Identificador	T2-U
Acción	Pulsar el botón “Perfil Principal” de la barra de navegación.
Resultado esperado	
El usuario regresará a su perfil principal.	
Resultado obtenido	
El usuario regresa a su perfil principal.	

Tabla 113: Prueba del botón “Perfil Principal”.

Identificador	T3-U
Acción	Pulsar el botón “Cerrar Sesión” de la barra de navegación.
Resultado esperado	
El usuario cerrará su sesión, regresando a su perfil principal.	
Resultado obtenido	
El usuario regresa a su perfil principal habiendo cerrado su sesión.	

Tabla 114: Prueba del botón “Cerrar Sesión”.

Identificador	T4-U
Acción	Iniciar sesión de forma incorrecta.
Resultado esperado	
El usuario no inicia sesión y salta un mensaje de error de credenciales.	
Resultado obtenido	
El usuario no inicia sesión, y salta un mensaje de error según la situación.	

Tabla 115: Prueba de inicio de sesión incorrecta.

Identificador	T5-U
Acción	Pulsar el botón de “Volver Atrás”.
Resultado esperado	
El usuario retrocede a la pantalla anterior respecto a la que se encontraba.	
Resultado obtenido	
El usuario vuelve a una pantalla anterior respecto a donde se encontraba.	

Tabla 116: Prueba del botón “Volver atrás”.

Identificador	T6-U
Acción	Escribir en la caja de texto “Buscar:” en las vistas que contengan tablas.
Resultado esperado	
La tabla filtrará los contenidos de las tablas según lo escrito por el usuario.	
Resultado obtenido	
La tabla filtra los contenidos de las tablas según lo escrito por el usuario.	

Tabla 117: Prueba para filtrar contenidos en las tablas.

Identificador	T1-M
Acción	Pulsar el botón “Modificar Archivo” en un informe con un archivo elegido.
Resultado esperado	

Se sustituirá el contenido del informe por el archivo elegido.
Resultado obtenido
Se sustituye correctamente el contenido del informe.

Tabla 118: Prueba para sustituir el contenido de un informe.

Identificador	T2-M
Acción	Pulsar el botón “Modificar Archivo” sin haber elegido un informe.
Resultado esperado	
Saltará un mensaje de error informando sobre la ausencia de un contenido.	
Resultado obtenido	
El mensaje aparece en la pantalla informando del problema.	

Tabla 119: Prueba para sustituir el contenido de un informe sin haber elegido uno con anterioridad.

Identificador	T1-AD
Acción	Crear un nuevo usuario alumno correctamente.
Resultado esperado	
El administrador creará un nuevo alumno.	
Resultado obtenido	
Se crea correctamente el nuevo alumno.	

Tabla 120: Prueba para crear un nuevo alumno.

Identificador	T2-AD
Acción	Crear un nuevo usuario profesor correctamente.
Resultado esperado	
El administrador creará un nuevo profesor.	
Resultado obtenido	
Se crea correctamente el nuevo profesor.	

Tabla 121: Prueba para crear un nuevo profesor.

Identificador	T3-AD
Acción	Crear un nuevo usuario médico correctamente.
Resultado esperado	
El administrador creará un nuevo médico.	
Resultado obtenido	
Se crea correctamente el nuevo médico.	

Tabla 122: Prueba para crear un nuevo médico.

Identificador	T4-AD
Acción	Usar un NIF ya utilizado por otro usuario para crear un nuevo usuario.
Resultado esperado	
Un mensaje de error se mostrará en el campo NIF informando de la situación.	
Resultado obtenido	
El mensaje de error se muestra correctamente debajo del atributo NIF en la	

creación de un usuario.

Tabla 123: Prueba para crear un nuevo usuario con un NIF que ya estaba usado.

Identificador	T5-AD
Acción	Usar un nombre de usuario ya utilizado por otro usuario para crear un nuevo usuario.
Resultado esperado	
Un mensaje de error se mostrará en el campo Usuario informando de la situación.	
Resultado obtenido	
El mensaje de error se muestra correctamente debajo del atributo Usuario en la creación de un usuario.	

Tabla 124: Prueba para crear un nuevo usuario con un nombre de usuario que ya estaba usado.

Identificador	T6-AD
Acción	Usar solo espacios en blancos para rellenar los atributos en la creación de un nuevo usuario.
Resultado esperado	
Un mensaje de error se mostrará en el campo formado por solo espacios en blanco.	
Resultado obtenido	
El mensaje de error se muestra correctamente debajo del atributo en la creación de un usuario.	

Tabla 125: Prueba para crear un nuevo usuario usando espacios en blanco.

Identificador	T7-AD
Acción	Crear un nuevo usuario con algún atributo vacío.
Resultado esperado	
Un mensaje de error se mostrará en la pantalla, informando de que algún atributo está vacío.	
Resultado obtenido	
El mensaje de error se muestra correctamente en la pantalla informando de la situación.	

Tabla 126: Prueba para crear un nuevo usuario dejando un atributo vacío.

Identificador	T8-AD
Acción	Crear un usuario con el atributo "Password" y "Confirmar Password" diferentes.
Resultado esperado	
Un mensaje de error se mostrará debajo del atributo "Confirmar Password" indicando que las contraseñas no coinciden.	
Resultado obtenido	
El mensaje de error se muestra correctamente debajo del atributo informando de la situación.	

Tabla 127: Prueba para verificar el funcionamiento de las contraseñas.

Identificador	T9-AD
Acción	Crear un nuevo informe correctamente.

Resultado esperado
Un administrador creará un informe de forma correcta.
Resultado obtenido
Se crea un nuevo informe de forma correcta.

Tabla 128: Prueba para crear un informe.

Identificador	T10-AD
Acción	Crear un nuevo informe con algún atributo vacío.
Resultado esperado	
Un mensaje de error se mostrará en la pantalla, informando de que algún atributo está vacío.	
Resultado obtenido	
El mensaje de error se muestra correctamente en la pantalla informando de la situación.	

Tabla 129: Prueba para crear un informe con algún atributo vacío.

Identificador	T11-AD
Acción	Crear una nueva docencia correctamente.
Resultado esperado	
Un administrador creará una docencia de forma correcta.	
Resultado obtenido	
Se crea una nueva docencia de forma correcta.	

Tabla 130: Prueba para crear una docencia.

Identificador	T12-AD
Acción	Crear una nueva docencia con algún atributo vacío.
Resultado esperado	
Un mensaje de error se mostrará en la pantalla, informando de que algún atributo está vacío.	
Resultado obtenido	
El mensaje de error se muestra correctamente en la pantalla informando de la situación.	

Tabla 131: Prueba para crear una docencia con algún atributo vacío.

Identificador	T13-AD
Acción	Crear una nueva matrícula correctamente.
Resultado esperado	
Un administrador creará una matrícula de forma correcta.	
Resultado obtenido	
Se crea una nueva matrícula de forma correcta.	

Tabla 132: Prueba para crear una matrícula.

Identificador	T14-AD
Acción	Crear una nueva docencia con algún atributo vacío.
Resultado esperado	
Un mensaje de error se mostrará en la pantalla, informando de que algún atributo está vacío.	

Resultado obtenido
El mensaje de error se muestra correctamente en la pantalla informando de la situación.

Tabla 133: Prueba para crear una matrícula con algún atributo vacío.

Identificador	T15-AD
Acción	Crear una nueva docencia ya existente
Resultado esperado	
Un mensaje de error se mostrará en la pantalla, informando de que esa docencia ya existe.	
Resultado obtenido	
El mensaje de error se muestra correctamente en la pantalla informando de la situación.	

Tabla 134: Prueba para crear una docencia existe.

Identificador	T16-AD
Acción	Crear una nueva matrícula ya existente
Resultado esperado	
Un mensaje de error se mostrará en la pantalla, informando de que esa matrícula ya existe.	
Resultado obtenido	
El mensaje de error se muestra correctamente en la pantalla informando de la situación.	

Tabla 135: Prueba para crear una matrícula existe.

Identificador	T17-AD
Acción	Crear una nueva asignatura correctamente.
Resultado esperado	
Un administrador creará una asignatura de forma correcta.	
Resultado obtenido	
Se crea una nueva asignatura de forma correcta.	

Tabla 136: Prueba para crear una asignatura.

Identificador	T18-AD
Acción	Crear una nueva asignatura con algún atributo vacío.
Resultado esperado	
Un mensaje de error se mostrará en la pantalla, informando de que algún atributo está vacío.	
Resultado obtenido	
El mensaje de error se muestra correctamente en la pantalla informando de la situación.	

Tabla 137: Prueba para crear una asignatura con algún atributo vacío.

Identificador	T19-AD
Acción	Crear una nueva asignatura con un nombre usado por otra asignatura.
Resultado esperado	
Un mensaje de error se mostrará en la pantalla, informando de que el nombre ya está siendo utilizado por otra asignatura.	

Resultado obtenido
El mensaje de error se muestra debajo del atributo “Nombre” de la asignatura, informando de la situación.

Tabla 138: Prueba para crear una asignatura con un nombre usado por otra asignatura.

Identificador	T20-AD
Acción	Borrar un usuario de forma definitiva.
Resultado esperado	
Se borrará un usuario.	
Resultado obtenido	
Un usuario se borra de forma correcta.	

Tabla 139: Prueba para borrar un usuario.

Identificador	T21-AD
Acción	Borrar un informe de forma definitiva.
Resultado esperado	
Se borrará un informe.	
Resultado obtenido	
Un informe se borra de forma correcta.	

Tabla 140: Prueba para borrar un usuario.

Identificador	T22-AD
Acción	Borrar el único informe que le queda a un alumno.

Resultado esperado
No se borrará el informe, y se mostrará un mensaje informando sobre cómo es el único informe que tiene.
Resultado obtenido
El informe no se borra, y un mensaje se muestra por pantalla informando de la situación.

Tabla 141: Prueba para borrar el único informe que le queda a un alumno.

Identificador	T23-AD
Acción	Borrar una docencia de forma definitiva.
Resultado esperado	
Se borrará una docencia.	
Resultado obtenido	
Una docencia se borra de forma correcta.	

