

# ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



## TRABAJO FIN DE GRADO

*APLICACIÓN WEB PARA LA  
COMUNICACIÓN ENTRE ALUMNOS,  
PADRES Y PROFESORES DE CENTROS  
ESCOLARES*

**Titulación:** Ingeniería en Tecnologías de la  
Telecomunicación  
**Mención:** Telemática  
**Autor/a:** Leticia Esther Santana Robaina  
**Tutor/a:** Dr. Luis Hernández Acosta  
**Fecha:** Julio 2024



## Resumen

La educación en España ha experimentado una significativa evolución a lo largo de los años, adaptándose en su mayoría a la tecnología. El sistema educativo español se estructura en varias etapas: educación infantil, primaria, secundaria obligatoria, bachillerato y formación profesional, así como la educación superior en universidades y otras instituciones.

La incorporación de la tecnología en el sistema educativo español ha sido una prioridad en las últimas décadas. El gobierno y diversas instituciones educativas han impulsado iniciativas para integrar las TIC (Tecnologías de la Información y la Comunicación) en las aulas, con el objetivo de mejorar la calidad educativa y preparar a los estudiantes para un mundo cada vez más digital.

En este contexto, surge *SchoolApp*, una plataforma web diseñada para fomentar la participación de padres, alumnos y personal docente y no docente en la educación secundaria. Este portal web permite la conexión entre todos los integrantes de un instituto, así como la gestión digitalizada de los perfiles de usuario, los cursos horarios y demás servicios.

Con *SchoolApp*, se pretende crear un espacio web inteligente y seguro donde los alumnos aprenderán a hacer un uso responsable de la tecnología, donde los padres podrán estar al día de todas las actividades y necesidades de sus hijos y donde los profesores tendrán una herramienta que mejorará la calidad del sector educativo en este país.



# Abstract

Education in Spain has experienced significant evolution over the years, mostly adapting to technology. The Spanish educational system is structured into several stages: early childhood education, primary education, compulsory secondary education, high school, and vocational training, as well as higher education in universities and other institutions.

In recent decades, incorporating technology into the Spanish educational system has been a priority. The government and various educational institutions have promoted initiatives to integrate ICT (Information and Communication Technologies) into classrooms, aiming to improve educational quality and prepare students for an increasingly digital world.

In this context, *SchoolApp* emerges, a web platform designed to encourage the participation of parents, students, and both teaching and non-teaching staff in secondary education. This web portal facilitates the connection among all members of a school and allows the digital management of user profiles, course schedules, and other services.

With *SchoolApp*, the goal is to create an intelligent and secure web space where students will learn how to use technology responsibly, parents will stay updated on their children's activities and needs, and teachers will find a tool that enhances the quality of the education sector in this country.



## Tabla de Contenido

Capítulo 1. Introducción.....	1
1.1 Contexto.....	2
1.2 Motivación.....	4
1.3 Objetivos.....	4
1.4 Comparativa de aplicaciones.....	6
1.4.1 LiveKid.....	6
1.4.2 SM Educamos.....	7
1.4.3 Addittio App.....	7
1.4.4 Ventajas y desventajas.....	8
1.4.5 Funcionalidades de las aplicaciones.....	9
1.5 Estructura del documento.....	10
Capítulo 2. Tecnologías Software.....	<b>¡Error! Marcador no definido.</b>
2.1 Angular.....	13
2.2 Firebase.....	13
2.3 TeleportHQ.....	15
2.4 Visual Studio Code.....	15
2.5 TypeScript.....	16
2.6 DBDiagram.io.....	17
Capítulo 3. Análisis.....	18
3.1 Contexto de la aplicación.....	19
Capítulo 4. Diseño.....	25
4.1 Contexto del diseño.....	26
4.2 Diseño de la base datos.....	26
4.2.1 Colección Usuarios.....	28
4.2.2 Colección Chat.....	30
4.2.3 Colección Municipios.....	31

4.2.4 Colección Roles .....	31
4.2.5 Colección Anuncios.....	32
4.2.6 Colección Recordatorios .....	33
4.2.7 Colección Grupos.....	33
4.2.8 Colección Curso .....	34
4.2.9 Colección Horarios .....	34
4.2.10 Colección Boletines .....	35
4.3 Mockups.....	35
4.3.1 Pantalla Home .....	36
4.3.2 Pantalla Presentación.....	37
4.3.3 Pantalla Identificación.....	38
4.3.4 Pantalla de Menú .....	38
4.3.5 Pantalla de Recordatorios .....	39
4.3.6 Pantalla Alta Empleado .....	40
4.3.7 Pantalla Alta Alumno .....	41
4.3.8 Pantalla Buscador .....	41
4.3.9 Pantalla Configurar Horarios .....	42
4.3.10 Pantalla Nuevo Chat.....	42
4.3.11 Pantalla Chat.....	43
4.3.11 Pantalla Boletines.....	43
4.3.11 Pantalla Cursos .....	44
4.3.12 Pantalla Anuncios.....	45
4.3.13 Pantalla Incidencias .....	46
4.4 Diseño del logo de la aplicación.....	46
4.5 Patrón de diseño.....	47
Capitulo 5. Implementación .....	48
5.1 Contexto de la implementación.....	49



5.2 Estructura del proyecto .....	49
5.3 Interfaces .....	52
5.3.1 Interfaz de Usuario .....	52
5.3.2 Interfaz de Chatl .....	54
5.3.3 Interfaz de Anuncio .....	54
5.3.4 Interfaz de Curso .....	55
5.3.5 Interfaz de Grupo .....	55
5.3.6 Interfaz de Municipio .....	55
5.3.7 Interfaz de Horariol .....	56
5.3.8 Interfaz de Incidencia .....	56
5.3.9 Interfaz de Notas .....	57
5.3.10 Interfaz de Recordatoriol .....	57
5.3.11 Interfaz de Rol .....	58
5.4 Servicios .....	58
5.4.1 Servicio de Usuario .....	58
5.4.2 Servicio de Chat .....	60
5.4.3 Servicio de Auth .....	60
5.4.4 Servicio de Cursos .....	61
5.4.5 Servicio de Grupos .....	61
5.4.6 Servicio de Municipios .....	61
5.4.7 Servicio de Anuncio .....	62
5.4.8 Servicio de Horario .....	62
5.4.9 Servicio de Incidencia .....	63
5.4.10 Servicio de Recordatorios .....	63
5.5 Componentes principales (páginas) .....	64
5.5.1 Componente Login .....	65
5.5.2 Componente Menú .....	65

5.5.3	Componente <i>AltaEmpleado</i> .....	66
5.5.4	Componente <i>AltaAlumno</i> .....	68
5.5.5	Componente <i>BuscadorPersonal</i> .....	69
5.5.6	Componente <i>Recordatorios</i> .....	70
5.5.7	Componente <i>PlantillaHorario</i> .....	71
5.5.8	Componente <i>Chat</i> .....	72
5.5.9	Componente <i>Horario</i> .....	73
5.5.10	Componente <i>Incidencias</i> .....	74
5.5.11	Componente <i>TablonAnuncios</i> .....	75
5.5.12	Componente <i>Cursos</i> .....	76
5.6	Componentes específicos.....	77
5.6.1	Componente <i>HeaderNoLogin</i> .....	77
5.6.2	Componente <i>HeaderLogin</i> .....	78
5.6.3	Componente <i>Alumno-Curso</i> .....	78
5.6.4	Componente <i>Anuncio</i> .....	78
5.6.5	Componente <i>Bloque-Chat</i> .....	78
5.6.6	Componente <i>Identificación</i> .....	78
5.6.7	Componente <i>Recordatorio</i> .....	79
5.6.8	Componente <i>User-Welcome</i> .....	79
5.6.9	Componente <i>usuario-alumno</i> .....	79
5.6.10	Componente <i>usuario-chat</i> .....	79
Capítulo 6.	Guía de usuario.....	80
6.1	Contexto de la guía de usuario.....	81
6.2	Guía de usuario.....	81
6.2.1	Inicio.....	81
6.2.2	¿Quiénes somos?.....	82
6.2.3	Login.....	85

6.2.4 Menú Administrador .....	85
6.2.5 Dar de alta a un empleado .....	86
6.2.6 Dar de alta a un alumno .....	87
6.2.7 Buscador .....	88
6.2.8 Alta alumno (perfil) .....	90
6.2.8 Recordatorios .....	91
6.2.9 Plantilla Horario .....	92
6.2.10 Nuevo Chat .....	93
6.2.11 Menú Alumno .....	94
6.2.12 Mi perfil.....	95
6.2.13 Horario.....	96
6.2.14 Incidencias (alumno) .....	96
6.2.15 Tablón anuncios (alumno).....	97
6.2.16 Menú Conserjería .....	98
6.2.17 Menú Jefatura .....	98
6.2.18 Acceso a cursos .....	99
6.2.19 Añadir incidencia .....	100
6.2.20 Mi perfil (empleado).....	101
6.2.21 Tablón Anuncios (Añadir) .....	102
6.2.22 Menú Padres .....	103
6.2.23 Incidencias (justificar).....	104
6.2.24 Menú Profesores .....	105
Capítulo 7. Conclusiones y líneas futuras .....	106
7.1 Contexto de la Conclusión .....	107
7.2 Conclusiones .....	108
7.3 Líneas futuras .....	110
Presupuesto .....	112

Desglose del presupuesto .....	113
Recursos materiales .....	113
Trabajo tarifado por tiempo empleado.....	114
Costes asociados con la redacción del documento .....	115
Derechos de visado COITT .....	116
Gastos de tramitación y envío.....	116
Aplicación de impuestos .....	117



## Índice de Figuras

Figura 1 - Logo LiveKid .....	6
Figura 2 - Logo SM Educamos .....	7
Figura 3 - Logo AdditioApp .....	7
Figura 4 - Logo Angular .....	13
Figura 5 - Logo Firebase .....	14
Figura 6 - Logo TeleportHQ.....	15
Figura 7 - Logo Visual Studio Code.....	16
Figura 8 - Logo TypeScript .....	16
Figura 9 - Logo DBDiagram.io.....	17
Figura 10 - Esquema de Gestión de Datos .....	19
Figura 11 - Diagrama de funcionalidades comunes .....	23
Figura 12 - Diagrama de funcionalidades específicas .....	24
Figura 13 - Esquema de base de datos.....	27
Figura 14 - Colección de Usuarios .....	29
Figura 15 - Colección Chat.....	30
Figura 16 - Colección de Municipios.....	31
Figura 17 - Colección de Roles .....	32
Figura 18 - Colección de Anuncios.....	32
Figura 19 - Colección de Recordatorios .....	33
Figura 20 - Colección de Grupos.....	33
Figura 21 - Colección de Curso .....	34
Figura 22 - Colección de Horarios .....	34
Figura 23 - Colección de Boletines.....	35
Figura 24 - Diseño pantalla Home .....	37
Figura 25 - Diseño de pantalla Presentación.....	37
Figura 26 - Diseño de Formulario de Contacto.....	38
Figura 27 - Diseño de Pantalla de Identificación .....	38
Figura 28 - Diseño de Menú principal de Administradores .....	39
Figura 29 – Diseño de Pantalla de Lista de Recordatorios.....	40
Figura 30 - Diseño de Pantalla de Formulario de Alta de Empleado .....	40
Figura 31 - Diseño de Pantalla de Formulario de Dar de Alta a Alumno .....	41
Figura 32 - Diseño de Pantalla de Buscador de Usuarios .....	42

Figura 33 - Diseño de Pantalla de Configuración de Horarios.....	42
Figura 34 - Diseño de Pantalla de Buscador de Usuarios para Chat .....	43
Figura 35 - Diseño de Pantalla de Chat.....	43
Figura 36 - Diseño de Pantalla de Boletín de Notas.....	44
Figura 37 - Diseño de Pantalla de Lista de Alumnos Por Curso.....	45
Figura 38 - Diseño de Pantalla de Anuncios .....	45
Figura 39 - Diseño de Pantalla de Incidencias del Alumno .....	46
Figura 40 - Logo SchoolApp.....	46
Figura 41 - Diagrama de Flujo de Datos en Patrón M-V-VM.....	47
Figura 42 - Estructura de Carpetas del Proyecto.....	50
Figura 43 - Pantalla Principal .....	82
Figura 44 - Pantalla Principal Características.....	82
Figura 45 - Presentación Parte 1.....	83
Figura 46 - Presentación Parte 2.....	83
Figura 47 - Presentación Parte 3.....	84
Figura 48 - Presentación Formulario De Contacto .....	84
Figura 49 - Respuesta al Formulario de Contacto .....	85
Figura 50 - Identificación .....	85
Figura 51 - Menú de Administrador .....	86
Figura 52 - Ruta de Navegación en Acceso a Menú de Administrador .....	86
Figura 53 - Formulario para dar de Alta a un Empleado.....	87
Figura 54 - Respuesta al Registrar Empleado Correctamente .....	87
Figura 55 - Pantalla de Formulario para Dar de Alta a un Alumno .....	88
Figura 56 - Respuesta de Alumno Registrado Correctamente .....	88
Figura 57 - Buscador de Usuarios .....	89
Figura 58 - Buscador de Usuarios por nombre.....	89
Figura 59 - Ruta de navegación para que un Adminsitrador acceda al perfil de un Alumno .....	90
Figura 60 - Respuesta de datos Actualizados Correctamente .....	90
Figura 61 - Confirmar si se desea Dar de Baja al Empleado.....	90
Figura 62 - Confirmación de eliminación de datos.....	91
Figura 63 - Lista de Recordatorios .....	91
Figura 64 - Añadir Recordatorio .....	92
Figura 65 - Respuesta a Recordatorio creado correctamente .....	92

Figura 66 - Plantilla de Horario .....	93
Figura 67 - Respuesta al Guardar Horario Correctamente .....	93
Figura 68 - Chat con Usuario .....	94
Figura 69 - Menú de Alumnos .....	95
Figura 70 - Alumno accediendo a su Perfil .....	95
Figura 71 - Alumno Accediendo a su Horario .....	96
Figura 72 - Incidencias del Alumno .....	97
Figura 73 - Anuncios en el Tablón.....	97
Figura 74 - Menú para Conserjes .....	98
Figura 75 - Menú para Jefes de Estudio.....	99
Figura 76 - Listado de Alumnos por Curso .....	99
Figura 77 - Añadir Incidencia.....	100
Figura 78 - Formulario para Crear Incidencia.....	101
Figura 79 - Respuesta a Incidencia enviada correctamente.....	101
Figura 80 - Empleado accediendo a su Perfil .....	102
Figura 81 - Jefe de Estudios añadiendo un Anuncio al tablón.....	102
Figura 82 - Respuesta a Anuncio añadido correctamente.....	103
Figura 83 - Menú para Padres.....	103
Figura 84 - Incidencias desde Perfil de Padre .....	104
Figura 85 - Justificación de Faltas .....	104
Figura 86 - Menú de Profesores .....	105





## Índice de Tablas

Tabla 1 - Comparativa de aplicaciones .....	10
Tabla 2 - Descripción de los campos de la colección Usuarios.....	30
Tabla 3 - Descripción de los campos de la colección Chat .....	31
Tabla 4 - Descripción de los campos de la colección Anuncios .....	33
Tabla 5 - Métodos del Servicio de Usuarios .....	59
Tabla 6 - Métodos del Servicio de Chat.....	60
Tabla 7 - Métodos del Servicio de Auth.....	60
Tabla 8 - Métodos del Servicio de Cursos.....	61
Tabla 9 - Métodos del Servicio de Grupos .....	61
Tabla 10 - Métodos del Servicio de Municipios .....	62
Tabla 11 - Métodos del Servicio de Anuncio .....	62
Tabla 12 - Métodos del Servicio de Horario .....	63
Tabla 13 - Métodos del Servicio de Incidencias .....	63
Tabla 14 - Métodos del Servicio de Recordatorios.....	64
Tabla 15 - Métodos del Componente Login.....	65
Tabla 16 - Métodos de los componentes de Menú.....	66
Tabla 17 - Métodos del Componente Alta Empleado .....	68
Tabla 18 - Métodos del Componente Alta Alumno .....	69
Tabla 19 - Métodos del Componente Buscador .....	70
Tabla 20 - Métodos del Componente Recordatorio.....	70
Tabla 21 - Métodos del Componente Nuevo Recordatorio.....	71
Tabla 22 - Métodos del Componente Plantilla Horario .....	72
Tabla 23 - Métodos del Componente Buscador Chat.....	72
Tabla 24 - Métodos del Componente Chat.....	73
Tabla 25 - Métodos del Componente Horario.....	74
Tabla 26 - Métodos del componente Incidencias .....	74
Tabla 27 - Métodos del Componente Anuncios.....	75
Tabla 28 - Métodos del componente Cursos.....	76
Tabla 29 - Métodos del Componente Añadir Incidencia.....	77
Tabla 30 - Cálculo de Recursos Materiales.....	114
Tabla 31 - Factor de Corrección según el COITT.....	115
Tabla 32 - Coste Total.....	117



## Índice de Ecuaciones

Ecuación 1 - Coste amortización .....	113
Ecuación 2 - Trabajo tarificado por tiempo empleado .....	114
Ecuación 3 - Costes de Redacción.....	115
Ecuación 4 - Derechos de Visado COITT.....	116



# Capítulo 1. Introducción

EN ESTE CAPÍTULO, SE PRESENTA LA CONTEXTUALIZACIÓN DEL PROYECTO, PROPORCIONANDO UN MARCO DE REFERENCIA INTEGRAL QUE ABARCA LOS ANTECEDENTES HISTÓRICOS, LA SITUACIÓN ACTUAL, Y LOS FACTORES SOCIALES, ECONÓMICOS, LEGALES Y TECNOLÓGICOS QUE INFLUYEN EN EL DESARROLLO DE LA APLICACIÓN WEB PARA LA GESTIÓN INTEGRAL DE INSTITUTOS EDUCATIVOS. ADEMÁS, SE ESTABLECERÁN LAS BASES PARA COMPRENDER LA ESTRUCTURA DEL PROYECTO Y SU RELEVANCIA EN EL CONTEXTO EDUCATIVO ACTUAL Y FUTURO.

## 1.1 Contexto

La educación ha evolucionado a lo largo de la historia, desde las primeras formas de escritura en tablas de arcilla, seguidas por los papiros en Egipto datados en el año 2200 a.C, hasta la invención del papel en China durante el siglo II a.C. Desde entonces el papel ha experimentado mejoras continuas y fue durante muchos siglos el soporte principal para la transmisión del conocimiento.

A medida que avanzó la era digital, surgieron nuevas tecnologías que gradualmente reemplazaron al papel en muchos ámbitos. La popularización de las computadoras personales, dispositivos móviles y la conectividad a Internet transformaron la manera en que la información se crea, comparte y almacena. La digitalización ha permitido que documentos que anteriormente requerían papel, como informes, libros y comunicaciones escritas, sean ahora accesibles en formatos electrónicos, facilitando el almacenamiento, la búsqueda y la distribución instantánea de información en todo el mundo.[1]

En la actualidad, la tecnología digital ha transformado radicalmente el panorama educativo. Las plataformas digitales han reemplazado los métodos tradicionales de enseñanza y gestión escolar, ofreciendo herramientas que facilitan la comunicación instantánea entre estudiantes, padres y docentes, la gestión de horarios, la administración de recursos educativos y la evaluación del rendimiento académico. Este cambio ha sido impulsado por la necesidad de adaptarse a un entorno globalizado y tecnológicamente avanzado, donde la información y el aprendizaje están al alcance de un clic.

Después de la reciente pandemia, se ha acelerado aún más esta transformación digital, evidenciando la necesidad de herramientas digitales que permitan la continuidad educativa en situaciones de crisis. La virtualización de clases, la implementación de plataformas de aprendizaje en línea y la digitalización de procesos administrativos se convirtieron en prioridades para los sistemas educativos de todo el mundo.

Además, a partir de esta situación se ha buscado el desarrollo de herramientas que simplifiquen las tareas administrativas en los centros educativos. Estas soluciones permiten reducir la necesidad de hacer colas para la organización de reuniones de padres y tutorías presenciales entre los padres o los propios alumnos, así como una fluida comunicación entre el equipo educativo de los diferentes cursos. Con la tecnología adecuada, se optimiza el tiempo disponible para padres y profesores, quienes pueden coordinar encuentros virtuales de manera conveniente, eliminando las barreras de horarios restrictivos y facilitando la comunicación directa en cualquier momento. Esta flexibilidad no solo mejora la eficiencia de la gestión escolar, sino que también crea un entorno más cómodo y accesible para todos los involucrados en la educación de los estudiantes.[2]

El sector de la educación ha adoptado progresivamente medidas que potencian y agilizan el aprendizaje, haciendo además esta tarea más amena para los alumnos. Sin embargo, aunque se han adoptado medidas tecnológicas en las aulas, muchas veces los medios disponibles son fragmentados y no abordan todas las necesidades administrativas y pedagógicas de manera global. La falta de productos o servicios específicamente diseñados para la gestión integral de institutos limita la eficiencia y la efectividad en la coordinación entre padres, alumnos y personal implicado.

Por otro lado, hoy en día es innegable el avance del desarrollo web en términos de usabilidad y accesibilidad para los usuarios. Es por eso, que de prácticamente cualquier necesidad que se presente surgirá una aplicación que pueda abordarla, facilitarla y simplificarla.

Esta evolución ha creado una demanda sin precedentes para sitios web y aplicaciones que no solo informen y entretengan, sino que también faciliten transacciones comerciales, educación a distancia, comunicación instantánea y ayuden al usuario en todo lo posible.[3]



## 1.2 Motivación

La motivación de la realización de este Trabajo de Fin de Grado reside en la posibilidad de contribuir a la sociedad, mediante el diseño e implementación de una plataforma web que englobe, agilice y mejore la gestión y la comunicación en los servicios educativos, más concretamente, en la educación secundaria obligatoria, mediante la creación de una aplicación web destinada a todos los miembros que componen un centro docente.

Se pretende ofrecer diversas funcionalidades de fácil manejo y uso que por un lado acompañen al alumno y a sus tutores durante esta etapa del aprendizaje, que además sirva como ejemplo de uso responsable de la tecnología y que además involucre los procesos de administración a los que se puede enfrentar un instituto.

De manera resumida, el objetivo es proporcionar una herramienta sólida y eficaz que apoye el aprendizaje continuo y el desarrollo integral de los estudiantes en el contexto educativo actual y futuro.

## 1.3 Objetivos

El objetivo principal de este proyecto es el desarrollo de una aplicación web, que permita la gestión e interacción de alumnos y padres con el personal docente y no docente de los centros educativos, creando una solución íntegra y única para la educación secundaria obligatoria, promoviendo un uso apropiado de las nuevas tecnologías y evitando el uso de multitud de herramientas para un mismo fin. Para alcanzar este objetivo general se planean los siguientes objetivos específicos:

- **O1: Analizar las ventajas que podría aportar la aplicación.** Para ello se hará un estudio de los beneficios que puede tener que todos los miembros de un centro educativo, desde alumnos y padres, hasta profesores, administradores y conserjes, puedan hacer uso de una misma herramienta para facilitar, cada uno,

sus tareas. Para ello, destacaremos qué ventajas y qué comodidades aportaría a cada grupo de usuarios.

- **O2: Analizar la oferta actual del mercado.** Tener referencias sobre otras aplicaciones cuyo objetivo sea similar al nuestro será beneficioso a la hora de pensar como un usuario para conocer que necesidades nos gustaría tener cubiertas. Además, nos ayudará a distinguir qué tareas del desarrollo podrían presentar inconvenientes y qué tareas serán más sencillas de implementar.
- **O3: Establecer las fases que requiere el desarrollo.** Conocer que pasos debemos seguir en cuanto a planificación, análisis, diseño e implementación nos ayudará en cuanto a la organización y división del trabajo. Cuanto más simplificadas y detalladas tengamos las tareas, más fácil será de abordar el desarrollo del proyecto.
- **O4: Establecer y diseñar las funcionalidades que se van a implementar.** Es importante ponerse en el lugar de cada usuario y definir las necesidades que queremos cubrir. Posteriormente, se hará un diseño que recoja de qué manera se van a presentar dichas funcionalidades al usuario.
- **O5: Definir la estructura de datos.** Antes de empezar a desarrollar la aplicación será conveniente definir qué datos vamos a necesitar y cómo los vamos a almacenar para su posterior tratamiento.
- **O6: Implementar la interfaz de usuario.** Una vez se ha definido qué necesitamos y de qué manera, es el momento de hacer uso de las herramientas pertinentes para llevar a cabo la implementación necesaria para crear una interfaz de usuario intuitiva y fácil de manejar para el usuario.
- **O7: Cargar los datos reales en la aplicación.** Cuando ya esté listo el esqueleto de la aplicación, es decir, todo el diseño, pero sin la funcionalidad completa todavía, será el momento de gestionar como se llevará a cabo la creación, lectura, actualización y eliminación de los datos reales.

## 1.4 Comparativa de aplicaciones

Para ponernos en contexto de las herramientas que ya se ofertan en el mercado, se ha llevado a cabo un análisis acerca de aplicaciones existentes en la actualidad para el mismo servicio. Con este análisis se pretende reforzar los objetivos de la aplicación teniendo en cuenta, qué queremos ofrecer y qué podemos aportar que otras aplicaciones no lo hagan, o simplemente cómo podemos contribuir de una manera más completa. A continuación, se muestran los resultados obtenidos:

### 1.4.1 LiveKid



*Figura 1 - Logo LiveKid*

LiveKid es un reciente sistema integral que ayuda en la gestión de las escuelas infantiles. Abarca procesos desde la facturación automática a la gestión completa del centro, la comunicación con las familias, clases virtuales, tableros de anuncios e informes personalizados.

A pesar de ser una herramienta potente, está enfocado a las escuelas infantiles que comprenden la enseñanza infantil y primaria, cuyo rango de edad oscila entre los 3 y los 10 años. Esta puede ser una edad temprana para esperar un uso apropiado por parte de los alumnos.

Además, maneja un amplio abanico de posibilidades, lo que podría ir en detrimento de su facilidad de uso. Sin embargo, las tareas de gestión y administración la convierten en una herramienta muy completa. [4]

### 1.4.2 SM Educamos



*Figura 2 - Logo SM Educamos*

Es un ecosistema educativo digital abierto, flexible y en continua evolución. Ofrece una atención personalizada a alumnos y a padres e incorpora contenidos digitales al aula y optimiza los procesos de evaluación. Una gran ventaja es la actualización de la información en tiempo real debido a su base de datos compartida.