Tabla 142: Prueba para borrar una docencia.

Identificador	T24-AD
Acción	Borrar una matrícula de forma definitiva.
Resultado esperado	
Se borrará una matrícula.	
Resultado obtenido	
Una matrícula se borra de forma correcta.	

Tabla 143: Prueba para borrar una matrícula.

Identificador	T25-AD
Acción	Borrar una asignatura de forma definitiva.
Resultado esperado	
Se borrará una asignatura.	
Resultado obtenido	
Una asignatura se borra de forma correcta.	

Tabla 144: Prueba para borrar una asignatura.

Identificador	T26-AD
Acción	Modificar un atributo a un usuario.
Resultado esperado	
Se guardarán los cambios realizados a algún atributo de un usuario.	
Resultado obtenido	
Se guardan de forma correcta las modificaciones en el usuario.	

Tabla 145: Prueba para modificar un atributo a un usuario.

Identificador	T27-AD
Acción	Modificar un atributo poniendo solo espacios en blancos.
Resultado esperado	
Un mensaje aparecerá debajo del atributo advirtiéndole que solo tiene espacios en blanco.	
Resultado obtenido	
El atributo no se modifica, además de aparecer un mensaje advirtiéndole de la	

situación debajo del atributo.

Tabla 146: Prueba para modificar un atributo a un usuario, pero solo poniendo espacios en blanco.

Identificador	T28-AD
Acción	Dejar algún atributo en blanco en la modificación de un usuario.
Resultado esperado	
Un mensaje aparecerá debajo del atributo advirtiendo que está vacío.	
Resultado obtenido	
El atributo no se modifica, además de aparecer un mensaje advirtiendo de la situación debajo del atributo.	

Tabla 147: Prueba para dejar vacío un atributo de un usuario.

Identificador	T29-AD
Acción	Modificar el nombre de una asignatura.
Resultado esperado	
Se guardarán los cambios realizados al nombre de una asignatura.	
Resultado obtenido	
Se guardan de forma correcta las modificaciones en la asignatura.	

Tabla 148: Prueba para modificar el nombre de una asignatura.

Identificador	T30-AD
----------------------	--------

Acción	Dejar el nombre en blanco en la modificación de una asignatura.
Resultado esperado	
Un mensaje aparecerá debajo del atributo advirtiendo que está formado solo por espacios en blanco.	
Resultado obtenido	
El atributo no se modifica, además de aparecer un mensaje advirtiendo de la situación debajo del atributo.	

Tabla 149: Prueba para introducir el nombre de una asignatura con solo espacios en blanco.

Identificador	T31-AD
Acción	Dejar el nombre vacío en la modificación de una asignatura.
Resultado esperado	
Un mensaje aparecerá debajo del nombre advirtiendo que está vacío.	
Resultado obtenido	
El nombre no se modifica, además de aparecer un mensaje advirtiendo de la situación debajo del atributo.	

Tabla 150: Prueba para dejar vacío el nombre de una asignatura.

Identificador	T32-AD
Acción	Cambiar el médico titular de un informe correctamente.
Resultado esperado	

Se cambiará el médico titular de un informe.
Resultado obtenido
El médico titular de un informe se ha modificado.

Tabla 151: Prueba para cambiar el médico titular de un informe.

Identificador	T33-AD
Acción	Escribir un médico inexistente en una lista de médicos.
Resultado esperado	
Un mensaje de error aparecerá en la lista.	
Resultado obtenido	
El mensaje de error se muestra debajo de la lista de forma correcta informando de la situación.	

Tabla 152: Prueba para verificar el funcionamiento de las listas desplegables de los médicos.

Identificador	T34-AD
Acción	Escribir un médico con una estructura diferente a la esperada en una lista de médicos.
Resultado esperado	
Un mensaje de error aparecerá en la lista.	
Resultado obtenido	
El mensaje de error se muestra debajo de la lista de forma correcta informando de la situación.	

Tabla 153: Prueba para verificar la estructura de las listas desplegables de los médicos.

Identificador	T35-AD
Acción	Escribir un profesor inexistente en una lista de profesores.
Resultado esperado	
Un mensaje de error aparecerá en la lista.	
Resultado obtenido	
El mensaje de error se muestra debajo de la lista de forma correcta informando de la situación.	

Tabla 154: Prueba para verificar el funcionamiento de las listas desplegables de los profesores.

Identificador	T36-AD
Acción	Escribir un profesor con una estructura diferente a la esperada en una lista de profesor.
Resultado esperado	
Un mensaje de error aparecerá en la lista.	
Resultado obtenido	
El mensaje de error se muestra debajo de la lista de forma correcta informando de la situación.	

Tabla 155: Prueba para verificar la estructura de las listas desplegables de los profesores.

Identificador	T37-AD
Acción	Escribir un alumno inexistente en una lista de alumnos.
Resultado esperado	
Un mensaje de error aparecerá en la lista.	
Resultado obtenido	
El mensaje de error se muestra debajo de la lista de forma correcta informando de la situación.	

Tabla 156: Prueba para verificar el funcionamiento de las listas desplegables de los alumnos.

Identificador	T38-AD
Acción	Escribir un alumno con una estructura diferente a la esperada en una lista de alumno.
Resultado esperado	
Un mensaje de error aparecerá en la lista.	
Resultado obtenido	
El mensaje de error se muestra debajo de la lista de forma correcta informando de la situación.	

Tabla 157: Prueba para verificar la estructura de las listas desplegables de los alumnos.

Identificador	T39-AD
Acción	Escribir una asignatura inexistente en una lista de asignaturas.
Resultado esperado	

Un mensaje de error aparecerá en la lista.
Resultado obtenido
El mensaje de error se muestra debajo de la lista de forma correcta informando de la situación.

Tabla 158: Prueba para verificar el funcionamiento de las listas desplegadas de las asignaturas.

Identificador	T40-AD
Acción	Escribir una asignatura con una estructura diferente a la esperada en una lista de asignaturas.
Resultado esperado	
Un mensaje de error aparecerá en la lista.	
Resultado obtenido	
El mensaje de error se muestra debajo de la lista de forma correcta informando de la situación.	

Tabla 159: Prueba para verificar la estructura de las listas desplegadas de las asignaturas.

5

Conclusiones y líneas futuras

5.1 Conclusiones

La versión final de la aplicación web ha resultado satisfactoria: Se han implementado todos los requisitos planteados inicialmente, además de algunos aspectos extras que han ido surgiendo según se ha ido avanzando en el proyecto.

En cuanto a la interfaz, se ha intentado realizar algo intuitivo, pero a la vez simple, sin que esté muy cargado o sea muy complejo a la vista. En este punto se ha tenido especial cuidado, ya que no es un aspecto en el que se tenía un alto grado de dominancia.

Gracias a este proyecto, se ha podido desarrollar software en nuevas tecnologías, ampliando la experiencia y conocimientos adquiridos anteriormente. Además, se ha utilizado metodologías y tecnologías que se usarían en el mundo laboral, por lo que se ha considerado una experiencia tanto positiva como enriquecedora.

5.2 Líneas futuras

Aunque la aplicación web cumple con las características planteadas, siempre puede haber un margen de mejora. Por eso, se van a proponer estas líneas de desarrollo que se podrían realizar a futuro para mejorar la aplicación:

1. El orden de las tablas:

Aunque las tablas muestran correctamente los datos intencionados, y

filtra con precisión cuando se les proporciona algún carácter en la caja de texto, en cuanto al orden de los elementos cuando se pulsa en la cabecera de las columnas, es cierto que no ordena como se esperaría: caracteres especiales, independientemente de su orden en el alfabeto, siempre van en último lugar. Algo similar sucede con las fechas, pues las tablas ordenan las fechas por orden numérico, y no por tiempo, (aunque aparecen en orden gracias al formato de la fecha que se ha escogido, el cual es: *año/mes/día horas:minutos:segundos*). Esta característica, aunque se ha intentado arreglar, no se ha obtenido un resultado satisfactorio, y se tendría que hacer una investigación más avanzada, a la vez que profundizar más en el lenguaje de JavaScript para poder darle una solución definitiva.

2. **Mayor rango de sistemas operativos:**

La aplicación web está diseñada para funcionar exclusivamente en Windows, (preferiblemente Windows 10 o superior), por lo que en otros sistemas operativos como Mac OS o Linux, la aplicación web puede funcionar, pero podría tener comportamientos inesperados y no controlados. También se han realizado pruebas en móviles, y aunque sigue siendo una aplicación web, parece que se puede funcionar en algunos dispositivos telefónicos, pero el objetivo de este proyecto es el de una aplicación web, no el de una aplicación móvil.

3. **Sesión más sofisticada con los administradores y la navegación:**

El sistema de navegación de la aplicación web funciona de forma correcta, y la sesión guarda los valores como debería. Sin embargo, específicamente para los usuarios del tipo administrador, se debería tener especialmente cuidado con esta característica, ya que este usuario tiene la capacidad de modificar atributos y contenidos de otros usuarios a tiempo real. Por ejemplo, si un administrador borra a un usuario que haya iniciado sesión en ese momento, debería haber un conflicto.

Técnicamente esto no debería ser posible, ya que sabiendo esto, los administradores operarían con la aplicación en modo de mantenimiento, pero sería un caso extraordinario que podría mejorarse.

Además, la aplicación web está diseñada para navegar en ella mediante los botones proporcionados en la pantalla, por lo que utilizar las flechas de los navegadores para desplazarse entre pantallas no es aconsejable, pues podría llevar a errores en la sesión. Se intentó arreglar este aspecto, pero debido a las diferencias de comportamiento entre navegadores, se dejó como una posible opción a futuro el investigar más a fondo esta característica.

4. **Detalles varios:**

Como toda aplicación web, siempre se puede mejorar y añadir funcionalidades, como la capacidad de poder cambiarse de idioma, o una mejora del rendimiento de la aplicación. También se podría intentar mejorar su adaptabilidad para que otros navegadores pueden acceder a esta aplicación web. Hay que recordar que ninguna aplicación web es perfecta, y que siempre se puede mejorar algunos aspectos de ella, o simplemente añadir más cosas que no se ha tenido en cuenta en su desarrollo.

Referencias

- [1] Microsoft. (accedido el 29 de Junio de 2023). *dotnet.microsoft.com: csharp language*. Obtenido de dotnet.microsoft.com: <https://dotnet.microsoft.com/es-es/languages/csharp>
- [2] developer.mozilla.org. (accedido el 29 de Junio de 2023). *developer.mozilla.org: ¿ Qué es el JavaScript ?* Obtenido de developer.mozilla.org: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- [3] Álvarez, M. A. (accedido el 29 de Junio de 2023). *desarrolloweb.com: Qué es el HTML*. Obtenido de desarrolloweb.com: <https://desarrolloweb.com/articulos/que-es-html.html>
- [4] developer.mozilla.org. (accedido el 29 de Junio de 2023). *developer.mozilla.org: ¿ Qué es el CSS ?* Obtenido de developer.mozilla.org: ¿ Qué es el CSS ?: https://developer.mozilla.org/es/docs/Learn/CSS/First_steps/What_is_CSS
- [5] Deyimar, A. (accedido el 29 de Junio de 2023). *www.hostinger.mx: ¿ Qué es Bootstrap ?* Obtenido de www.hostinger.mx: <https://www.hostinger.mx/tutoriales/que-es-bootstrap>
- [6] Koidan, K. (accedido el 29 de Junio de 2023). *learnsql.es: ¿ Qué es SQL ?* Obtenido de learnsql.es: <https://learnsql.es/blog/que-es-sql/>
- [7] Microsoft. (accedido el 29 de Junio de 2023). *learn.microsoft.com: Language Integrated Query (LINQ)*. Obtenido de learn.microsoft.com: <https://learn.microsoft.com/es-es/dotnet/csharp/programming-guide/concepts/linq/>
- [8] Flores, F. (accedido el 29 de Junio de 2023). *openwebinars.net: Qué es Visual Studio y qué ventajas ofrece*. Obtenido de openwebinars.net: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- [9] Fernández, Y. (accedido el 29 de Junio de 2023). *www.xataka.com: Qué es Github y qué es lo que le ofrece a los desarrolladores*. Obtenido de www.xataka.com: <https://www.xataka.com/basics/que-github-que-le-ofrece-a-desarrolladores>
- [10] Github. (accedido el 29 de Junio de 2023). *docs.github.com: Comenzar con GitHub Desktop*. Obtenido de docs.github.com:

<https://docs.github.com/es/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop>

- [11] Microsoft. (accedido el 29 de Junio de 2023). *learn.microsoft.com: ¿Qué es SQL Server Management Studio (SSMS)?* Obtenido de learn.microsoft.com: <https://learn.microsoft.com/es-es/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>
- [12] Keepcoding. (accedido el 29 de Junio de 2023). *keepcoding.io: ¿Qué es Draw.io?* Obtenido de keepcoding.io: <https://keepcoding.io/blog/que-es-drawio/>
- [13] Lucidchart. (accedido el 29 de Junio de 2023). *https://www.lucidchart.com/pages/es: ¿Qué es Lucidchart ?* Obtenido de <https://www.lucidchart.com/pages/es>
- [14] Microsoft. (accedido el 29 de Junio de 2023). *https://www.microsoft.com: Instalar SQL Server 2022.* Obtenido de <https://www.microsoft.com/es-es/sql-server/sql-server-downloads>
- [15] Microsoft. (accedido el 29 de Junio de 2023). *azure.microsoft.com: Sitio Oficial Microsoft Azure.* Obtenido de azure.microsoft.com: https://azure.microsoft.com/es-es/free/search/?ef_id=k_447d2cd194a01293da7c484c135ea3fe_k_&OCID=AIDcmm68ejnsa0_SEM_k_447d2cd194a01293da7c484c135ea3fe_k_&msclkid=447d2cd194a01293da7c484c135ea3fe
- [16] Pmoinformatica. (accedido el 29 de Junio de 2023). *pmoinformatica.com: Elementos del diagrama de casos de uso.* Obtenido de pmoinformatica.com: <http://www.pmoinformatica.com/2021/02/elementos-diagrama-casos-de-uso.html>
- [17] Hubspot. (accedido el 29 de Junio de 2023). *blog.hubspot.es.* Obtenido de blog.hubspot.es: <https://blog.hubspot.es/marketing/metodologia-scrum>

Apéndice A

Manual de despliegue e instalación

Para utilizar la aplicación web, no se requiere de ninguna instalación ni despliegue extra, pues todo ya está preparado y funcionando en Internet, (específicamente, gracias a Azure). Lo único necesario sería tener una conexión estable a Internet, un ordenador con Windows 10 o una versión mayor, y un navegador web compatible con la aplicación web.

Aquí una lista de los posibles navegadores compatibles:



Google Chrome



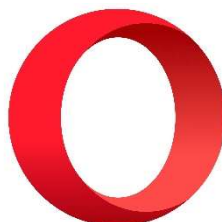
Microsoft Edge



Mozilla Firefox



Brave



Ópera



Vivaldi

Hay que recordar que es una aplicación web, y no está diseñada ni pensada para que sea una aplicación móvil, por lo que no se recomienda su uso en teléfono móviles.

También se hace hincapié en el sistema operativo: esta aplicación web se ha desarrollado para Windows 10, (o versiones posteriores), por lo que usar otro sistema operativo para acceder a la aplicación podría provocar conflictos y problemas.

Aquí está el enlace necesario para acceder a la aplicación web: (Se recomienda hacer ctrl + click izquierdo para acceder al enlace)

<https://gestiondiversidad.azurewebsites.net>

Apéndice B

Manual de usuario

B.1 Introducción

Este documento tiene como objetivo facilitar a los usuarios de la aplicación web su uso de una forma correcta, resolviendo las posibles dudas que puedan surgir en el momento de su empleo.

Este manual está dirigido sobre todo a los usuarios que no tienen ningún tipo de experiencia con la aplicación, y se explicará paso a paso qué acciones se deben realizar para completar las distintas tareas que la aplicación ofrece, partiendo del inicio de sesión en un navegador web.

Si surge algún tipo de duda o comentario, no dude en enviar sus dudas vía [correo electrónico](#). (Ctr + click izquierdo en “correo electrónico” para acceder al correo en cuestión).

B.2 Inicio sesión como alumno

Para poder iniciar sesión como alumno, se necesita tener una cuenta con dicho rol. Si no la tiene, deberá contactar a los encargados de la aplicación web o a algún administrador para que se le proporcione una cuenta. En el caso de poseer una, solo tiene que poner su nombre de usuario en la caja de texto “Usuario:” y la contraseña correspondiente en la caja de texto “Password:”.

Una vez introducido los datos, solo tiene que pulsar el botón **“Iniciar Sesión”** para entrar en el perfil principal del alumno.

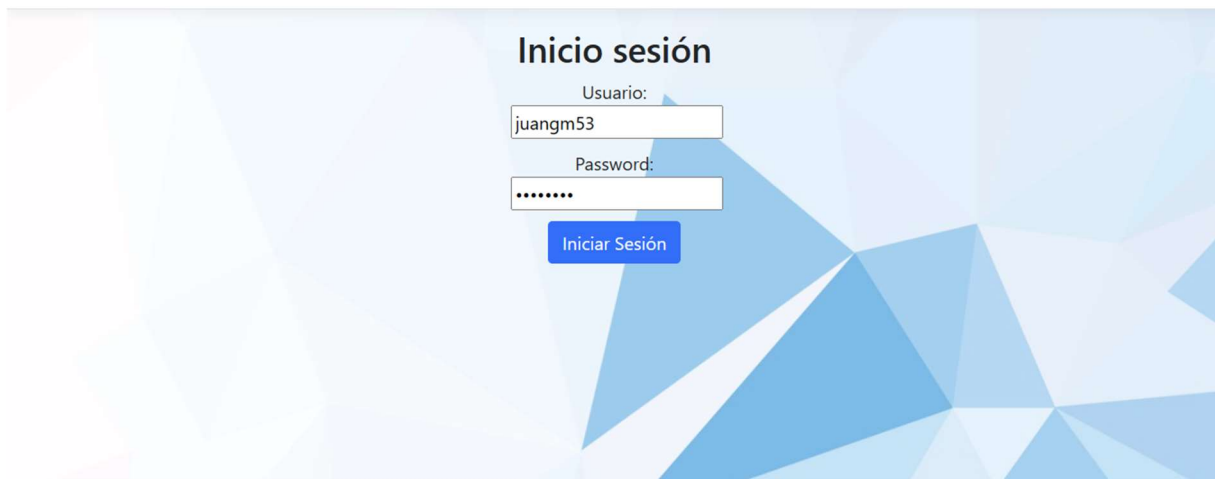


Imagen 56: Inicio de sesión de un alumno cuyo nombre de usuario y contraseña son juangm53.



Imagen 57: Perfil principal del alumno Juan García Martínez.

B.3 Ver las asignaturas de un alumno

Para poder ver las asignaturas en las que un alumno está matriculado, solo hay que pulsar el botón “**Asignaturas**” en el perfil principal de un alumno. La aplicación web redirigirá al usuario a una tabla con la lista de asignaturas en las que el alumno está matriculado.



Lista de asignaturas: Alumno/a Juan (NIF: 1111111A)

Mostrar asignaturas

Buscar:

Nombre
Cálculo
Java
Álgebra

Mostrando 1 a 3 de 3 asignaturas

Anterior **1** Siguiente

Volver atrás

Imagen 58: Asignatura en las que está matriculado el alumno Juan con NIF 11111111.