El principal problema es que, según las reseñas, no cuenta con un gran servicio de soporte. Además, algunos usuarios refieren que es una aplicación poco intuitiva en algunos aspectos.[5]

### 1.4.3 Addittio App



*Figura 3 - Logo AddittioApp*

Es una plataforma que ofrece soluciones integrales para centros educativos, que permite unificar la gestión, las evaluaciones y la comunicación. Unifica toda la gestión académica en una solución, genera informes, boletines y analíticas de aprendizaje, entre otras funcionalidades. Esta plataforma está disponible en España y varios países de Latinoamérica.

En este caso, el único detalle que le faltaría a esta aplicación es que tenga un rol para cada componente del centro, es decir, que incluyera, por ejemplo, el perfil de consejería. Sin embargo, es una de las aplicaciones más completas disponibles en el mercado.[6]

#### 1.4.4 Ventajas y desventajas

Para resumir los puntos más y menos destacados de las aplicaciones que acabamos de ver, recogeremos las características más relevantes en la siguiente tabla:

##### **Ventajas:**

- Facilitan a las instituciones la incorporación de nuevas tecnologías, ya que con una sola herramienta se puede abarcar la mayor parte de la gestión del centro.
- Inculcan el buen uso de la tecnología a edades tempranas. Además, las aplicaciones que cuentan con chat entre usuarios podrían servir de herramienta útil ante casos de acoso escolar.
- Agiliza y simplifica el trabajo de los profesores.
- Promueve la implicación de padres en la educación de sus hijos.

##### **Desventajas:**

- La principal desventaja ante aplicaciones de este tipo es la responsabilidad con el tratamiento y almacenamiento de datos sensibles. Por este motivo, es fundamental atender a lo establecido por la Ley Orgánica de Protección de Datos Personales y Garantía de los Derechos Digitales (LOPDGDD), que incorpora directrices establecidas por el Reglamento General de Protección de Datos (RGPD) de la Unión Europea. Entre los aspectos más relevantes a considerar se encuentran el consentimiento por parte de los usuarios para el tratamiento de datos sensibles o el cumplimiento y la responsabilidad con la normativa.[7]
- La dependencia de la conectividad y la infraestructura tecnológica supone otra desventaja, ya que la actividad está sujeta a fallos en los servidores que podrían derivar en períodos de desconexión o dificultades de acceso.
- La sobreprotección que algunos padres pudieran ejercer sobre sus hijos a través de estas aplicaciones. Al tener acceso directo y continuo a información detallada

sobre el desempeño escolar, comunicaciones con los profesores y actividades de sus hijos, algunos padres podrían tener una tentación de monitorear excesivamente o intervenir constantemente en la vida académica y social de sus hijos.

En conclusión, las aplicaciones web diseñadas para la gestión educativa ofrecen indudables ventajas al agilizar procesos administrativos, fomentar la participación de padres y facilitar el trabajo de los educadores. Sin embargo, es crucial manejar con responsabilidad este tipo de herramientas para sacar la mejor experiencia de usuario posible.

#### 1.4.5 Funcionalidades de las aplicaciones

Tras el análisis del mercado, podemos hacer una comparativa de las funcionalidades que ofrecen estas tres potentes herramientas. A continuación, se muestra una tabla en la que se recogen las principales funcionalidades de estas aplicaciones.

Funcionalidades	LiveKid	SM Educamos	Additio App
Registro de usuarios	Sí	Sí	Sí
Perfiles para padres	Sí	Sí	Sí
Perfil para alumnos	No	Sí	Sí
Administración	Sí	Sí	Sí
Perfil para personal no docente	No	No	No
Chat entre usuarios	Sí	Sí*	Sí
Horario	No	Sí	Sí
Tablón de Anuncios	No	Sí	Sí

<b>Recordatorios personales</b>	No	No	Sí
<b>Incidencias</b>	Sí	Sí	Sí

*Tabla 1 - Comparativa de aplicaciones*

Chat entre usuarios\*: solo especifica notificaciones a padres.

Llegados a este punto podremos ya concretar las bases que debemos tener en cuenta a la hora de realizar nuestro proyecto si queremos aspirar al desarrollo de una aplicación de utilidad. Resumiendo, es necesario hacer de la aplicación un entorno fácil de entender y de utilizar para el usuario, segura y comprometida con el tratamiento de los datos sensibles y que permita la interacción de usuarios.

El siguiente paso será entonces la redacción de los objetivos que se debe cumplir nuestro proyecto para que efectivamente sea una aplicación útil para el sector.

## 1.5 Estructura del documento

Hasta este momento, se ha detallado el contexto en el que tendrá lugar este Trabajo de Fin de Título. En adelante, se detallará el desarrollo propio del proyecto. Para ello, se organizará la información en una serie de capítulos:

- **Capítulo 1: Introducción.** Se presentará una introducción general sobre el proyecto, se incluirán si es necesario los antecedentes más relevantes. También incluye un análisis en el que se especifiquen objetivos concretos que se pretenden alcanzar con el proyecto.
- **Capítulo 2: Tecnologías software.** Se recogerán las tecnologías software y herramientas de las que se debe disponer a lo largo del desarrollo del proyecto.
- **Capítulo 3: Análisis.** Se ofrecerá el planteamiento general de la aplicación web mediante diagramas de bloques que detallen el flujo de esta.

- **Capítulo 4: Diseño.** En este capítulo se mostrará el diseño que inicialmente se había planteado para la aplicación, acompañado del logo que llevará y de los mockups definidos.
- **Capítulo 5: Implementación.** En este capítulo se van a detallar en profundidad las acciones llevadas a cabo para la implementación del diseño propuesto inicialmente, así como la gestión y el tratamiento de los datos.
- **Capítulo 6: Guía de usuario.** En este apartado se va a ofrecer una descripción detallada del manejo de la aplicación desde el punto de vista del usuario.
- **Capítulo 7: Conclusiones.** Por último, se hará retrospectiva del proyecto realizado, en el que se recogerán los problemas enfrentados, los cambios del proyecto con respecto de la idea inicial y la escalabilidad que podría tener el proyecto.



## Capítulo 2. Tecnologías Software

A LO LARGO DE ESTE CAPÍTULO, SE EXPLICARÁN BREVEMENTE LAS HERRAMIENTAS Y TECNOLOGÍAS QUE HAN SIDO NECESARIAS PARA LLEVAR A CABO EL PROYECTO.

## 2.1 Angular

*Angular* es un *framework* de desarrollo web de código abierto, mantenido principalmente por Google, que se utiliza en la creación de aplicaciones web de una sola página (SPA) y aplicaciones web dinámicas. En sus inicios fue lanzado como *AngularJS*, en 2010 y posteriormente, en 2016 se lanzó una versión reescrita, Angular.

Se trata de un *framework* basado en componentes para crear aplicaciones web escalables, además, su inyección de dependencias promueve un código limpio y modular. Aportando una colección de bibliotecas bien integradas que cubren una amplia variedad de características, que incluyen enrutamiento, administración de formularios, comunicación cliente-servidor y más. Las herramientas como Angular CLI y las características del *framework* permiten un desarrollo rápido y eficiente.[8]



Figura 4 - Logo Angular

## 2.2 Firebase

*Firebase* es una plataforma de desarrollo de aplicaciones móviles y web en la nube que proporciona una amplia gama de herramientas y servicios para ayudar a los desarrolladores a construir, mejorar y escalar sus aplicaciones. Originalmente fue desarrollada por *Firebase, Inc.*, y posteriormente adquirida por *Google* en 2014.

*Firestore* se integra perfectamente con otros servicios de *Google Cloud* y ofrece soluciones que cubren una variedad de necesidades en el ciclo de vida de desarrollo de una aplicación.

Sus herramientas son variadas y de fácil uso, considerando que su agrupación simplifica las tareas de gestión a una misma plataforma. Las finalidades de las mismas se pueden dividir en cuatro grupos: desarrollo, crecimiento, monetización y análisis. Es especialmente interesante para no dedicarle tanto tiempo al desarrollo del servidor, tanto en cuestiones de desarrollo como de mantenimiento.

En este proyecto, el producto de *Firestore* que se utiliza es *Cloud Firestore*. Se trata de una base de datos de documentos *NoSQL* en tiempo real, es decir, no relacional, que permite almacenar, sincronizar y consultar fácilmente datos en las aplicaciones web y para dispositivos móviles a escala global.

Este producto utiliza colecciones y documentos para estructurar datos con facilidad. Crea jerarquías para almacena datos relacionados y recuperar los datos necesarios mediante consultas expresivas de manera sencilla. [9]



Figura 5 - Logo Firebase

## 2.3 TeleportHQ

*TeleportHQ* es una potente plataforma de diseño y desarrollo de las interfaces de usuario que facilita la creación de sitios web y aplicaciones web a través de una interfaz visual intuitiva, está impulsada por inteligencia artificial que utiliza código generado por *OpenAI*. Su objetivo principal es acelerar el proceso de desarrollo, permitiendo a los usuarios generar código limpio y eficiente sin necesidad de escribirlo manualmente.

A través de la *API* de Visión de *TeleportHQ*, se pueden convertir los bocetos realizados a mano en diseños digitales pasando a prototipos completamente funcionales en cuestión de minutos. Esta herramienta facilita la exportación del código generado para su integración en proyectos existentes o para su implementación en servidores.[10]



Figura 6 - Logo TeleportHQ

## 2.4 Visual Studio Code

Se trata de un editor de código fuente desarrollado por *Microsoft* que ofrece una interfaz limpia y fácil de usar que soporta múltiples lenguajes de programación. Este editor se ejecuta en el escritorio y está disponible para múltiples sistemas operativos.

Además, ofrece una gran variedad de extensiones que se pueden instalar para tener acceso a nuevas funcionalidades y mejorar el flujo de trabajo. Proporciona también herramientas de depuración para diferentes lenguajes, para facilitar la detección de errores y la comprensión del funcionamiento del código.[11]

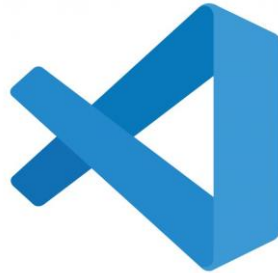


Figura 7 - Logo Visual Studio Code

## 2.5 TypeScript

*TypeScript* es un lenguaje de programación desarrollado y mantenido por *Microsoft*. Es un superconjunto de *JavaScript*, lo que significa que extiende las capacidades de *JavaScript* al agregarle características adicionales, es decir, los programas de *JavaScript* son válidos de *TypeScript*.

La característica principal es que tiene un tipado estático ya que las variables cuentan con un tipo específico de dato, al menos en su mayoría y además los valores sólo se pueden asignar a variables del tipo correspondiente.[12]



Figura 8 - Logo TypeScript

## 2.6 DBDiagram.io

Se trata de un diseñador de bases de datos, eficaz, intuitivo y rápido que está optimizado para el análisis de datos y para ayudar en el diseño de los diagramas de bases de datos.

Permite documentar y planificar las estructuras de datos de una manera eficiente, además de representar las entidades y las relaciones entre ellas. Para ello se puede definir tablas, columnas, tipo de datos y la relaciones (uno a uno, uno a muchos o muchos a muchos).[13]



*Figura 9 - Logo DBDiagram.io*

## Capítulo 3. Análisis.

A LO LARGO DE ESTE CAPÍTULO SE PRESTARÁ UNA DESCRIPCIÓN GENÉRICA DEL FUNCIONAMIENTO QUE PRETENDE TENER LA APLICACIÓN PARA COMPRENDER SUS UTILIDADES. SIN EMBARGO, NO SE PRESTARÁN DETALLES TÉCNICOS ACERCA DE LA IMPLEMENTACIÓN DEL PROYECTO.

### 3.1 Contexto de la aplicación

Para cumplir con los objetivos marcados, debemos implementar una plataforma web que ofrezca diferentes funcionalidades según las necesidades que pueda tener cada uno de los perfiles que conforman un centro educativo de secundaria.

De manera que, tendremos entonces que nuestra plataforma web estará compuesta, por un lado, por el *backend*, formado por el servidor junto con la base de datos, en este caso proporcionado por *Firebase* y *Cloud Firestore*, que se encargará del almacenamiento de los datos de todos los usuarios de la aplicación. Por otro lado, el *frontend*, una interfaz de usuario que, en constante comunicación con la base de datos, gestionará el manejo de estos datos.

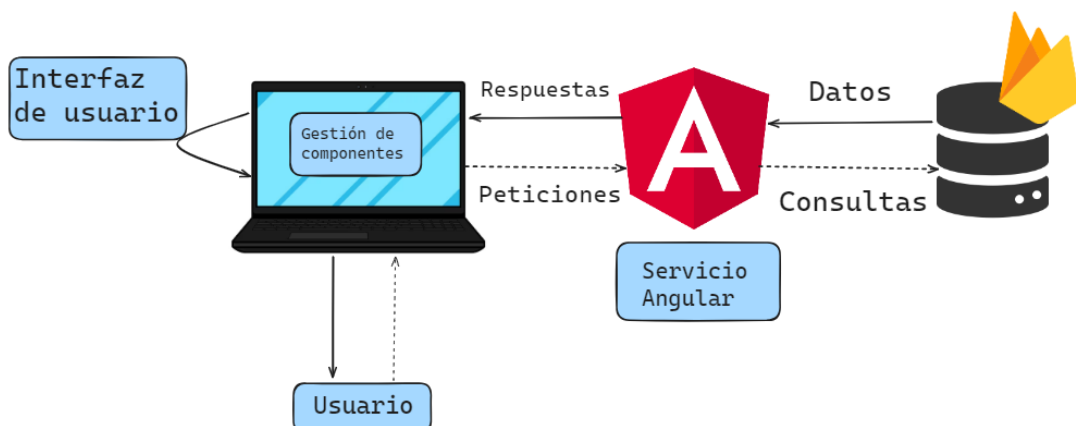


Figura 10 - Esquema de Gestión de Datos

Para llevar a cabo un desarrollo que cumpla con lo establecido anteriormente, lo primero que debemos permitir, es el inicio y cierre de sesión por parte de cualquiera de los usuarios. Es a partir del inicio de sesión que comienza la petición de datos a *Firestore*, que estará controlada a través de los servicios de *Angular*.

Una vez iniciada la sesión, cada uno accederá al perfil correspondiente según su rol, distinguiéndose hasta cinco roles: 'Alumno', 'Padre', 'Administrador', 'Profesor',



‘Jefatura’ y ‘Conserje’. Para ello haremos uso de *Cloud Firestore*, descrito en el apartado anterior, que nos permitirá el tratamiento de datos a lo largo de todo el desarrollo.

La aplicación está definida de manera que, una vez iniciada la sesión, no se pueda acceder directamente a una ruta, si no se tiene el rol apropiado.

Una vez se haya accedido al menú principal, según el rol tendremos acceso a unas funcionalidades u otras. Sin embargo, existen varias funcionalidades comunes a todos los perfiles, las cuales son:

- **Chat de usuarios:** una vez se haya seleccionado el usuario con el que se quiere hablar o bien se haya introducido su nombre en el buscador, accederemos al chat con esa persona y se cargarán los mensajes anteriores si procede.
- **Visualización del perfil:** cada usuario tendrá acceso sus propios datos pudiendo notificar al administrador en caso de fallo, si fuera necesario.
- **Recordatorios:** todos los perfiles tienen un apartado de recordatorio en el que pueden añadir notas con información a la que solamente ellos tienen acceso.
- **Tablón de anuncios:** en este caso tenemos un sistema de notificaciones compartidos, los únicos perfiles con permisos para generar una notificación son administradores, jefes de estudio y profesores, aunque estos últimos solo las pueden definir por curso. Sin embargo, todos los perfiles tienen acceso al tablón de anuncios, una vez dentro, verán los anuncios dependiendo de si están dentro de la lista de destinatarios o no.

Por otro lado, cada perfil específicamente tendrá acceso funcionalidades determinadas:

- **Administradores:**
  - Tienen acceso a los formularios para dar de alta a empleados, según el rol que vayan a tener en el centro docente, y a alumnos, pudiendo asignarles directamente un curso y un grupo.
  - Tienen acceso a un buscador para casos en los que necesiten acceder a los datos de alta de algún usuario.

- Son los encargados de configurar y actualizar los horarios de los cursos.
- **Alumnos:**
  - Tienen acceso a sus boletines de notas.
  - Tienen acceso al horario semanal según el curso y el grupo, que puede ser actualizado en cualquier momento.
  - Tienen acceso a sus incidencias del curso, donde podrán ver la lista de faltas que han recibido, la asignatura y el motivo, sin embargo, no las podrán justificar.
- **Consejería:**
  - Tienen acceso al horario de todos los cursos, pero no tienen permisos para editarlos.
- **Jefatura:**
  - Tienen permisos para añadir anuncios al tablón y filtrar a los destinatarios según el rol.
  - Son los encargados de editar los boletines de notas de los alumnos.
- **Padres:**
  - Tienen acceso al perfil de su hijo y, en caso de tener más de un hijo en el centro educativo, podrá acceder a ambos perfiles desde la misma cuenta, simplemente seleccionando el nombre del hijo al que quiere acceder en cualquier momento.
  - Tienen acceso a los boletines de notas de sus hijos.
  - Tienen acceso a las incidencias del curso de sus hijos y cuentan con permisos para justificarlas.
  - Tienen acceso al horario semanal.

- **Profesores:**
  - Tienen acceso a la lista de cursos que tienen asignados. Dentro de cada curso aparecerá una lista con los alumnos del curso y un acceso directo para poner una incidencia, para acceder al chat con el alumno y para acceder al chat con los padres del alumno. Dentro de cada curso, podrán añadir un anuncio al tablón, donde los destinatarios serán todos los alumnos de ese curso.
  - Tienen acceso al horario de todos los cursos y grupos.
  - Tienen acceso al tablón de anuncios.

Una vez detalladas las funcionalidades que comprende este proyecto, pasaremos a detallar las mismas de manera esquematizada por medio de un diagrama de flujo.

La Figura 11, muestra el flujo de inicio y cierre de sesión, por un lado, las funcionalidades comunes a todos los perfiles, descritas al principio de este apartado y aquellas funcionalidades que, aun no comportándose igual para todos los perfiles, dependen únicamente de los permisos que tenga el usuario.

Además, en el acceso al menú principal en función del rol, se pretende que, todos los usuarios después de iniciar la sesión sean redirigidos al menú principal, sin embargo, el contenido de ese menú va a depender de las funcionalidades que deba tener cada perfil.

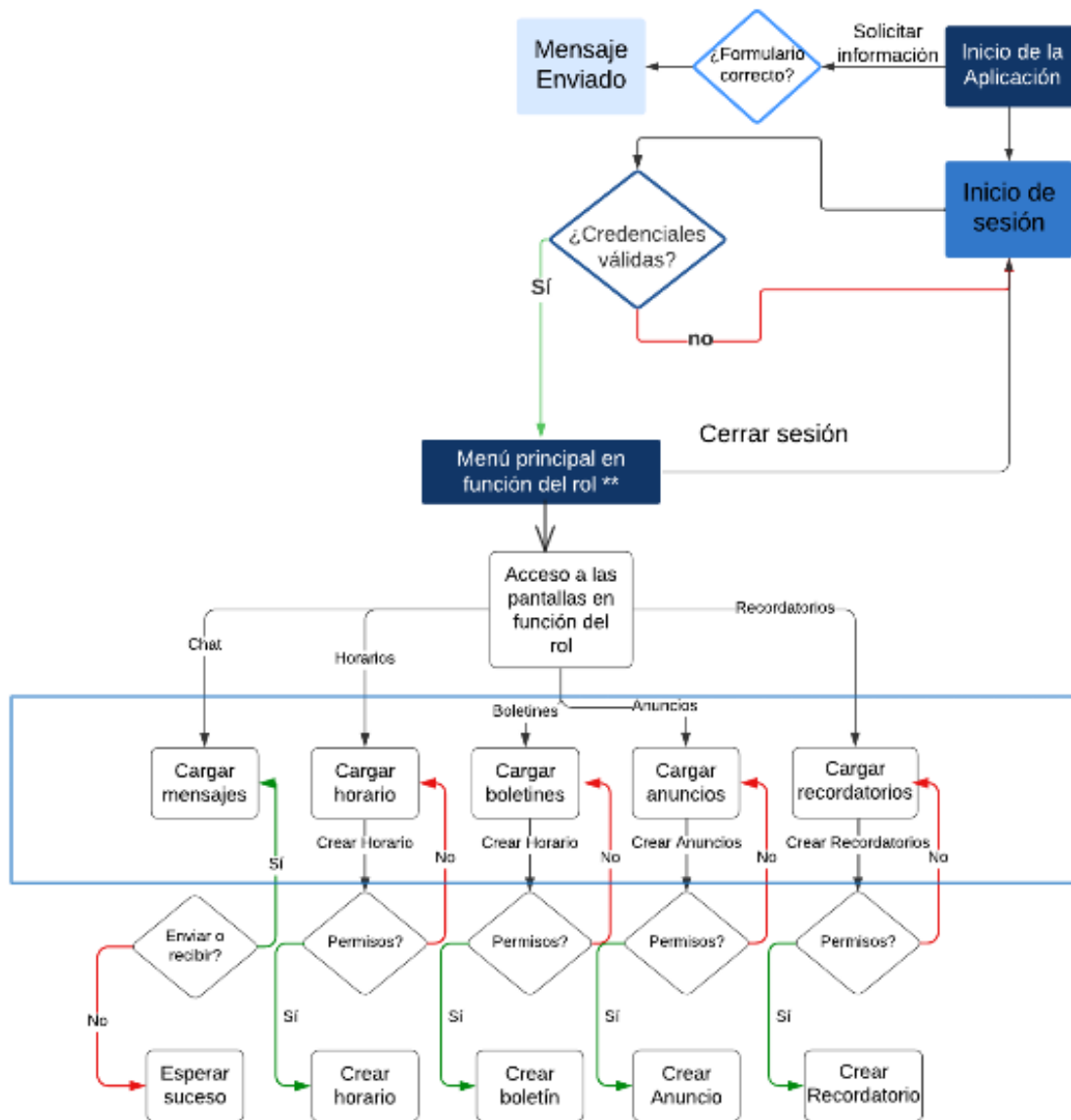


Figura 11 - Diagrama de funcionalidades comunes

A continuación, se mostrarán el diagrama de aquellas funciones que no son comunes a todos los perfiles de usuario. Cabe destacar que en la Figura 13, a pesar de que se muestran tres pantallas diferentes, todas parten del bloque “Acceso a las pantallas en función del rol” descrito en la Figura 12.

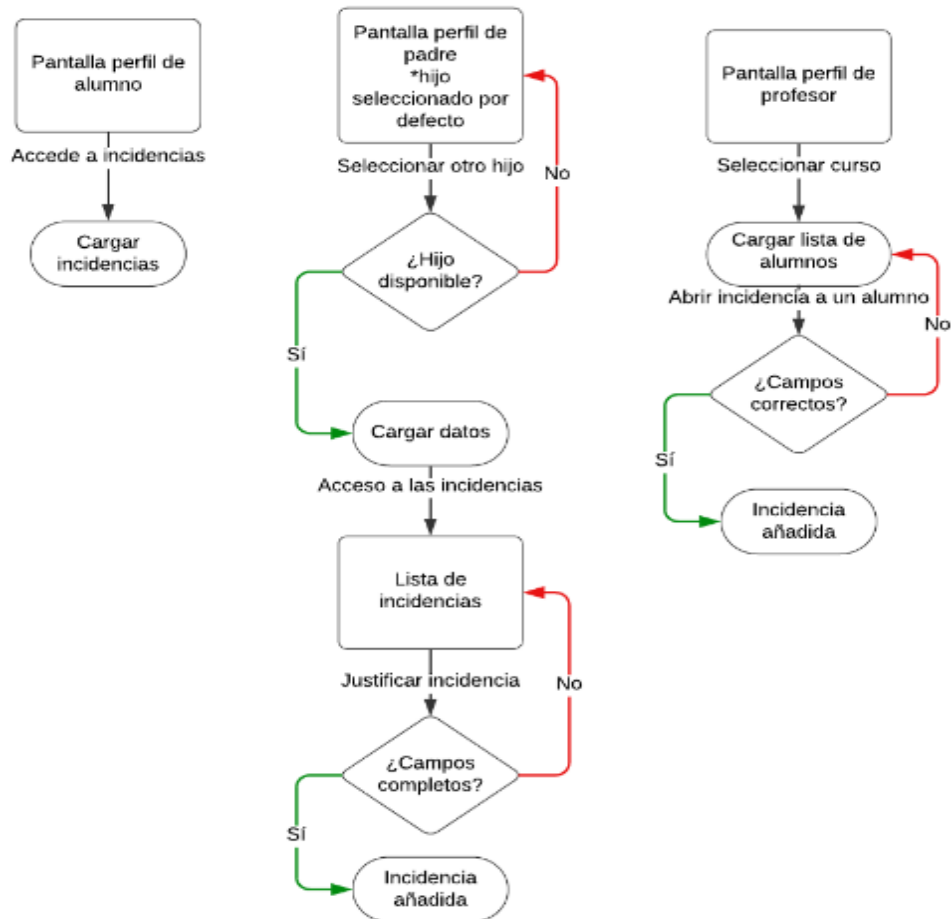


Figura 12 - Diagrama de funcionalidades específicas

En resumen, el objetivo de este apartado es ofrecer una versión esclarecida de cuál será el comportamiento de los diferentes elementos interactivos con los que pretende contar el desarrollo web.

## Capítulo 4. Diseño

DURANTE ESTE CAPÍTULO SE MOSTRARÁ EL DISEÑO REALIZADO INICIALMENTE CON LA INTENCIÓN DE TENER UNA VISIÓN GLOBAL DE LO QUE SE IBA A IMPLEMENTAR, TANTO CON RELACIÓN A COMO SE TENÍA QUE DEFINIR LA ESTRUCTURA DE DATOS EN LA BASE DE DATOS COMO EL ASPECTO PROPIO QUE DEBERÍA TENER LA INTERFAZ DE USUARIO PROPIA DE LA APLICACIÓN.