B.4 Ver los informes de un alumno

Para poder ver los informes asignados a un alumno, primero hay que pulsar el botón **“Informes”** en el perfil principal de un alumno. La aplicación web redirigirá al usuario a una tabla con los informes que pertenecen a dicho alumno. Una vez en dicha pantalla, se tiene que pulsar en el botón **“Detalles”** de un informe para acceder a su información completa. En la nueva pantalla del informe, se podrá apreciar un botón que pone **“Ver informe”**. Si se pulsa dicho botón, una nueva ventana emergerá en el navegador, siendo esta el contenido del informe en formato PDF.

Lista de informes: Alumno/a Juan (NIF: 1111111A)

Mostrar informes

Buscar:

Alumno(NIF)	Médico(NIF)	Fecha	Acción: Detalles del informe
Juan (1111111A)	Ricardo (1111111M)	2023/05/31 23:06:54	Detalles
Juan (1111111A)	Carlos (3333333M)	2023/06/06 23:00:51	Detalles

Mostrando 1 a 2 de 2 informes

Anterior **1** Siguiente

Volver atrás

Imagen 59: Informes del alumno Juan con NIF 11111111A.



Imagen 60: Información principal de un informe.



Imagen 61: Nueva pestaña con el contenido del informe.

B.5 Inicio sesión como profesor

Para poder iniciar sesión como profesor, se necesita tener una cuenta con el rol de profesor asignado. Si no la tiene, deberá contactar a los encargados de la aplicación web o a algún administrador para que se le proporcione una cuenta. En el caso de poseer una, se procederá igual que el punto **B.2**. En este caso, tanto el nombre de usuario como contraseña son “**alegl24**”

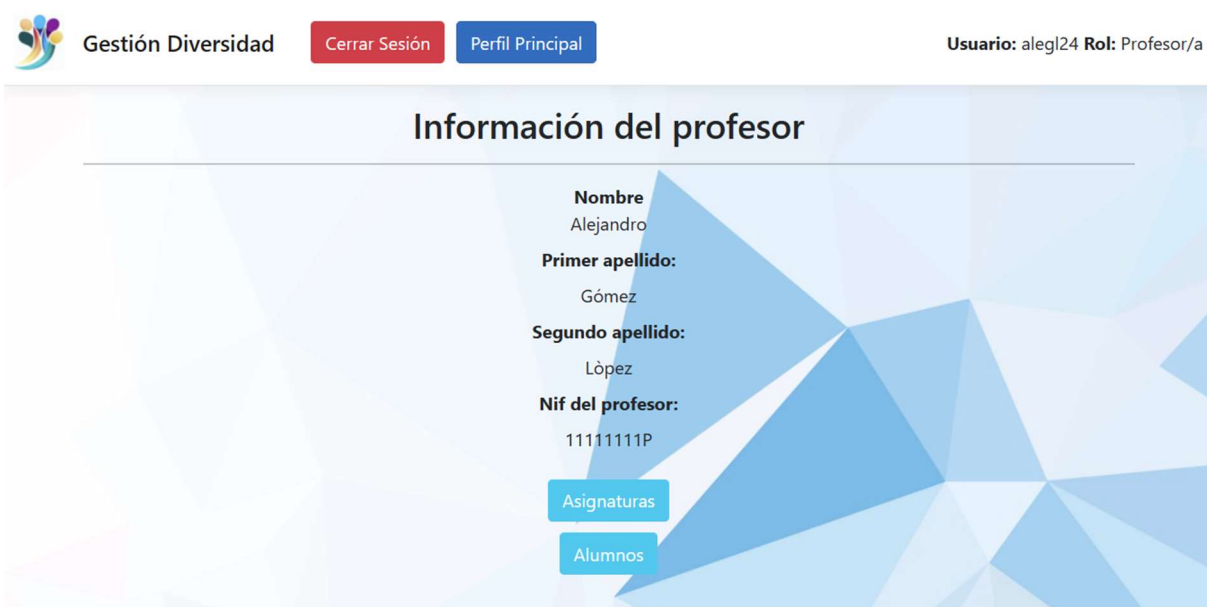


Imagen 62: Perfil principal del profesor Alejandro Gómez López.

B.6 Ver las asignaturas de un profesor

Para poder ver las asignaturas que un profesor imparte, solo hay que pulsar el botón “**Asignaturas**” en el perfil principal de profesor. El procedimiento es similar al punto B.3.

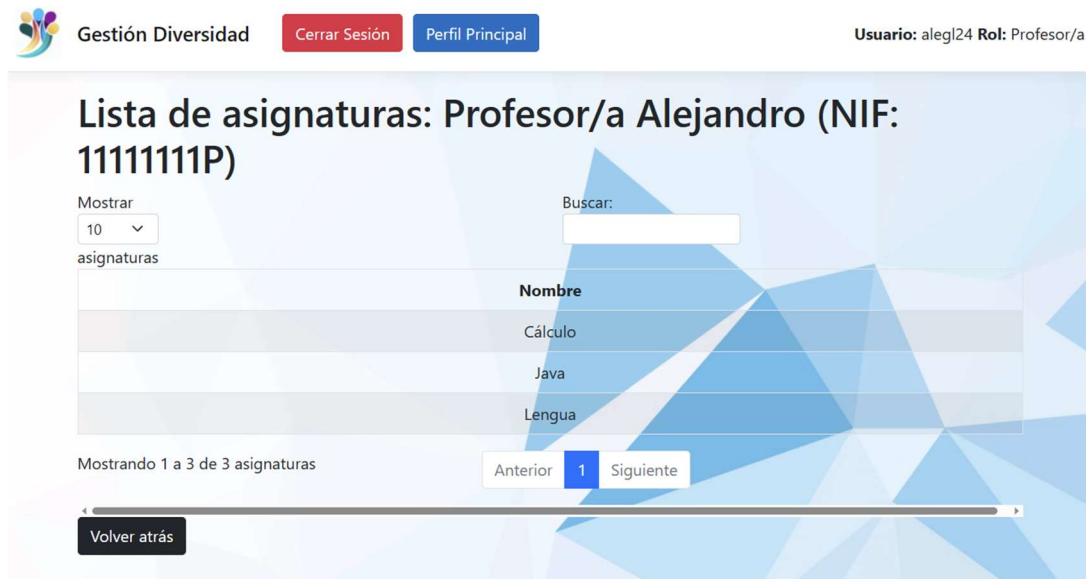


Imagen 63: Asignaturas que imparte el profesor Alejandro con NIF 11111111P.

B.7 Acceder al perfil principal de los alumnos de un profesor

Para poder ver los alumnos a los que imparte un profesor, primero hay que pulsar el botón “**Alumnos**” en el perfil principal de un profesor. La aplicación web redirigirá al usuario a una tabla con los alumnos a los que imparte el profesor. Una vez en dicha pantalla, se tiene que pulsar en el botón “**Detalles**” para acceder al perfil principal de un alumno. Un profesor podrá acceder tanto a la información relativa a las asignaturas del alumno, como a sus informes.

Mostrar 10 alumnos

Buscar:

Nombre	Primer apellido	Segundo apellido	Nif	Acción: Detalles del alumno
Juan	García	Martínez	11111111A	Detalles
Marta	Fernández	González	22222222A	Detalles
Nacho	Monte	Romero	33333333A	Detalles

Mostrando 1 a 3 de 3 alumnos

Anterior 1 Siguiente

Volver atrás

Imagen 64: Alumnos del profesor Alejandro con NIF 11111111P.

B.8 Inicio sesión como médico

Para poder iniciar sesión como médico, se necesita tener una cuenta con el rol de médico asignado. Si no la tiene, deberá contactar a los encargados de la aplicación web o a algún administrador para que se le proporcione una cuenta. En el caso de poseer una, se procederá igual que el punto B.2. En este caso, tanto el nombre de usuario como contraseña son “carlosrn22”.

Mostrar 10 alumnos

Buscar:

Mostrando 1 a 3 de 3 alumnos

Anterior 1 Siguiente

Volver atrás

Información de médico

Nombre:
Carlos

Primer apellido:
Ramírez

Segundo apellido:
Navarro

Nif:
33333333M

Alumnos

Informes

Imagen 65: Perfil principal del médico Carlos Ramírez Navarro.

B.9 Acceder al perfil principal de los alumnos de un médico

Para poder ver los alumnos que tiene asignado un médico, primero hay que pulsar el botón “**Alumnos**” en el perfil principal de un médico. Los siguientes pasos son similares a los del punto B.7. Sin embargo, al contrario que los demás roles, los médicos no tienen permitido ver las asignaturas de los alumnos.

Gestión Diversidad [Cerrar Sesión](#) [Perfil Principal](#) Usuario: carlosrn22 Rol: Médico

Lista de alumnos: Médico Carlos (NIF: 33333333M)

Mostrar alumnos

Buscar:

Nombre	Primer apellido	Segundo apellido	Nif	Acción: Detalles del alumno
Juan	García	Martínez	11111111A	Detalles
Nacho	Monte	Romero	33333333A	Detalles

Mostrando 1 a 2 de 2 alumnos

[Anterior](#) [1](#) [Siguiete](#)

[Volver atrás](#)

Imagen 66: Alumnos del médico Carlos con NIF 33333333M.

Gestión Diversidad [Cerrar Sesión](#) [Perfil Principal](#) Usuario: carlosrn22 Rol: Médico

Información del/a alumno/a

Nombre:
Nacho

Primer apellido:
Monte

Segundo apellido:
Romero

Nif:
33333333A

[Informes](#)

[Volver atrás](#)

Imagen 67: Perfil principal de un alumno accedido desde un usuario médico.

B.10 Ver los informes de un médico

Para poder ver los informes de un médico, primero hay que pulsar el botón “**Informes**” en el perfil principal de un médico. Los siguientes pasos son similares a los del punto B.4.

Gestión Diversidad Cerrar Sesión Perfil Principal Usuario: carlosrn22 Rol: Médico

Lista de informes: Médico Carlos (NIF: 33333333M)

Mostrar: 10

informes

Alumno(NIF)	Médico(NIF)	Fecha	Acción: Detalles del informe
Juan (11111111A)	Carlos (33333333M)	2023/06/06 23:00:51	Detalles
Nacho (33333333A)	Carlos (33333333M)	2023/05/31 23:09:51	Detalles

Mostrando 1 a 2 de 2 informes

Anterior 1 Siguiente

Volver atrás

Imagen 68: Informes del médico Carlos con NIF 33333333M.

B.11 Modificar el contenido de un informe

Para modificar el contenido de un informe, el usuario médico deberá encontrarse en la pantalla que contiene los detalles del informe. (Véase punto B.4). Una vez en esa pantalla, y solo si el médico en cuestión es el médico titular del informe, podrá ver un nuevo botón llamado “**Elegir archivo**”. Una vez seleccionado el botón, el médico podrá elegir solo archivos en PDF para sustituir el contenido actual del informe por el nuevo PDF seleccionado.

Gestión Diversidad Cerrar Sesión Perfil Principal Usuario: carlosrn22 Rol: Médico

Información del informe

Nombre alumno: Nacho
Nombre médico: Carlos
Fecha de creación: 2023/05/31 23:09:51

Ver informe

Seleccione el archivo PDF: No se ha seleccionado ningún archivo

Modificar archivo

Volver atrás

Imagen 69: Informe accedido desde el usuario de su médico titular.

B.12 Inicio sesión como administrador

Para poder iniciar sesión como administrador, se necesita tener una cuenta con el rol de administrador asignado. Si no la tiene, deberá contactar a los encargados de la aplicación web para que se le proporcione una cuenta. En el caso de poseer una, se procederá igual que el punto B.2. En este caso, tanto el nombre de usuario como contraseña son “**admin1**”.



Imagen 70: Perfil principal de un administrador.

B.13 Ver la lista de profesores y acceder a su perfil principal

Para acceder a la lista de todos los profesores, solo hay que pulsar el botón “**Profesores**” en el perfil principal de un administrador. El sistema llevará al usuario a una tabla con todos los profesores registrados en la aplicación.

Una vez en esa tabla, si se pulsa el botón “**Detalles**”, el sistema redirigirá al usuario al perfil principal del profesor seleccionado.

Gestión Diversidad Cerrar Sesión Perfil Principal Usuario: admin1 Rol: Administrador/a

Lista de profesores

Mostrar: 10 profesores

Buscar:

Nombre	Primer apellido	Segundo apellido	Nif	Acción: Detalles	Acción: Borrar
Alejandro	Gómez	López	11111111P	Detalles	Borrar
Carmen	Torres	Martin	22222222P	Detalles	Borrar
Daniel	Herrera	Rodriguez	33333333P	Detalles	Borrar
Laura	Pueblos	Salinas	44444444P	Detalles	Borrar

Mostrando 1 a 4 de 4 profesores

[Anterior](#) [1](#) [Siguiente](#)

[Crear nuevo profesor](#) [Volver atrás](#)

Imagen 71: Lista de todos los profesores.

B.14 Ver lista de médicos y acceder a su perfil principal

Para acceder a la lista de todos los médicos, solo hay que pulsar el botón “**Médicos**” en el perfil principal de un administrador. El sistema llevará al usuario a una tabla con todos los médicos registrados en la aplicación.

Una vez en esa tabla, si se pulsa el botón “**Detalles**”, el sistema redirigirá al usuario al perfil principal del médico seleccionado.

Gestión Diversidad Cerrar Sesión Perfil Principal Usuario: admin1 Rol: Administrador/a

Lista de médicos

Mostrar: 10 medicos

Buscar:

Nombre	Primer apellido	Segundo apellido	Nif	Acción: Detalles del médico	Acción: Borrar médico
Carlos	Ramírez	Navarro	33333333M	Detalles	Borrar
Javier	Gómez	Castro	22222222M	Detalles	Borrar
Medico_Temporal	Medico_Temporal	Medico_Temporal	Medico_Temporal	Detalles	Borrar
Ricardo	Medina	Cruz	11111111M	Detalles	Borrar

Mostrando 1 a 4 de 4 medicos

[Anterior](#) [1](#) [Siguiente](#)

[Crear nuevo médico](#) [Volver atrás](#)

Imagen 72: Lista de todos los médicos.

B.15 Ver lista de todos los alumnos y acceder a su perfil principal

Para acceder a la lista de todos los alumnos, solo hay que pulsar el botón “**Alumnos**” en el perfil principal de un administrador. El sistema llevará al usuario a una tabla con todos los alumnos registrados en la aplicación.

Una vez en esa tabla, si se pulsa el botón “**Detalles**”, el sistema redirigirá al usuario al perfil principal del alumno seleccionado.

Gestión Diversidad Cerrar Sesión Perfil Principal Usuario: admin1 Rol: Administrador/a

Lista de alumnos: Administrador/a leafeon (NIF: 111AD)

Mostrar 10 alumnos

Buscar:

Nombre	Primer apellido	Segundo apellido	Nif	Acción: Detalles del alumno	Acción: Borrar alumno
Juan	García	Martínez	11111111A	Detalles	Borrar
Marta	Fernández	González	22222222A	Detalles	Borrar
Nacho	Monte	Romero	33333333A	Detalles	Borrar

Mostrando 1 a 3 de 3 alumnos

Anterior 1 Siguiente

Crear nuevo alumno

Volver atrás

Imagen 73: Lista de todos los alumnos.

B.16 Ver lista de todas las asignaturas

Para acceder a la lista de todas las asignaturas, solo hay que pulsar el botón “**Asignaturas**” en el perfil principal de un administrador. El sistema llevará al usuario a una tabla con todas las asignaturas registradas en la aplicación.

Gestión Diversidad Cerrar Sesión Perfil Principal Usuario: admin1 Rol: Administrador/a

Lista de asignaturas: Administrador/a leafeon (NIF: 111AD)

Mostrar 10 asignaturas

Buscar:

Nombre	Acción: Borrar Asignatura	Acción: Modificar Asignatura
Cálculo	Borrar	Modificar
Java	Borrar	Modificar
Lengua	Borrar	Modificar
Álgebra	Borrar	Modificar

Mostrando 1 a 4 de 4 asignaturas

Anterior 1 Siguiente

Crear nueva asignatura

Volver atrás

Imagen 74: Lista de todas las asignaturas.

B.17 Ver lista de todas las matrículas

Para acceder a la lista de todas las matrículas, solo hay que pulsar el botón “**Matrículas**” en el perfil principal de un administrador. El sistema llevará al usuario a una tabla con todas las matrículas creadas en la aplicación.

Mostrar 10 matrículas

Buscar:

Alumno	Nif	Asignatura	Acción: Borrar matrícula
Juan	11111111A	Cálculo	Borrar
Juan	11111111A	Álgebra	Borrar
Juan	11111111A	Java	Borrar
Marta	22222222A	Cálculo	Borrar
Nacho	33333333A	Lengua	Borrar
Nacho	33333333A	Álgebra	Borrar

Mostrando 1 a 6 de 6 matrículas

Anterior 1 Siguiente

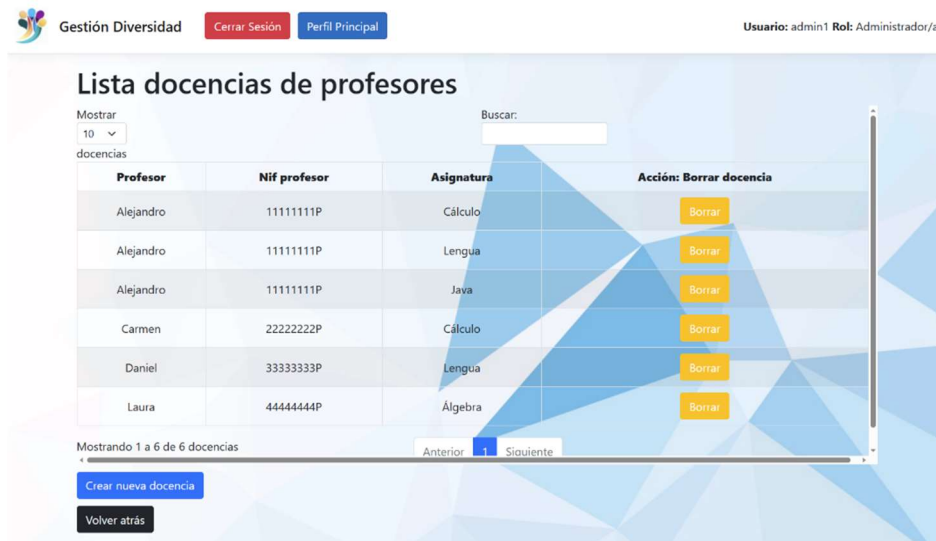
Crear nueva matrícula

Volver atrás

Imagen 75: Lista de todas las matrículas.

B.18 Ver listas de todas las docencias

Para acceder a la lista de todas las docencias, solo hay que pulsar el botón “**Docencias**” en el perfil principal de un administrador. El sistema llevará al usuario a una tabla con todas las docencias creadas en la aplicación.



Mostrar 10 docencias

Buscar:

Profesor	Nif profesor	Asignatura	Acción: Borrar docencia
Alejandro	11111111P	Cálculo	Borrar
Alejandro	11111111P	Lengua	Borrar
Alejandro	11111111P	Java	Borrar
Carmen	22222222P	Cálculo	Borrar
Daniel	33333333P	Lengua	Borrar
Laura	44444444P	Álgebra	Borrar

Mostrando 1 a 6 de 6 docencias

Anterior 1 Siguiente

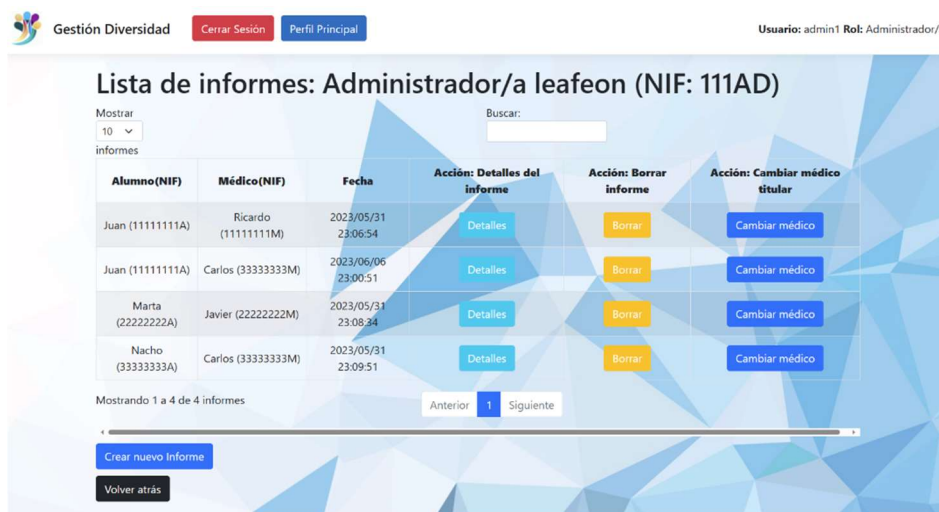
Crear nueva docencia

Volver atrás

Imagen 76: Lista de todas las docencias.