## 4.1 Contexto del diseño

Anteriormente, se ha analizado el comportamiento que queremos que tenga la aplicación, a través de los distintos elementos e interacciones. Para llevar a cabo esa comunicación entre componentes de manera que den lugar a funcionalidades es imprescindible definir la lógica interna, a través de la cual se llevará a cabo la implementación.

A continuación, se mostrará el procedimiento seguido para fijar cada uno de estos componentes esenciales en la aplicación web.

## 4.2 Diseño de la base datos

*Firestore* ha sido la herramienta elegida para el almacenamiento de datos de la aplicación, más concretamente, *Cloud Firestore*, que proporciona un almacenamiento de grandes cantidades de datos en la nube, y su actualización en tiempo real.[14][15]

Teniendo en cuenta que esta plataforma está destinada al tratamiento de datos en centros educativos, es de esperar que se vayan a manejar grandes cantidades de datos. En ese sentido, la opción de almacenamiento seleccionada es realmente útil al mismo tiempo que manejable y escalable. Además, cuenta con una interfaz de usuario por medio de su consola, realmente sencilla de utilizar e intuitiva para el cliente.

Cabe destacar que, al tratarse de una base de datos No Relacional, no es obligatorio mantener los mismos campos para todos los documentos de una misma colección. Esto aporta una gran comodidad en el manejo de datos y un desarrollo ágil y adaptable.[16]

Para ser capaces de definir la estructura de datos, se debe empezar a partir el análisis que previamente se ha hecho acerca del comportamiento. Es de utilidad ayudarse también, del diagrama de flujo representado en el apartado anterior, y

entonces, partiendo de cada bloque del diagrama, determinar qué datos necesitaremos para llevarlo a cabo.

En la Figura 13, podemos ver la estructura de tablas definiendo la información y clasificación que en principio vamos a necesitar en nuestra aplicación.

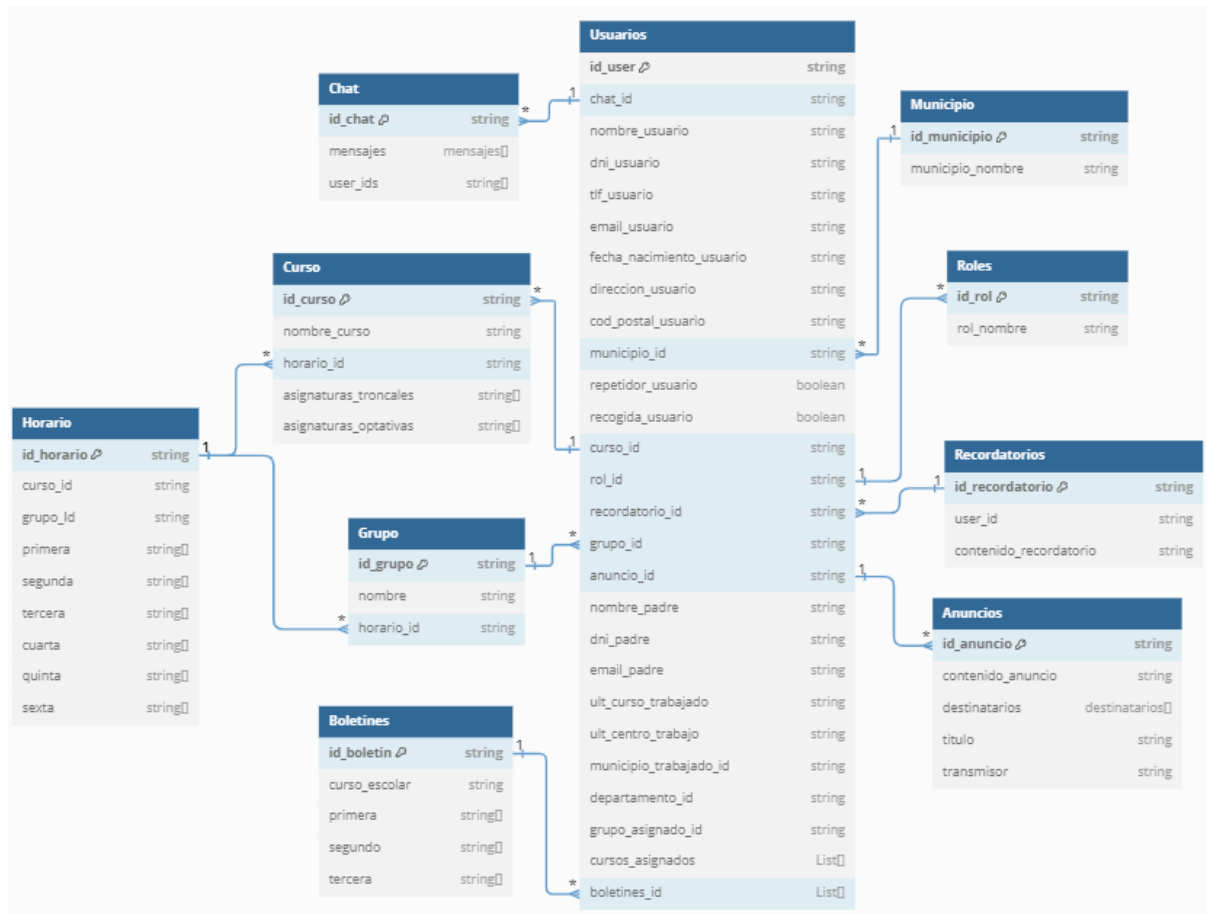


Figura 13 - Esquema de base de datos

En este esquema se muestra de manera completa la estructura que vamos a emplear para el almacenamiento de datos. A continuación, analizaremos cada tabla con más detalle, haciendo hincapié en las diferencias que podrá tener una misma tabla para diferentes perfiles de usuario.

Es importante tener en cuenta que, a pesar de que por comodidad y por motivos de presentación, el esquema se ha realizado por medio de tablas, cuando nos referimos



a bases de datos no relacionales, en este caso concreto que se utiliza *Firestore*, no hablamos de tablas.

En lugar de tablas, la información se almacenará en colecciones, podemos tener tantas colecciones como se consideren oportunas. Cada colección estará compuesta por documentos, que conformarían los registros de una tabla convencional. Dentro de cada documento se definirán los campos pertinentes, teniendo en cuenta que no todos los documentos requieren los mismos campos.

#### 4.2.1 Colección Usuarios

Obviamente, se trata de la colección más importante del proyecto. Cada documento de esta colección contará con una amplia variedad de datos, ya que servirán para dar de alta a los usuarios. Estos datos deben ser tratados con responsabilidad por los motivos mencionados en apartados anteriores.

Usuarios	
id_user	string
chat_id	string
nombre_usuario	string
dni_usuario	string
tif_usuario	string
email_usuario	string
fecha_nacimiento_usuario	string
direccion_usuario	string
cod_postal_usuario	string
municipio_id	string
repetidor_usuario	boolean
recogida_usuario	boolean
curso_id	string
rol_id	string
recordatorio_id	string
grupo_id	string
anuncio_id	string
nombre_padre	string
dni_padre	string
email_padre	string
ult_curso_trabajado	string
ult_centro_trabajo	string
municipio_trabajado_id	string
departamento_id	string
grupo_asignado_id	string
cursos_asignados	List[]
boletines_id	List[]

Figura 14 - Colección de Usuarios

En la Figura 14, podemos ver los campos que comprenderían principalmente los datos de matrícula para el registro de altas ya sean de alumnos o del resto del personal del centro. Algunos de estos campos serán necesarios en la comunicación con otras colecciones y otros pues simplemente están para abordar la matriculación y no tendrán mayor repercusión en el desarrollo.

En la siguiente tabla, mostramos una descripción de aquellos campos que resulten más relevantes para el desarrollo:

Campo	Descripción
<b>Id_user</b>	Identificador único generado por Firestore
<b>Nombre_usuario</b>	Nombre completo del usuario registrado
<b>Email_usuario</b>	Será esencial en el inicio de sesión

<b>Curso_id</b>	Apuntará al identificador del curso al que pertenece (solo alumnos)
<b>Rol_id</b>	Refiere el rol que va a tener el usuario
<b>Dni_padre</b>	Será importante a la hora de determinar si ya el padre del alumno está registrado. (solo alumnos)
<b>Ult_curso_trabajado</b>	Datos exclusivos de un perfil que no sea ni alumno ni padre
<b>Cursos_asignados</b>	Lista de objetos formados por identificadores de cursos y de grupos exclusivo del rol de profesor

Tabla 2 - Descripción de los campos de la colección Usuarios

#### 4.2.2 Colección Chat

Esta colección se encarga de almacenar los datos y el historial del chat entre usuarios. Habrá un documento por cada chat, dentro de ese documento se encontrarán los datos para identificar a los usuarios implicados y sus mensajes.


Chat	
id_chat 	string
mensajes	mensajes[]
user_ids	string[]

Figura 15 - Colección Chat

Esta colección se relacionará con la colección de usuarios por medio del identificador único que *Firestore* asignará a cada documento en el caso en el que dos usuarios nunca hayan tenido una conversación. Es decir, si no existe ningún documento que incluya el identificador de los dos usuarios, pues se creará un documento nuevo, de lo contrario se cargarán los datos disponibles.

En la tabla 3, se detallan los datos de los campos más destacados que almacena la colección:

Campo	Descripción
<b>Id_chat</b>	Identificador único del chat entre usuarios
<b>Mensajes</b>	Será una lista formada por objetos con los campos, contenido (almacena el contenido del mensaje), userId (identifica qué usuario mandó el mensaje), timestamp (guarda la fecha de envío del mensaje)
<b>userIds</b>	Será una lista formada por un objeto que incluya el identificador de los dos usuarios

Tabla 3 - Descripción de los campos de la colección Chat

### 4.2.3 Colección Municipios

Esta colección almacenará los municipios, en este caso únicamente los de la isla de Gran Canaria. Se ha decidido manejar estos datos como una colección debido a que si mañana se utiliza la aplicación desde otra región del país, únicamente habría que añadir los municipios de la zona concreta.


Municipio	
id_municipio 	string
municipio_nombre	string

Figura 16 - Colección de Municipios

### 4.2.4 Colección Roles

Esta colección almacenará todas las opciones con las que se puede crear un perfil de usuario que son: 'Alumno', 'Padre', 'Profesor', 'Administrador', 'Jefatura' y 'Conserje'.

Roles	
id_rol	string
rol_nombre	string

Figura 17 - Colección de Roles

#### 4.2.5 Colección Anuncios

Esta colección almacenará los datos pertinentes para la creación y lectura de los anuncios que se añadan al tablón en función de sus destinatarios.

Anuncios	
id_anuncio	string
contenido_anuncio	string
destinatarios	destinatarios[]
titulo	string
transmisor	string

Figura 18 - Colección de Anuncios

El campo destinatario será una lista que podrá estar formada ya sea por los roles o bien por objetos que contengan el identificador de curso y grupo específicamente. De esta manera, la estructura de datos quedará preparada para que tanto profesores como roles de jefatura de estudios puedan hacer uso de la funcionalidad de añadir anuncios.

Campo	Descripción
<b>Id_ anuncio</b>	Identificador único de cada anuncio asignado por Firestore
<b>Contenido_anuncio</b>	Es el cuerpo del anuncio, donde va el mensaje
<b>Destinatarios</b>	Es una lista que podrá estar formada o bien por un rol o bien por un objeto

	formado por el identificador de curso y de grupo
<b>Título</b>	Descripción del anuncio
<b>Transmisor</b>	Es la persona que ha añadido el anuncio

Tabla 4 - Descripción de los campos de la colección Anuncios

## 4.2.6 Colección Recordatorios

Esta colección permite a cada uno de los usuarios añadir notas en su perfil que no serán vistas por ningún otro usuario, ni siquiera los padres tendrán acceso a los recordatorios de sus hijos. Almacena tanto el contenido del recordatorio como el usuario al que pertenece.

Recordatorios	
<b>id_recordatorio</b> 	string
user_id	string
contenido_recordatorio	string

Figura 19 - Colección de Recordatorios

## 4.2.7 Colección Grupos

Esta colección almacena las posibilidades que hay para elegir grupo, que por defecto se prestarán cuatro: 'A', 'B', 'C' y 'D'.

Grupo	
<b>id_grupo</b> 	string
nombre	string
horario_id	string

Figura 20 - Colección de Grupos

## 4.2.8 Colección Curso

Esta colección ofrece las cuatro opciones de cursos según la educación secundaria obligatoria, así como las asignaturas que incluye. Además, esta colección se relaciona con la colección 'Horarios' como veremos en el siguiente apartado.

Curso	
id_curso 	string
nombre_curso	string
horario_id	string
asignaturas_troncales	string[]
asignaturas_optativas	string[]

Figura 21 - Colección de Curso

## 4.2.9 Colección Horarios

Esta colección permitirá mostrar cada una de las asignaturas por franja horaria y día de un alumno. Para ello se ha compuesto el identificador del horario como la unión del identificador del curso y el identificador del grupo.


Horario	
id_horario 	string
curso_id	string
grupo_id	string
primera	string[]
segunda	string[]
tercera	string[]
cuarta	string[]
quinta	string[]
sexta	string[]

Figura 22 - Colección de Horarios

## 4.2.10 Colección Boletines

Esta colección tendrá almacenada la información de la evaluación del alumnado. Habrá un boletín por curso escolar y además estará más o menos completado en función del trimestre en el que se encuentre el alumno.

Boletines	
id_boletin 	string
curso_escolar	string
primera	string[]
segundo	string[]
tercera	string[]

Figura 23 - Colección de Boletines

En este caso, las listas que se encuentran en la colección almacenan el valor de la nota de los usuarios en cada una de las asignaturas de los tres trimestres.

En resumen, el diseño de la base de datos se ha analizado minuciosamente para satisfacer las necesidades específicas de la aplicación web. A través de la creación de colecciones bien definidas se ha logrado una organización clara y eficiente de los datos.

Con este enfoque se pretende no solo garantizar la seguridad de los datos de los usuarios sino lograr una información consistente de fácil acceso y manejo por parte de los usuarios de la aplicación.

## 4.3 Mockups

Este desarrollo web está destinado a los centros educativos del país, por ello se busca una interfaz de usuario intuitiva, adaptada a su entorno para satisfacer las necesidades de todos los usuarios.



Para llevar a cabo este apartado, en primera instancia, se llevará a cabo un sencillo diseño de la aplicación por medio de la herramienta *TeleportHQ*, descrita en el apartado 3 de este documento. Una vez tengamos definido en componentes el diseño de la interfaz de usuario, podremos exportar el código adaptado perfectamente al *framework* con el que vamos a trabajar, en este caso *Angular*.

A lo largo de este capítulo como también del siguiente, veremos que el resultado de la implementación no coincide en su totalidad, aunque sí en su mayoría, con el diseño planteado desde *TeleportHQ*. Esto es debido a que a medida que avanzaba la implementación surgían ciertos inconvenientes y otras necesidades que debíamos cubrir en el momento y que pueden resultar en cambios. Aclarar que, independientemente de los cambios que se hayan realizado finalmente, estarán todos recogidos en el documento.

### 4.3.1 Pantalla Home

Es la página inicial de la aplicación ('Home') se muestra una descripción breve del producto y algunas de las características con las que contarán los centros que lo utilicen para poner en contexto a los posibles futuros clientes.

Desde esta página tendremos acceso, a un apartado de presentación, y al inicio de sesión. Además, cuenta con el componente de la cabecera, en este caso concreto, la cabecera para usuarios que no han iniciado sesión, el cual será utilizado en algunas otras pantallas.



Figura 24 - Diseño pantalla Home

### 4.3.2 Pantalla Presentación

Desde la página 'Home' descrita en el apartado anterior, se puede acceder a través del enlace de '¿Quiénes somos?', a una página en la que se describe más en profundidad las funcionalidades de la aplicación web.



Figura 25 - Diseño de pantalla Presentación

Si seguimos avanzando en esta página llegaremos a un formulario de contacto desde el que los clientes podrían solicitar un presupuesto o preguntar sus dudas acerca de la aplicación.

**¿Quieres saber más? Contáctanos**

Nombre

Email

Teléfono

Mensaje

**Enviar**

Figura 26 - Diseño de Formulario de Contacto

### 4.3.3 Pantalla Identificación

Esta es la página desde la que el usuario podrá introducir las credenciales proporcionadas por su administrador.

**SchoolApp** Inicio ¿Quiénes somos? Iniciar Sesión

**Servicio de identificación**

Introduce tu correo electrónico

Introduce tu contraseña

**Iniciar Sesión**

Figura 27 - Diseño de Pantalla de Identificación

### 4.3.4 Pantalla de Menú

Una vez se haya iniciado la sesión, todos los usuarios llegarán al menú principal. Sin embargo, como ya se ha comentado en apartados anteriores, el resultado dependerá del tipo de rol de usuario, debido a que según el rol se podrá acceder a unas

funcionalidades o a otras. En principio, aunque se mostrará el diseño de todas las funcionalidades, no se repetirán las figuras comunes a todos los perfiles, ni aquellas figuras en las que el cambio sea poco apreciable desde el punto de vista del diseño. Por tanto, la Figura 28, pertenece al menú principal de un perfil de administrador.



Figura 28 - Diseño de Menú principal de Administradores

En este caso, el componente de la cabecera es diferente, ya que pertenece al de los usuarios que están dentro de la sesión. Desde esta figura se puede observar el acceso a numerosas funcionalidades. Cabe destacar, que el diseño para el resto de los roles es igual, diferenciándose únicamente en los accesos a las posibles funcionalidades, que varían entre un perfil y otro.

#### 4.3.5 Pantalla de Recordatorios

Esta página pertenece al apartado de “Personal”, en este apartado se recogen los datos sensibles y la información privada de cada usuario, en ella podemos ver una lista que mostrará tantos componentes como recordatorios tenga el usuario almacenado. Desde el propio componente, se ofrece la opción de eliminar el recordatorio.

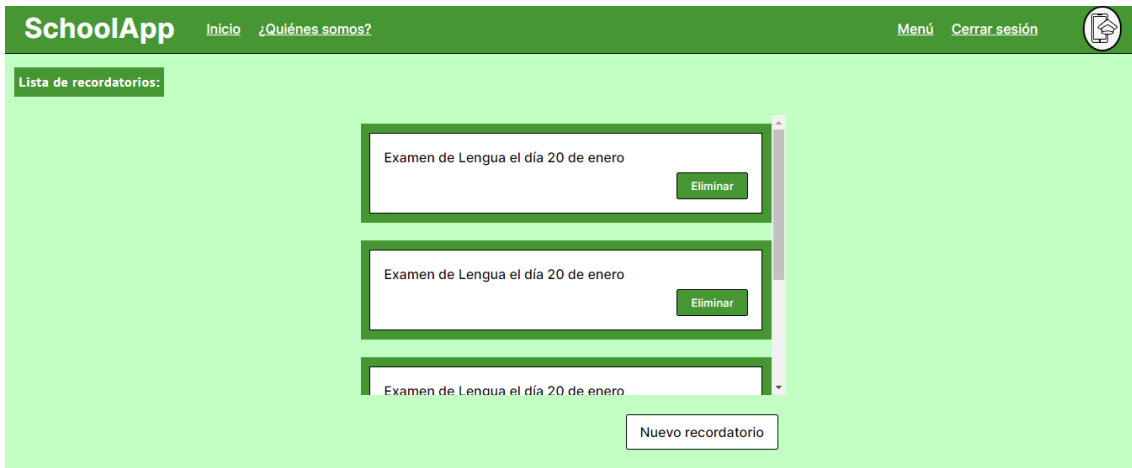


Figura 29 – Diseño de Pantalla de Lista de Recordatorios

Desde esta página si le damos a nuevo recordatorio seremos redirigidos a otra en la que, aportando los campos requeridos, seremos capaces de añadir un nuevo elemento a nuestra lista.

#### 4.3.6 Pantalla Alta Empleado

Desde esta página, los administradores serán los encargados de aportar los datos necesarios para dar de alta o de baja a un miembro en el centro y en la aplicación. Esta página será accesible también para que todos los empleados puedan ver su perfil, sin embargo, en este caso no tendrán permisos de edición.

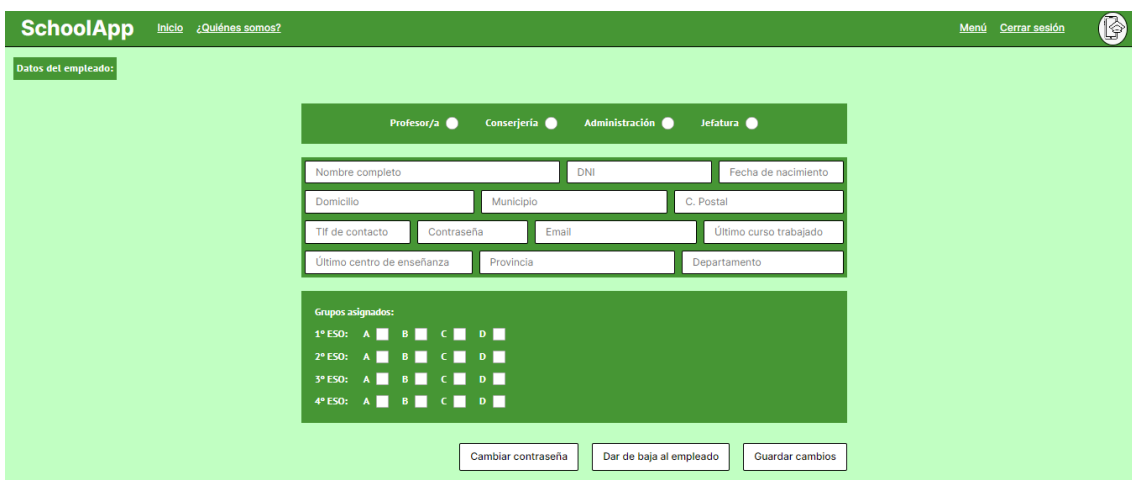


Figura 30 - Diseño de Pantalla de Formulario de Alta de Empleado

### 4.3.7 Pantalla Alta Alumno

De manera similar a la página anterior, los administradores podrán dar de alta o de baja en el sistema a los alumnos y consigo a sus padres, mientras que los alumnos y padres podrán acceder a esta página sin permisos de edición para revisar su perfil.

The screenshot shows the 'Dar de Alta a Alumno' form in the SchoolApp interface. The form is titled 'Datos del alumno/a:' and contains several input fields for personal and contact information. Below the fields are radio buttons for 'Repite curso' and 'Servicio de recogida'. At the bottom, there are four columns representing different grade levels (1º ESO to 4º ESO), each with a list of subjects and radio buttons for selection.

1º ESO	2º ESO	3º ESO	4º ESO
Atención Educativa	Atención Educativa	Atención Educativa	Atención Educativa
Religión	Religión	Religión	Religión
Ciencias de la naturaleza	Ciencias de la naturaleza	Biología y Geología	Biología y geología
Ciencias Sociales, geografía e historia	Ciencias Sociales, geografía e historia	Ciencias Sociales, geografía e historia	Francés
Historia de España	Historia de España	Historia de España	Historia de España
Educación física	Educación física	Educación física	Educación física
Educación para la ciudadanía	Educación para la ciudadanía	Física y Química	Física y Química
Educación plástica y visual	Educación plástica y visual	Educación plástica y visual	Educación plástica y visual
Lengua Castellana y Literatura	Lengua Castellana y Literatura	Lengua Castellana y Literatura	Lengua Castellana y Literatura
Lengua extranjera Inglés	Lengua extranjera Inglés	Lengua Castellana y Literatura	Lengua extranjera Inglés

Figura 31 - Diseño de Pantalla de Formulario de Dar de Alta a Alumno

### 4.3.8 Pantalla Buscador

Desde esta página tendremos acceso a todos los usuarios registrados en la aplicación. En ella veremos una lista que desplegará unos componentes a modo de tarjeta de usuario con el nombre y el rol, o grupo y curso en caso de los alumnos. En caso de seleccionar una de las tarjetas, o de elegir a un usuario introduciendo su nombre en el buscador, los administradores podrán acceder a su perfil.

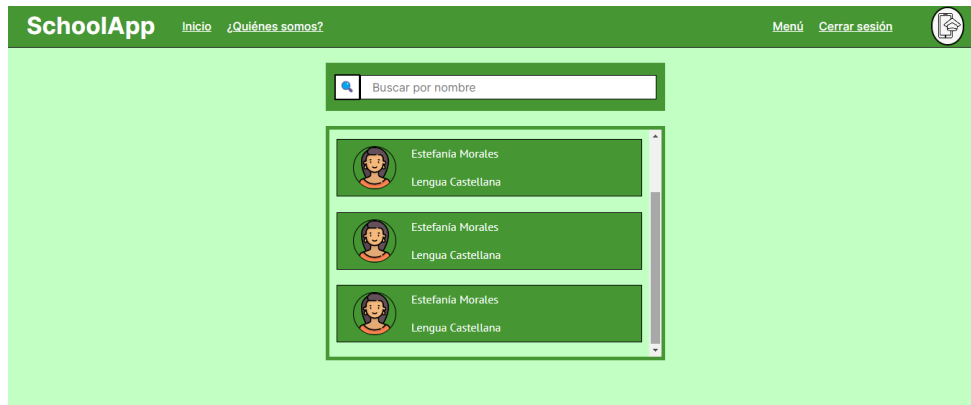


Figura 32 - Diseño de Pantalla de Buscador de Usuarios

### 4.3.9 Pantalla Configurar Horarios

Desde aquí, los administradores podrán seleccionar el curso del que desean editar o crear el horario y después lo podrán guardar. Esta pantalla será accesible además por profesores y conserjes sin contar con permisos de edición.