B.19 Ver lista de todos los informes

Para acceder a la lista de todos los informes, solo hay que pulsar el botón “**Informes**” en el perfil principal de un administrador. El sistema llevará al usuario a una tabla con todos los informes creados en la aplicación.



Mostrar 10 informes

Buscar:

Alumno(NIF)	Médico(NIF)	Fecha	Acción: Detalles del informe	Acción: Borrar informe	Acción: Cambiar médico titular
Juan (11111111A)	Ricardo (11111111M)	2023/05/31 23:06:54	Detalles	Borrar	Cambiar médico
Juan (11111111A)	Carlos (33333333M)	2023/06/06 23:00:51	Detalles	Borrar	Cambiar médico
Marta (22222222A)	Javier (22222222M)	2023/05/31 23:08:34	Detalles	Borrar	Cambiar médico
Nacho (33333333A)	Carlos (33333333M)	2023/05/31 23:09:51	Detalles	Borrar	Cambiar médico

Mostrando 1 a 4 de 4 informes

Anterior 1 Siguiente

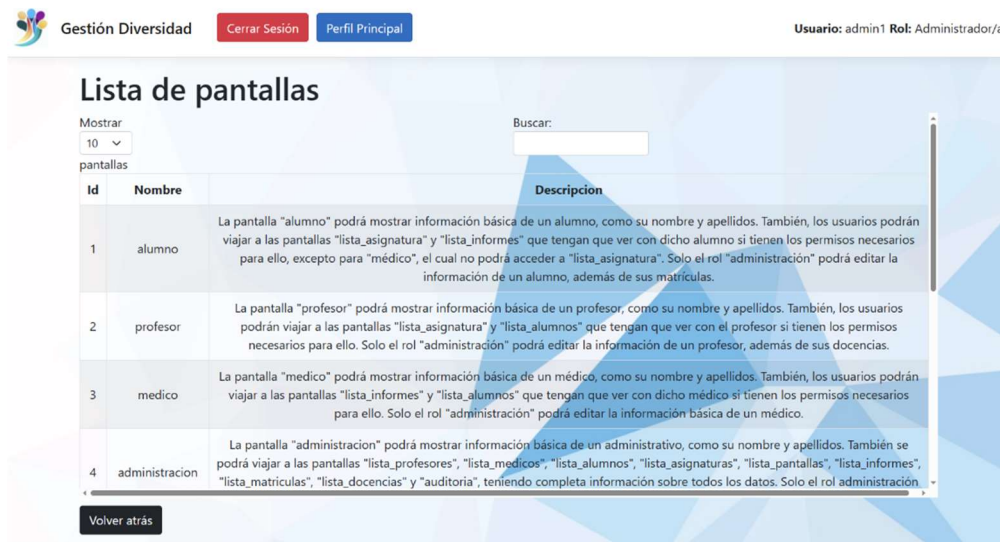
Crear nuevo Informe

Volver atrás

Imagen 77: Lista de todos los informes.

B.20 Lista de la información sobre las pantallas

Para acceder a la lista de la información sobre las pantallas, solo hay que pulsar el botón **“Pantallas”** en el perfil principal de un administrador. El sistema llevará al usuario a una tabla con la información en cuestión.



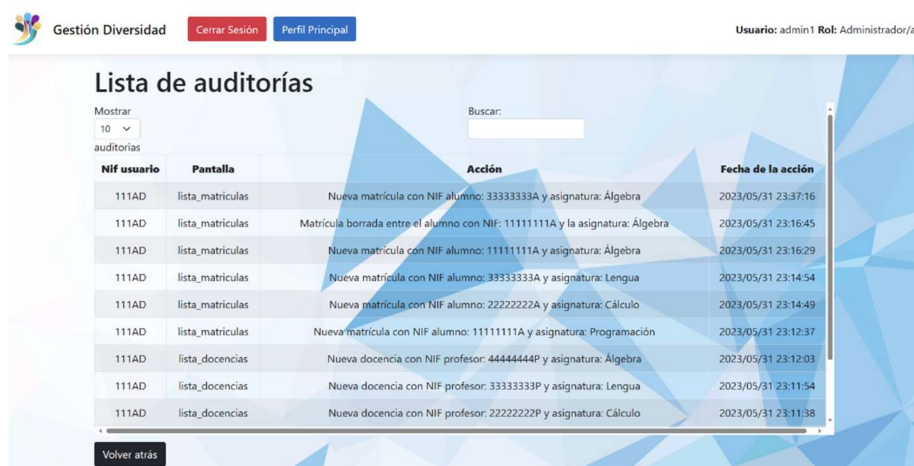
The screenshot shows the 'Lista de pantallas' page. At the top, there is a navigation bar with 'Gestión Diversidad', 'Cerrar Sesión', and 'Perfil Principal'. The user is identified as 'admin1 Rol: Administrador/a'. Below the title, there is a 'Mostrar' dropdown set to '10' and a search box. The table has the following data:

Id	Nombre	Descripción
1	alumno	La pantalla "alumno" podrá mostrar información básica de un alumno, como su nombre y apellidos. También, los usuarios podrán viajar a las pantallas "lista_asignatura" y "lista_informes" que tengan que ver con dicho alumno si tienen los permisos necesarios para ello, excepto para "médico", el cual no podrá acceder a "lista_asignatura". Solo el rol "administración" podrá editar la información de un alumno, además de sus matrículas.
2	profesor	La pantalla "profesor" podrá mostrar información básica de un profesor, como su nombre y apellidos. También, los usuarios podrán viajar a las pantallas "lista_asignatura" y "lista_alumnos" que tengan que ver con el profesor si tienen los permisos necesarios para ello. Solo el rol "administración" podrá editar la información de un profesor, además de sus docencias.
3	medico	La pantalla "medico" podrá mostrar información básica de un médico, como su nombre y apellidos. También, los usuarios podrán viajar a las pantallas "lista_informes" y "lista_alumnos" que tengan que ver con dicho médico si tienen los permisos necesarios para ello. Solo el rol "administración" podrá editar la información básica de un médico.
4	administracion	La pantalla "administracion" podrá mostrar información básica de un administrativo, como su nombre y apellidos. También se podrá viajar a las pantallas "lista_profesores", "lista_medicos", "lista_alumnos", "lista_asignaturas", "lista_pantallas", "lista_informes", "lista_matriculas", "lista_docencias" y "auditoria", teniendo completa información sobre todos los datos. Solo el rol "administración" podrá editar la información de un administrativo, además de sus docencias.

Imagen 78: Lista con la información sobre las pantallas de la aplicación.

B.21 Lista de todas las auditorías

Para acceder a la lista de todas las auditorías, solo hay que pulsar el botón **“Auditorías”** en el perfil principal de un administrador. El sistema llevará al usuario a una tabla con todas las auditorías registradas por la aplicación.



The screenshot shows the 'Lista de auditorías' page. At the top, there is a navigation bar with 'Gestión Diversidad', 'Cerrar Sesión', and 'Perfil Principal'. The user is identified as 'admin1 Rol: Administrador/a'. Below the title, there is a 'Mostrar' dropdown set to '10' and a search box. The table has the following data:

Nif usuario	Pantalla	Acción	Fecha de la acción
111AD	lista_matriculas	Nueva matrícula con NIF alumno: 33333333A y asignatura: Álgebra	2023/05/31 23:37:16
111AD	lista_matriculas	Matrícula borrada entre el alumno con NIF: 11111111A y la asignatura: Álgebra	2023/05/31 23:16:45
111AD	lista_matriculas	Nueva matrícula con NIF alumno: 11111111A y asignatura: Álgebra	2023/05/31 23:16:29
111AD	lista_matriculas	Nueva matrícula con NIF alumno: 33333333A y asignatura: Lengua	2023/05/31 23:14:54
111AD	lista_matriculas	Nueva matrícula con NIF alumno: 22222222A y asignatura: Cálculo	2023/05/31 23:14:49
111AD	lista_matriculas	Nueva matrícula con NIF alumno: 11111111A y asignatura: Programación	2023/05/31 23:12:37
111AD	lista_docencias	Nueva docencia con NIF profesor: 44444444P y asignatura: Álgebra	2023/05/31 23:12:03
111AD	lista_docencias	Nueva docencia con NIF profesor: 33333333P y asignatura: Lengua	2023/05/31 23:11:54
111AD	lista_docencias	Nueva docencia con NIF profesor: 22222222P y asignatura: Cálculo	2023/05/31 23:11:38

Imagen 79: Lista de todas las auditorías.

B.22 Crear un nuevo alumno

Para crear un nuevo alumno, el usuario debe encontrarse en una lista de alumnos (como en el punto B.15). Una vez en dicha pantalla, debajo de la tabla habrá un botón llamado “**Crear nuevo alumno**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde podrá rellenar los atributos necesarios para la creación de un nuevo alumno. Una vez rellenados los atributos de forma correcta, solo falta que el usuario pulse el botón “**Crear alumno**” para completar dicha acción.



The screenshot shows a web interface for creating a new student. At the top left, there is a logo and the text 'Gestión Diversidad'. To its right are two buttons: 'Cerrar Sesión' (red) and 'Perfil Principal' (blue). On the top right, the user information is displayed: 'Usuario: admin1 Rol: Administrador/a'. The main heading is 'Crear nuevo/a alumno/a', followed by a sub-heading: 'Un/a alumno/a deberá tener un informe asignado como mínimo'. The form contains several input fields: 'Nif:', 'Usuario:', 'Password:', 'Confirmar Password:', 'Nombre:', 'Primer apellido:', and 'Segundo apellido:'. Below these is a section for PDF uploads with a button 'Elegir archivo' and the text 'No se ha seleccionado ningún archivo'. The next field is 'Médico titular:' with a search input 'Buscar médico...'. At the bottom of the form are two buttons: 'Crear alumno' (blue) and 'Volver atrás' (black).

Imagen 80: Pantalla de creación de un alumno.

B.23 Borrar un alumno

Para borrar un alumno, el usuario debe encontrarse en una lista de alumnos (como en el punto B.15). Una vez en dicha pantalla, en las filas de la tabla, a la derecha del todo, se encuentra un botón llamado “**Borrar**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde preguntará al usuario si está seguro de borrar el alumno en cuestión. Si el usuario pulsa el botón “**Borrar Alumno/a**”, habrá eliminado con éxito el usuario de tipo alumno.



Imagen 81: Pantalla de borrado del alumno Juan con NIF: 11111111A.

B.24 Modificar un alumno

Para modificar un alumno, el administrador debe encontrarse en el perfil principal de un alumno (como en el punto B.9). Una vez en dicha pantalla, habrá un botón llamado “**Modificar alumno**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde mostrará los atributos con los valores actuales del alumno. El administrador puede modificar esos valores, y si desea que se guarden los cambios realizados, solo tendría que pulsar el botón “**Modificar**”.

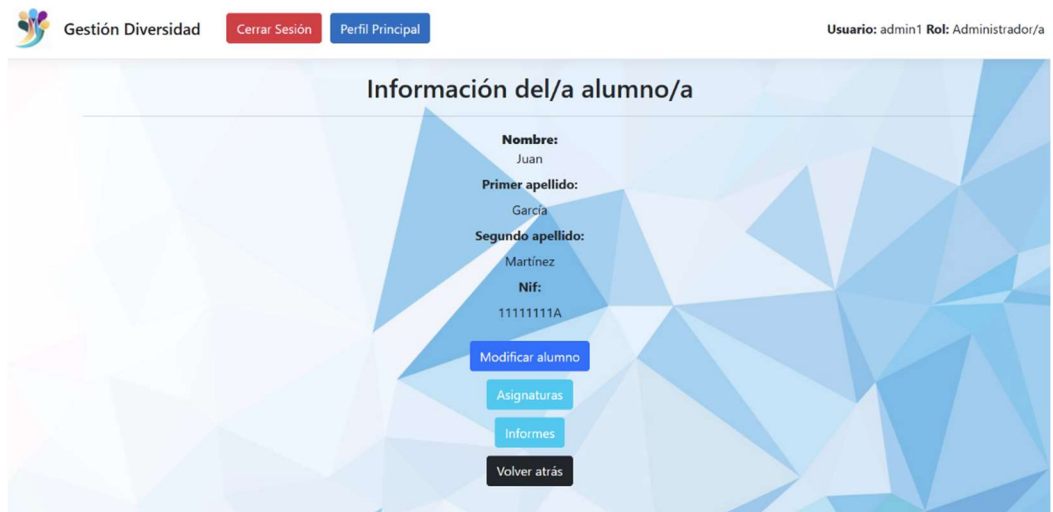


Imagen 82: Perfil principal de un alumno accedido como administrador.

The screenshot shows the 'Gestión Diversidad' interface. At the top left is the logo and the text 'Gestión Diversidad'. To its right are two buttons: 'Cerrar Sesión' (red) and 'Perfil Principal' (blue). At the top right, it says 'Usuario: admin1 Rol: Administrador/a'. The main content area is titled 'Modificar alumno/a'. It contains a form with the following fields: 'Nombre:' with the value 'Juan'; 'Primer Apellido:' with the value 'García'; 'Segundo Apellido:' with the value 'Martínez'; 'Usuario:' with the value 'juangm53'; 'Password:' with masked characters '*****'; and 'Confirmar Password:' with masked characters '*****'. Below the form are two buttons: 'Modificar' (blue) and 'Volver atrás' (black).

Imagen 83: Pantalla de modificación de un alumno.

B.25 Crear un nuevo profesor

Para crear un nuevo profesor, el usuario debe encontrarse en una lista de profesores (como en el punto B.13). Una vez en dicha pantalla, debajo de la tabla habrá un botón llamado “**Crear nuevo profesor**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde podrá rellenar los atributos necesarios para la creación de un nuevo profesor. Una vez rellenados los atributos de forma correcta, solo falta que el usuario pulse el botón “**Crear profesor**” para completar dicha acción.

The screenshot shows the 'Gestión Diversidad' interface. At the top left is the logo and the text 'Gestión Diversidad'. To its right are two buttons: 'Cerrar Sesión' (red) and 'Perfil Principal' (blue). At the top right, it says 'Usuario: admin1 Rol: Administrador/a'. The main content area is titled 'Crear nuevo/a profesor/a'. It contains a form with the following fields: 'Nif:', 'Usuario:', 'Password:', 'Confirmar Password:', 'Nombre:', 'Primer apellido:', and 'Segundo apellido:'. Below the form are two buttons: 'Crear profesor' (blue) and 'Volver atrás' (black).

Imagen 84: Pantalla de creación de un profesor.

B.26 Borrar un profesor

Para borrar un profesor, el administrador debe encontrarse en una lista de profesores (como en el punto B.13). Una vez en dicha pantalla, en las filas de la tabla, a la derecha del todo, se encuentra un botón llamado “**Borrar**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde preguntará al usuario si está seguro de borrar el profesor en cuestión. Si el usuario pulsa el botón “**Borrar Profesor**”, habrá eliminado con éxito el usuario de tipo profesor.



Imagen 85: Pantalla de borrado del profesor Alejandro con NIF: 11111111P.

B.27 Modificar un profesor

Para modificar un profesor, el administrador debe encontrarse en el perfil principal de un profesor (como al final del punto B.13). Una vez en dicha pantalla, habrá un botón llamado “**Modificar profesor**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde mostrará los atributos con los valores actuales del profesor. El administrador puede modificar esos valores, y si desea que se guarden los cambios realizados, solo tendría que pulsar el botón “**Modificar**”.

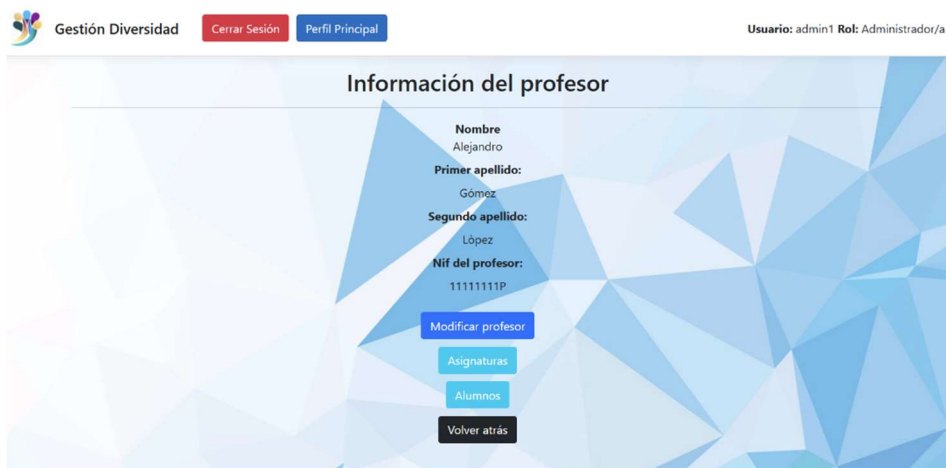


Imagen 86: Perfil principal de un profesor accedido como administrador.

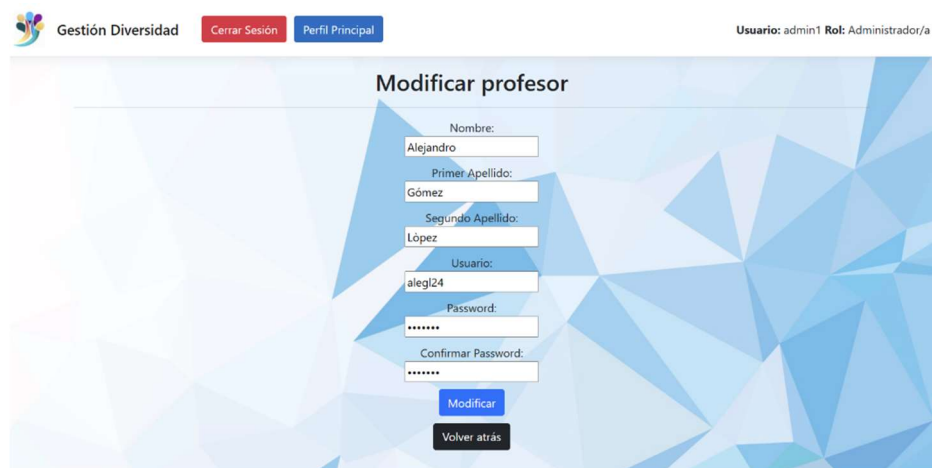


Imagen 87: Pantalla de modificación de un profesor.

B.28 Crear un nuevo médico

Para crear un nuevo médico, el usuario debe encontrarse en una lista de médicos (como en el punto **B.14**). Una vez en dicha pantalla, debajo de la tabla habrá un botón llamado “**Crear nuevo médico**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde podrá rellenar los atributos necesarios para la creación de un nuevo médico. Una vez rellenados los atributos de forma correcta, solo falta que el usuario pulse el botón “**Crear médico**” para completar dicha acción.