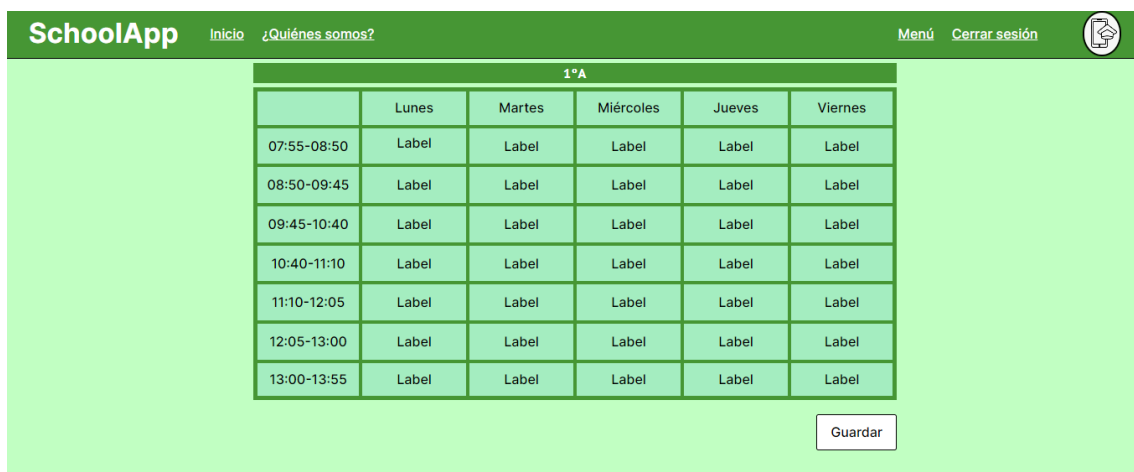


Figura 33 - Diseño de Pantalla de Configuración de Horarios

### 4.3.10 Pantalla Nuevo Chat

Desde esta página común a todos los perfiles, podremos acceder a un buscador que permitirá elegir al usuario con el que se quiere hablar. Una vez seleccionado nos iremos a la página del chat, detallada en el siguiente apartado.

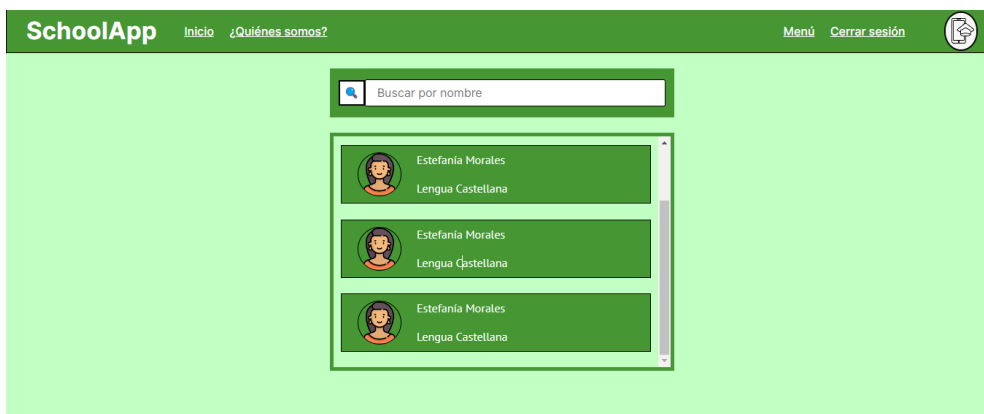


Figura 34 - Diseño de Pantalla de Buscador de Usuarios para Chat

### 4.3.11 Pantalla Chat

En esta página también accesible desde todos los perfiles, accederemos al chat con el usuario que se haya elegido previamente. Si ya hay mensajes en esa conversación se cargarán por orden y cada vez que el usuario escriba un mensaje bastará con pulsar la tecla *Enter* para enviarlo.

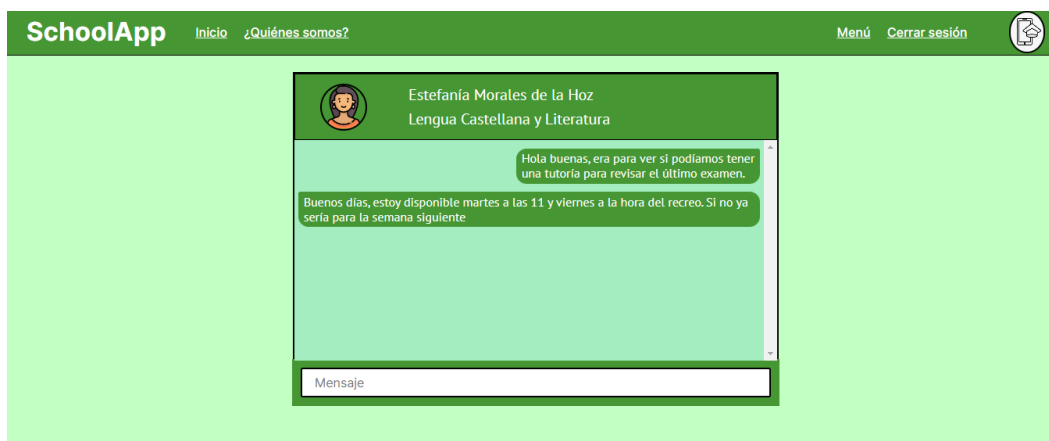


Figura 35 - Diseño de Pantalla de Chat

### 4.3.11 Pantalla Boletines

La siguiente página será accesible únicamente para aquellos usuarios que tengan un rol de 'Jefatura'. Desde aquí y habiendo seleccionado al usuario a través de



buscadores similares a los definidos en apartados anteriores se podrán añadir las notas de la evaluación a los alumnos.

The screenshot shows the 'SchoolApp' interface. At the top, there is a green header with the logo 'SchoolApp', navigation links 'Inicio' and '¿Quiénes somos?', and user options 'Menú' and 'Cerrar sesión'. Below the header is a form containing student information:

Alumno: Francisco López	Curso: 4º	Grupo: B	DNI: 11111111A	Código Postal: 33333
Dirección: Tafira, 23, Portal 2, 1º Izq	Email alumno: francisco.lopez@gmail.com			
Email padre/madre: madre.alumno@gmail.com	Teléfono: 666666666			
Año escolar: 2019/2020	Tutora: Francisco Gómez	Número de lista: 22		

Below the form is a table for recording grades:

	1º trimestre	2º trimestre	3º trimestre

Figura 36 - Diseño de Pantalla de Boletín de Notas

Desde una página similar, los usuarios con perfil de alumnos y de padres, podrán acceder a sus boletines correspondientes en función de los años que lleven cursados.

### 4.3.11 Pantalla Cursos

Desde esta página todos los usuarios con perfil de 'Jefatura' tendrán acceso al listado de alumnos que pertenecen a cada curso y grupo. Se trata una pantalla compartida entre 'Jefatura' y 'Profesorado', sin embargo, el profesorado no tiene la opción de seleccionar otro curso.

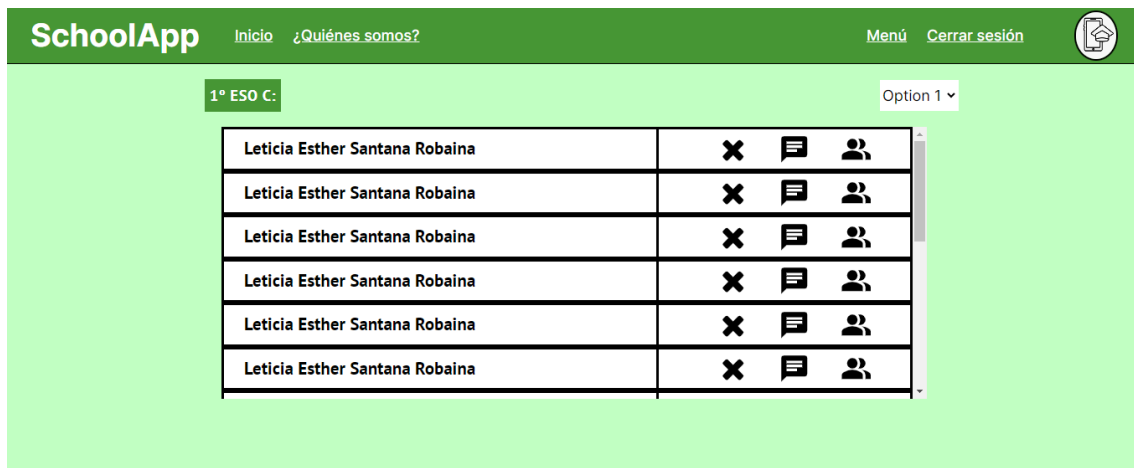


Figura 37 - Diseño de Pantalla de Lista de Alumnos Por Curso

Desde esta página, además, si se accede como profesor de un curso, se puede añadir un anuncio al tablón. Además, con cualquiera de los dos perfiles, se puede añadir una incidencia al alumno, ir al chat directo con él o ir al chat directo con sus padres.

#### 4.3.12 Pantalla Anuncios

Desde esta página veremos una lista de componentes para representar anuncios. Todos los perfiles tienen acceso al tablón de anuncios y verán aquellos que estén destinados a su rol o en el caso de alumnos, a su curso y grupo.

Además, los perfiles con permisos verán un botón para agregar un anuncio y en caso de pulsar aparecerá un formulario que deben completar para añadirlo.

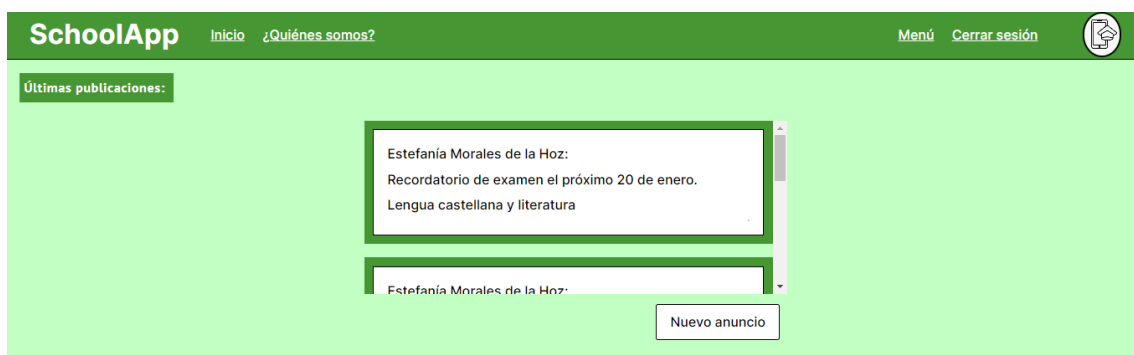


Figura 38 - Diseño de Pantalla de Anuncios

### 4.3.13 Pantalla Incidencias

Esta página pertenece a varios perfiles de usuario, pero sus funcionalidades se verán limitadas en función de los permisos. En este caso, los alumnos tendrán acceso al listado de incidencias que se les han puesto, mientras que los padres de estos alumnos tendrán la opción de justificarla desde un formulario. Una vez justificadas, aparecerán simplemente como leídas.

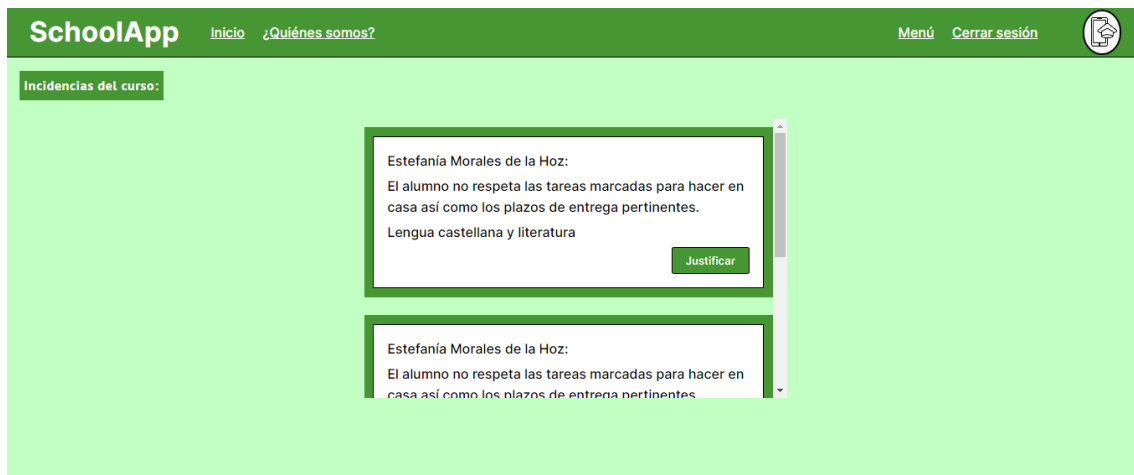


Figura 39 - Diseño de Pantalla de Incidencias del Alumno

### 4.4 Diseño del logo de la aplicación

Con motivo de decorar y aportar al diseño una marca de identidad se ha diseñado un logo para la aplicación que propone mostrar al usuario la combinación de tecnología y educación. A continuación, en la Figura 40, se muestra el logotipo junto con el nombre de la aplicación.



Figura 40 - Logo SchoolApp

## 4.5 Patrón de diseño

En este apartado, haremos referencia a la arquitectura software sobre la que ha planteado este desarrollo. La aplicación utiliza el patrón de diseño *Modelo-Vista-Modelo de Vista (MVVM)*, que propone una manera estructurada y eficiente de separar las responsabilidades en una aplicación, dividiéndola principalmente en tres componentes:

- **Modelo:** representa los datos y la lógica de la aplicación y no tiene conocimiento de la vista o del modelo de vista.
- **Vista:** representa la interfaz de usuario, es decir, se encarga de la presentación visual de los datos y la interacción con el usuario.
- **Modelo de Vista:** es el intermediario entre el Modelo y la Vista, que contiene la lógica de presentación y maneja la interacción de datos entre el Modelo y la Vista.

En resumen, este patrón facilita una separación clara de responsabilidades, mejorando la mantenibilidad y la escalabilidad del código. La elección de *Angular* como *framework* de desarrollo no es casual, ya que *Angular* está diseñado para trabajar de manera óptima con el patrón *MVVM*, proporcionando herramientas integradas para el *data binding* y la inyección de dependencias, lo cual simplifica el desarrollo de aplicaciones robustas y dinámicas. [17], [18]. A continuación, en la figura 41, se muestra un diagrama que refleja el flujo de datos en este patrón de diseño:

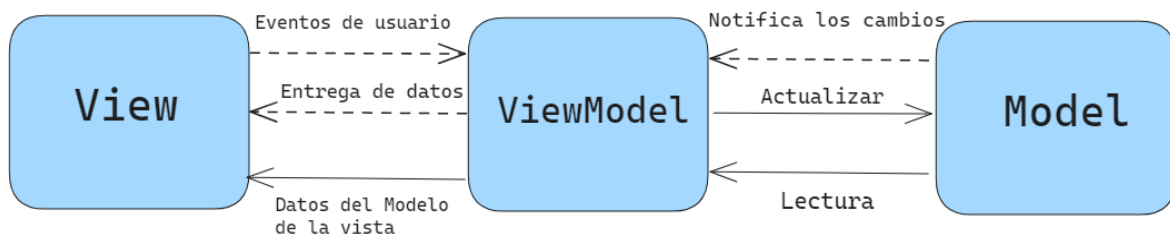


Figura 41 - Diagrama de Flujo de Datos en Patrón M-V-VM

## Capítulo 5. Implementación

A LO LARGO DE ESTE CAPÍTULO, SE EXPLICARÁ DETALLADAMENTE LAS FASES DE LA IMPLEMENTACIÓN DE LA APLICACIÓN A PARTIR DEL DISEÑO MOSTRADO EN EL CAPÍTULO ANTERIOR. SE PRESENTARÁ LA ESTRUCTURA DEL PROYECTO, DESTACANDO CÓMO SE HA ORGANIZADO EL CÓDIGO Y LOS COMPONENTES CLAVE DE LA APLICACIÓN. ESTE CAPÍTULO PROPORCIONARÁ UNA VISIÓN CLARA Y COMPRENSIVA DEL PROCESO DE IMPLEMENTACIÓN, SUBRAYANDO LAS DECISIONES TÉCNICAS Y METODOLÓGICAS QUE HAN GUIADO EL DESARROLLO DE LA PLATAFORMA.

## 5.1 Contexto de la implementación

Hasta llegar a este capítulo, hemos abordado el planteamiento de la aplicación, así como los objetivos que se pretenden lograr para hacer del desarrollo una aplicación funcional y la fase de diseño. Esta última nos ayudará a lograr los objetivos establecidos en apartados anteriores.

A continuación, nos adentramos en los aspectos técnicos de la implementación del desarrollo, mostrando la estructura y organización del código. Se detallará cómo se han llevado a cabo las diversas funcionalidades de la aplicación y se explicarán las decisiones tecnológicas adoptadas. Asimismo, se describirán los componentes principales del sistema, las interfaces de usuario, y cómo se ha asegurado la integración y el funcionamiento de todos los elementos.

Este capítulo será clave para entender cómo se ha materializado el proyecto desde su concepción hasta su estado funcional final.[19], [20]

## 5.2 Estructura del proyecto

En esta sección, se presenta la estructura del proyecto tal como se visualiza en el entorno de desarrollo en el que vamos a trabajar, *Visual Studio Code*. Este árbol de directorios muestra la organización de los distintos componentes de la aplicación, distribuidos en varias dentro del directorio principal.

A continuación, se detalla el propósito y contenido de cada una de estas carpetas para una mejor comprensión del diseño y desarrollo del proyecto.

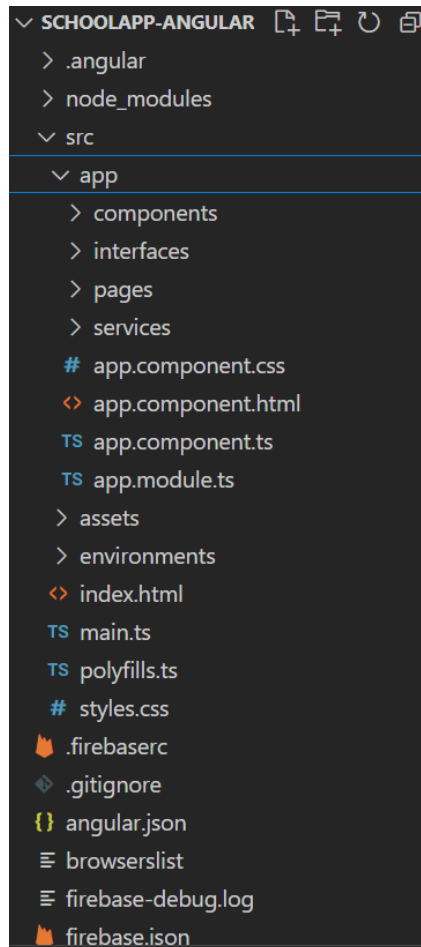


Figura 42 - Estructura de Carpetas del Proyecto

Según vemos en la Figura 42, el proyecto está contenido en el espacio de trabajo o *workspace SchoolApp-Angular*. Dentro, encontraremos las carpetas de *angular* y *node\_modules*, las cuales contienen en su mayoría archivos generados automáticamente o configuraciones específicas del proyecto *Angular* y las dependencias y módulos necesarios para ejecutar un proyecto en el caso de *Node.js*.

El siguiente directorio es el *src*, que es el encargado de almacenar el código fuente del proyecto, así como las configuraciones específicas que podamos necesitar. Una vez dentro, la primera carpeta que encontramos es la de aplicación, en la que la información se ha estructurado de la siguiente manera:

- **Components:** contiene los componentes específicos diseñados en *TeleportHQ*, que podrán ser accesibles desde varias páginas de la aplicación.
- **Interfaces:** en esta carpeta se almacenan las interfaces *TypeScript* que se han definido para estructurar los datos y definir la forma de los objetos que se utilizan en la aplicación.
- **Pages:** en este caso, también se encuentran los componentes de diseño que se definieron en *TeleportHQ*, sin embargo, en este caso son aquellos componentes que definen la vista de cada pantalla. Estas páginas, contienen tanto componentes específicos como componentes comunes desde la carpeta *components*.
- **Services:** desde aquí tendremos acceso a los archivos que comunican directamente con la base de datos, en este caso *Firestore*.

El resto de archivos de este directorio, incluyen un archivo *app.component*, con su *HTML* y su *CSS*, que en este caso, aunque no se han suprimido, no se hace uso de ellos ya que el diseño lo hemos importado directamente a través de la exportación del código adaptado el *framework* en *TeleportHQ*. Además, se incluye el archivo *app.modules*, en el que se encuentran todas las rutas principales de las páginas de la aplicación y se importa el *components.module*, para incluir también los componentes.

La siguiente carpeta es la de *assets*, en ella hemos almacenado el contenido multimedia que utilizaremos en algunos componentes de la aplicación. Mientras que la última carpeta de este apartado es la de *environments*, que contiene los archivos de configuración a través de los cuales vinculamos nuestro proyecto en *Firebase* con nuestro proyecto en local.

El resto de los archivos que se ven en la imagen son archivos generados automáticamente para la configuración del proyecto.



## 5.3 Interfaces

Como se mencionó en el apartado anterior, las interfaces son aquellos objetos en los que almacenaremos los datos que nos devuelva la base de datos después de una consulta, para después poder someterlos al tratamiento que requiera para enviarlos a la vista. Por todo ello, las interfaces tienen gran similitud con la estructura de la base de datos:

### 5.3.1 Interfaz de Usuario

La interfaz del 'Usuario' podría definirse como la principal de la aplicación. Cada una de las interfaces que hemos utilizado coinciden en su totalidad con las colecciones definidas anteriormente, con la particularidad de que en cada propiedad se especifica si el campo es obligatorio o no a la hora de crear un objeto. Por ejemplo, si el usuario que vamos a dar de alta es un alumno, no tiene por qué tener el campo hijos. La interfaz de usuario queda por tanto de la siguiente manera:

- **id**: identificador único del usuario, de tipo *string*.
- **nombre**: nombre y apellidos de cada usuario, de tipo *string*.
- **dni**: dni del usuario, de tipo *string*.
- **tlf**: teléfono para contactar con el usuario o con la familia de este en caso de un alumno, de tipo *string*.
- **email**: dirección de correo del usuario, de tipo *string*.
- **contrasena**: contraseña para iniciar sesión del usuario, de tipo *string*.
- **fecha\_nacimiento\_usuario**: de tipo *string*.
- **direccion**: dirección del domicilio, de tipo *string*.
- **cpostal**: código postal al que pertenece la dirección, de tipo *string*.
- **municipio\_id**: identificador del municipio al que pertenece el usuario, de tipo *string*.
- **repetidor\_usuario?**: campo obligatorio únicamente para perfiles de tipo alumno, de tipo *boolean*.

- **recogida\_usuario?**: campo obligatorio únicamente para perfiles de tipo Alumno, de tipo *boolean*.
- **curso?**: campo obligatorio únicamente para perfiles de tipo alumno, de tipo *string*.
- **grupo?**: campo obligatorio únicamente para perfiles de tipo alumno, de tipo *string*.
- **rol**: indica el perfil de usuario, de tipo *string*.
- **nombre\_padre**: campo obligatorio únicamente para perfiles de tipo alumno, de tipo *string*.
- **dni\_padre?**: campo obligatorio únicamente para perfiles de tipo alumno, de tipo *string*.
- **email\_padre?**: campo obligatorio únicamente para perfiles de tipo alumno, de tipo *string*.
- **contrasena\_padre?**: campo obligatorio únicamente para perfiles de tipo alumno, de tipo *string*.
- **ult\_curso\_trabajado?**: campo obligatorio únicamente para perfiles que no sean de tipo alumno o padre, de tipo *string*.
- **ult\_centro\_trabajado?**: campo obligatorio únicamente para perfiles que no sean de tipo alumno o padre, de tipo *string*.
- **departamento?**: campo obligatorio únicamente para perfiles de tipo profesor, de tipo *string*.
- **asignaturas?**: lista de tipo *string* que almacena las asignaturas del curso del alumno, exclusivo de los alumnos.
- **hijos?**: lista de *string*, que almacena los identificadores de los hijos que pueda tener un padre en centro, exclusivo de los padres.
- **boletines?**: lista de *string* que almacena las notas de los alumnos, exclusivo de los alumnos.
- **cursos\_asignados?**: lista de *string*, que almacena los cursos que tiene asignado un profesor, exclusivo de los profesores.
- **foto?**: campo que permite almacenar una foto del usuario, no es obligatorio.

### 5.3.2 Interfaz de Chatl

Esta interfaz tiene este nombre debido a que podría causar conflictos a la hora de importarla en el componente Chat.

Además, en este archivo se incluye la interfaz de mensaje para manejar de manera más cómoda los objetos que van a almacenarse en la colección de 'Chat'. La estructura de la interfaz 'Chatl' es la siguiente:

- **id:** es el identificador del chat entre dos usuarios, de tipo *string*.
- **userIds:** lista de *string*, que incluye de los identificadores de los usuarios que conforman el chat.
- **mensajes:** lista de objetos de tipo Mensaje.

Por otro lado, la estructura de la interfaz Mensaje es la siguiente:

- **userId:** es el identificador del usuario que envía el mensaje, de tipo *string*.
- **content:** es el cuerpo del mensaje, de tipo *string*.
- **timeStamp:** fecha en la que envía el mensaje para su posterior orden en el chat, de tipo *Date*.

### 5.3.3 Interfaz de Anuncio

La interfaz de 'Anuncio' permite manejar y almacenar de forma sencilla los objetos en la base de datos. La estructura es la siguiente:

- **id:** identificador único del anuncio, de tipo *string*.
- **nombre:** título del anuncio, de tipo *string*.
- **contenido:** cuerpo del anuncio, de tipo *string*.
- **transmisor:** nombre de la persona que envía el anuncio, de tipo *string*.
- **destinatarios:** lista de *string* en el que se guardarán los identificadores de los usuarios que deben recibir ese anuncio en función de su rol.

### 5.3.4 Interfaz de Curso

La interfaz de 'Curso' permite manejar los datos del curso de los usuarios. La estructura de la interfaz es la siguiente:

- **id:** identificador del curso, de tipo *string*.
- **nombre:** nombre que se le asigna el curso, de tipo *string*.
- **troncales:** lista de objetos de tipo *string* que hace referencia a las asignaturas troncales que tendrá el curso.
- **optativas:** lista de objetos de tipo *string* que hace referencia a las asignaturas optativas que hay disponibles en un curso.

### 5.3.5 Interfaz de Grupo

La interfaz de 'Grupo' se prepara para recibir desde *Firestore* una lista de las opciones de grupo que hay y presentarlas en un selector para su posterior elección. La estructura que presenta es la siguiente:

- **id:** identificador del grupo, de tipo *string*.
- **nombre:** nombre que se le asigna al grupo, de tipo *string*.

### 5.3.6 Interfaz de Municipio

De manera análoga, se encuentra la interfaz de 'Municipio', una entidad preparada para recibir los objetos procedentes de la colección de municipios en *Firestore* y presentarlos en un desplegable para seleccionar la opción apropiada. La estructura que presenta es la siguiente:

- **id:** identificador único del municipio, de tipo *string*.
- **nombre:** nombre del municipio, de tipo *string*.

### 5.3.7 Interfaz de HorarioI

La interfaz de 'HorarioI', se denota con la "I" al final para evitar conflictos con el componente 'Horario'. La estructura presenta las siguientes propiedades:

- **id**: identificador del horario, de tipo *string*,
- **curso\_id**: almacena el identificador del curso al que representa, de tipo *string*.
- **grupo\_id**: almacena el identificador del grupo al que representa, de tipo *string*.
- **primera**: lista de objetos de tipo *string* que guarda el conjunto de asignaturas que durante la semana tienen lugar a primera hora.
- **segunda**: lista de objetos de tipo *string* que guarda el conjunto de asignaturas que durante la semana tienen lugar a segunda hora.
- **tercera**: lista de objetos de tipo *string* que guarda el conjunto de asignaturas que durante la semana tienen lugar a tercera hora.
- **cuarta**: lista de objetos de tipo *string* que guarda el conjunto de asignaturas que durante la semana tienen lugar a cuarta hora.
- **quinta**: lista de objetos de tipo *string* que guarda el conjunto de asignaturas que durante la semana tienen lugar a quinta hora.
- **sexta**: lista de objetos de tipo *string* que guarda el conjunto de asignaturas que durante la semana tienen lugar a sexta hora.

### 5.3.8 Interfaz de Incidencia

Por su parte, la interfaz de 'Incidencia' permite manejar los objetos que almacenarán los datos de una incidencia para mostrarlos después al alumno y padre correspondiente. La estructura definida es la siguiente:

- **id**: identificador de la incidencia, de tipo *string*.
- **userId**: identificador del alumno al que se está apercibiendo, de tipo *string*.
- **profesorNombre**: nombre del profesor que está poniendo la incidencia, de tipo *string*.
- **asignatura**: nombre de la asignatura en la que el alumno recibió la amonestación, de tipo *string*.

- **justificación?**: este campo no es requerido inicialmente, debido a que es el padre del alumno el que debe aportar dicha justificación, de tipo *string*.
- **mostrarJustificacion?**: en este caso, este campo se ha añadido a la interfaz para servir de ayuda a la hora de determinar si la justificación se debe mostrar o no, pues, aunque un padre la haya justificado, no debe aparecer en el perfil del alumno, de tipo *boolean*.

### 5.3.9 Interfaz de Notas

La interfaz de ‘Nota’ será de gran utilidad a la hora de crear y presentar los boletines de los alumnos. La estructura es la siguiente.

- **id**: identificador único del boletín, de tipo *string*.
- **userId**: identificador del usuario al que pertenece el boletín, de tipo *string*.
- **cursoEscolar**: año escolar al que pertenece el boletín, de tipo *string*.
- **primera**: lista de objetos de tipo *string* que contienen las notas de la primera evaluación.
- **segunda**: lista de objetos de tipo *string* que contienen las notas de la segunda evaluación.
- **tercera**: lista de objetos de tipo *string* que contienen las notas de la tercera evaluación.

### 5.3.10 Interfaz de Recordatorio

El título de la interfaz, al igual que en el caso de la interfaz ‘Chat’, se debe a evitar conflictos con el componente ‘Recordatorio’ a la hora de realizar las diferentes importaciones. La estructura de un recordatorio tendrá los siguientes campos:

- **id**: identificador único del recordatorio, de tipo *string*.
- **userId**: identificador del usuario al que pertenece el recordatorio, de tipo *string*.
- **contenido**: cuerpo de la nota que el usuario ha añadido al recordatorio, de tipo *string*.

### 5.3.11 Interfaz de Rol

En esta última interfaz, se describirá la estructura que permitirá almacenar los tipos de roles en su colección. La estructura es la siguiente:

- **id**: identificador único del rol, de tipo *string*.
- **nombre**: es el nombre del rol, de tipo *string*.

## 5.4 Servicios

Una vez hemos definido las interfaces de las entidades que nos ayudarán a estructurar la información intercambiada entre la base de datos y la interfaz de usuario, procederemos a identificar los servicios necesarios para esta comunicación.

Los servicios se utilizan para implementar toda la lógica que comunica la interfaz de usuario con la base de datos y manejan toda la información que tiene lugar en esa comunicación. Si bien todas las funciones podrían agruparse en un solo servicio es altamente recomendable mantener unas buenas prácticas y definir un servicio para cada conjunto de responsabilidades específicas, de manera que al final se obtenga un código más claro y modular.

### 5.4.1 Servicio de Usuario

En el servicio de 'Usuario' encontraremos todas aquellas funciones que permiten la interacción de los datos de usuario desde la base de datos. Es el servicio que más funcionalidades contiene y a continuación analizaremos cada una de ellas:

<b>Método</b>	<b>Parámetros</b>	<b>Descripción</b>
<b>agregarAlumno</b>	usuario: Usuario	Este método añade un usuario de tipo alumno a la colección Usuarios
<b>agregarEmpleado</b>	usuario: Usuario	Similar al anterior para el resto de los roles
<b>actualizarUsuario</b>	usuario: Usuario, id: string	Actualiza los datos de un usuario existente
<b>crearPerfilPadre</b>	usuario: Usuario, alumnoid: string	Método llamado desde agregar alumno para determinar si un perfil de padre existe o no y crearlo si procede
<b>obtenerAlumnosOrdenados</b>	-	Devuelve por orden alfabético a todos los usuarios de rol alumno
<b>buscarAlumnosPorNombre</b>	nombre: string	Devuelve los resultados que coincidan con la búsqueda del usuario a medida que introduce un nombre en el buscador
<b>obtenerUsuarioPorId</b>	id: string	Devuelve el objeto usuario que coincida con el identificador pasado por parámetro
<b>eliminarUsuario</b>	id: string	Elimina de la colección usuarios al usuario con el id pasado por parámetro
<b>obtenerAlumnosPorCursoY Grupo</b>	cursoId: string, grupoId: string	Devuelve una lista con todos los usuarios cuyos identificadores de curso y grupo coincidan con los pasados por parámetro
<b>obtenerIdPorDniPadre</b>	dniPadre: string	Devuelve el id del usuario de tipo padre si existe previamente en la colección usuarios

Tabla 5 - Métodos del Servicio de Usuarios



## 5.4.2 Servicio de Chat

A continuación, se detallarán las funciones implementadas para lograr el intercambio de datos con la colección de 'Chat'.

Método	Parámetros	Descripción
<b>getChat</b>	userIds: string[]	Devuelve, si existe, el chat, con sus datos, entre los usuarios pasados por parámetros
<b>createChat</b>	userIds: string[]	Crea un objeto de tipo Chat entre los usuarios pasados por parámetro
<b>addMessage</b>	chatId: string, message: Mensaje	Añade mensajes al chat entre dos usuarios

Tabla 6 - Métodos del Servicio de Chat

## 5.4.3 Servicio de Auth

Este servicio contiene los métodos para manejar el inicio de sesión. A pesar de que sus funciones interactúan directamente con la colección de usuarios, se ha optado por hacer un servicio diferenciado del de Usuarios para mayor limpieza en el código.

Método	Parámetros	Descripción
<b>login</b>	email: string, password: string	Comprueba las credenciales y redirige al usuario en función del rol
<b>logout</b>	-	Cierra la sesión impidiendo retroceder y redirige a la pantalla Login
<b>setUserId</b>	userId: string	Guarda el identificador del usuario al iniciar sesión

Tabla 7 - Métodos del Servicio de Auth

## 5.4.4 Servicio de Cursos

El servicio de 'Cursos' gestiona las funciones que comunican directamente con la colección de 'Cursos'. Presenta, los siguientes métodos:

Métodos	Parámetros	Descripción
<b>obtenerCursoPorNombre</b>	nombreCurso: string	Devuelve el objeto curso que coincida con el nombre pasado
<b>obtenerCombinaciones CursoGrupo</b>	-	Devuelve todas las combinaciones entre cursos y grupos que resultan en todos los cursos y grupos del centro

Tabla 8 - Métodos del Servicio de Cursos

## 5.4.5 Servicio de Grupos

El servicio de 'Grupos' gestiona la obtención de la disponibilidad de los grupos desde la colección pertinente en *Firestore*.

Métodos	Parámetros	Descripción
<b>obtenerGrupos</b>	-	Devuelve una lista con los nombres de los grupos disponibles

Tabla 9 - Métodos del Servicio de Grupos

## 5.4.6 Servicio de Municipios

El servicio de 'Municipios' funciona de manera similar al de 'Grupos' gestiona la obtención los municipios desde la colección pertinente en *Firestore*.

Métodos	Parámetros	Descripción
<b>obtenerMunicipios</b>	-	Devuelve una lista con los nombres de los municipios disponibles

Tabla 10 - Métodos del Servicio de Municipios

### 5.4.7 Servicio de Anuncio

El servicio de 'Anuncio', llamado 'Tablonanuncios' se prepara para gestionar todas las peticiones que permitan la interacción con la colección 'Anuncios', para que desde la aplicación se puedan consultar y crear lo anuncios necesarios.

Métodos	Parámetros	Descripción
<b>obtenerAnunciosPorUsuarioRol</b>	userRol: string	Devuelve una lista con los anuncios que pertenezcan al rol pasado por parámetro
<b>obtenerAnunciosPorUsuario CursoYGrupo</b>	cursoId: string, grupoId: string	Devuelve una lista de anuncios que incluyan en su destinatario el curso y el grupo pasados por parámetros

Tabla 11 - Métodos del Servicio de Anuncio

### 5.4.8 Servicio de Horario

El servicio de 'Horario' es el encargado de permitir tanto la creación y lectura de los horarios, está compuesto por los siguientes métodos:

Métodos	Parámetros	Descripción
<b>obtenerHorarioPorCurso YGrupo</b>	cursoId: string, grupoId: string	Devuelve las asignaturas y horas del curso y grupo elegido

<b>crearHorario</b>	docId: string, horarioData:HorarioI	Crea el horario según el id, que será la combinación del curso y grupo y añade los datos que haya elegido el usuario
---------------------	--	--

Tabla 12 - Métodos del Servicio de Horario

### 5.4.9 Servicio de Incidencia

En este servicio se implementan los métodos que comunican con la colección de 'Incidencia' y gestionan el intercambio de datos al crear o recibir un listado de incidencias. Los métodos que incluye son los siguientes:

Método	Parámetros	Descripción
<b>agregarIncidencia</b>	Incidencia: Incidencia	Añade un documento a la colección con los datos pasados en el objeto
<b>obtenerIncidenciasPorUsuario</b>	userId: string	Devuelve la lista de incidencias que tiene cada usuario, en este caso de tipo alumno
<b>actualizarIncidencia</b>	id: string, incidencia: Incidencia	Permite añadir la justificación de la incidencia

Tabla 13 - Métodos del Servicio de Incidencias

### 5.4.10 Servicio de Recordatorios

Este es el servicio encargado del intercambio de datos entre la aplicación y la colección de Recordatorios en *Firestore*, permitiendo así que los usuarios tengan acceso a sus notas personales.

Métodos	Parámetros	Descripción
<b>agregarRecordatorio</b>	Recordatorio: Recordatorio	Crea un nuevo recordatorio con los datos prestados por el usuario y los guarda en la colección
<b>getRecordatoriosByUserId</b>	userId: string	Devuelve la lista de recordatorios que tiene un usuario
<b>eliminarRecordatorio</b>	recordatorioid: string	Eliminar el recordatorio del usuario en función del identificador

Tabla 14 - Métodos del Servicio de Recordatorios

## 5.5 Componentes principales (páginas)

Una vez que hemos visto a través de qué objetos nos vamos a comunicar con la base de datos, y cómo los servicios nos proporcionan los métodos necesarios para que esa comunicación se lleve a cabo, es el momento de detallar cómo hacen las páginas para presentar esa información. Cabe recordar, que reciben el nombre de páginas porque alguna de ellas ya contiene componentes más específicos, pero realmente, por su composición y sus propiedades, de igual manera son componentes.

La estructura tanto de páginas como de componentes están compuestas por cuatro archivos:

- **Archivo Component Typescript (.component.ts):** en este archivo tiene lugar toda la lógica del componente y su comunicación con los servicios en interfaces mencionados anteriormente.
- **Archivo Module Typescript (.module.ts):** en este archivo se organizan y agrupan los componentes, servicios y otras directivas que puedan hacer falta en el componente principal.
- **Archivo HTML (.component.html):** se trata del archivo de la vista del componente, encargado de presentar la información que le pasa el componente.

- **Archivo CSS (.component.css):** es el encargado de dar formato y diseño a la vista, aplicando los estilos a cada componente de la vista.

A continuación, veremos qué componentes hacen uso de los servicios que acabamos de repasar.

### 5.5.1 Componente Login

En este componente, la vista le pasa al componente *login* las credenciales que el usuario introduce, comprueba si tienen el formato válido y lo pasan al servicio *Auth*. En caso de que la respuesta sea correcta, se inicia la sesión guardando el identificador del usuario y redirigiendo al usuario a una página concreta en función de su rol y en el caso contrario se alerta al usuario que sus credenciales son incorrectas. En la siguiente tabla podemos ver las funciones específicas del componente.

Método	Descripción
<b>onSubmit</b>	Se ejecuta cuando el usuario le da al botón de inicio de sesión
<b>rediretBasedOnRole</b>	Se llama desde el método anterior cuando las credenciales del usuario son correctas.

Tabla 15 - Métodos del Componente Login

### 5.5.2 Componente Menú

En este caso, se ha decidido hacer una pantalla *Main* por cada rol, de manera que no se explicarán cada uno de los menús porque el funcionamiento es el mismo.

Desde la vista se muestran las rutas de acceso a las funcionalidades que tiene cada perfil, para acceder a cada ruta se debe añadir el identificador de cada usuario en la ruta de navegación. Para ello, desde el componente cada vez que se ingrese en el menú principal, se cogerá el identificador del usuario que previamente el método del componente *Login* ha incluido en la ruta de navegación.

Como se ha mencionado anteriormente, los accesos a las funcionalidades que tenga cada menú dependen fundamentalmente del tipo de rol. En el caso de los padres, el menú principal incluye un selector desde el que seleccionar el perfil de qué hijo quieren revisar. Por su parte, desde el perfil de profesores, en el apartado de cursos aparecerán tantos enlaces como cursos y grupos tengan asignados.

Para permitir el acceso a un componente añadiendo un identificador en su ruta, es necesario hacer la configuración de dicha ruta desde el módulo del componente. A continuación, se muestran los métodos del componente y las rutas definidas en el módulo.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede al menú principal. Se encarga de pasar el identificador a la vista para que esta lo envíe con cada ruta que pueda seleccionar el usuario
<b>onHijoChange</b>	Accesible únicamente desde un perfil de padre, detecta el cambio de hijo desde el selector y carga el identificador del hijo seleccionado para pasarlo a las rutas.
<b>ngOnInit* (profesores)</b>	Carga los cursos asignados que tiene un profesor y los pasa a la vista para mostrarlos en el panel de cursos.

*Tabla 16 - Métodos de los componentes de Menú*

### 5.5.3 Componente *AltaEmpleado*

Se trata de la primera ruta accesible desde el perfil de administrador. Este componente permite al administrador dar de alta a un empleado. También será accesible desde el propio administrador cuando acceda a los datos de un empleado concreto y desde el perfil del propio empleado para ver sus datos. Por este motivo es importante detectar desde qué perfil y en qué condiciones se está accediendo y para ello según se accede a este componente se guardan los identificadores que puedan venir

en la ruta. Para poder acceder al componente tanto con un identificador como con dos en la ruta de navegación, es necesario configurar ambas rutas en el módulo.

Cuando el administrador pretende dar de alta a un usuario, en la ruta solamente aparece el perfil del administrador. Cuando se accede al perfil de un usuario ya dado de alta, en la ruta se verá primero el identificador del administrador y después el del empleado. Y cuando se accede desde el propio perfil del empleado únicamente se ve el identificador del empleado.

Esto es importante tenerlo en cuenta ya que no todas las funcionalidades van a ser accesibles desde un perfil que no sea el de administrador, pues el resto de los roles no tienen permisos de edición.

Además, según se ingresa en este componente, se reciben de los servicios pertinentes tanto la combinación de todos los cursos y grupos como los municipios y se crea el formulario.

Por otro lado, existen métodos que manejan el formulario según en qué condiciones y que se encargan de recoger los datos procedentes de la vista y mandarlos al método del servicio que permite almacenarlos en la colección 'Usuarios'.

Método	Descripción
ngOnInit	Método que se ejecuta cada vez que el usuario accede al componente altaEmpleado. Se encarga de obtener los identificadores de la ruta de navegación y según la procedencia de estos, de cargar los datos del usuario. También se encarga de obtener los municipios e inicializa los formularios para que se pueda interactuar con ellos a través de la vista.



<b>setUpCursoForm</b>	Es el encargado de generar las combinaciones de cursos y grupo para, desde la vista, añadir tantos <i>checkboxes</i> como combinaciones haya.
<b>toggleCheckboxes</b>	Permite bloquear la edición de los <i>checkboxes</i> cuando se han seleccionado perfiles distintos al de profesor.
<b>onSubmit</b>	Confirma que los datos del formulario sean válidos y según si ya existe o no el usuario se acuden al método agregar o actualizar del servicio.
<b>resetCheckboxes</b>	Permite resetear los <i>checkboxes</i> que se han seleccionado al crear un usuario
<b>darDeBajaEmpleado</b>	Pregunta al usuario si está seguro de eliminar al usuario y si es así envía su identificador al servicio.

Tabla 17 - Métodos del Componente Alta Empleado

#### 5.5.4 Componente *AltaAlumno*

Se trata de un componente muy parecido en cuanto a lógica al anterior. Si bien comparten la llamada a métodos del servicio, la vista del componente alumno contempla la recogida de datos para crear un perfil exclusivo de 'Alumno'.

De nuevo en este componente, se debe hacer una comprobación previa en la ruta de navegación para determinar, si el administrador ha entrado a dar de alta a un alumno, si ha entrado a consultar el perfil de un alumno ya dado de alta, si ha entrado un alumno a ver su propio perfil o si ha entrado un padre a ver el perfil de su hijo. En cada uno de estos casos, la ruta de navegación será diferente.

Al acceder a este componente, se cargan los municipios y los grupos, que serán pasados a la vista para presentarlos en el selector y que el administrador pueda asignar el correcto. Desde el módulo del componente hay que definir las rutas tanto para permitir el acceso a través de un identificador como de dos, como ya se explicó antes.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede al componente altaAlumno. Se encarga de obtener los identificadores de la ruta de navegación y según la procedencia de estos, de cargar los datos del usuario. También se encarga de obtener los municipios y grupos e inicializa los formularios para que se pueda interactuar con ellos a través de la vista.
<b>onSubmit</b>	Se ejecuta al darle al botón guardar cambios. Comprueba si el formulario es correcto y si es así comprueba si el usuario ya existe o no para saber a qué método del servicio llamar
<b>darDeBajaAlumno</b>	Se ejecuta cuando desde la vista se hace clic en el botón dar de baja y llama a los métodos necesarios del servicio para, además de eliminar al alumno, eliminar al padre si este no tiene más hijos.

Tabla 18 - Métodos del Componente Alta Alumno

### 5.5.5 Componente *BuscadorPersonal*

Este componente es accesible desde el enlace 'Buscador' del menú principal de administradores y nos redirige a un buscador en el que aparecerá una lista de tarjetas de usuarios con sus nombres y sus roles o cursos y grupos en el caso de los alumnos.

Además, esta lista estará encabezada por un buscador desde el que se recibirá a través de la vista el contenido que el usuario proporcione, para solicitar los datos necesarios a través del servicio.

Para acceder a este componente es necesario configurar la ruta en el módulo, para que permita acceder vía un identificador en la ruta de navegación.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede al buscador. Se encarga de obtener la lista de todos los usuarios por orden alfabético y de obtener el identificador de la ruta de navegación.
<b>buscarUsuarios</b>	Se encarga de hacer la llamada al servicio pasando los datos que introduce el usuario para proporcionar a la vista los resultados
<b>seleccionarAlumno</b>	Se ejecuta cuando el administrador hace clic sobre la tarjeta del usuario deseado y se encarga de redirigirlo a su perfil de alumno o empleado con los identificadores necesarios.

Tabla 19 - Métodos del Componente Buscador

### 5.5.6 Componente *Recordatorios*

Se trata de un componente al que pueden acceder todos los perfiles. Para ello, se debe configurar el módulo y añadirle una ruta mediante un identificador. Una vez se haya accedido, se presentará la lista de recordatorios que tenga cada usuario.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede a sus recordatorios. Se encarga de obtener la lista de recordatorios y pasarlos a la vista y obtener el identificador de la ruta de navegación.

Tabla 20 - Métodos del Componente Recordatorio

A su vez, desde este componente, si el usuario hace clic en el botón “Nuevo Recordatorio”, se dirigirá al componente *nuevoRecordatorio*, al que solo se puede acceder de esa forma.

Una vez aquí, de nuevo se cogerá la ruta del usuario que accede y se cargará el formulario para que se pueda añadir un nuevo recordatorio.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede a crear un nuevo recordatorio. Obtiene el identificador del usuario de la ruta y carga el formulario.
<b>onSubmit</b>	Crea el recordatorio si el formulario es correcto

Tabla 21 - Métodos del Componente Nuevo Recordatorio

### 5.5.7 Componente *PlantillaHorario*

A este componente se puede acceder a través del enlace “Gestión de horarios” del perfil de administrador. Se encarga, por un lado, de obtener los horarios de todos los cursos y pasarlos a la vista para presentarlos a demanda del usuario y, por otro lado, permite al administrador crear o actualizar los horarios.

Además, este componente será accesible desde los perfiles de ‘Conserjería’ y de ‘Profesorado’, pero sin permisos de edición. Para lograr esto de nuevo es necesario gestionar los identificadores que vienen en la ruta de acceso para saber qué rol tiene el usuario que trata de acceder.

La configuración de rutas en el módulo del componente se mantiene igual que las que se han definido hasta ahora en el resto.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede a la gestión de horarios o bien a consultar horarios desde los perfiles de profesor y conserjería. Obtiene el identificador del usuario de la ruta y las combinaciones de curso y grupo para presentarlas en un selector a través de la vista.
<b>onSelectChange</b>	Se ejecuta cuando hay se cambia la opción del selector

<b>cargarHorario</b>	Se encarga de cargar un primer horario por defecto, correspondiente al curso "1ºA", por ser el primero. Además, se ejecuta cuando se selecciona un curso diferente en el selector.
<b>guardarHorario</b>	Se ejecuta cuando el administrador le da al botón guardar, recogiendo los datos que envía la vista y pasándolos al método del servicio correspondiente.

Tabla 22 - Métodos del Componente Plantilla Horario

### 5.5.8 Componente Chat

Este componente, al igual que el anterior, también es accesible desde todos los perfiles de la aplicación. En primer lugar, el usuario se dirigirá a un buscador, similar al explicado en apartados anteriores. Desde esta vista, en la ruta de navegación debe aparecer el identificador del usuario, por lo que el módulo debe tener definida una ruta que le permita acceder con un identificador.

<b>Método</b>	<b>Descripción</b>
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede seleccionar chat. Además, obtiene el identificador del usuario a través de la ruta de navegación.
<b>onSelectChange</b>	Se ejecuta cuando hay se cambia la opción del selector
<b>loadAlumnos</b>	A través del mismo método del servicio usuarios que en el componente buscador, obtiene la lista de todos los usuarios de la aplicación y la pasa a la vista para generar un componente usuarioChat por cada uno de ellos
<b>buscarAlumnos</b>	Permite responder a la búsqueda concreta que hace el usuario a través de la vista

Tabla 23 - Métodos del Componente Buscador Chat

Una vez que se haya seleccionado a un usuario, se dirigirá al componente 'Chat'. La ruta que se define desde el módulo debe contener los identificadores de los dos usuarios implicados en el chat.

Cuando se accede al componente, se deben coger ambos identificadores y comprobar si existe algún documento en la tabla 'Chat' que contenga ambos, eso significará que los usuarios ya han iniciado su conversación y deberán traerse los datos desde el servicio para pasarlos a la vista. De lo contrario se debe generar un nuevo documento.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede al chat con un alumno. Obtiene los identificadores de ambos usuarios para recuperar o crear el chat y obtiene el objeto usuario del receptor para pasar a la vista los datos de identificación del usuario con el que se va a chatear.
<b>sendMessage</b>	Recoge los mensajes enviados por el usuario. Al tratarse de una base de datos en tiempo real, cada usuario recibe los mensajes en el momento en el que el otro usuario los envía.

*Tabla 24 - Métodos del Componente Chat*

### 5.5.9 Componente *Horario*

Se trata de un componente similar al de 'Plantilla horario'. Sin embargo, este es accesible únicamente por alumnos y padres y recibe el Horario que previamente ha creado en administrado por medio de la plantilla. En el caso de este componente tanto los alumnos como sus padres, solo tendrán acceso al horario del curso y grupo al que pertenece el alumno. Para poder acceder desde ambos perfiles se deben configurar ambas rutas desde el módulo del componente.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede al horario. Almacena los identificadores de la ruta de navegación.
<b>cargarHorario</b>	Carga el horario según el curso y el grupo del usuario que recibe por parámetro. Cuando tiene los datos los almacena en una variable que pasa a la vista para presentarlos.

Tabla 25 - Métodos del Componente Horario

### 5.5.10 Componente *Incidencias*

Se trata de un componente accesible desde el perfil de padres y de alumnos, con la diferencia de que, si accede un padre, tendrá la opción de justificar la falta mientras que el alumno solamente puede verla. Para acceder, es necesario definir dos rutas en el módulo, una que permita acceder con un identificador para cuando acceda el alumno, y otra que permita acceder con dos identificadores cuando accede el padre.

Si un padre quiere justificar la falta, bastará con darle al botón 'Justificar' en la vista y aportar un texto a modo de justificación.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede a las incidencias. Almacena los identificadores de la ruta de navegación y carga las incidencias del identificador que pertenezca a un alumno. Además, en el caso de las que ya estén justificadas, oculta esta justificación a los alumnos.
<b>mostrarJustificacion</b>	Se encarga de gestionar cuando hay que mostrar y cuando no el botón justificar para notificárselo a la vista
<b>justificarIncidencia</b>	Permite recoger los datos de justificación que el usuario ha escrito, mandarlos al servicio y ocultar el cuadro que permite añadir dicha justificación.

Tabla 26 - Métodos del componente Incidencias

### 5.5.11 Componente *TablonAnuncios*

Este componente es accesible desde todos los perfiles y permite ver aquellos anuncios de los que son destinatarios. En este caso, el módulo solo guarda una ruta configurada para acceder mediante un identificador.

Desde este componente solo se podrá añadir un anuncio si se tiene un rol de 'Jefatura' o de 'Administrador'. Además, los profesores podrán añadir anuncios a los cursos que tienen asignados.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede al tablón de anuncios. Se encarga de obtener los roles para poder presentarlos como <i>checkboxes</i> a la vista para poder seleccionar a los destinatarios. También se encarga de pedir los anuncios según el rol o según el curso y el grupo de los alumnos para pasárselos a la vista y que el usuario los pueda ver.
<b>createForm</b>	Este componente carga un formulario a través del cual los usuarios con permisos añadan un nuevo anuncio.
<b>agregarAnuncio</b>	Comprueba que los datos del formulario sean válidos, los manda al servicio y limpia los campos del formulario.
<b>destinatariosFormArray</b>	Recoge los destinatarios seleccionados por el anuncio para pasarlos al formulario en un formato apropiado.
<b>toggleNuevoAnuncioForm</b>	Se encarga de gestionar cuando se debe mostrar el formulario y cuando no según la interacción del usuario con la vista.

Tabla 27 - Métodos del Componente Anuncios



## 5.5.12 Componente *Cursos*

Este componente es accesible desde el perfil de 'Jefatura' y desde el de 'Profesor', con ciertas diferencias para cada uno. El profesor puede añadir un anuncio al tablón del curso y el jefe de estudios puede seleccionar el curso y grupo del que quiere ver la lista de alumnos. Para acceder a este componente se debe definir una ruta en el módulo que permita el acceso con un identificador en la ruta de navegación.

Este componente permite, además, desde cada alumno, acceder al chat directo con el alumno, acceder al chat con sus padres y añadir una incidencia.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede a cursos. Se encarga de obtener la lista de alumnos que tengan el curso y grupo al que ha accedido un profesor o bien el que ha seleccionado el jefe de estudios a través del selector. Para poder generar este selector en la vista, también se encarga de obtener la lista de combinaciones de cursos y grupos
<b>onSelectChange</b>	Se ejecuta cada vez que el jefe de estudios selecciona un curso.
<b>createForm</b>	El formulario se carga al acceder al componente para recoger los datos en caso de que un profesor quiera añadir un anuncio.
<b>agregarAnuncio</b>	Comprueba el formulario y lo manda al servicio para crear el anuncio.
<b>toggleNuevoAnuncioForm</b>	Se encarga de gestionar cuando se debe mostrar el formulario y cuando no según la interacción del usuario con la vista.

Tabla 28 - Métodos del componente *Cursos*

Como se dijo anteriormente, desde este componente se puede añadir una incidencia al alumno, para ello el usuario debe hacer clic sobre el icono de la cruz y será redirigido al componente *mensajesEnviados*. La ruta de navegación contiene el identificador del profesor y del alumno amonestado por lo que el módulo debe tener la ruta configurada adecuadamente.

Método	Descripción
<b>ngOnInit</b>	Método que se ejecuta cada vez que el usuario accede a <i>mensajesEnviados</i> para poner una incidencia, recoge el identificador del alumno de la ruta de navegación
<b>enviarIncidencia</b>	Comprueba que estén completos todos los campos de la vista y envía los datos al servicio.

Tabla 29 - Métodos del Componente Añadir Incidencia

## 5.6 Componentes específicos

En este apartado definiremos a los componentes que se usan dentro de los componentes principales descritos en el apartado anterior. Al no tratarse de componentes independientes, ya que forman parte de las páginas, se han agrupado en otra carpeta.

### 5.6.1 Componente *HeaderNoLogin*

Se trata del componente de la cabecera que forma parte de aquellas páginas en las que el usuario todavía no está dentro de la sesión. La vista contiene los enlaces a 'Inicio', a la página de presentación a través de '¿Quiénes somos?', y al inicio de sesión. Además, incluye el logotipo y el nombre de la aplicación.

### 5.6.2 Componente *HeaderLogin*

Es un componente similar al anterior pero utilizado en aquellas páginas en las que el usuario ya ha iniciado sesión, la diferencia que tiene es que, en lugar del enlace a iniciar sesión, lo tiene a cerrar sesión y además cuenta con un enlace 'menú' que redirige al usuario a su menú principal.

### 5.6.3 Componente *Alumno-Curso*

Se trata del componente que aparece dentro de la página 'Curso', en concreto, uno por cada alumno que compone el curso y el grupo. Muestra el nombre del alumno y permite la redirección a las páginas de 'Incidencias' y del 'Chat'.

### 5.6.4 Componente *Anuncio*

Es el componente que crea el cuadro en el que se representan los anuncios que se añaden al tablón.

### 5.6.5 Componente *Bloque-Chat*

Se utiliza dentro del componente 'Chat' y forma el componente en el que se incluye la cabecera que identifica al usuario, con su foto, su nombre y su rol o curso y grupo y además los mensajes.

### 5.6.6 Componente *Identificación*

Se emplea en el inicio de sesión, desde donde los usuarios introducen sus credenciales para su posterior tratamiento y comprobación.

### 5.6.7 Componente *Recordatorio*

Similar al componente 'Anuncio', muestra el contenido de los recordatorios de los usuarios y tiene además un botón que permite su eliminación.

### 5.6.8 Componente *User-Welcome*

Es el componente que incluye la bienvenida al usuario, junto con su nombre y apellidos desde el menú principal independientemente del tipo de perfil.

### 5.6.9 Componente *usuario-alumno*

Se trata del componente que muestra la lista de usuarios desde el buscador del administrador. Ofrece la foto de perfil, el nombre, el curso y el grupo o el rol en el caso de perfiles que no sean alumnos.

### 5.6.10 Componente *usuario-chat*

Es un componente igual al anterior, salvo por cambios en el estilo. Se utiliza para que el usuario pueda seleccionar al usuario con el que quiera chatear. Presenta el nombre del usuario y su rol, o su curso y grupo en el caso de los alumnos.

## Capítulo 6. Guía de usuario

EN ESTE CAPÍTULO SE EXPLICARÁ EN DETALLE QUÉ USO TENDRÍA QUE HACER EL USUARIO DE LA APLICACIÓN PARA SACAR SU MÁXIMO PARTIDO, INDEPENDIENTEMENTE DEL PERFIL DE USUARIO DEL QUE VAYAN A HACER USO.

## 6.1 Contexto de la guía de usuario

Ya hemos examinado en detalle los mockups diseñados que representan nuestra visión y objetivo final para la aplicación. Estos mockups nos han proporcionado una imagen clara y concreta de la interfaz de usuario que deseamos lograr. A continuación, profundizamos en la compleja lógica que sustenta esta interfaz, abordando los mecanismos y procesos necesarios para convertir ese diseño visual en una herramienta práctica y funcional. Esta lógica abarca desde la gestión de datos hasta la implementación de las interacciones del usuario.

Ahora, presentaremos el resultado de esta implementación. En este punto del desarrollo, la aplicación ha alcanzado un nivel de solidez mayor y tiene capacidad suficiente para manejar datos de manera eficiente y ofrecer una experiencia de usuario integrada y fluida. A continuación, exploraremos cómo todas estas piezas se han ensamblado para dar lugar a una aplicación funcional y dinámica. Cabe destacar que debido a la herramienta de diseño que hemos utilizado (*TeleportHQ*), el resultado de la implementación será muy similar con el diseño ya mostrado en capítulos anteriores.

## 6.2 Guía de usuario

### 6.2.1 Inicio

Esta vista ha recibido un número mínimo de cambios durante la implementación, ya que no implica ningún flujo de datos. Los cambios que se han realizado sobre ella han sido meramente de estilo.



Figura 43 - Pantalla Principal

Si seguimos leyendo hacia abajo haciendo *scrolling* nos encontraremos con la siguiente información.



Figura 44 - Pantalla Principal Características

En este panel inferior, el usuario podrá leer acerca de las ventajas y comodidades que ofrece *SchoolApp*.

### 6.2.2 ¿Quiénes somos?

A esta pantalla podemos acceder desde la cabecera. Nos mostrará una breve explicación acerca del funcionamiento de la herramienta y para quién está destinada.



Figura 45 - Presentación Parte 1

Para ver todo el contenido de este apartado es necesario que el usuario siga navegando hacia abajo.



Figura 46 - Presentación Parte 2

A lo largo de la explicación el usuario encontrará esta imagen que pretende denotar la importancia de la incorporación de la tecnología en las aulas.




## ¿Cómo funciona?

Tras iniciar sesión se abrirá el perfil correspondiente a cada usuario, entre los que se encuentran "alumnos", "padres/madres", "profesorado", "administración" y "conserjería". Cada uno de ellos cuenta con funciones específicas diseñadas para optimizar su participación y contribución al entorno educativo. Desde la gestión académica y la comunicación fluida hasta la colaboración en proyectos y la supervisión administrativa, esta aplicación ofrece un conjunto diverso de herramientas intuitivas que se adaptan a las necesidades individuales de cada usuario.

Figura 47 - Presentación Parte 3

Si el usuario continúa haciendo *scrolling*, seguirá leyendo información acerca de la plataforma, como vemos en la Figura--, hasta llegar al formulario de contacto.



¿Quieres saber más? Contáctanos

Fernando Sánchez

fernando.sanchez@gmail.com

637023640

Buenos días, quería más información sobre la aplicación. Estoy interesado en incorporarla en mi centro de enseñanza. Gracias.

Enviar

Figura 48 - Presentación Formulario De Contacto

Por último, el usuario encontrará este formulario, en el cual deberá rellenar todos los campos y enviar para que posteriormente un especialista se ponga en contacto con él y le brinde la información que necesita. Una vez que se ha enviado la solicitud el usuario verá el siguiente mensaje.

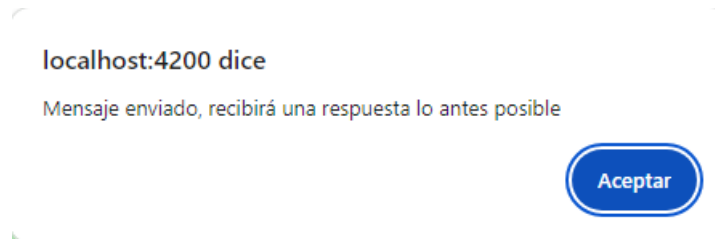


Figura 49 - Respuesta al Formulario de Contacto

### 6.2.3 Login

Esta pantalla es accesible tanto desde el enlace “Iniciar Sesión” de la cabecera como desde el botón “Empezar” en la página de inicio. Además, será la pantalla a la que se redirigirá al usuario cuando este cierre la sesión.

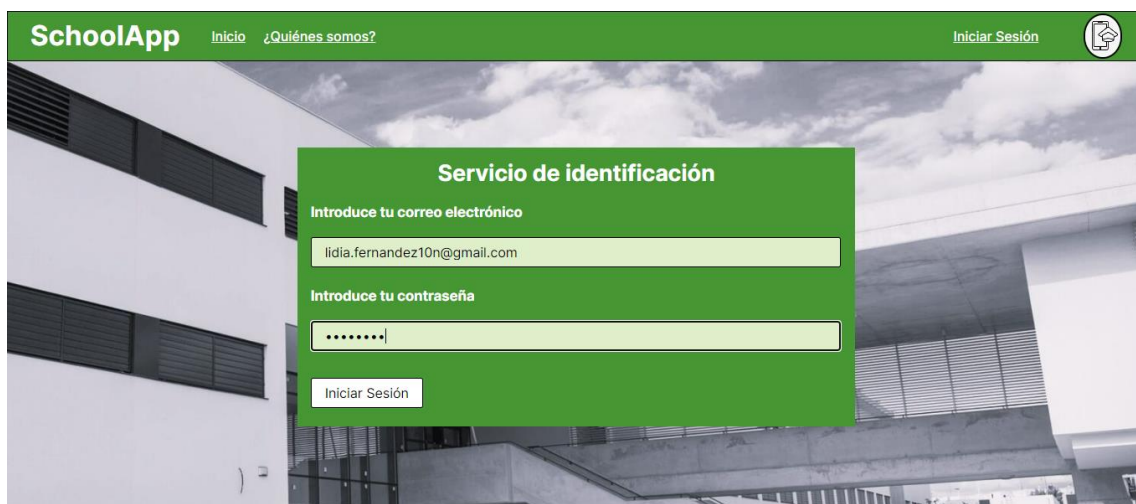


Figura 50 - Identificación

### 6.2.4 Menú Administrador

Una vez que el usuario haya introducido sus credenciales, y si estas son correctas, será redirigido al menú principal que le corresponda según su rol.

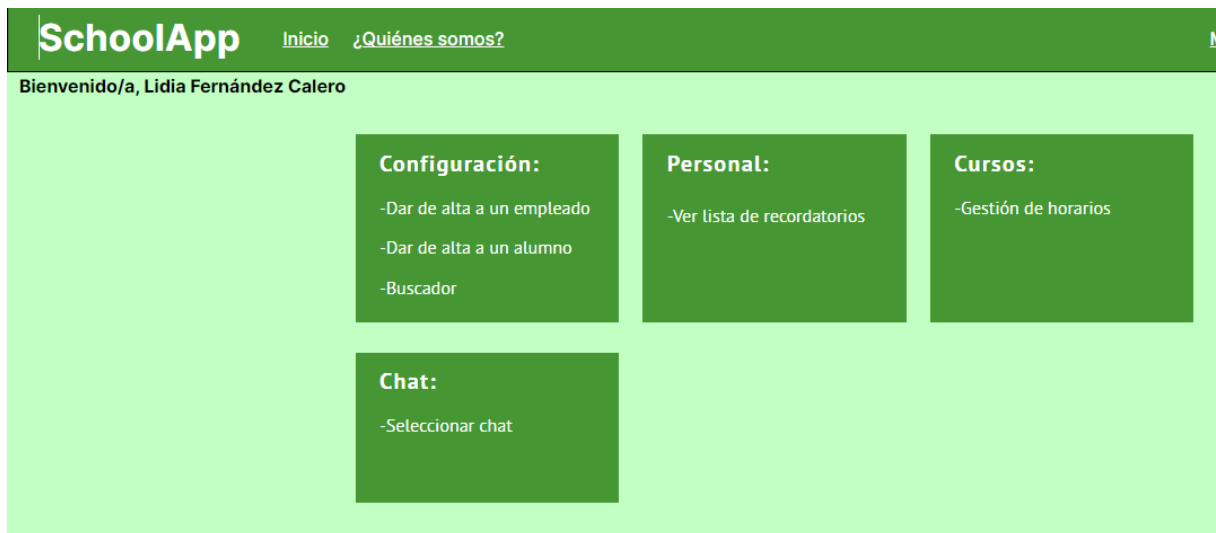


Figura 51 - Menú de Administrador

En este caso, se trata del menú principal de los administradores. En él podemos ver los módulos a los que tiene acceso que son: 'Configuración', 'Personal', 'Cursos' y 'Chat'. Cada uno con sus funcionalidades.

Además, desde esta pantalla, el componente de la cabecera cambia y en vez de mostrar el enlace para iniciar sesión, pasa a contar con un enlace para volver al menú principal y otro para cerrar la sesión.

Por último, desde esta pantalla, en la barra de navegación se mostrará tanto la ruta del componente como el identificador del usuario.

```
localhost:4200/mainadmin/SRPABVtkktGE6ymR7scS|
```

Figura 52 - Ruta de Navegación en Acceso a Menú de Administrador

### 6.2.5 Dar de alta a un empleado

En esta pantalla será indispensable completar los campos mínimos requeridos en función del rol que se haya elegido para el usuario.

**SchoolApp** Inicio ¿Quiénes somos? Menú Cerrar

**Datos del empleado:**

Profesor/a
  Conserjería
  Administración
  Jefatura

Dunia Gómez Gómez 44444444A 26/06/1985  
 Calle Tomás Morales, 48, 3º Izc Las Palmas de Gran Canaria 35004  
 637023640 DG44krd dunia.gomez@gmail.com 2023/2024  
 IES Pérez Galdós Las Palmas de Gran Canaria Matemáticas

**Grupos asignados:**

1º - A	1º - B	1º - C	1º - D	2º - A	2º - B	2º - C	2º - D	3º - A	3º - B	3º - C	3º - D	4º - A	4º - B	4º - C	4º - D
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 53 - Formulario para dar de Alta a un Empleado

En el caso de dar de alta a un usuario que no sea profesor, el cuadro ‘Grupos asignados’ no será editable. Además, el campo ‘Departamento’, no será obligatorio. En caso de que el usuario elija guardar los cambios y todos los datos tengan su formato correcto, se mostrará el siguiente mensaje y se limpiarán los campos del formulario.

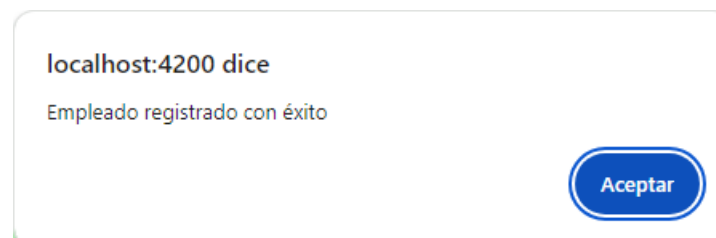


Figura 54 - Respuesta al Registrar Empleado Correctamente

## 6.2.6 Dar de alta a un alumno

Se trata de una pantalla con un caso de uso similar al anterior. El usuario debe rellenar todos los campos, de no ser así no se podrá guardar al usuario y automáticamente se crearán los perfiles de padre y de alumno.

**SchoolApp** Inicio ¿Quiénes somos? Menú Cer

**Datos del alumno/a:**

Leticia Esther Santana Robaina	45365704K	637023640
Urb. Casa Blanca, Blq 3, Pt 7	Las Palmas de Gran Canaria	24/10/2000
35016	leticia.santana@gmail.com	12345678
Alexander Santana Pérez	45555555A	
alexander.santana@gmail.com	12345678	
Repite curso: Sí <input type="radio"/> No <input checked="" type="radio"/>	Servicio de recogida: Sí <input checked="" type="radio"/> No <input type="radio"/>	

1º ESO	2º ESO	3º ESO	4º ESO
Atención Educativa <input type="radio"/> Religión <input checked="" type="radio"/>	Atención Educativa <input type="radio"/> Religión <input type="radio"/>	Atención Educativa <input type="radio"/> Religión <input type="radio"/>	Atención Educativa <input type="radio"/> Religión <input type="radio"/>
Ciencias de la naturaleza	Ciencias de la naturaleza	Biología y Geología	Biología y geología
Ciencias Sociales, geografía e historia	Ciencias Sociales, geografía e historia	Ciencias Sociales, geografía e historia	Francés <input type="radio"/> Historia de España <input type="radio"/>

Figura 55 - Pantalla de Formulario para Dar de Alta a un Alumno

Si desplazamos hasta el final de la página, veremos los mismos botones que en el caso anterior. Si queremos guardar los cambios y los datos son correctos veremos el siguiente mensaje y se reiniciarán los campos del formulario.

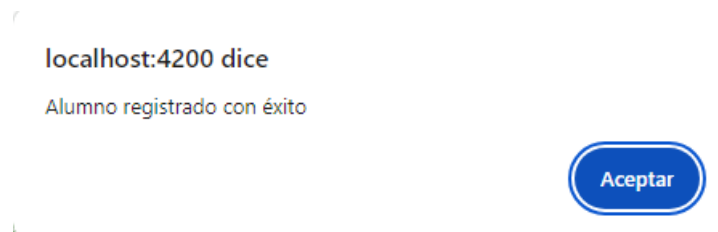
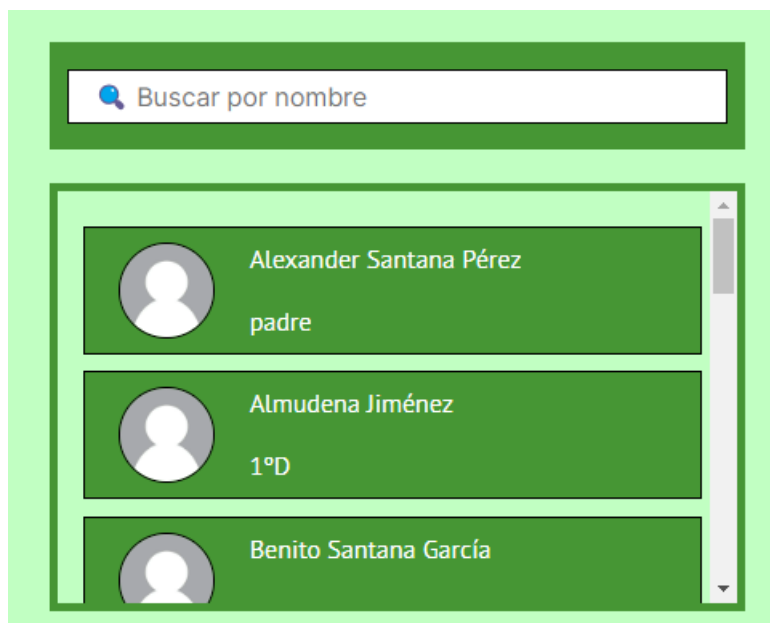


Figura 56 - Respuesta de Alumno Registrado Correctamente

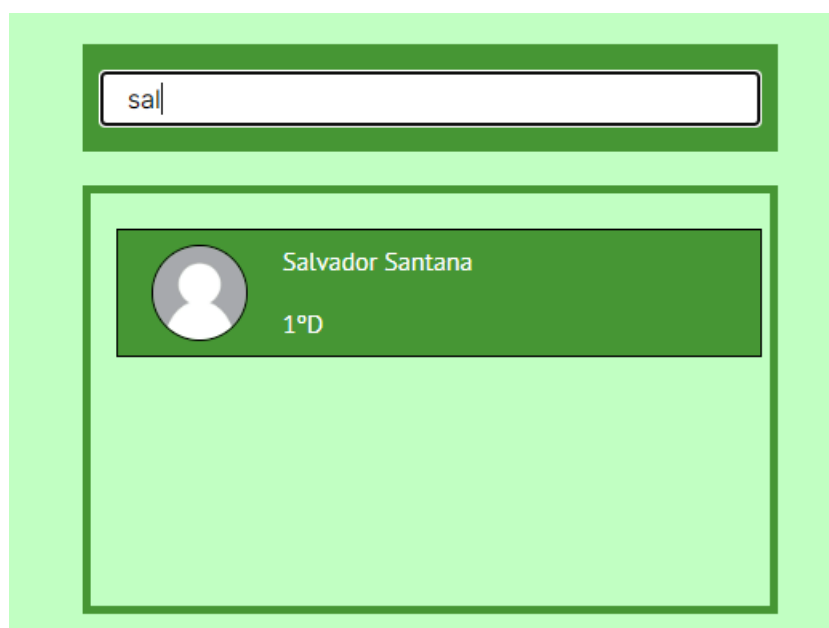
## 6.2.7 Buscador

Desde esta pantalla, se verá principalmente la lista de usuarios del centro por orden alfabético.



*Figura 57 - Buscador de Usuarios*

Si queremos concretar la búsqueda, se debe introducir un nombre en el buscador y automáticamente irán apareciendo los resultados que coincidan.

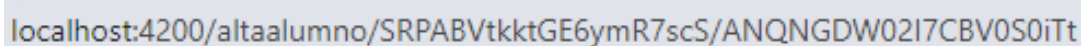


*Figura 58 - Buscador de Usuarios por nombre*

Cuando se encuentre el alumno deseado bastará con hacer clic sobre la tarjeta.

## 6.2.8 Alta alumno (perfil)

Si hemos seguido los pasos desde el buscador y hemos seleccionado a un usuario, nos encontraremos ahora en su perfil independientemente de si era un alumno, un padre o cualquier otro rol. La ruta de navegación, por tanto, incluye el identificador del administrador y el del alumno en concreto.



localhost:4200/altaalumno/SRPABVtkktGE6ymR7scS/ANQNGDW02I7CBV0S0iItt

Figura 59 - Ruta de navegación para que un Adminsitrador acceda al perfil de un Alumno

Desde aquí podemos actualizar algún dato o dar de baja al alumno. En cualquiera de los casos, el mensaje que nos aparecerá será el siguiente:

-Actualizar datos:

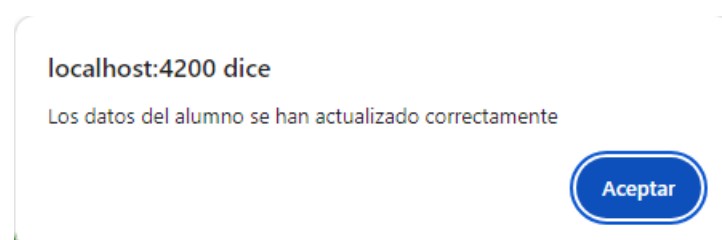


Figura 60 - Respuesta de datos Actualizados Correctamente

-Dar de baja:

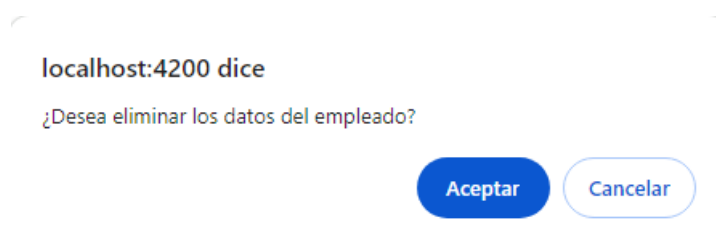


Figura 61 - Confirmar si se desea Dar de Baja al Empleado

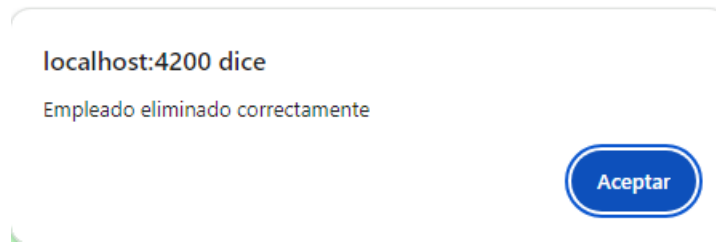


Figura 62 - Confirmación de eliminación de datos

## 6.2.8 Recordatorios

Al llegar a esta pantalla lo primero que veremos será la lista de recordatorios que ya tenga el usuario.

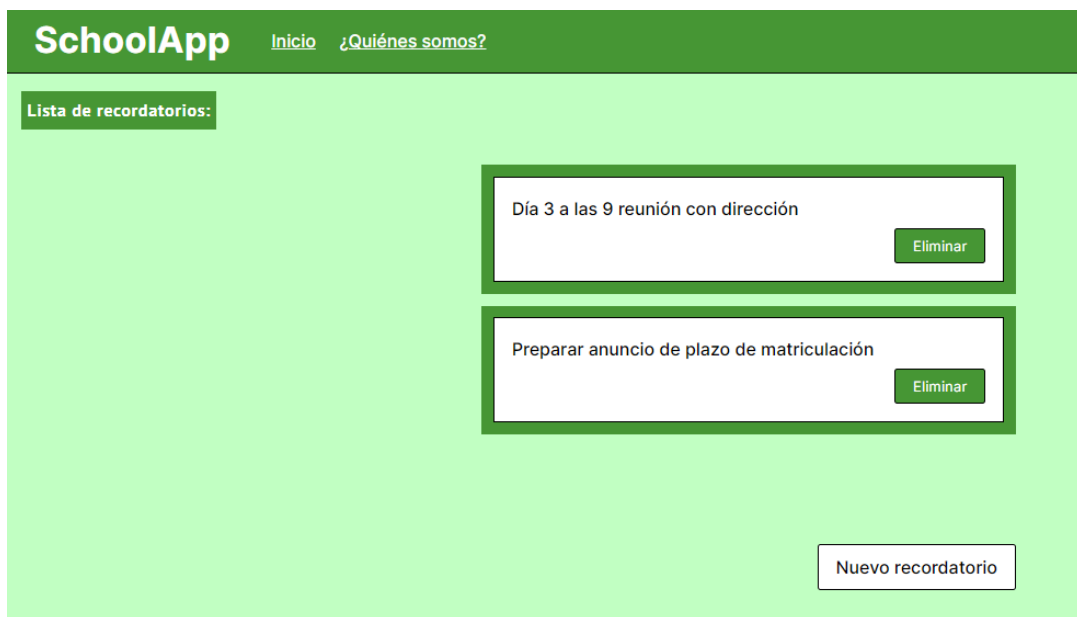


Figura 63 - Lista de Recordatorios

Si se le da al botón “Nuevo recordatorio” nos dirigiremos a la siguiente página.



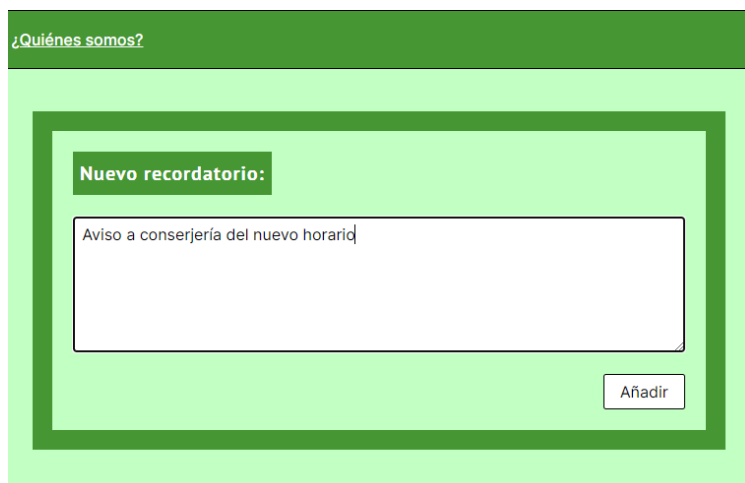


Figura 64 - Añadir Recordatorio

El mensaje que veremos cuando se cree el recordatorio será:

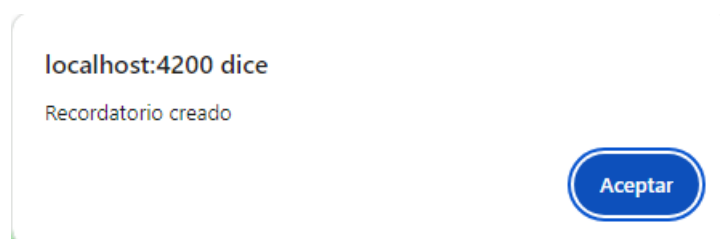


Figura 65 - Respuesta a Recordatorio creado correctamente

Desde la lista principal, si le damos a eliminar un recordatorio veremos de nuevo un mensaje de pregunta y uno de confirmación.

## 6.2.9 Plantilla Horario

Para llegar aquí debemos acceder desde el menú principal a 'Gestión de horarios'. Desde aquí veremos la plantilla editable y un selector para elegir el curso que se debe configurar. Puede darse la posibilidad de que ya esté el horario completo y se vaya a realizar el cambio o que la plantilla esté vacía y se deban incluir las asignaturas.

Hay que recordar que este perfil es accesible desde el perfil de 'Conserjería' y desde el de 'Profesorado', sin embargo, solo tendrán permisos de lectura y no de edición de los horarios.

	Lunes	Martes	Miércoles	Jueves	Viernes
07:55-08:50	Inglés	Lengua	Naturales	Ed. Física	Matemáticas
08:50-09:45	Lengua	C sociales	Inglés	Tecnología	Inglés
09:45-10:40	C Sociales	Naturales	Matemáticas	Matemáticas	Lengua
11:10-12:05	Matemáticas	Tecnología	Lengua	Plástica	C Sociales
12:05-13:00	Tecnología	Inlgés	Ed Física	Naturales	Plástica
13:00-13:55	Francés	Francés	Plástica	Religión	Religión

Figura 66 - Plantilla de Horario

En caso de guardar el horario, si todo ha ido bien veremos el siguiente mensaje:

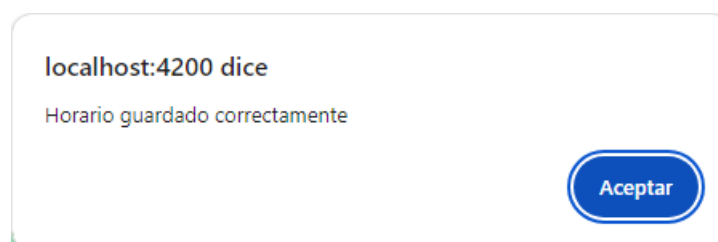


Figura 67 - Respuesta al Guardar Horario Correctamente

## 6.2.10 Nuevo Chat

En este caso veremos un buscador similar al que vimos en componentes atrás. Su funcionamiento es igual salvo que al elegir a un usuario iremos al chat con ese

usuario. Desde el chat, en la ruta de navegación se verá tanto el identificador del usuario destinatario y el del usuario actual.

Una vez dentro, se cargarán los mensajes existentes y según se van escribiendo será suficiente con darle a la tecla *Enter*.

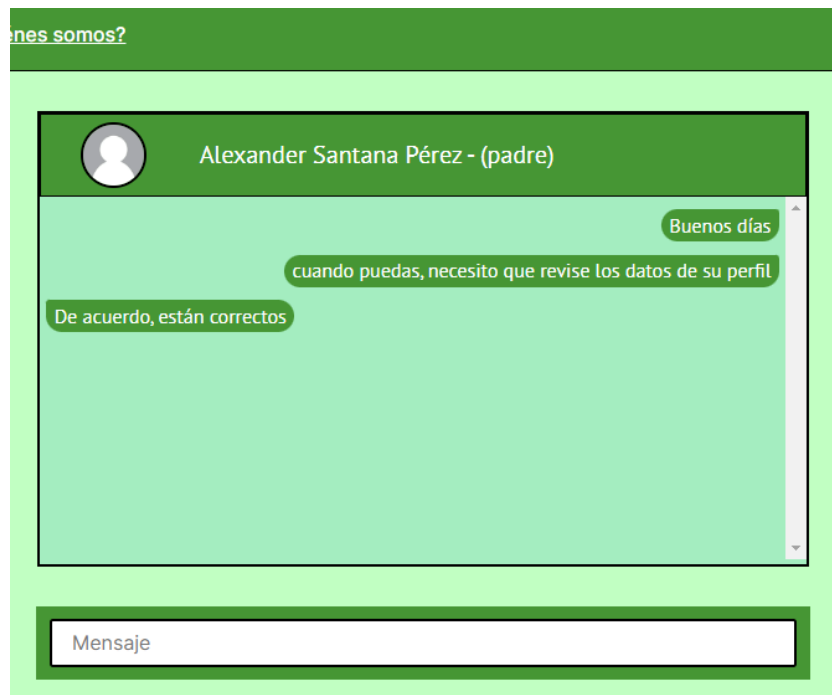


Figura 68 - Chat con Usuario

### 6.2.11 Menú Alumno

Se trata de la pantalla principal a la que llegan los usuarios que tienen el perfil de alumno. En este caso hay funcionalidades que ya se han documentado así que nos centraremos en aquellas que no se hayan descrito con el perfil anterior.



Figura 69 - Menú de Alumnos

## 6.2.12 Mi perfil

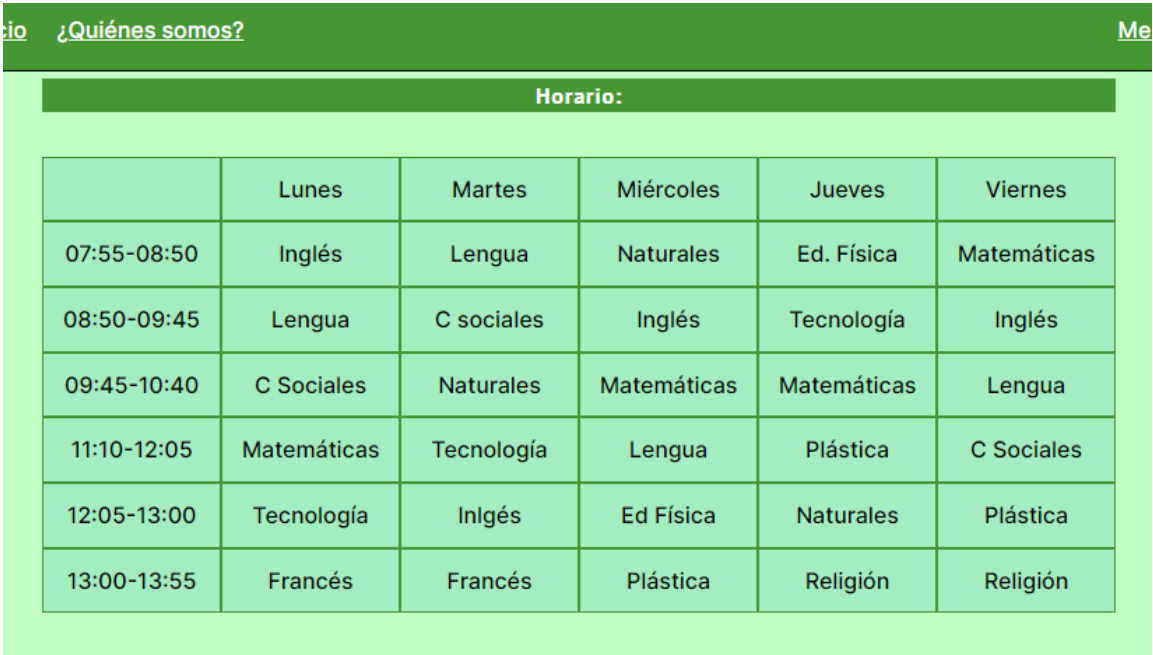
Desde esta pantalla el alumno tiene acceso sin edición a sus datos de registro. En este caso, la ruta de navegación contiene solo el identificador del alumno.

Figura 70 - Alumno accediendo a su Perfil

Si el alumno sigue navegando hasta el final de la página en este caso no estarán los botones de edición. En cambio, si viera algún error en los datos de su perfil, podría directamente contactar con el administrador a través del chat.

## 6.2.13 Horario

El alumno solo puede ver el horario de su curso y grupo. Como se puede ver en la Figura-- en este caso el alumno pertenece al curso '1ºD', por lo que directamente al acceder al horario verá sus asignaturas semanales.



	Lunes	Martes	Miércoles	Jueves	Viernes
07:55-08:50	Inglés	Lengua	Naturales	Ed. Física	Matemáticas
08:50-09:45	Lengua	C sociales	Inglés	Tecnología	Inglés
09:45-10:40	C Sociales	Naturales	Matemáticas	Matemáticas	Lengua
11:10-12:05	Matemáticas	Tecnología	Lengua	Plástica	C Sociales
12:05-13:00	Tecnología	Inglés	Ed Física	Naturales	Plástica
13:00-13:55	Francés	Francés	Plástica	Religión	Religión

Figura 71 - Alumno Accediendo a su Horario

## 6.2.14 Incidencias (alumno)

A continuación, veremos cómo queda el acceso al apartado de incidencias desde el perfil del alumno. En este caso, solo podrá ver la incidencia y en ningún momento podrá justificarlas o ver la justificación que su padre ha añadido.

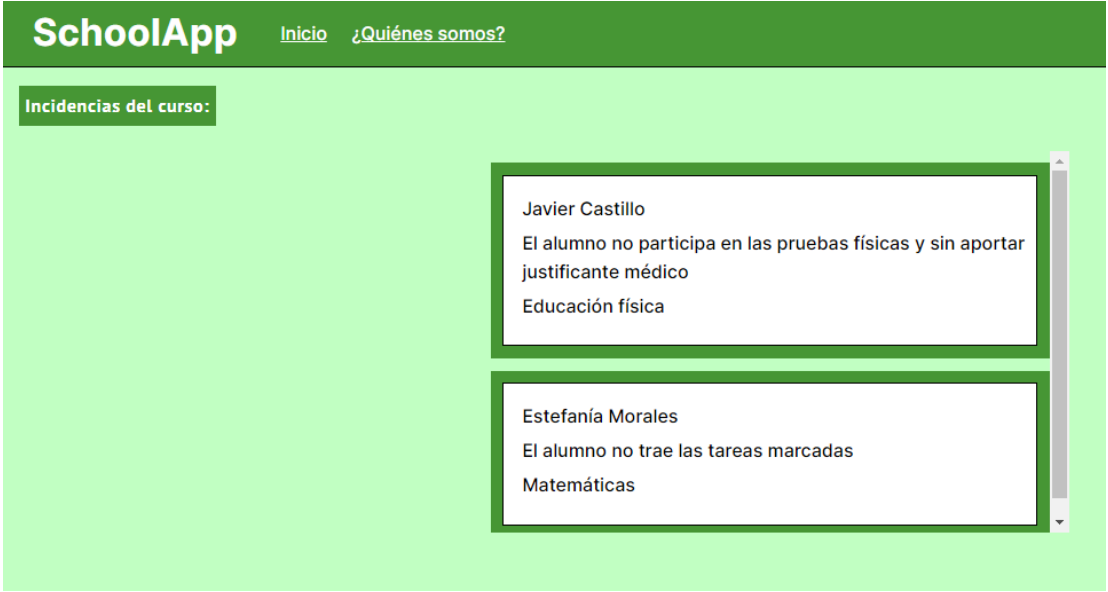


Figura 72 - Incidencias del Alumno

### 6.2.15 Tablón anuncios (alumno)

Aunque en este caso, el ejemplo se muestra desde el perfil de alumno, tendrá el mismo comportamiento desde un perfil de padre, profesor, conserje y jefe de estudios. Al entrar se verá una lista con los anuncios que haya recibido, sin embargo, no tendrá permisos para redactar y añadir un anuncio al tablón.

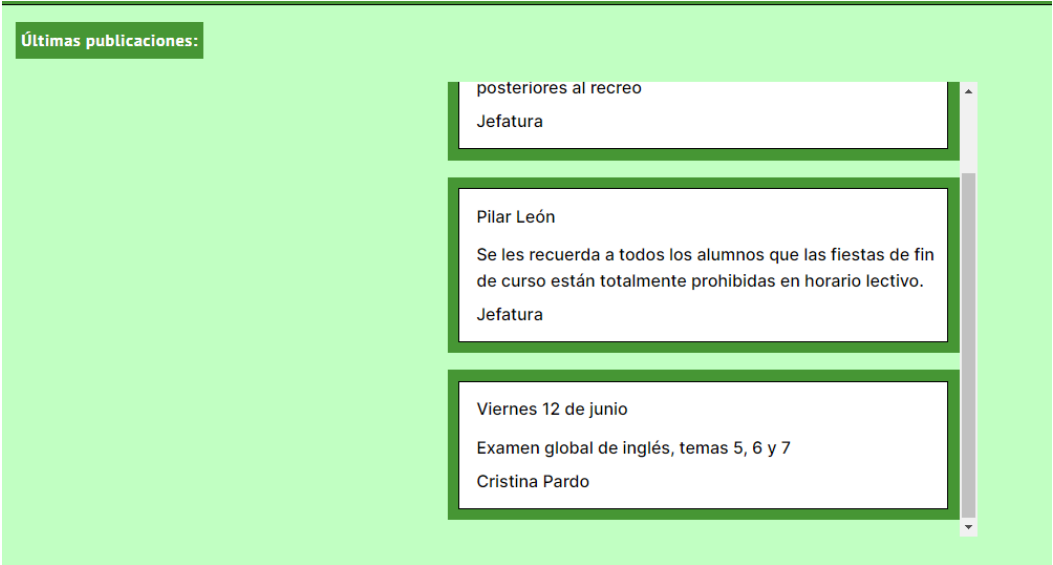


Figura 73 - Anuncios en el Tablón

En la Figura-- podemos ver que el alumno ha recibido anuncios por parte de la jefatura y además por parte de un profesor determinado que ha añadido un anuncio al curso completo.

## 6.2.16 Menú Conserjería

Desde el menú principal de perfil de 'Conserjería' veremos funcionalidades que ya se han comentado. Se trata del perfil con los accesos más generalizados y reducidos, pues tiene acceso a todos los horarios, al chat con cualquier usuario y al tablón de anuncios.



Figura 74 - Menú para Conserjes

## 6.2.17 Menú Jefatura

A continuación, se mostrará la Figura 75 que representa al menú principal de un perfil de 'Jefatura' y se detallarán aquellas funcionalidades que no se hayan explicado anteriormente.



Figura 75 - Menú para Jefes de Estudio

## 6.2.18 Acceso a cursos

Desde este apartado el jefe de estudios tiene acceso a la lista de alumnos de cualquier curso. Se trata de una página accesible también desde un perfil de 'Profesor'. Con la diferencia de que los profesores podrán añadir anuncios al tablón del curso en el que se encuentren y los jefes de estudio podrán seleccionar cualquier curso y ver la lista.

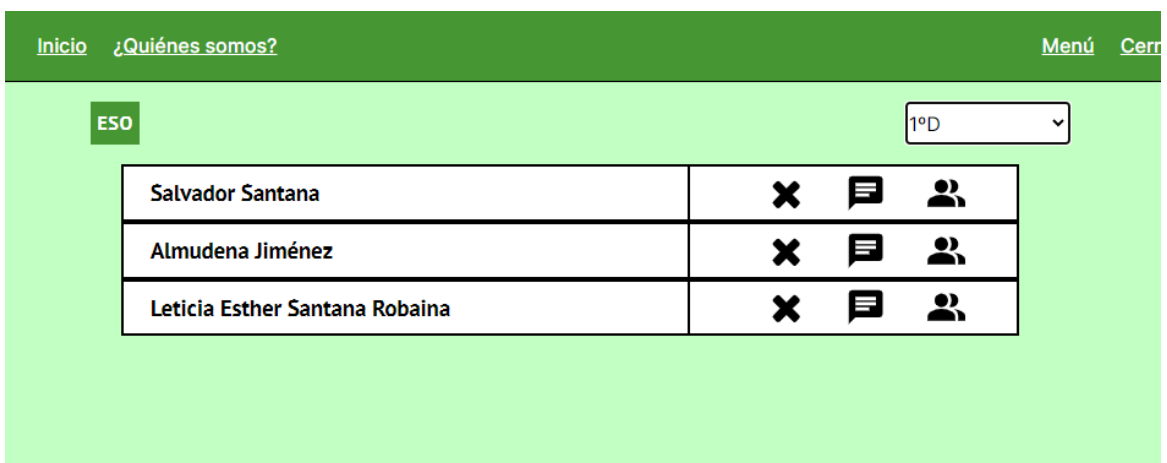


Figura 76 - Listado de Alumnos por Curso

Como se puede ver en la Figura 76, por cada alumno hay un acceso directo a “Añadir Incidencia”, “Chat con el alumno” y “Chat con los padres del alumno”.



## 6.2.19 Añadir incidencia

Se trata de una página accesible tanto por profesores como por jefes de estudio. El usuario accede a una página en la que la ruta de navegación incluye el identificador del alumno amonestado y el del usuario que pone la incidencia.

Para acceder, se debe ir al listado de alumnos por cursos y pulsar sobre la 'X', entonces, veremos la lista de incidencias pendientes que tiene el usuario y podremos ver las justificaciones de aquellas que las tengan.

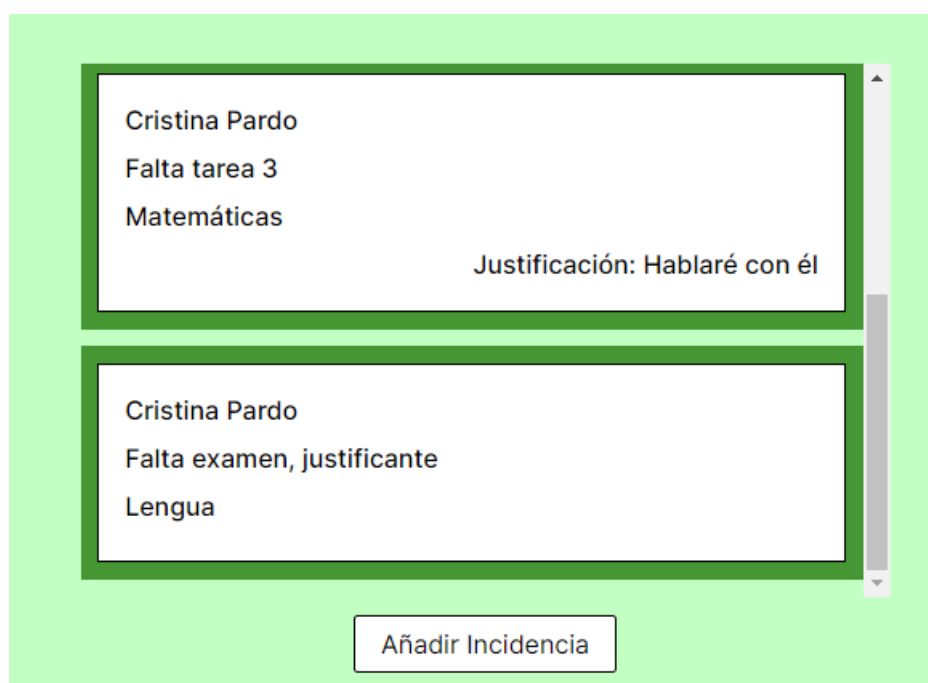


Figura 77 - Añadir Incidencia

Si posteriormente, decidimos añadir una incidencia, debemos completar el siguiente formulario:

Inicio ¿Quiénes somos? Menú

**Enviar incidencia**

General

Buenos días,  
Informarle de que durante las últimas semanas, su hijo a llegado alrededor de las 08:20. Recordarle que la primer hora comienza a las 07:55.  
Gracias

María Monzón

Enviar

Figura 78 - Formulario para Crear Incidencia

Si se han rellenado todos los campos del formulario, al enviar veremos el siguiente mensaje:

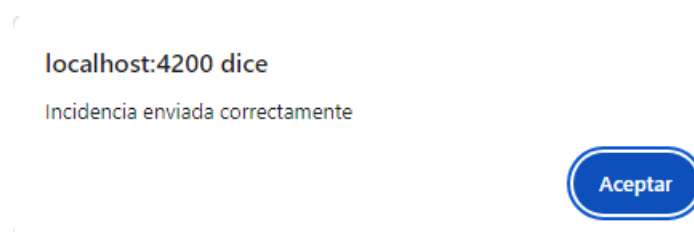


Figura 79 - Respuesta a Incidencia enviada correctamente

### 6.2.20 Mi perfil (empleado)

Se trata del acceso al perfil por parte de un usuario que no tenga el rol de alumno o el de padre. En este caso, la ruta de acceso cuenta únicamente con el identificador del trabajador. De nuevo, al no ser administrador, no cuenta con permisos de edición por lo que solo puede leer y revisar sus datos y si encuentra alguna incidencia podrá comunicársela al administrador vía chat.

Inicio ¿Quiénes somos? Menú Cel

Profesor/a
  Conserjería
  Administración
  Jefatura

María Monzón Trujillo 55554111D 08/01/1963

Calle Pío XII, 245 Las Palmas de Gran Canaria 35005

637023640 MPMG4948D maria.monzon49d@gmail.com 2023/2024

IES Alonso Quesada Las Palmas de Gran Canaria Departamento

Grupos asignados:

1º - A
  1º - B
  1º - C
  1º - D
  2º - A
  2º - B
  2º - C
  2º - D
  3º - A
  3º - B
  3º - C
  3º - D
  4º - A
  4º - B
  4º - C
  4º - D

Figura 80 - Empleado accediendo a su Perfil

### 6.2.21 Tablón Anuncios (Añadir)

El único perfil que puede enviar anuncios a los demás perfiles es el de 'Jefatura'. Por su parte, los profesores podrán acceder a esta página para añadir un anuncio a un curso concreto. Una vez que se haya accedido al tablón y se haga clic sobre "Nuevo anuncio" aparecerá en la pantalla el siguiente cuadro.

alumno
  conserje
  jefatura
  padre
  profesor

Entrega de notas

La entrega de notas del tercer trimestre será el 10 de junio de 16 a 18.  
Gracias

María Monzon

Añadir

Figura 81 - Jefe de Estudios añadiendo un Anuncio al tablón

Una vez enviado, si se han completado todos los campos veremos el siguiente mensaje.

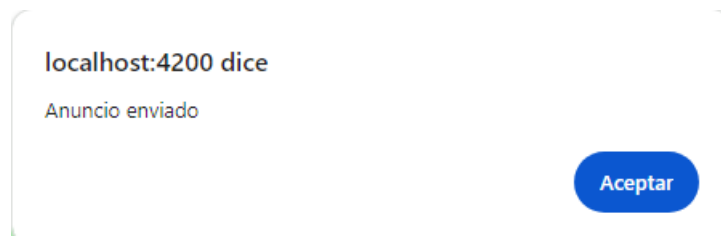


Figura 82 - Respuesta a Anuncio añadido correctamente

## 6.2.22 Menú Padres

Al acceder con un perfil de 'Padre', encontraremos un selector que nos permitirá seleccionar el hijo del que queremos obtener la información.



Figura 83 - Menú para Padres

De esta manera, cuando acceda a las funcionalidades de 'Perfil', 'Incidencias' o 'Ver horario semanal', accederá concretamente a la del hijo que haya seleccionado. Para ello, cuando acceda, las rutas de navegación incluirán los identificadores de padre y de hijo.

## 6.2.23 Incidencias (justificar)

Desde esta pantalla los padres conocerán las incidencias de sus hijos y podrán añadir una justificación.

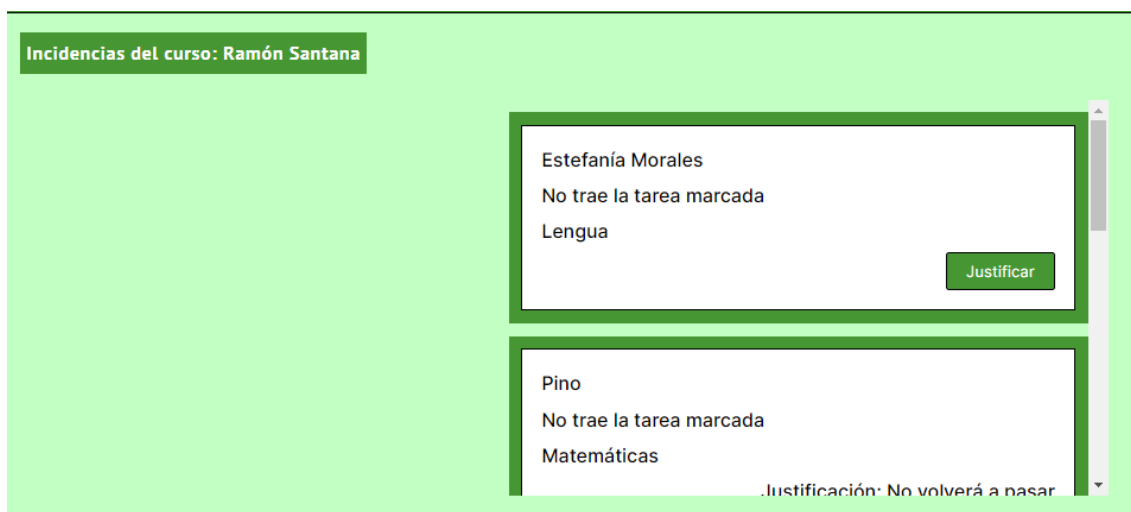


Figura 84 - Incidencias desde Perfil de Padre

En la Figura 85, podremos observar de qué manera un padre puede añadir una justificación a la falta.

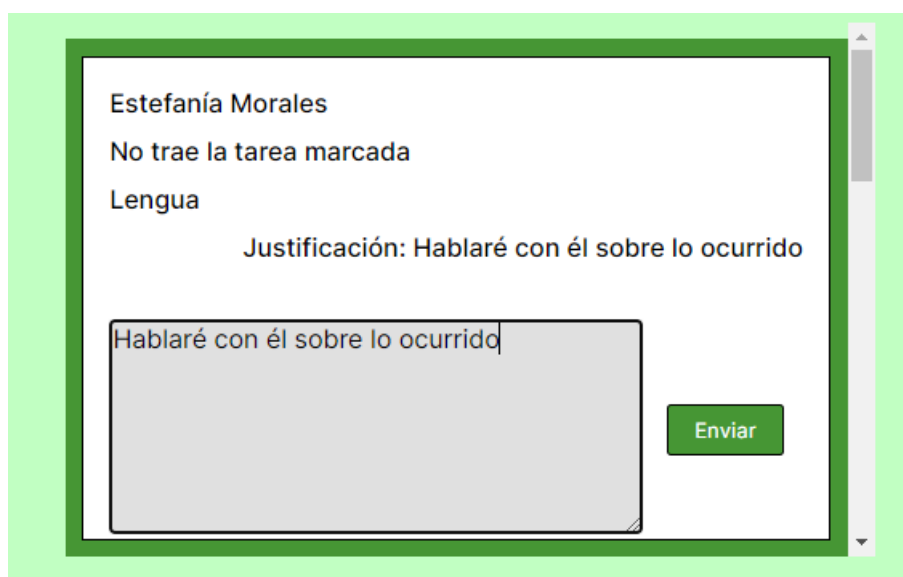


Figura 85 - Justificación de Faltas

## 6.2.24 Menú Profesores

Este es el último menú disponible en la plataforma, para aquellos perfiles de 'Profesor'. La diferencia es que los profesores tienen acceso a la lista de cursos que tiene asignados, los cuales los llevarán a la pantalla 'Cursos', definida en apartados anteriores. La ruta de navegación de cada uno de esos cursos incluirá tanto el identificador del profesor como el identificador del curso y del grupo.

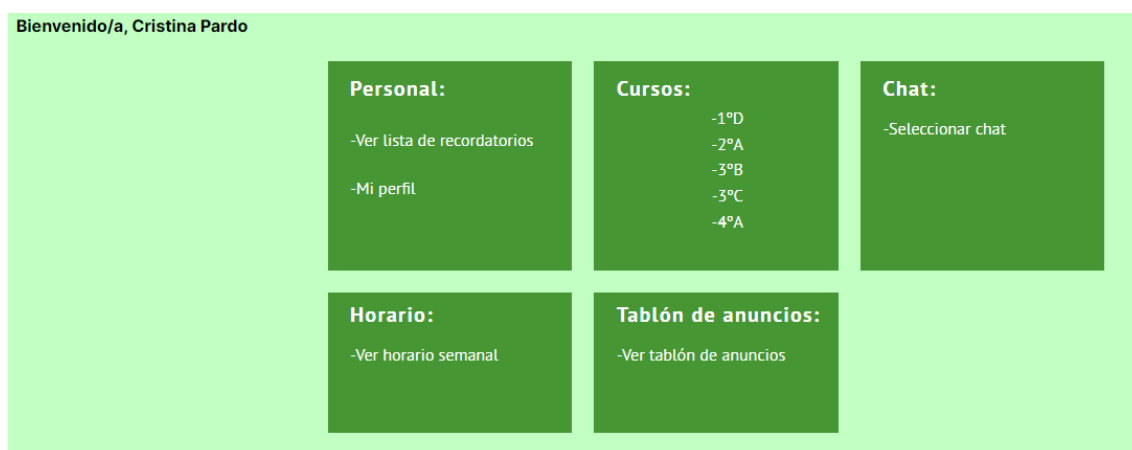


Figura 86 - Menú de Profesores

## Capítulo 7. Conclusiones y líneas futuras

EN ESTA SECCIÓN, ANALIZAREMOS LOS RESULTADOS OBTENIDOS CON LA IMPLEMENTACIÓN DE LA APLICACIÓN, EVALUANDO SU RENDIMIENTO Y FUNCIONALIDAD. ADEMÁS, EXPLORAREMOS POSIBLES MEJORAS Y LÍNEAS FUTURAS DE DESARROLLO PARA SEGUIR OPTIMIZANDO Y EXPANDIENDO SUS CAPACIDADES.

## 7.1 Contexto de la Conclusión

Metas como la introducción integral de la tecnología en centros educativos, la facilitación de la comunicación entre las entidades que componen un instituto, y la definición de una herramienta que cubra las necesidades emergentes durante el período de enseñanza eran los objetivos que se pretendían lograr.

Es por eso que el proyecto comenzó con la definición y estructuración de objetivos sólidos. Esto no solo nos permitió organizar los recursos de manera ordenada, sino también identificar el momento adecuado y la tarea específica a abordar en cada fase del proyecto.

Según se han seguido los planes y fases de diseño e implementación, el resultado de los objetivos definidos al comienzo del proyecto es el siguiente:

- **O1: Analizar las ventajas que podría aportar la aplicación.** Hemos visto que realmente la aplicación cubre muchas necesidades a la hora de gestionar y de comunicarse en un centro de educación secundaria. Todo ello ha ayudado a definir los problemas que pretendemos solucionar con nuestro proyecto.
- **O2: Analizar la oferta actual del mercado.** Esto nos ayudó a tener unas referencias contundentes y a definir una serie de funcionalidades mínimas con las que debía contar la plataforma web a la hora de contribuir a la sociedad.
- **O3: Establece las fases que requiere el desarrollo.** Se trató de un objetivo fundamental a la hora de proceder con el proyecto debido a qué en todo momento se conocía la tarea que había que abordar. Además, supuso una optimización del tiempo de desarrollo.
- **O4: Establecer y diseñar las funcionalidades que se van a implementar.** Supuso una ayuda similar a la del objetivo anterior, pero fundamentándonos en aspectos más técnicos. Realizar un diseño de la aplicación prepara el camino que debemos seguir a través de la implementación para lograr nuestro propósito.



- **O5: Definir la estructura de datos.** Gracias al diseño implementado, pudimos definir qué datos íbamos a necesitar manejar en cada momento. Por ello, se conocían los pasos que se debían seguir para preparar el proyecto y que pudiera recibir los datos a través de entidades, implementar los servicios a través de los cuales tendría lugar el intercambio de datos entre la base de datos y nuestro proyecto y cómo manejar esa estructura de manera que las vistas tuvieran un fácil manejo y acceso a los datos.
- **O6: Implementar la interfaz de usuario.** En este objetivo es justo destacar la facilidad que ofrecen las herramientas que hemos decidido utilizar, ya que gracias a ellas partiendo del propio diseño hemos sido capaces de obtener en su mayoría las vistas que conforman la interfaz de usuario. De lo contrario no solo tendríamos que haber diseñado la interfaz, sino que tendríamos que haber dedicado mucho más tiempo a su implementación y podría haber resultado en un decremento de las funcionalidades añadidas.
- **O7: Cargar los datos reales en la aplicación.** De nuevo, cada vez que se iba implementando alguna funcionalidad, la herramienta que nos proporciona los datos, en este caso *Firestore*, aportaba muchas facilidades a la hora de probar la funcionalidad y de manejar los datos.

## 7.2 Conclusiones

El Trabajo De Fin De Título propuesto pretendía desarrollar una aplicación web destinada a mejorar las tareas de gestión, administración y comunicación en los centros educativos de enseñanza obligatoria.

Se proponía un espacio web integral en que cada entidad de un instituto tuviera su propio espacio y donde cada perfil jugara un papel fundamental, pues todos los integrantes de un centro educativo son fundamentales a la hora de realizar las tareas propias de educación.

En el apartado anterior, se echaba la vista atrás a la evolución de los objetivos propuestos en primera instancia. Si bien los objetivos están cumplidos en su gran mayoría, cabe mencionar que por motivos de tiempo y de dificultades presentadas en la implementación, no se pudo concluir la funcionalidad que se había propuesto en relación con los boletines de notas.

Por otro lado, se debe mencionar que las herramientas elegidas han sido las verdaderas determinantes para que el desarrollo propuesto haya salido adelante. Mencionar que algunas de ellas fueron recomendadas por el tutor, lo cual implica que al no conocerlas anteriormente se haya tenido que destinar tiempo a estudiarlas para sacar el máximo partido posible de ellas.

En cuanto a la interfaz de usuario, la herramienta *TeleportHQ*, supuso, una vez superada su curva de aprendizaje, un adelanto, una optimización del tiempo y una facilidad en su manejo sin olvidar que nos permitió descargar el esqueleto del proyecto completamente desarrollado en cuanto a vistas y a estilos se refiere, ya preparado para integrarlo en el editor *Visual Studio Code* a través del *framework Angular*. Dicho *framework*, gracias a su composición nos permite aplicar el patrón de diseño *model-view-viewmodel*, de manera que podamos mantener el código limpio y estructurado en todo momento.

Por su parte, *Firebase* y su producto *Cloud Firestore*, completó nuestro proyecto gracias a su *backend* sólido y robusto y un sistema de almacenamiento en tiempo real que, además, ofrece una consola de fácil acceso y manejo a través de la cual comprobar y editar los datos que necesitemos.

Por tanto, se concluye que este desarrollo es no solo útil, sino también altamente escalable. Puede desempeñar un papel crucial en la introducción temprana de la tecnología en la educación, enseñando a los estudiantes a hacer un uso responsable de estas herramientas. Este proyecto demuestra que una aplicación bien definida y ejecutada puede mejorar significativamente las tareas de gestión, administración y comunicación en los centros educativos, proporcionando una solución integral y eficaz para las necesidades actuales y futuras del entorno educativo.

## 7.3 Líneas futuras

Es natural que a medida que avanzaba el desarrollo de este proyecto surgieran nuevas ideas que, bien sea por falta de tiempo o bien por falta de conocimientos técnicos en el momento de la implementación no se han podido llevar a cabo. Las más significativas se recogen a continuación:

- **Boletines de notas:** si bien como se comentó en el apartado anterior no fue posible su implementación en este desarrollo, se mantiene como línea futura inmediata. Sin embargo, no podría sustituir la tradicional recogida de notas, ya que esta debe estar acompañada de una firma del tutor legal del alumno, podría servir de sistema de almacenamiento de calificaciones.
- **Estructura de datos:** gracias a la herramienta *TeleportHQ*, se nos facilita en gran medida la implementación de componentes y páginas. Esto puede albergar resultados tanto positivos como negativos, debido a que en el caso de abordar un proyecto de estas dimensiones sin una base sólida y experimentada puede hacer pasar por alto la posibilidad de reutilizar código. Por todo esto, una posible línea futura sería basar la aplicación en módulos y usuarios. Donde todos los paneles que hemos visto como “Configuración”, “Horarios”, “Chat”, entre otros, podrían ser vistos como módulos, y cada uno de sus enlaces podría ser submódulos. De esta manera un usuario podría tener acceso a unos módulos u otros en función de su rol, lo cual resultaría en eliminar la necesidad de implementar seis pantallas diferentes para el menú principal.
- **Campus virtual:** otra posibilidad de escalar la aplicación sería la de añadir un sistema de subida de archivos de manera que resulte en un campus virtual para los alumnos, según sus asignaturas.
- **Sistemas de videoconferencia:** para reforzar la comunicación, podría darse un paso más allá del chat actual desarrollado e implementar la opción de tener una llamada o una vídeo llamada vía una plataforma web entre usuarios.

- **Mejoras de seguridad:** al ser una aplicación que maneja una gran cantidad de datos sensibles, toda seguridad es poca. Por lo que una importante línea futura sería la de aplicar mayores y más complejos niveles de seguridad.
- **Segregar las tareas:** podría cambiar el enfoque del sistema de matriculación y en vez de ser el administrador el que prepare todos los datos, se podría añadir más robustez a los formularios y permitir que sean los propios alumnos los que soliciten esa matriculación y, entonces, desde una entidad interna del centro aceptarlo o rechazarlo.

# Presupuesto

A CONTINUACIÓN, SE DETALLARÁ EL CÁLCULO DE LOS GASTOS ASOCIADOS AL DESARROLLO DE ESTE TRABAJO DE FIN DE TÍTULO.

## Desglose del presupuesto

El presupuesto se puede dividir en los siguientes apartados:

- Recursos materiales.
- Trabajo tarifado por tiempo empleado.
- Costes de redacción de la documentación.
- Derechos del visado del COITT.
- Gastos de tramitación y envío.
- Aplicación de impuestos

## Recursos materiales

Para la realización de este proyecto, se ha hecho uso de elementos físicos o hardware como de programas o software, que podrían tener un costo asociado en función de su versión. Los recursos hardware se resumen en ordenador portátil desde el que poder llevar a cabo la implementación y sus pruebas y la conexión a Internet. Mientras que los recursos software en este caso, no han supuesto un coste debido a que hemos hecho uso de versiones gratuitas.

Para llevar a cabo el cálculo del coste de amortización se deberá tener en cuenta por tanto el valor de estimación de adquisición, esto es, el precio de compra del portátil. Posteriormente, se hará una ponderación de su valor residual, que hace referencia al valor estimado de un activo al final de su vida útil, teniendo en cuenta factores como desgaste o depreciación. Además, debe tenerse en cuenta el tiempo estimado de vida útil. El cálculo del coste de amortización se representa por tanto como se ve en la Ecuación 1.

$$\textit{Coste de amortización} = \frac{\textit{Valor de adquisición} - \textit{Valor residual}}{\textit{Número de años de vida útil}}$$

*Ecuación 1 - Coste amortización*

A continuación, se reflejan los costes de amortización derivados del recurso hardware empleado, teniendo en cuenta los medios materiales

- Ordenador Acer TravelMate p215-53 series.
- Conexión a Internet.

Hardware	Valor de adquisición (€)	Valor residual (€)	Tiempo de vida útil (mese)	Tiempo Empleado (meses)	Coste mensual (€)	Importe Final (€)
Ordenador Personal	776.93	383	42	3	18.49	55.47
Conexión a Internet				3	50	150
<b>Total</b>						<b>205.47</b>

Tabla 30 - Cálculo de Recursos Materiales

Por tanto, se establece como costo global de los recursos materiales del proyecto de **205,47€**.

## Trabajo tarificado por tiempo empleado

Este apartado hace referencia al cálculo del importe a percibir por horas trabajadas basado en un Ingeniero en Telecomunicaciones. Para ello no hemos guiado por lo establecido en las recomendaciones del Colegio Oficial de Ingenieros Técnicos de Telecomunicación (COITT), que establece el cálculo según:

$$H = C_t \cdot 74,88 \cdot H_n + 96.72 \cdot H_e$$

Ecuación 2 - Trabajo tarificado por tiempo empleado

Donde:

- $C_t$  indica un factor de corrección en función del número de horas trabajadas.
- $H_n$  indica las horas trabajadas dentro de una jornada laboral.

- He indica las horas trabajadas fuera de la jornada laboral.

De acuerdo con lo establecido en el proyecto docente de la asignatura de Trabajo de Fin de Título de la Universidad de Las Palmas de Gran Canaria, se obtiene un total de 12 créditos ECTS en la asignatura, lo que resulta en un total de 300 horas. Por tanto, según lo regulado por el COITT, el coeficiente  $C_t$  tendrá el siguiente valor.

Horas empleadas (Hn)	Factor de corrección (Ct)
180-360	0.6

*Tabla 31 - Factor de Corrección según el COITT*

Por lo tanto, el resultado de la ecuación viene dado por:

$$H = 0.6 \cdot 74.8 \cdot 300 + 96.7 \cdot 0 = 13.478.40\text{€}$$

La suma total de los honorarios por tiempo dedicado asciende por tanto a **13.478,40€**.

## Costes asociados con la redacción del documento

La cantidad asociada a los costes de la documentación del proyecto se calculan según la siguiente ecuación:

$$R = 0.07 \cdot P \cdot Cn$$

*Ecuación 3 - Costes de Redacción*

Donde:

- P indica el total del presupuesto estimado hasta ahora.
- Cn es el coeficiente de ponderación en función del presupuesto.



Hasta ahora el presupuesto asciende a 13.683.47€. Según el COITT para aquellos presupuestos inferiores a 30.050€, el coeficiente de ponderación  $C_n$  se fija en 1. De esta forma, los costes de redacción quedarían en:

$$R = 0.07 \cdot 13.683.47 \cdot 1 = 957.84 \text{ €}$$

Por tanto, los costes de redacción ascienden a **957.84€**.

## Derechos de visado COITT

A continuación, debemos calcular el costo del visado del COITT en base a la siguiente ecuación:

$$V = 0.006 \cdot P \cdot C_v$$

*Ecuación 4 - Derechos de Visado COITT*

Donde:

- P es el presupuesto actual del proyecto.
- $C_v$  es el coeficiente reductor en función del presupuesto del trabajo.

El presupuesto total, tiene un coste de 14.641.31€. El coeficiente reductor de nuevo es 1. Por lo que los costes del visado quedan:

$$V = 0.006 \cdot 14.641.31 \cdot 1 = 87.84 \text{ €}$$

De la ecuación anterior se extrae que los costes por derechos de visados del COITT resultan en 87.33€.

## Gastos de tramitación y envío

Los gastos correspondientes a tramitación y envío han resultado en una cantidad de 6.01€.

## Aplicación de impuestos

En resumen, el gasto global de Trabajo de Fin de Título asciende a 14.734,76€. Teniendo en cuenta, el régimen fiscal en vigencia en Canarias, incorporaremos el Impuesto General Indirecto Canario o IGIC del 7%. Por tanto, el resultado final establece que:

<b>Recurso</b>	<b>Coste (€)</b>
Recursos materiales	205,47
Trabajo tarifado por tiempo empleado	13.478,40
Costes asociados a la redacción del documento	957,84
Derecho de visado del COITT	87,33
Gastos de tramitación y envío	6,01
Aplicación de impuestos (7%)	1.031,43
<b>Total</b>	<b>15.766,48</b>

*Tabla 32 - Coste Total*

## Bibliografía

- [1] UNIVERSIDAD ISABEL I, “Evolución de las tecnologías educativas desde su origen (08/2022)”. Accedido el 20 de abril de 2024. [En línea]. Disponible: <https://www.ui1.es/blog-ui1/evolucion-tecnologias-educativas>
- [2] UNESCO, “España | Tecnología (05/2023)”. Accedido el 20 de abril de 2024. [En línea]. Disponible: <https://education-profiles.org/es/europa-y-america-del-norte/espana/~tecnologia>
- [3] EDUCACIÓN 3.0, “Plataformas para la gestión de centros educativos”. Accedido el 20 de abril de 2024. [En línea]. Disponible: <https://www.educaciontrespuntocero.com/recursos/plataformas-gestion-escolar/>
- [4] LIVEKID, “La agenda digital perfecta para tu escuela infantil”. Accedido el 21 de abril de 2024. [En línea]. Disponible: <https://livekid.com/es/>
- [5] SM EDUCAMOS, “SM Educamos, el ecosistema educativo digital que acerca y conecta personas, soluciones de aprendizaje y de gestión educativa”. Accedido el 21 de abril de 2024. [En línea] Disponible: <https://www.grupo-sm.com/es/sm-educamos/>
- [6] ADDITIO APP, “Plataforma de gestión escolar para escuelas de secundaria”. Accedido el 21 de abril de 2024. [En línea] Disponible: <https://additioapp.com/plataforma-de-gestion-escolar-para-secundaria-y-ciclos/>
- [7] LOPD/RGPD, “¿Qué es el RGPD? | ¿Qué es el LOPD?”. Accedido el 21 de abril de 2024. [En línea]. Disponible: <https://solucionesqes.com/servicios/lopd-rgpd/#:~:text=La%20principal%20diferencia%20entre%20RGPD,y%20la%20finalidad%20de%20esta>
- [8] ANGULAR, “What is Angular?”. Accedido el 22 de abril de 2024. [En línea]. Disponible: <https://angular.dev/overview>
- [9] FIREBASE, “¿Qué es, para qué sirve?, funcionalidades y ventajas (05/2020)”. Accedido el 21 de abril de 2024. [En línea] Disponible: <https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/>
- [10] TELEPORTHQ, “Create. Code. Publish”. Accedido el 21 de abril de 2024. [En línea] Disponible: <https://teleporthq.io/>

- [11] VISUAL STUDIO CODE, “Code, Editing, Redefined.”. Accedido el 21 de abril de 2024. [En línea] Disponible: [https://code.visualstudio.com/?wt.mc\\_id=DX\\_841432](https://code.visualstudio.com/?wt.mc_id=DX_841432)
- [12] TYPESCRIPT, “¿Qué es TypeScript? (06/2018) “ Accedido el 21 de abril de 2024. [En línea]. Disponible: <https://codigofacilito.com/articulos/typescript>
- [13] DBDIAGRAM.IO, “Introduction DbDiagram Docs”. Accedido el 21 de abril de 2024. [En línea]. Disponible: <https://dbdiagram.io/docs/>
- [14] UNIVERSIDAD FRANCISCO DE VITORIA, “Bases de datos no relacionales”. Accedido el 26 de abril de 2024. [En línea] Disponible: <https://www.ufv.es/cetys/blog/bases-de-datos-no-relacionales/>
- [15] MEDIUM, “Conectar Firebase a proyecto de Angular (06/2023)”. Accedido el 22 de abril de 2024. [En línea]. Disponible: <https://medium.com/@dgongoragamboa/conectar-firebase-a-proyecto-de-angular-feb86ff61a5a>
- [16] FIRESTORE, “Primeros pasos con Cloud Firestore”. Accedido el 23 de abril de 2024. [En línea] Disponible: <https://firebase.google.com/docs/firestore/quickstart?hl=es-419>
- [17] MVVM, “Modelo-Vista-Modelo de vista”. Accedido el 26 de abril de 2024 [En línea] Disponible: <https://learn.microsoft.com/es-es/dotnet/architecture/maui/mvvm>
- [18] MVVM, “¿Qué es el patrón de arquitectura MVVM?”. Accedido el 26 de abril de 2024. [En línea] Disponible: <https://keepcoding.io/blog/que-es-el-patron-de-arquitectura-mvvm/>
- [19] BEZKODER, “Angular 16 Firebase CRUD example (12/2023)”. Accedido el 27 de abril de 2024. [En línea] Disponible: <https://www.bezkoder.com/angular-16-firebase-crud/>
- [20] CRUD ANGULAR, “CRUD con Angular y Firebase (Firestore) paso a paso (2022)”. Accedido el 28 de abril de 2024. [Vídeo en línea] Disponible: [https://www.youtube.com/watch?v=t\\_YSrxj0wGY](https://www.youtube.com/watch?v=t_YSrxj0wGY)
- [21] FRONTEND, “What is Frontend Development?”. Accedido el 25 de abril de 2024. [En línea] Disponible: <https://roadmap.sh/frontend>
- [22] TYPESCRIPT, “Roadmap”. Accedido el 25 de abril de 2024. [En línea] Disponible: <https://roadmap.sh/typescript>
- [23] ANGULAR, “Roadmap”. Accedido el 25 de abril de 2024. [En línea]. Disponible: <https://roadmap.sh/angular>