The screenshot shows a web interface for creating a new doctor. At the top left, there is a logo and the text 'Gestión Diversidad'. To its right are two buttons: 'Cerrar Sesión' (red) and 'Perfil Principal' (blue). At the top right, the user information is displayed: 'Usuario: admin1 Rol: Administrador/a'. The main heading is 'Crear nuevo médico'. Below this, there is a form with the following fields: 'Nif', 'Usuario', 'Password', 'Confirmar Password', 'Nombre', 'Apellido1', and 'Apellido2'. At the bottom of the form are two buttons: 'Crear médico' (blue) and 'Volver atrás' (black).

Imagen 88: Pantalla de creación de un médico.

B.29 Borrar un médico

Para borrar un médico, el administrador debe encontrarse en una lista de médicos (como en el punto **B.14**). Una vez en dicha pantalla, en las filas de la tabla, a la derecha del todo, se encuentra un botón llamado **“Borrar”**. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde preguntará al usuario si está seguro de borrar el médico en cuestión. Si el usuario pulsa el botón **“Borrar Médico”**, habrá eliminado con éxito el usuario de tipo médico.

The screenshot shows a confirmation dialog box titled '¿Seguro que quieres borrar el médico?'. It displays the details of the doctor to be deleted: 'Nombre: Carlos', 'Primer Apellido: Ramirez', 'Segundo Apellido: Navarro', and 'Nif: 33333333M'. At the bottom of the dialog are two buttons: 'Borrar Médico' (red) and 'Volver atrás' (black).

Imagen 89: Pantalla de borrado del médico Carlos con NIF: 33333333M.

B.30 Modificar un médico

Para modificar un médico, el administrador debe encontrarse en el perfil principal de un médico (como al final del punto B.14). Una vez en dicha pantalla, habrá un botón llamado “**Modificar médico**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde, mostrará los atributos con los valores actuales del médico. El administrador puede modificar esos valores, y si desea que se guarden los cambios realizados, solo tendría que pulsar el botón “**Modificar**”.



Imagen 90: Perfil principal de un médico accedido como administrador.

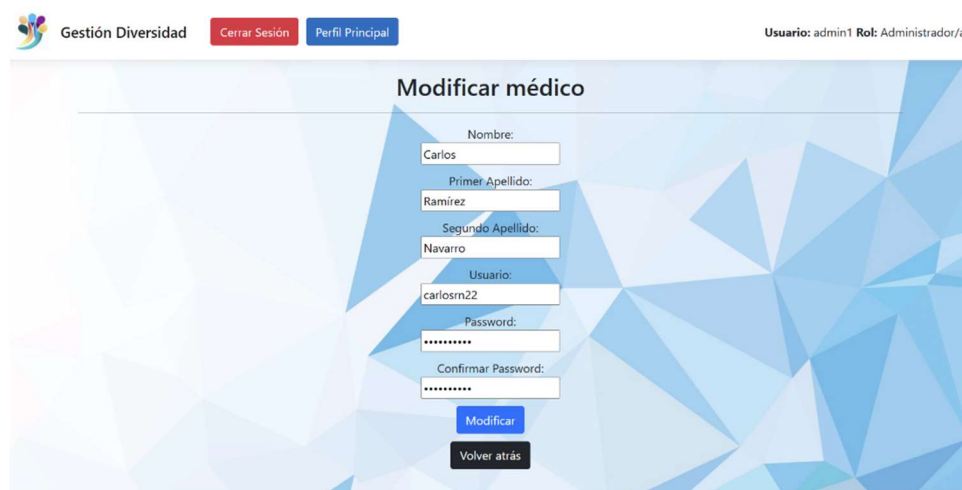


Imagen 91: Pantalla de modificación de un médico.

B.31 Crear una nueva asignatura

Para crear una asignatura, el administrador debe encontrarse en una lista de asignaturas (como en el punto **B.16**). Una vez en dicha pantalla, debajo de la tabla habrá un botón llamado “**Crear nueva asignatura**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde podrá rellenar los atributos necesarios para la creación de una nueva asignatura. Una vez rellenados los atributos de forma correcta, solo falta que el usuario pulse el botón “**Crear asignatura**” para completar dicha acción.



Imagen 92: Pantalla de creación de una asignatura.

B.32 Borrar una asignatura

Para borrar una asignatura, el administrador debe encontrarse en una lista de asignaturas (como en el punto **B.16**). Una vez en dicha pantalla, se encuentra un botón llamado “**Borrar**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde preguntará al usuario si está seguro de borrar la asignatura en cuestión. Si el usuario pulsa el botón “**Borrar Asignatura**”, habrá eliminado con éxito la asignatura.



Imagen 93: Pantalla de borrado de la asignatura con nombre: Cálculo.

B.33 Modificar nombre de una asignatura

Para modificar una asignatura, el administrador debe encontrarse en una lista de asignaturas (como en el punto B.16). Una vez en dicha pantalla, habrá un botón llamado **“Modificar”**. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde, mostrará el nombre actual de la asignatura. El administrador puede modificar este valor, y si desea que se guarden los cambios realizados, solo tendría que pulsar el botón **“Modificar”**.



Imagen 94: Pantalla de modificación de una asignatura.

B.34 Crear una nueva matrícula

Para crear una matrícula, el administrador debe encontrarse en la lista de matrículas (específicamente el punto B.17). Una vez en dicha pantalla, debajo de la tabla habrá un botón llamado **“Crear nueva matrícula”**. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde podrá buscar los alumnos y asignaturas necesarias para la creación de una nueva matrícula. Una vez seleccionado tanto el alumno como la asignatura, solo falta que el usuario pulse el botón **“Crear matrícula”** para completar dicha acción.



Imagen 95: Pantalla de creación de una matrícula.

B.35 Borrar una matrícula

Para borrar una matrícula, el administrador debe encontrarse en la lista de matrículas (específicamente el punto **B.17**). Una vez en dicha pantalla, se encuentra un botón llamado “**Borrar**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde preguntará al usuario si está seguro de borrar la matrícula en cuestión. Si el usuario pulsa el botón “**Borrar matrícula**”, habrá eliminado con éxito la matrícula.



Imagen 96: Pantalla de borrado de una matrícula con el alumno Juan (NIF: 11111111A) y asignatura Cálculo.

B.36 Crear una nueva docencia

Para crear una docencia, el administrador debe encontrarse en la lista de docencias (específicamente el punto **B.18**). Una vez en dicha pantalla, debajo de la tabla habrá un botón llamado “**Crear nueva docencia**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde podrá buscar los profesores y asignaturas necesarias para la creación de una nueva docencia. Una vez seleccionado tanto el profesor como la asignatura, solo falta que el usuario pulse el botón “**Crear docencia**” para completar dicha acción.



Imagen 97: Pantalla de creación de una docencia.

B.37 Borrar una docencia

Para borrar una docencia, el administrador debe encontrarse en la lista de docencias (específicamente el punto **B.18**). Una vez en dicha pantalla, se encuentra un botón llamado “**Borrar**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde preguntará al usuario si está seguro de borrar la docencia en cuestión. Si el usuario pulsa el botón “**Borrar docencia**”, habrá eliminado con éxito la docencia.



Imagen 98: Pantalla de borrado de una docencia con el profesor Alejandro (NIF: 11111111P) y asignatura Cálculo.

B.38 Crear un nuevo informe

Para insertar un nuevo informe, el administrador debe encontrarse en una lista de informes (como en el punto **B.19**, aunque puede ser cualquier lista de informes). Una vez en dicha pantalla, debajo de la tabla habrá un botón llamado “**Crear nuevo Informe**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde podrá buscar los médicos y alumnos necesarios para la creación de un nuevo informe. Además, tendrá que proporcionar un documento en formato PDF para el contenido del informe mediante el uso del botón “**Elegir archivo**”. Una vez rellenados todos los campos, solo falta que el usuario pulse el botón “**Crear informe**” para completar dicha acción.



Imagen 99: Pantalla de creación de un informe.

B.39 Borrar un informe

Para borrar un informe, el administrador debe encontrarse en una lista de informes (como en el punto **B.19**, aunque puede ser cualquier lista de informes). Una vez en dicha pantalla, se encuentra un botón llamado “**Borrar**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde preguntará al usuario si está seguro de borrar el informe en cuestión. Si el usuario pulsa el botón “**Borrar Informe**”, habrá eliminado con éxito el informe.



Imagen 100: Pantalla de borrado de un informe.

B.40 Cambiar el médico titular de un informe

Para cambiar el médico titular de un informe, el administrador debe encontrarse en una lista de informes (como en el punto **B.19**, aunque puede ser cualquier lista de informes). Una vez en dicha pantalla, se encuentra un botón llamado “**Cambiar médico**”. Si el usuario pulsa ese botón, el sistema llevará al usuario a otra vista donde podrá seleccionar otro médico titular para un informe. Si el usuario pulsa el botón “**Cambiar médico**” después de haber elegido un nuevo médico, habrá cambiado el médico titular de un informe con éxito.



Imagen 101: Pantalla de cambio de médico titular.

B.41 Cerrar sesión de un usuario

Para cerrar sesión, lo único que tiene realizar el usuario es pulsar el botón llamado “**Cerrar Sesión**”. Una vez pulsado dicho botón, el sistema redirigirá al usuario a la pantalla de inicio de sesión. Da igual en que vista se encuentre el usuario mientras haya iniciado sesión, pues el botón siempre estará arriba a la izquierda en la barra de navegación, al lado de del título de la aplicación web, (el botón de color rojo).

B.42 Volver al perfil principal de un usuario

Para volver al perfil principal del usuario que haya iniciado sesión, lo único que tiene que hacer el usuario es pulsar el botón llamado “**Perfil Principal**”. Una vez pulsado dicho botón, el sistema redirigirá al usuario a su perfil principal. Da igual en que vista se encuentre el usuario mientras haya iniciado sesión, pues el botón siempre estará arriba a la izquierda en la barra de navegación, al lado del botón “Cerrar Sesión”, (es el botón de color azul).



UNIVERSIDAD
DE MÁLAGA

| uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA