



UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍA INFORMÁTICA DE SEGOVIA

Grado en Ingeniería Informática de Servicios y Aplicaciones

GOMI: Gestión Online de Matrículas para un Instituto

Autor: Carlos Rascón Herrero

Tutor: José Vicente Álvarez Bravo

AGRADECIMIENTOS

En primer lugar, a mi familia, especialmente a mis padres y a mi prima Marta, con quién convivo, puesto que me han apoyado, ayudado y animado durante toda la realización de este proyecto.

En segundo lugar, a todos los profesores de la titulación de Ingeniería informática que me han enseñado e inculcado tanto conocimientos sobre la carrera como valores útiles para toda la vida.

Y por supuesto, dedico un agradecimiento especial a José Vicente Álvarez Bravo, mi tutor durante el desarrollo de este proyecto, que siempre ha estado dispuesto a ayudarme y asesorarme en cualquier duda que tuviese.

Por otro lado, quiero agradecer el apoyo a mis compañeros de la universidad, en especial a Rodrigo Álvaro, Yaser Eldabete y Diego Borja, puesto que me han ayudado mucho, tanto con dudas técnicas que tenían que ver con la implementación de la aplicación, como con trámites o formalidades previas a la defensa.

Por último, quiero agradecer el apoyo del resto de mis amigos porque siempre han estado a mi lado, animándome y apoyándome en los mejores y peores momentos por los que he pasado durante la realización de este proyecto.

RESUMEN

El propósito principal de este Trabajo de Fin de Grado consiste en la implementación de una aplicación web que permite a los usuarios matricularse en un centro de enseñanza sin necesidad de realizar gestiones de manera física.

La idea principal es realizar una herramienta sencilla y que se pueda utilizar desde cualquier dispositivo con conexión a internet, para facilitar todo el proceso a los alumnos que quieran matricularse.

Además, se ha desarrollado una parte de administrador, que ayudará a los gestores del centro a tener una visión global sobre qué y cuántas matrículas hay para cada curso u optativa.

En definitiva, se pretende ofrecer una gestión global de los alumnos matriculados en un centro de enseñanza, tanto para usuarios consumidores de la aplicación como para usuarios gestores de la misma.

PALABRAS CLAVE

Aplicación web, matrícula, Angular, Node Js, Cloud firestore, Javascript, Typescript.

Índice general

Índice de Tablas.....	8
Índice de figuras	10
1. Introducción	12
1.1. Motivación	13
1.2. Objetivos	13
1.3. Alcance	13
1.4. Organización del documento	14
2. Estado del arte	16
2.1. Introducción	17
2.2. Matriculasescolares.com.....	17
2.3. PDF rellenable	17
2.4. Comparativa de propuestas	18
3. Herramientas utilizadas	20
3.1. Herramientas para el desarrollo	21
3.2. Herramientas de soporte	21
3.3. Tecnologías empleadas	21
4. Planificación del proyecto	24
4.1. Metodología de trabajo	25
4.2. Fases de trabajo y planificación temporal	27
4.3. Costes del proyecto.....	28
4.3.1. Costes de personal	28
4.3.2. Costes Hardware	29
4.3.3. Costes software.....	29
4.3.4. Costes Añadidos	29
4.4. Estimaciones Iniciales.....	29
4.4.1. Estimación por puntos de función (Método Albrecht)	29
4.5. Presupuestos.....	32
4.5.1. Costes hardware y software.....	32
4.5.2. Costes Recursos humanos.....	33
4.5.3. Costes añadidos.....	34
4.5.4. Presupuesto final.....	35
5. Análisis del sistema	38
5.1. Tipos de usuario del sistema	39
5.2. Requisitos de Usuario.....	39
5.2.1. Relativos a gestión de usuarios	39

5.2.2.	Relativos al proceso de matriculación	39
5.2.3.	Relativos a la gestión de datos de matrículas	39
5.3.	Casos de uso	40
5.3.1.	Relativos al usuario	40
5.3.2.	Relativos al alumno	42
5.3.3.	Relativos al administrador.....	49
5.4.	Reglas de Negocio	54
5.5.	Requisitos funcionales.....	55
5.6.	Requisitos No Funcionales	57
5.6.1.	Requisitos de Interfaz Externa	57
5.6.2.	Atributos de Calidad.....	57
5.7.	Requisitos de Información	58
6.	Diseño.....	60
6.1.	Arquitectura	61
6.1.1.	Arquitectura lógica.....	61
6.1.2.	Arquitectura física	62
6.2.	Modelos de Diseño.....	63
6.2.1.	Diagrama de secuencia	63
6.2.2.	Modelo Lógico de Datos.....	64
6.2.3.	Diccionario de datos.....	66
6.3.	Diseño de interfaces.....	72
7.	Implementación	78
7.1.	Consideraciones de implementación	79
7.2.	Servidores de bases de datos utilizados.....	79
7.3.	Organización interna del proyecto.....	79
8.	Pruebas.....	82
8.1.	Pruebas de Caja Blanca	83
8.2.	Pruebas de Caja Negra	83
9.	Conclusiones y posibles mejoras.....	88
9.1.	Conclusiones.....	89
9.2.	Posibles mejoras.....	89
10.	Webgrafía	91

Índice de Tablas

Tabla 1: Comparativa de herramientas del mercado.....	18
Tabla 2: Distribución temporal de las etapas.....	28
Tabla 3: Costes hardware.....	29
Tabla 4: Costes Software.....	29
Tabla 5: Costes añadidos.....	29
Tabla 6: Complejidad entradas de usuarios.....	30
Tabla 7: Complejidad salidas externas.....	30
Tabla 8: Complejidad consultas externas.....	30
Tabla 9: Complejidad ficheros externos.....	31
Tabla 10: Puntos de función sin ajustar.....	31
Tabla 11: Grado de complejidad de los factores.....	32
Tabla 12: Coste real Albrecht.....	32
Tabla 13: Coste real planificación inicial.....	33
Tabla 14: Coste real software.....	33
Tabla 15: Coste recursos humanos Albrecht.....	34
Tabla 16: Coste recursos humanos planificación inicial.....	34
Tabla 17: Costes añadidos Albrecht.....	34
Tabla 18: Costes añadidos planificación inicial.....	34
Tabla 19: Coste total duraciones estimadas.....	35
Tabla 20: Presupuesto final hardware.....	35
Tabla 21: Presupuesto final software.....	35
Tabla 22: Presupuesto final recursos humanos.....	36
Tabla 23: Presupuesto final conexiones y electricidad.....	36
Tabla 24: Caso de uso "Iniciar Sesión".....	41
Tabla 25: Caso de uso "Cerrar Sesión".....	42
Tabla 26: Caso de uso "Registrarse".....	43
Tabla 27: Caso de uso "Matricularse".....	44
Tabla 28: Caso de uso "Seleccionar asignaturas de la matrícula".....	45
Tabla 29: Caso de uso "Imprimir justificante de la matrícula".....	46
Tabla 30: Caso de uso "Visualizar matrícula".....	47
Tabla 31: Caso de uso "Modificar matrícula".....	48
Tabla 32: Caso de uso "Visualizar datos relativos a las matrículas de los alumnos".....	50
Tabla 33: Caso de uso "Visualizar estadísticas relativas a los alumnos matriculados en cada asignatura".....	51
Tabla 34: Caso de uso "Visualizar estadísticas relativas a los alumnos matriculados en cada curso".....	52
Tabla 35: Caso de uso "Visualizar estadísticas relativas a los alumnos que repiten curso".....	53
Tabla 36: Caso de uso "Generar grupos por curso".....	54
Tabla 37: Requisitos funcionales.....	57
Tabla 38: Colecciones de datos.....	66
Tabla 39: Diccionario Datos-Colección "Usuarios".....	67
Tabla 40: Diccionario Datos-Colección "Datos personales".....	67
Tabla 41: Diccionario Datos-Colección "Datos parentales".....	69
Tabla 42: Diccionario Datos-Colección "Datos de dirección".....	69
Tabla 43: Diccionario Datos-Colección "Datos de matrícula".....	70
Tabla 44: Diccionario Datos-Colección "Primer curso".....	70

Tabla 45: Diccionario Datos-Colección "Segundo curso"	70
Tabla 46: Diccionario Datos-Colección "Cuarto curso"	71
Tabla 47: Diccionario Datos-Colección "Cuarto curso"	71
Tabla 48: Diccionario Datos-Colección "Quinto curso"	71
Tabla 49: Diccionario Datos-Colección "Sexto curso"	72
Tabla 50: Interfaz "Pantalla principal"	72
Tabla 51: Interfaz "Pantalla de login"	73
Tabla 52: Interfaz "Pantalla de registro"	73
Tabla 53: Interfaz "Pantalla de logout"	74
Tabla 54: Interfaz "Pantalla del formulario personal"	74
Tabla 55: Interfaz "Pantalla del formulario de datos de tutores legales"	75
Tabla 56: Interfaz "Pantalla del formulario de datos de dirección"	75
Tabla 57: Interfaz "Pantalla del formulario de datos de matrícula"	76
Tabla 58: Interfaz "Página de datos de optativas"	76
Tabla 59: Interfaz "Página de generación de grupos"	77
Tabla 60: Prueba Caja Negra "Inicio de sesión"	83
Tabla 61: Prueba Caja Negra "Registro"	83
Tabla 62: Prueba Caja Negra "Filtrado de matrículas por curso"	84
Tabla 63: Prueba Caja Negra "Filtrado de matrículas por optativa de un determinado curso" ..	84
Tabla 64: Prueba Caja Negra "Filtrado de matrículas que repiten curso"	84
Tabla 65: Prueba Caja Negra "Generación de grupos por curso"	84
Tabla 66: Prueba Caja Negra "Formulario de datos personales"	85
Tabla 67: Prueba Caja Negra "Formulario de datos parentales"	85
Tabla 68: Prueba Caja Negra "Formulario de datos de dirección"	85
Tabla 69: Prueba Caja Negra "Formulario de matrícula"	85
Tabla 70: Prueba Caja Negra "El formulario de registro no acepta un email incorrecto"	86
Tabla 71: Prueba Caja Negra "El usuario no puede acceder con claves incorrectas"	86
Tabla 72: Prueba Caja Negra "El sistema no puede avanzar en el registro sin todos los datos rellenos"	86

Índice de figuras

Figura 1: Árbol de características	14
Figura 2: Metodología de desarrollo iterativo e incremental. Jeff Patton (oil and bits).....	25
Figura 3: Usuarios del sistema	39
Figura 4: Casos de uso del usuario	40
Figura 5: Casos de uso del alumno	42
Figura 6: Casos de uso del administrador	49
Figura 7: Arquitectura lógica	61
Figura 8: Arquitectura física	62
Figura 9: Diagrama de secuencia "Introducir/Modificar datos personales"	63
Figura 10: Diagrama de secuencia "Login"	64
Figura 11: Diagrama de secuencia "Obtener datos por curso"	64
Figura 12: Estructura de base de datos "Cloud firestore"	65
Figura 13: Equivalencia de modelo de datos en Typescript.....	66
Figura 14: Estructura de documentos en base de datos.....	66

Capítulo 1.

Introducción

1. Introducción

En este apartado se va a realizar la introducción del proyecto para poner en contexto sobre qué y por qué se ha realizado este proyecto, cuál es su motivación, sus objetivos, su alcance y, por último, se explica la organización que se va a seguir a lo largo de este documento.

Los subapartados de los que consta esta sección son los siguientes:

- **Motivación:** en el que se explican los motivos que han llevado a la realización del proyecto.
- **Objetivos:** en el que se desarrollan los objetivos iniciales.
- **Alcance:** en el que se explica el alcance del proyecto, y se incluye el árbol de características.
- **Organización del documento:** en el que se enumeran los distintos apartados de los que va a constar el documento.

1.1. Motivación

El mundo cada vez está más digitalizado en cualquier aspecto de la sociedad, y los institutos de enseñanza secundaria no deberían ser menos.

Este proyecto surge como petición de un conocido del autor para que se creara una aplicación que evitara a los alumnos desplazarse al centro para realizar su matrícula en varias ocasiones, puesto que actualmente, tienen que ir hasta dos veces, una a recoger el formulario de matrícula y otra a entregar el mismo firmado y rellenado.

Cuando se planteó este proyecto se pensó que sería suficiente con crear un formulario que generara un *PDF* para que el alumno se ahorrara desplazarse al centro varias veces, pero tras darle algunas vueltas se decidió ir más allá y almacenar la matrícula y todos los datos personales para que la presencialidad en el centro fuera prescindible, tan solo se necesitaría una confirmación por parte del alumno, que podría ser por vía telefónica o, incluso, por email.

Además, se pensó que podría ser útil a la hora de generar horarios (descartado de este proyecto por la magnitud de la problemática), grupos y distribuir las horas o clases que el centro desee. A colación de esto, viene la parte de administrador, en la que se intenta ayudar al centro lo más posible.

Por supuesto, el proyecto queda abierto a posteriores mejoras requeridas por el centro petionario.

1.2. Objetivos

El objetivo principal es digitalizar el proceso de matriculación para, en este caso, un centro en concreto, pero el proyecto se ha enfocado para poder ser implantado en cualquier centro de enseñanza, tanto primaria como secundaria, con unos simples retoques de configuración y modificar las asignaturas, cursos y demás.

Otro de los objetivos del proyecto es ayudar a los administradores del centro de enseñanza a tener una visión global de las elecciones de los alumnos, tanto de las distintas optativas de cada curso como de los alumnos repetidores.

Además, se ha intentado ayudar al centro a dividir a los alumnos en grupos por curso, procurando juntar aquellos que cursen optativas similares.

Por último, otro objetivo de la aplicación es facilitar los procesos de matriculación, ofreciendo oportunidades tales como revisar o modificar las matrículas después de haberlas rellenado o generar un *PDF* con su matrícula rellena siempre que se quiera.

1.3. Alcance

Este proyecto, en principio, va destinado a un centro de enseñanza en concreto que me hizo la petición, pero, como se ha dicho antes, el proyecto podría ser utilizado e implantado en cualquier centro de enseñanza modificando unos cuantos archivos de configuración para ajustarlo a las necesidades.

Este proyecto está pensado para cualquier usuario que disponga de un dispositivo con conexión a Internet y un navegador para poder acceder a la aplicación.

En un primer momento, la aplicación estará configurada para ofrecer las optativas del curso 2021-2022 para el centro petionario de esta aplicación, pero existe la posibilidad de, si el

resultado de la implantación del proyecto es el deseado, actualizarlo de manera que quede permanente para los futuros cursos escolares.

Las funcionalidades principales de la aplicación serán:

- **Gestión de usuarios:** Se encarga de todo lo relativo a la gestión de usuarios, tanto la creación de una nueva cuenta, como el inicio y cierre de sesión.
- **Gestión de alumnos matriculados:** Se encarga de ofrecer datos a los usuarios administradores de las matrículas generadas. Así mismo, ofrece estadísticas sobre los alumnos matriculados por curso, las optativas elegidas para un determinado curso, los alumnos repetidores de cada curso y ofrece la posibilidad de generar los grupos de un determinado nivel, intentando realizar agrupaciones por elección de optativas similares.
- **Gestión de matrícula:** Se encarga de todo lo relativo al proceso de matriculación de un alumno. Comprende desde los datos personales hasta la elección de curso y asignaturas, pasando por los datos de los tutores y la dirección del alumno. Además, tiene una opción para importar la matrícula desde un archivo CSV con la intención de prescindir de pasos innecesarios.

A continuación, se presenta el árbol de características de la aplicación:

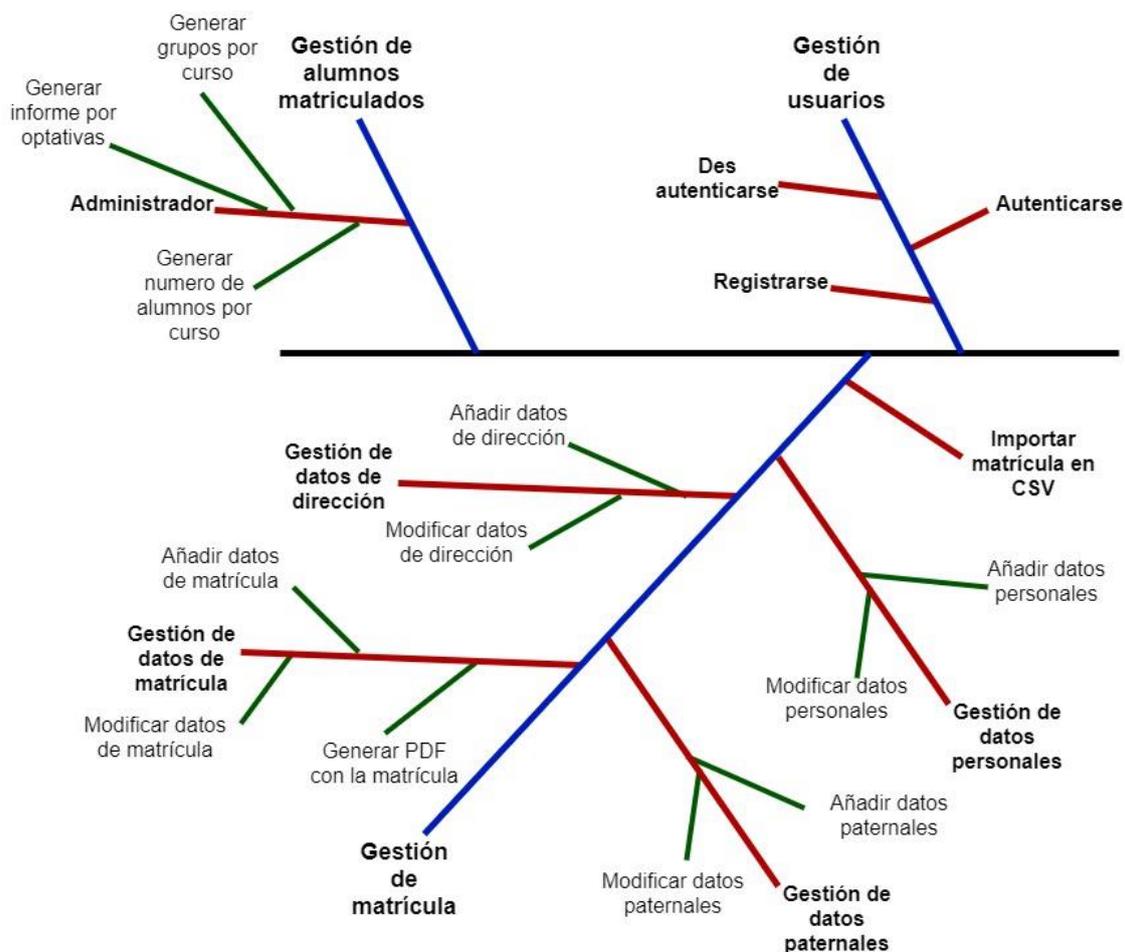


Figura 1: Árbol de características

1.4. Organización del documento

En este apartado se describe la estructura que se va a seguir durante todo este documento. Consta de 10 secciones principales:

1. **Introducción:** se describen conceptos generales del proyecto, así como los objetivos, motivación, alcance y características principales.
2. **Estado del arte:** se contemplan otras herramientas similares del mercado, para observar las diferencias con nuestro proyecto.
3. **Herramientas utilizadas:** se enumeran las herramientas y tecnologías utilizadas para el desarrollo de la aplicación.
4. **Planificación del proyecto:** se refleja la metodología empleada, la planificación temporal y el presupuesto planificado para este proyecto.
5. **Análisis:** se describen los requisitos que debe cumplir la aplicación, así como los actores y casos de uso de esta.
6. **Diseño:** se describe la arquitectura, tanto lógica como física de la aplicación junto con el diseño de la interfaz.
7. **Implementación:** se explican cuestiones relativas al desarrollo de la aplicación, centrándonos en algunos conceptos técnicos que son importantes dentro del proyecto.
8. **Pruebas:** se muestran las pruebas de caja blanca y negra realizadas para garantizar el correcto funcionamiento de la aplicación.
9. **Manuales:** Esta sección ha sido omitida, puesto que nuestra aplicación es de tipo web, por lo que no implica instalación alguna.
10. **Conclusiones:** se refleja la opinión final del autor de este proyecto, así como posibles mejoras para llevar a cabo en un futuro sobre este proyecto base.
11. **Bibliografía:** se citan las fuentes usadas tanto para llevar a cabo la aplicación como la documentación.

Capítulo 2.

Estado del Arte

2. Estado del arte

Este apartado incluye una comparativa de mercado entre nuestra aplicación, y otras aplicaciones similares que existen en internet, comparando todas las funcionalidades que ofrece.

Los subapartados de los que consta este apartado son los siguientes:

- **Introducción:** se introduce el tema que vamos a tratar en la sección.
- **Matriculasescolares.com:** se explica cómo funciona la primera de las dos aplicaciones con las que se ha hecho la comparativa.
- **PDF rellenable:** se explica el segundo ejemplo que se ha tomado para realizar esta comparación.
- **Comparativa de propuestas:** la comparativa real, funcionalidad a funcionalidad entre nuestra aplicación y las dos que se han tomado.

2.1. Introducción

En el caso de nuestro proyecto, el mercado ofrece infinidad de aplicaciones para realizar la labor que pretende desempeñar nuestra aplicación. Muchos centros educativos, ya sean de educación primaria, secundaria o universitaria tiene una aplicación similar a la que se quiere desarrollar en este proyecto.

El problema que surge al llevar a cabo la investigación previa a la escritura de esta sección es que para poder acceder a las aplicaciones que permiten matricularte en algún centro educativo es necesario contar con un perfil de usuario en ese centro, lo que hace imposible la observación y estudio de muchas de ellas.

Debido al problema anteriormente citado se han seleccionado dos aplicaciones web que permiten acceder al alta de matrícula sin necesidad de tener cuenta de usuario creada en el centro.

La primera de ellas es la plataforma “[Matriculasescolares.com](https://matriculasescolares.com)”, dominio que contiene las páginas web de alta de matrícula de muchos institutos de la comunidad de Madrid.

Otra de las alternativas que he encontrado son los archivos *PDF* rellenables que ofrecen otros institutos para matricularse.

2.2. [Matriculasescolares.com](https://matriculasescolares.com)

Es un dominio que contiene las páginas web para la matriculación en centros de enseñanza secundaria de la comunidad de Madrid. Para observar las características de dicha aplicación web nos vamos a centrar en la página web del instituto IES Santamarca (<https://iessantamarca.matriculasescolares.com>).

Los cambios que se observan entre los distintos centros que poseen una aplicación web con este dominio son el logo del instituto, las distintas optativas de cada centro y la sección de autorizaciones, en la que cada centro dispone de distintas secciones.

En esta aplicación, podemos acceder al formulario de matrícula sin estar logado en la aplicación, pero nos pide el usuario y la contraseña a la hora de guardar la matrícula, creando un usuario nuevo si no tienes cuenta.

La aplicación se divide en tres pantallas, una para rellenar los datos personales, de tutores legales y de dirección en una misma pantalla, los datos de matrícula en otra pantalla, y, por último, una serie de autorizaciones necesarias para el centro. Por último, tienes acceso a un documento *PDF* con la matrícula rellena.

Por otro lado, ofrece la posibilidad de iniciar sesión en la aplicación para modificar los datos de la matrícula previamente introducidos.

Como no tengo cuenta de administrador no puedo saber si existe esa parte dentro de esta aplicación o, si existe, de qué funcionalidades consta.

2.3. *PDF* rellenable

En esta opción, la aplicación ofrece un archivo *PDF* rellenable para que el alumno introduzca sus datos y seleccione las opciones de su matrícula.

Como ejemplo he seleccionado los *PDF* rellenables del instituto I.E.S “Ciudad los Ángeles” de Madrid (https://instituto.iescla.org/?page_id=1911). En la *url* anterior se ofrecen los archivos para rellenar la matrícula para cada uno de los cursos ofrecidos en dicho centro.

En estas matrículas se piden los datos personales, de tutores legales y de dirección del alumno, además de las asignaturas seleccionadas.

Tras rellenar este archivo, el alumno debe hacerlo llegar al centro de enseñanza, ya sea físicamente o enviándolo por medios telemáticos, ya sea la secretaria virtual o la página web de la comunidad de Madrid.

Respecto a la parte de administrador, creo que no existirá, puesto que los centros tienen que introducir manualmente los datos de los alumnos y sus matrículas.

2.4. Comparativa de propuestas

En este apartado se va a realizar una comparativa de características entre las distintas alternativas anteriormente citadas y este proyecto.

	GOMI	Matriculasescolares.com	PDF rellenable
Posibilidad de matriculación totalmente online	✓	✓	✗
Posibilidad de modificación de los datos introducidos	✓	✓	✗
Almacenamiento de los datos introducidos de manera automática	✓	✓	✗
Impresión de matrícula rellena	✓	✓	✓
Zona de administrador	✓	Sin información	✗
Necesidad de tener cuenta	✓	✓	✗

Tabla 1: Comparativa de herramientas del mercado

Capítulo 3.

Herramientas Utilizadas.

3. Herramientas utilizadas

Esta sección describe todas las herramientas, tanto propias del desarrollo a nivel de código, como de apoyo usadas en nuestra aplicación. Además, enumera y explica las tecnologías y lenguajes de programación usados.

Los subapartados de los que consta esta sección son los siguientes:

- **Herramientas para el desarrollo:** se enumeran los programas y tecnologías usadas para desarrollar el proyecto.
- **Herramientas de soporte:** se enumeran las herramientas que se han usado para realizar tanto la aplicación como la memoria, pero no son propias del desarrollo.
- **Tecnologías empleadas:** se explican los lenguajes de programación, *frameworks* y tecnologías usadas para el desarrollo de la aplicación web que se ha desarrollado.

3.1. Herramientas para el desarrollo

- **Visual Studio Code:** es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto.
- **Firebase:** es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles lanzada en 2011 y adquirida por Google en 2014. Está compuesta por varios servicios, para nuestro proyecto hemos usado el servicio “Cloud Firestore”, que sirve para suministrar a nuestra aplicación una base de datos en tiempo real con localización en la “nube”. Se trata de una base de datos “NoSQL” capaz de almacenar gran cantidad de información, y fácilmente integrable con el entorno de programación Android Studio. Además, goza de gran escalabilidad para poder soportar una demanda creciente de usuarios.

3.2. Herramientas de soporte

- **Microsoft Word:** es un programa informático orientado al procesamiento de textos. Fue creado por la empresa Microsoft, y viene integrado de manera predeterminada en el paquete ofimático denominado Microsoft Office.
- **Diagrams.net:** es una aplicación web gratuita y de código abierto que te permite crear una gran variedad de diagramas.
- **GitHub:** es una compañía sin ánimo de lucro que ofrece un servicio de hosting de repositorios almacenados en la nube. En nuestro proyecto se ha empleado para salvaguardar las distintas versiones desarrolladas de nuestro sistema.

3.3. Tecnologías empleadas

- **Node Js:** es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con entrada/salida de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. En nuestro proyecto se ha utilizado para desarrollar la parte de servidor, utilizando el framework **Express Js** sobre Node Js para desarrollar una *Web API*.
- **Angular:** es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. En nuestro proyecto se ha usado la versión 8 para desarrollar la parte de cliente de nuestra aplicación web.
- **Html:** hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. En nuestro caso se usa junto con *Angular* para generar las vistas.
- **Css:** es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. En nuestro proyecto se ha usado para modificar los estilos de las distintas vistas.
- **Json:** es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia

adopción como alternativa a XML, se considera un formato independiente del lenguaje. En nuestro proyecto se ha utilizado para almacenar configuración.

Capítulo 4.

Planificación del Proyecto.

4. Planificación del proyecto

En esta sección se describirá la metodología de trabajo usada para este proyecto, la organización de las fases de trabajo, y, por último, las estimaciones iniciales del presupuesto y los costes que ha conllevado este proyecto.

Los subapartados de los que consta esta sección son los siguientes:

- **Metodología de trabajo:** se explica la metodología de trabajo usada para el desarrollo del proyecto.
- **Fases de trabajo y planificación temporal:** se enumeran las fases en las que se ha dividido todo el trabajo que se ha realizado y se lleva a cabo una planificación temporal del proyecto al inicio de este.
- **Costes del proyecto:** se establecen los costes iniciales del proyecto.
 - **Costes de personal**
 - **Costes Hardware**
 - **Costes Software**
 - **Costes añadidos**
- **Estimaciones iniciales:** se explican las estimaciones que se han llevado a cabo al inicio del proyecto para compararlo con la planificación inicial.
 - **Estimación por puntos de función (Método Albrecht)**
- **Presupuestos:** se lleva a cabo la presupuestación del proyecto tanto usando las fechas que hemos obtenido en la planificación inicial como en la estimación por puntos de función.
 - **Costes Hardware y Software**
 - **Costes de personal**
 - **Costes añadidos**
 - **Presupuesto final**

4.1. Metodología de trabajo

En este proyecto se ha utilizado el modelo de “Desarrollo iterativo e incremental”, que consiste en desarrollar el proyecto dividiéndolo en pequeñas etapas repetitivas (iteraciones) que aporten nuevo valor al sistema y además realizar mejoras sobre las iteraciones anteriores en cada una de ellas. Lo que se busca es que en cada iteración los componentes logren evolucionar el producto dependiendo de los completados de las iteraciones antecesoras, agregando más opciones de requisitos y logrando así un mejoramiento mucho más completo. Para ello, cada requisito se debe completar en una única iteración: el equipo debe realizar todas las tareas necesarias para completarlo (incluyendo pruebas y documentación) y que esté preparado para ser entregado al cliente con el mínimo esfuerzo necesario.

Una manera adecuada para dirigir al proceso iterativo incremental es la de priorizar los objetivos y requerimientos en función del valor que se ofrece al cliente.

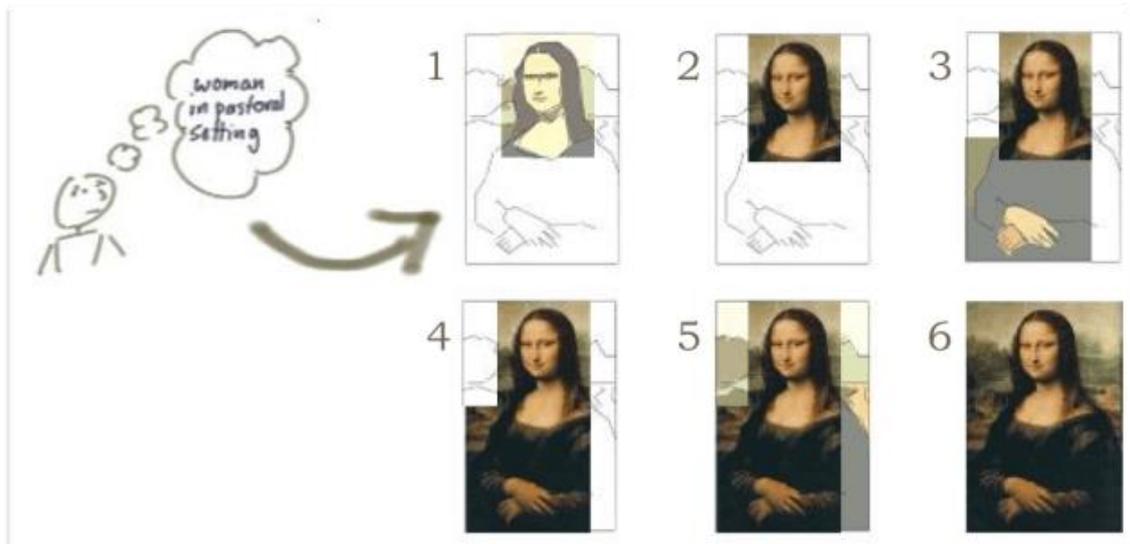


Figura 2: Metodología de desarrollo iterativo e incremental. Jeff Patton (oil and bits)

Los pasos claves en el proceso son comenzar con una implementación simple de los requerimientos del sistema, e iterativamente mejorar la secuencia evolutiva de versiones hasta que el sistema completo esté implementado. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema.

Básicamente este modelo se basa en dos premisas:

- Los usuarios nunca saben bien que es lo que necesitan para satisfacer sus necesidades.
- En el desarrollo, los procesos tienden a cambiar.

El proceso en sí mismo consiste en:

- Etapa de inicialización, en la que se crea un producto con el que el usuario pueda interactuar, y por ende retroalimentar el proceso. Debe ofrecer una muestra de los aspectos claves del problema y proveer una solución lo suficientemente simple para ser comprendida e implementada fácilmente.
- Etapa de iteración, que involucra el rediseño e implementación de algunas de las tareas proyecto, y el análisis de la versión más reciente del sistema. La meta del diseño e implementación de cualquier iteración es ser simple, directa y modular, para poder soportar el rediseño de la etapa o como una tarea añadida a la lista

de control de proyecto. El código puede, en ciertos casos, representar la mayor fuente de documentación del sistema.

Las principales ventajas de esta metodología de desarrollo son:

- En este modelo los usuarios no tienen que esperar hasta que el sistema completo se entregue para hacer uso de él. El primer incremento cumple los requerimientos más importantes de tal forma que pueden utilizar el software al instante.
- Los usuarios pueden utilizar los incrementos iniciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.
- Existe muy pocas probabilidades de riesgo en el sistema. Aunque se pueden encontrar problemas en algunos incrementos, lo normal es que el sistema se entregue sin inconvenientes al usuario.
- Ya que los sistemas de más alta prioridad se entregan primero, y los incrementos posteriores se integran entre ellos, es muy probable que los sistemas más importantes sean a los que se les hagan más pruebas. Esto quiere decir que es menos probable que los usuarios encuentren fallas de funcionamiento del software en las partes más importantes del sistema.
- En el desarrollo de este modelo se da la retroalimentación muy temprano a los usuarios.
- Permite separar la complejidad del proyecto, gracias a su desarrollo por parte de cada iteración o bloque.
- El producto es consistente y puntual en el desarrollo.
- Los productos desarrollados con este modelo tienen una menor probabilidad de fallar.
- Se obtiene un aprendizaje en cada iteración que es aplicado en el desarrollo del producto y aumenta las experiencias para próximos proyectos

Por todo esto, además de ser una metodología con la que ya tengo experiencia, considero que esta metodología de trabajo se ajusta muy bien al modelo de proyecto que voy a llevar a cabo.

Las iteraciones que se han llevado a cabo, puntualizando que, en cada una de aquellas que implican desarrollo mejoraban la anterior, son:

- Iteración para el análisis de requisitos
- Iteración para crear la base de datos y los proyectos iniciales
- Iteración para el desarrollo del login, registro y formularios iniciales
- Iteración para generar un pdf imprimible con la matrícula
- Iteración para almacenar y tratar las matrículas en base de datos
- Iteración para generar estadísticas relativas a los alumnos matriculados por curso y asignatura
- Iteración para generar los grupos de alumnos por curso
- Iteración de mejora de la seguridad de la aplicación
- Iteración de mejora de la interfaz de la aplicación
- Iteración para revisión final y documentación de la aplicación

Por último, hay que explicar que estas iteraciones se decidieron de este modo para que lo primero que se pudiera probar fuera el proceso de matrícula, después se pudiera rellenar la matrícula y, con el pdf generado, se pudiera llevar al centro de estudios y evitar ir dos veces (a por el sobre de matrícula y a entregarlo), y, por último, la parte de administrador y los estilos y seguridad, puesto que aún no está desplegada ni contiene datos personales.

4.2. Fases de trabajo y planificación temporal

Para la planificación de este proyecto cabe destacar algunas cuestiones previas que influyen en gran medida al desarrollo y, por tanto, a la planificación de este.

- La comunicación de inicio del proyecto se ubica en torno al mes de marzo de 2020, aunque, realmente, el inicio de este no se acomete hasta una fecha posterior.
- La planificación del proyecto no se inicia hasta enero de 2021, puesto que, por motivos personales, se decidió no llevar a cabo una planificación exhaustiva.
- Los días estipulados para trabajar son de lunes a viernes, salvo excepciones.
- El horario de trabajo del que se dispone es entre las 6 de la tarde y las 10 de la noche, por otros motivos laborales.
- La media de horas trabajadas por día se establece en 2 horas, por imposibilidad de aumentar dicha cifra por otros compromisos y, teniendo en cuenta que hay días entre semana que no hay posibilidad de dedicar tiempo al proyecto.
- Los días de vacaciones nacionales, y festivos autonómicos o provinciales fueron descartados de esta planificación, resultando una cifra cercana a 18 días aproximadamente.
- El desconocimiento, o falta de familiarización con algunas herramientas utilizadas en el proyecto implica un aumento del tiempo de planificación para algunas iteraciones del proyecto, puesto que conlleva aprendizaje y desarrollo más lento.
- Por último, las primeras iteraciones, sobre todo la tercera, conllevará un mayor tiempo de desarrollo, puesto que implica crear un producto utilizable y entregable, aunque con pocas funcionalidades.

Por lo tanto, en nuestro caso plantearemos el día 9 de enero de 2022 como fecha de finalización estimada así que nuestro proyecto habría tenido una duración global de 400 días aproximadamente, cifra muy alta pero que disminuye considerablemente si tenemos en cuenta los días en los que se ha podido dar dedicación al proyecto como veremos en la siguiente tabla.

A continuación, procedo a reflejar la distribución temporal (siempre de una manera aproximada) de cada una de las etapas seguidas en el desarrollo del proyecto (tabla X):

Etapa	Descripción	Fecha Inicio	Fecha Fin	Nº de días
Etapa 1	Análisis de requisitos	1/01/2021	05/02/2021	36
Etapa 2	Creación de base de datos y proyectos iniciales	06/02/2021	16/02/2021	10
Etapa 3	Login, registro y formularios iniciales	17/02/2021	10/03/2021	21
Etapa 4	Generación de pdf imprimible	11/03/2021	06/04/2021	25
Etapa 5	Almacenamiento y tratamiento de matrículas	07/04/2021	01/06/2021	54
Etapa 6	Generación de estadísticas relativas a los alumnos por curso y asignatura	02/06/2021	10/08/2021	69
Etapa 7	Generación de grupos por curso	11/08/2021	25/09/2021	45
Etapa 8	Mejora de la seguridad	26/09/2021	31/10/2021	35
Etapa 9	Mejora de la interfaz	01/11/2021	28/11/2021	27
Etapa 10	Revisión final y documentación	28/11/2021	20/12/2021	32
Nº de días totales				352

Tabla 2: Distribución temporal de las etapas

Por último, hemos de destacar que las estimaciones anteriores pueden sufrir alteraciones que hagan que aumenten o disminuyan por inconvenientes o problemas surgidos durante la realización del desarrollo de estas.

4.3. Costes del proyecto

4.3.1. Costes de personal

Son los costes que hacen referencia a los recursos humanos que se emplean durante el desarrollo del proyecto. Para este proyecto tan solo tendremos un recurso humano, el autor del proyecto, puesto que es la persona que va a hacerse cargo de todas las partes del desarrollo del proyecto, aunque tomando diferentes cargos en cada una de las fases. Los cargos que va a desempeñar son:

- Jefe de proyecto
- Analista de requisitos
- Desarrollador “Full Stack” (tanto “Front-End” como “Back-End”)
- Diseñador de interfaces
- Responsable del control de calidad

Los costes reales de estos cargos se desglosarán en el apartado XX en función del tiempo que conlleve cada uno de ellos y del salario que se decida para cada uno.

4.3.2. Costes Hardware

Estos costes hacen referencia a los distintos equipos tecnológicos empleados para este proyecto. Los costes asociados corresponden con:

Equipo tecnológico	Precio (€)	Vida útil (años)
Ordenador portátil	500	6
Teclado inalámbrico	15	4
Ratón inalámbrico	10	3

Tabla 3: Costes hardware

4.3.3. Costes software

Estos costes hacen referencia a los programas y herramientas usadas para realizar este proyecto. En nuestro caso no han supuesto ningún coste puesto que todos son gratuitos:

Programas/Herramienta	Precio (€)
Windows 10 Home	Gratuito
Office 365	Gratuito
Visual Studio Code	Gratuito
Firebase	Gratuito
Github	Gratuito
Draw Io	Gratuito

Tabla 4: Costes Software

4.3.4. Costes Añadidos

Estos costes hacen referencia a algunos costes que se incluyen en el proyecto, pero no se pueden categorizar en ninguna de las secciones:

Descripción	Coste mensual (€)
Conexión a internet (WiFi)	50
Electricidad	20 aproximadamente

Tabla 5: Costes añadidos

4.4. Estimaciones Iniciales

En esta sección se va a detallar la estimación inicial del proyecto en cuanto a sus costes y su duración. Existen diferentes métodos de estimación empleados en el desarrollo software, por lo que en nuestro proyecto nos vamos a decantar por el método de Albrecht con puntos de función. El objetivo que se persigue es obtener una estimación precisa y así poder compararlo con la planificación inicial realizada para obtener una visión más exhaustiva y precisa con relación a los costes temporales y presupuestarios finales del proyecto.

4.4.1. Estimación por puntos de función (Método Albrecht)

Este método se basa en una métrica que cuantifica la funcionalidad que se debe entregar al usuario para construir la aplicación.

Los parámetros que sirven para evaluar dicha funcionalidad son:

- **Número de entradas:** datos que aporta el usuario al sistema (nombre de ficheros, menú de selección). Siempre actualizan algún fichero interno.
- **Número de salidas:** datos que el sistema aporta al usuario (informes, mensajes)

- **Número de ficheros lógicos internos:** ficheros o bases de datos internos al sistema (es decir sólo los utiliza el sistema, ficheros maestros).
- **Número de ficheros externos:** ficheros o bases de datos externos a los que accede el sistema con el único objetivo de obtener información (es decir son mantenidos por otras aplicaciones).
- **Número de consultas externas:** entradas que requieren de una respuesta por parte del sistema. No modifica los datos del sistema.

Por lo tanto, se necesita detallar los diferentes elementos organizados en sus respectivas categorías (tablas X, X, X, X y X) presentes en nuestro sistema (junto con la complejidad de cada uno de ellos).

Entradas de usuario	
Descripción	Complejidad
Datos de inicio de sesión de usuario	Media
Datos de creación de cuenta de usuario	Media
Datos personales de usuario	Media
Datos de los progenitores de usuario	Media
Datos de la dirección de usuario	Media
Datos de la matrícula de usuario	Alta
Datos en formato CSV de la matrícula (personales, parentales, dirección y matrícula) de usuario	Alta
Parámetros de búsqueda para datos de alumnos por asignatura	Baja
Parámetros de búsqueda para la generación de grupos por curso	Baja

Tabla 6: Complejidad entradas de usuarios

Salidas externas	
Descripción	Complejidad
Mensajes al usuario	Baja
Mensajes de error en los formularios	Baja

Tabla 7: Complejidad salidas externas

Consultas externas	
Descripción	Complejidad
Pdf de matrícula generado	Alta
Datos de los alumnos filtrados por curso y asignatura	Media
Grupos generados por curso	Alta
Datos personales de usuario	Baja
Datos de los progenitores de usuario	Baja
Datos de la dirección de usuario	Baja
Datos de la matrícula de usuario	Media

Tabla 8: Complejidad consultas externas

Ficheros lógicos externos	
Descripción	Complejidad
Cloud Firestore	Alta

Tabla 9: Complejidad ficheros externos

A continuación, debemos calcular los puntos de función sin ajustar (PFNA) en base a (tabla X):

Dominio	Complejidad	Peso x Complejidad	Nº total de funciones	Total
Entradas	Baja	x3	5	15
	Media	x4	2	8
	Alta	x6	2	12
Salidas	Baja	x4	2	8
	Media	x5	0	0
	Alta	x7	0	0
Consultas	Baja	x3	3	9
	Media	x4	2	8
	Alta	x6	2	12
Ficheros lógicos internos	Baja	x7	0	0
	Media	x10	0	0
	Alta	x15	0	0
Ficheros lógicos externos	Baja	x5	0	0
	Media	x7	0	0
	Alta	x10	1	10
Puntos de función sin ajustar (PFNA)				82

Tabla 10: Puntos de función sin ajustar

Una vez calculados los puntos de función sin ajustar, tenemos que asignar el grado de complejidad a cada uno de los factores de complejidad que afectan a la funcionalidad del sistema (tabla X). Cada uno de esos factores se evalúan del 0 al 5 en función de su incidencia:

- 0: Sin influencia
- 1: Incidental
- 2: Moderado
- 3: Medio
- 4: Significativo
- 5: Esencial

Factor	Valor
1.-Frecuencia de transacciones	4
2.-Comunicación de datos	4
3.-Funciones distribuidas	0
4.-Rendimiento	3
5.-Diseño para la eficiencia del usuario final	3
6.-Entrada de datos en línea	4
7.-Actualización de datos en línea	4
8.-Procesos complejos	3

9.-Facilidad de instalación	0
10.-Reusabilidad	1
11.-Configuraciones fuertemente utilizadas	0
12.-Instalaciones en múltiples lugares	0
13.-Facilidad de cambio	3
14.-Facilidad de operación	4
Total	33

Tabla 11: Grado de complejidad de los factores

El factor de ajuste (FA) se obtiene a partir de la suma de los factores de complejidad anteriormente descritos gracias a la siguiente ecuación:

$$FA = (0.01 \times FC) + 0.65 = (0.01 \times 33) + 0.65 = 0.98$$

Para obtener **los puntos de función ajustados** bastaría con aplicar a los puntos de función no ajustados el factor de ajuste obtenido:

$$PF = PFNA \times FA = 82 * 0.98 = 80.36$$

Por último, debemos calcular la duración estimada del proyecto en base a los resultados obtenidos. La equivalencia en tiempo basándose en valores medios de la industria informática respecto a los puntos de función es: 1 mes (21 días laborables) es igual a 13 puntos de función.

Por lo tanto, la duración del proyecto estimada es igual a: **80.36/13 = 6.18 meses**

4.5. Presupuestos

En este apartado se detallarán los costes totales del proyecto en función de la estimación obtenida por los puntos de función (Método Albrecht).

Por otro lado, se calcularán también dichos costes según la planificación inicial realizada (XXX días que equivalen a XX meses aproximadamente), y en caso de no respetar los plazos establecidos se detallará el incremento/decremento de coste en el presupuesto.

Para ello se aglutinarán los costes relativos a los componentes hardware y software, junto con los costes asociados a los recursos humanos y a los costes añadidos por otras cuestiones.

4.5.1. Costes hardware y software

Para evaluar dichos costes se ha tenido en cuenta el porcentaje de uso de los componentes para el desarrollo del proyecto en relación con la vida útil media de cada uno de ellos, y así obtener el coste real asociado al proyecto.

Albrecht

Componente Hardware	Precio (€)	Vida útil media (años)	Porcentaje de uso (%)	Coste real (€)
Ordenador portátil	500	6	8,56%	42,8
Teclado inalámbrico	15	4	12,87%	1,94
Ratón inalámbrico	10	3	17,17%	1,72
Total (€)				46,46

Tabla 12: Coste real Albrecht

Planificación inicial

Componente Hardware	Precio (€)	Vida útil media (años)	Porcentaje de uso (%)	Coste real (€)
Ordenador portátil	500	6	15,93%	79,65
Teclado inalámbrico	15	4	23,9%	3,59
Ratón inalámbrico	10	3	31,86%	3,19
Total (€)				86,43

Tabla 13: Coste real planificación inicial

En el caso de los costes software consideramos que son nulos en ambos presupuestos ya que todos los componentes se obtuvieron de manera gratuita.

Componente software	Precio
Windows 10 Home	Gratuito
Office 365	Gratuito
Visual Studio Code	Gratuito
Github	Gratuito
Draw Io	Gratuito
Total (€)	0

Tabla 14: Coste real software

4.5.2. Costes Recursos humanos

Teniendo en cuenta que el proyecto ha sido llevado a cabo por una sola persona, desempeñando distintos roles (especificados en el apartado “Costes de personal”) vamos a agrupar los roles en tres categorías más concretas:

- **Analista:** se encarga de realizar las tareas de gestión del equipo, análisis de requisitos, diseño de interfaces y elaboración de documentación.
- **Programador:** se encarga de realizar las tareas de implementación, tanto a nivel visual como funcional del proyecto.
- **Jefe de control de calidad:** se encarga de realizar las pruebas necesarias y asegurar la calidad del proyecto.

Ya definidos los roles, necesitamos definir el tiempo que va a dedicar cada uno de ellos a la realización del proyecto, ya que el coste de cada uno de ellos varía.

En nuestro caso, el analista va a desempeñar un 30% del tiempo, el programador un 55% del tiempo y el jefe de control de calidad el 15% sobrante. Como no sabemos las horas diarias que trabaja cada persona, fijamos la media de 2 horas que hemos establecido anteriormente.

En este punto vamos a definir los salarios base de cada uno de los roles propuesto anteriormente, tomando como referencia la página web **jobted.es**:

Albrecht

Rol	Salario mensual (€)	Salario real (2h) (€)	Duración (mes)	Porcentaje de trabajo (%)	Coste total (€)
Analista	1.890	472,5	6,18	30%	876,01
Programador	1.610	402,5	6,18	55%	1.368,1
Control de calidad	1.844	461	6,18	15%	427,35
TOTAL (€)					2.671,46

Tabla 15: Coste recursos humanos Albrecht

Planificación inicial

Rol	Salario mensual (€)	Salario real (2h) (€)	Duración (mes)	Porcentaje de trabajo (%)	Coste total (€)
Analista	1.890	472,5	11,57	30%	1.640,05
Programador	1.610	402,5	11,57	55%	2.561,31
Control de calidad	1.844	461	11,57	15%	800,06
TOTAL (€)					5.001,42

Tabla 16: Coste recursos humanos planificación inicial

4.5.3. Costes añadidos

Albrecht

Descripción	Coste mensual (€)	Duración proyecto (mes)	Total (€)
Conexión a internet (WiFi)	50	6,18	309
Electricidad	20 aproximadamente	6,18	123,6
Total (€)			432,6

Tabla 17: Costes añadidos Albrecht

Planificación inicial

Descripción	Coste mensual (€)	Duración proyecto (mes)	Total (€)
Conexión a internet (WiFi)	50	11,57	573,5
Electricidad	20 aproximadamente	11,57	231,4
Total (€)			804,9

Tabla 18: Costes añadidos planificación inicial

A continuación, hay que detallar el coste total en cada una de las duraciones estimadas del proyecto:

Modelo	Duración (mes)	Coste (€)	Coste final (€)
Planificación inicial	11,57	Personal: 5.001,42 Añadidos: 804,9 Hardware: 86,43 Software: 0	5.892,75
Método de Albrecht	6,18	Personal: 2.671,46 Añadidos: 432,6 Hardware: 46,46 Software: 0	3.150,52

Tabla 19: Coste total duraciones estimadas

4.5.4. Presupuesto final

En este punto se van a calcular los costes reales del proyecto con relación a la fecha prevista de entrega de este, el día 31 de enero de 2022, es decir, unos 40 días más de lo previsto en la planificación inicial, lo que ha hecho que el coste previsto se vea incrementado.

El motivo principal de este retraso es la falta de conocimiento en varios ámbitos que hacen que las estimaciones iniciales fueran demasiado optimistas, tanto en la parte de desarrollo como en la parte de documentación. Además, tendremos que contar con imprevistos de última hora en el desarrollo que hicieron que hubiese que añadir nuevas tareas de implementación y, por supuesto, sus pruebas complementarias.

A continuación, se muestran los costes de hardware, software, personal y añadidos reales del proyecto usando la fecha de finalización de este:

Componente Hardware	Precio (€)	Vida útil media (años)	Porcentaje de uso (%)	Coste real (€)
Ordenador portátil	500	6	17,9%	89,5
Teclado inalámbrico	15	4	26,85%	4,03
Ratón inalámbrico	10	3	35,8%	3,58
Total (€)				97,11

Tabla 20: Presupuesto final hardware

Componente software	Precio
Windows 10 Home	Gratuito
Office 365	Gratuito
Visual Studio Code	Gratuito
Github	Gratuito
Draw io	Gratuito
Total (€)	0

Tabla 21: Presupuesto final software

Rol	Salario mensual (€)	Salario real (2h) (€)	Duración (mes)	Porcentaje de trabajo (%)	Coste total (€)
Analista	1.890	472,5	13,06	30%	1.851,25
Programador	1.610	402,5	13,06	55%	2.891,16
Control de calidad	1.844	461	13,06	15%	903,01
TOTAL (€)					5.645,28

Tabla 22: Presupuesto final recursos humanos

Descripción	Coste mensual (€)	Duración proyecto (mes)	Total (€)
Conexión a internet (WiFi)	50	13,06	653
Electricidad	20 aproximadamente	13,06	261,2
Total (€)			914,2

Tabla 23: Presupuesto final conexiones y electricidad

Por tanto, la suma total de los costes reales del proyecto asciende a la cifra de **6.656,59€**. Comparando esta cifra con la planificación inicial (**5.892,75€**) supone un incremento de **763,84€ (12,9%)**. Dicho incremento se entiende asumible, teniendo en cuenta los motivos que han llevado a generar este incremento.

Capítulo 5.

Análisis del Sistema.

5. Análisis del sistema

En este apartado se van a describir los distintos aspectos que se han considerado previamente a la realización del diseño e implementación del sistema. Esos aspectos abarcan desde los distintos tipos de usuario del sistema, hasta los requisitos que debe cumplir el sistema, acompañados de las especificaciones y diagramas que los representan.

Los subapartados de los que se compone este apartado son:

- **Tipos de usuario:** enumera y explica los tipos de usuarios que puede tener el sistema.
- **Requisitos de usuario:** enumera los requisitos de usuario del proyecto.
 - **Relativos a la gestión de usuario**
 - **Relativos al proceso de matriculación**
 - **Relativos a la gestión de datos de matrículas**
- **Casos de uso:** se explican los casos de uso del proyecto agrupados por tipo de usuario.
 - **Relativos al usuario**
 - **Relativos al alumno**
 - **Relativos al administrador**
- **Reglas de negocio:** se recogen las reglas de negocio.
- **Requisitos funcionales:** se desarrollan los requisitos funcionales.
- **Requisitos no funcionales:** se explican los distintos requisitos no funcionales divididos en:
 - **Requisitos de interfaz externa**
 - **Atributos de calidad**
- **Requisitos de información:** se explican los distintos requisitos de información del proyecto.

5.1. Tipos de usuario del sistema

- **Usuario:** Cualquier persona que accede a la aplicación web
- **Administrador:** Persona que pertenece al centro que gestiona esta aplicación
- **Alumno:** Persona que estudia o quiere estudiar en el centro de esta aplicación

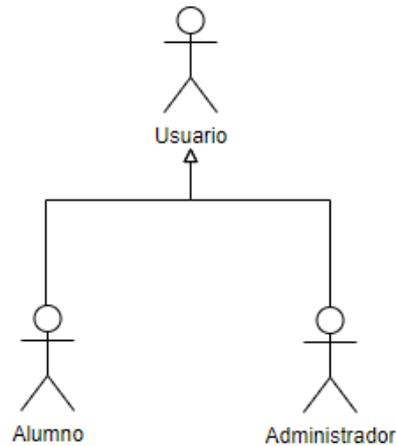


Figura 3: Usuarios del sistema

5.2. Requisitos de Usuario

5.2.1. Relativos a gestión de usuarios

- (RU-01) Todos los usuarios podrán iniciar sesión en el sistema.
- (RU-02) Todos los usuarios del sistema podrán cerrar sesión.
- (RU-03) Los alumnos podrán registrarse en el sistema.

5.2.2. Relativos al proceso de matriculación

- (RU-04) Los alumnos podrán matricularse
- (RU-05) Los alumnos podrán seleccionar las asignaturas de su matrícula
- (RU-06) Los alumnos podrán imprimir un justificante de su matrícula
- (RU-07) Los alumnos podrán visualizar su matrícula
- (RU-08) Los alumnos podrán modificar su matrícula.

5.2.3. Relativos a la gestión de datos de matrículas

- (RU-09) Los administrados podrán visualizar datos relativos a las matrículas de los alumnos
- (RU-10) Los administradores podrán obtener estadísticas relativas a los alumnos matriculados en cada asignatura
- (RU-11) Los administradores podrán obtener estadísticas relativas a los alumnos matriculados en cada curso
- (RU-12) Los administradores podrán obtener estadísticas de los alumnos que repiten curso
- (RU-13) Los administradores podrán generar los grupos de alumnos por curso

5.3. Casos de uso

5.3.1. Relativos al usuario

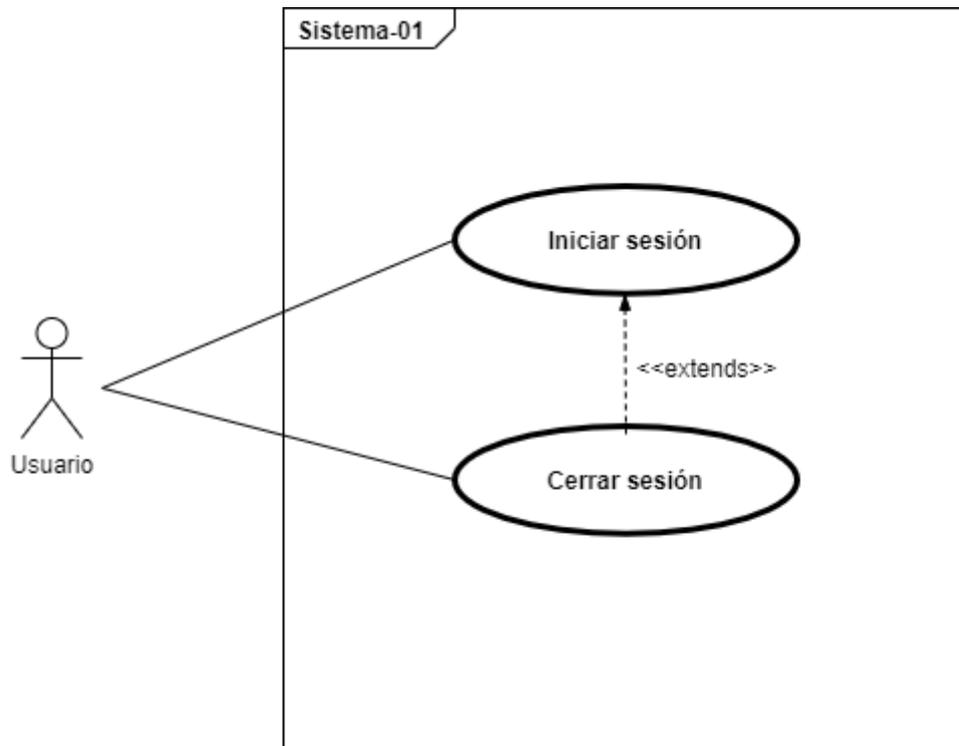


Figura 4: Casos de uso del usuario

Identificador	RU-01: Iniciar Sesión		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Usuario	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-02, RU-03
Descripción	El sistema permitirá al usuario acceder a la herramienta mediante un formulario único de inicio de sesión		
Pre-condiciones	El usuario debe de estar registrado en la herramienta		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita acceder a su cuenta 2. El sistema muestra al usuario los datos que debe rellenar (TODO y contraseña) 3. El usuario rellena los datos y los envía al sistema 4. El sistema comprueba que los datos existan y sean correctos 5. El sistema redirige al usuario a la página principal de su cuenta personal 		
Secuencia Alternativa			
Excepciones en la secuencia	4.1. Los datos de acceso son incorrectos, muestra un mensaje de error y vuelve al paso 2.		
Post-condiciones	El usuario accede a su cuenta de usuario para la realización de otras acciones.		
Importancia	Crítica	Prioridad	Alta

Tabla 24: Caso de uso " Iniciar Sesión "

Identificador	RU-02: Cerrar Sesión		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Usuario	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-01
Descripción	El sistema permitirá al usuario cerrar sesión		
Pre-condiciones	El usuario debe haber iniciado sesión en el sistema		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita cerrar sesión por medio de un botón 2. El sistema cierra la sesión del usuario que lo solicita 		
Secuencia Alternativa			
Excepciones en la secuencia			
Post-condiciones	El usuario no puede acceder al sistema porque ha cerrado su sesión		
Importancia	Moderada	Prioridad	Media

Tabla 25: Caso de uso "Cerrar Sesión"

5.3.2. Relativos al alumno

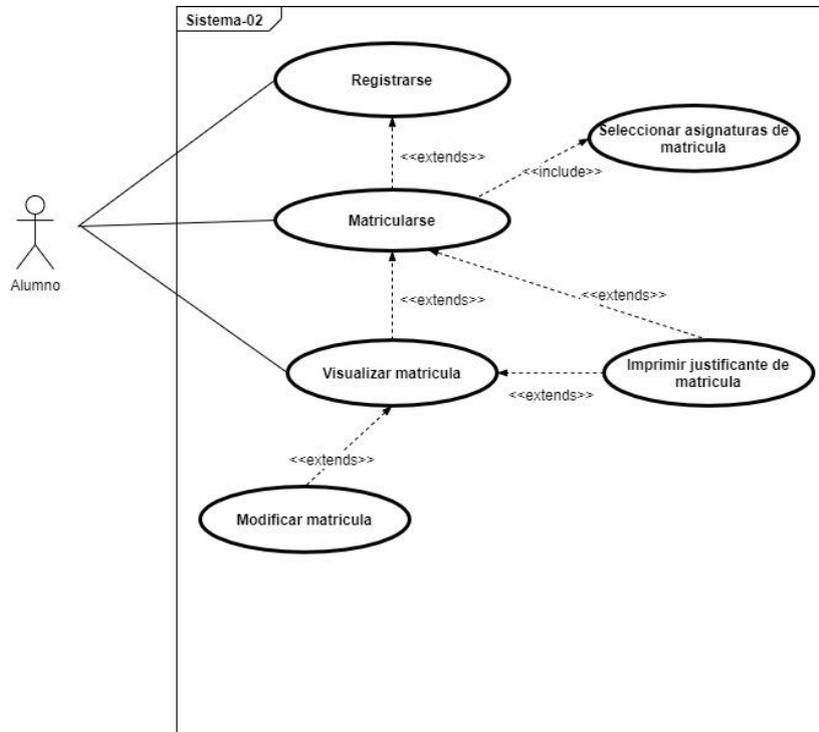


Figura 5: Casos de uso del alumno

Identificador	RU-03: Registrarse		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Alumno	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-01
Descripción	El sistema permitirá al usuario crear una cuenta para acceder a la herramienta.		
Pre-condiciones			
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita crear una cuenta en la herramienta 2. El sistema muestra al usuario los datos que debe rellenar 3. El usuario rellena los datos y los envía al sistema 4. El sistema comprueba que los datos sean correctos 5. El sistema crea el nuevo usuario 		
Secuencia Alternativa			
Excepciones en la secuencia	<p>4.1. Los datos de acceso ya existen en la herramienta, muestra un mensaje de error y vuelve al paso 2.</p> <p>4.2. Los datos de acceso son erróneos, muestra un mensaje de error y vuelve al paso 2.</p>		
Post-condiciones	La información se guarda en el sistema y pasa a estar registrado.		
Importancia	Crítica	Prioridad	Alta

Tabla 26: Caso de uso "Registrarse"

Identificador	RU-04: Matricularse		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Alumno	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-03, RU-05, RU-06, RU-07, RU-08
Descripción	El sistema permitirá al usuario crear una matrícula		
Pre-condiciones	El usuario debe de estar registrado en la herramienta		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita crear una matrícula en la herramienta 2. El sistema muestra el formulario de datos personales 3. El sistema muestra el formulario de datos parentales 4. El sistema muestra el formulario de datos de dirección 5. El sistema muestra al usuario las asignaturas que puede cursar 6. El usuario selecciona las asignaturas 7. El usuario confirma que ha finalizado su matrícula 8. El sistema comprueba que los datos sean correctos 9. El sistema crea una nueva matrícula 		
Secuencia Alternativa			
Excepciones en la secuencia	<p>6.1. Alguno de los datos obligatorios del usuario no está relleno, el sistema muestra un mensaje y vuelve al paso 2</p> <p>6.2. El usuario no ha seleccionado el número de asignaturas que la matrícula exige, el sistema muestra un mensaje y vuelve al paso 3</p>		
Post-condiciones	La matrícula se guarda en el sistema		
Importancia	Crítica	Prioridad	Alta

Tabla 27: Caso de uso "Matricularse"

Identificador	RU-05: Seleccionar asignaturas de la matrícula		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Alumno	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-04
Descripción	El sistema permitirá al usuario seleccionar las asignaturas de su matrícula		
Pre-condiciones	El usuario debe de estar registrado en la herramienta y haber comenzado el proceso de matriculación		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita crear una matrícula en la herramienta 2. El sistema muestra los datos del alumno 3. El sistema muestra al usuario las asignaturas que puede cursar 4. El usuario selecciona las asignaturas que desea cursar 5. El proceso de matriculación continua 		
Secuencia Alternativa			
Excepciones en la secuencia			
Post-condiciones	Las asignaturas seleccionadas por el usuario se almacenan en la matrícula		
Importancia	Crítica	Prioridad	Alta

Tabla 28: Caso de uso "Seleccionar asignaturas de la matrícula"

Identificador	RU-06: Imprimir justificante de la matrícula		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Alumno	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-04, RU-08
Descripción	El sistema permitirá al usuario imprimir un justificante de su matrícula		
Pre-condiciones	El usuario debe de estar registrado en la herramienta y haber completado el proceso de matriculación		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario finaliza el proceso de matriculación 2. El sistema muestra la opción de imprimir un justificante de su matrícula 3. El usuario acepta imprimir un justificante de su matrícula 		
Secuencia Alternativa	<ol style="list-style-type: none"> a. El usuario accede a la visualización de su matrícula b. El sistema le ofrece la posibilidad de imprimir un justificante de su matrícula 		
Excepciones en la secuencia			
Post-condiciones	El sistema genera un justificante de matrícula que el usuario puede imprimir		
Importancia	Crítica	Prioridad	Alta

Tabla 29: Caso de uso "Imprimir justificante de la matrícula"

Identificador	RU-07: Visualizar matrícula		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Alumno	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-04
Descripción	El sistema permitirá al usuario visualizar su matrícula		
Pre-condiciones	El usuario debe de estar registrado en la herramienta y haber completado el proceso de matriculación		
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario solicita visualizar su matrícula 2. El sistema muestra la matrícula guardada en el sistema 		
Secuencia Alternativa			
Excepciones en la secuencia			
Post-condiciones			
Importancia	Crítica	Prioridad	Alta

Tabla 30: Caso de uso "Visualizar matrícula"

Identificador	RU-08: Modificar matrícula		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Alumno	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-08
Descripción	El sistema permitirá al usuario modificar su matrícula		
Pre-condiciones	El usuario debe de estar registrado en la herramienta y haber completado el proceso de matriculación		
Secuencia Normal	<ol style="list-style-type: none"> 3. El usuario accede a la visualización de su matrícula 4. El sistema ofrece la posibilidad de modificar la matrícula 5. El usuario modifica su matrícula 6. El usuario confirma que ha finalizado la modificación de su matrícula 7. El sistema comprueba que los datos sean correctos 8. El sistema guarda la matrícula modificada 		
Secuencia Alternativa			
Excepciones en la secuencia	5.1. El usuario ha modificado algún dato de su matrícula incorrectamente, el sistema muestra un mensaje y vuelve al paso 3		
Post-condiciones	El sistema guarda la matrícula con los datos actualizados		
Importancia	Crítica	Prioridad	Alta

Tabla 31: Caso de uso "Modificar matrícula"

5.3.3. Relativos al administrador

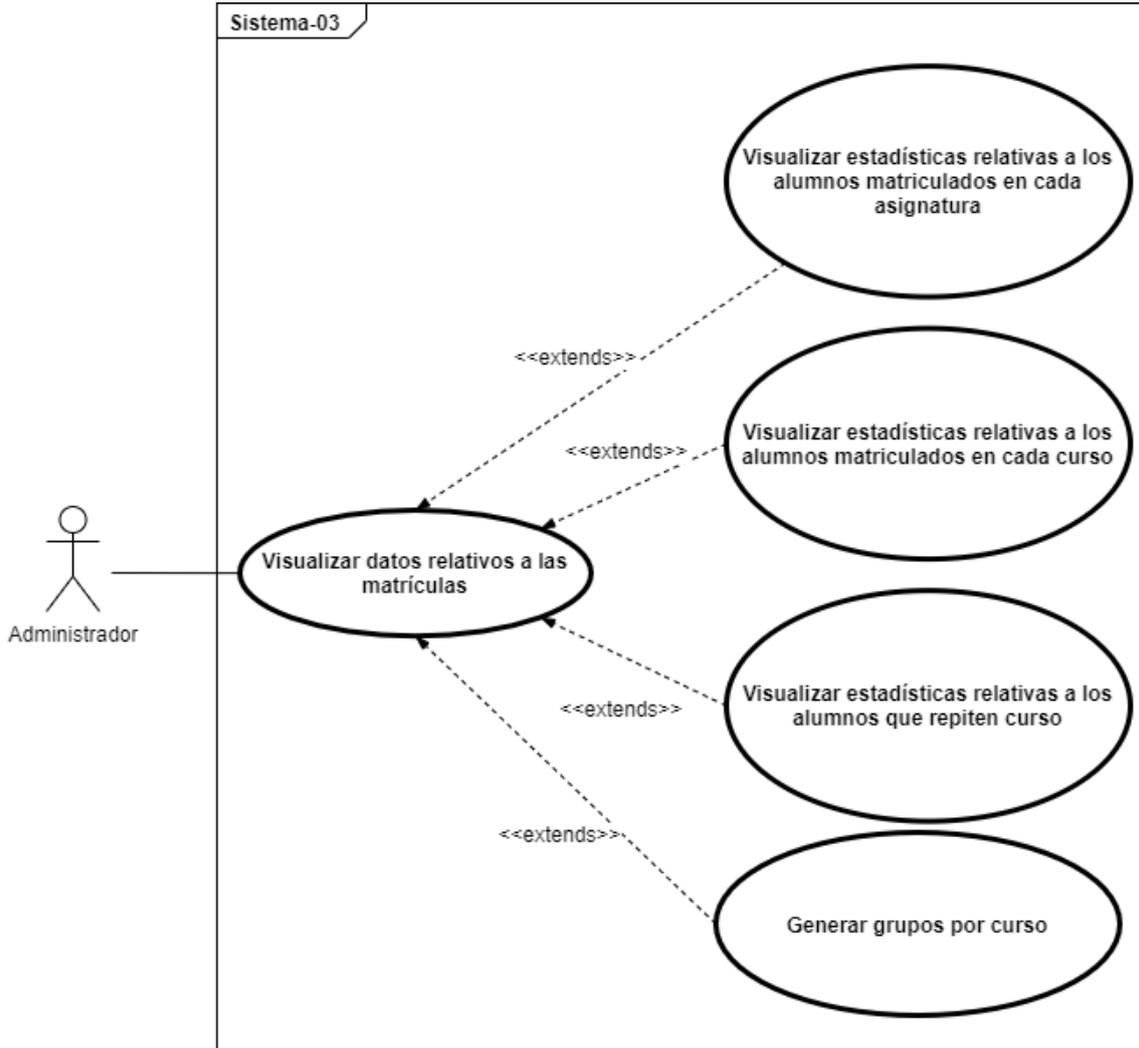


Figura 6: Casos de uso del administrador

Identificador	RU-09: Visualizar datos relativos a las matrículas de los alumnos		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Administrador	Actores Implicados	
Obj. Asociados		Req. Asociados	
Descripción	El sistema permitirá al administrador visualizar datos relativos a las matrículas de los alumnos		
Pre-condiciones	El administrador debe haber iniciado sesión en el sistema		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita visualizar datos relativos a las matrículas de los alumnos 2. El sistema muestra una pantalla con las diferentes opciones para generar estadísticas sobre los alumnos matriculados 		
Secuencia Alternativa			
Excepciones en la secuencia	2.1. El sistema no tiene ninguna matrícula, el sistema muestra un mensaje		
Post-condiciones	El sistema muestra la pantalla para generar estadísticas		
Importancia	Crítica	Prioridad	Alta

Tabla 32: Caso de uso "Visualizar datos relativos a las matrículas de los alumnos"

Identificador	RU-10: Visualizar estadísticas relativas a los alumnos matriculados en cada asignatura		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Administrador	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-10
Descripción	El sistema permitirá al administrador filtrar la visualización de matrículas por asignatura		
Pre-condiciones	El administrador debe haber iniciado sesión en el sistema		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita visualizar datos relativos a las matrículas de los alumnos 2. El administrador filtra las matrículas por asignatura 3. El sistema muestra las matrículas por asignatura 		
Secuencia Alternativa			
Excepciones en la secuencia	2.1. El sistema no tiene ninguna matrícula, el sistema muestra un mensaje		
Post-condiciones	El sistema muestra las matrículas almacenadas por asignatura		
Importancia	Crítica	Prioridad	Alta

Tabla 33: Caso de uso "Visualizar estadísticas relativas a los alumnos matriculados en cada asignatura"

Identificador	RU-11: Visualizar estadísticas relativas a los alumnos matriculados en cada curso		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Administrador	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-10
Descripción	El sistema permitirá al administrador filtrar la visualización de matrículas por curso		
Pre-condiciones	El administrador debe haber iniciado sesión en el sistema		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita visualizar datos relativos a las matrículas de los alumnos 2. El administrador filtra las matrículas por curso y optativa 3. El sistema muestra las matrículas por curso y optativa 		
Secuencia Alternativa			
Excepciones en la secuencia	2.1. El sistema no tiene ninguna matrícula, el sistema muestra un mensaje		
Post-condiciones	El sistema muestra las matrículas almacenadas por curso		
Importancia	Crítica	Prioridad	Alta

Tabla 34: Caso de uso "Visualizar estadísticas relativas a los alumnos matriculados en cada curso"

Identificador	RU-12: Visualizar estadísticas relativas a los alumnos que repiten curso		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Administrador	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-10
Descripción	El sistema permitirá al administrador filtrar la visualización de matrículas de alumnos que repiten curso		
Pre-condiciones	El administrador debe haber iniciado sesión en el sistema		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita visualizar datos relativos a las matrículas de los alumnos 2. El administrador filtra las matrículas por alumnos que repiten curso en cada curso 3. El sistema muestra las matrículas por alumnos que repiten curso en cada curso 		
Secuencia Alternativa			
Excepciones en la secuencia	2.1. El sistema no tiene ninguna matrícula, el sistema muestra un mensaje		
Post-condiciones	El sistema muestra las matrículas almacenadas por alumnos que repiten curso		
Importancia	Crítica	Prioridad	Alta

Tabla 35: Caso de uso "Visualizar estadísticas relativas a los alumnos que repiten curso"

Identificador	RU-13: Generar grupos por curso		
Versión Actual	1.0	Autor	Carlos Rascón
Actor Principal	Administrador	Actores Implicados	
Obj. Asociados		Req. Asociados	RU-10
Descripción	El sistema permitirá al administrador generar grupos por curso juntando a los alumnos por similitud de optativas elegidas.		
Pre-condiciones	El administrador debe haber iniciado sesión en el sistema		
Secuencia Normal	<ol style="list-style-type: none"> 1. El administrador solicita visualizar datos relativos a las matrículas de los alumnos 2. El administrador selecciona el curso del que quiere generar los grupos 3. El sistema genera los grupos y los muestra en pantalla 		
Secuencia Alternativa			
Excepciones en la secuencia	2.1. El sistema no tiene ninguna matrícula, el sistema muestra un mensaje		
Post-condiciones	El sistema muestra los grupos generados para el curso seleccionado		
Importancia	Crítica	Prioridad	Alta

Tabla 36: Caso de uso "Generar grupos por curso"

5.4. Reglas de Negocio

Las reglas de negocio no son requisitos de software por sí mismas, pero si son requisitos que hay que tener en cuenta ya que todo proyecto debe estar sujeto a un conjunto de normas, mandatos o restricciones de carácter obligatorio. Estas reglas por lo tanto serán la base y afectan al resto de requisitos que encontramos en nuestro proyecto, ya que establecen ciertas limitaciones para algunos de ellos.

Por lo tanto, podemos definir las reglas de negocio como aquellas que establecen todas las regulaciones y políticas de negocio que deberá seguir "GOMI: Gestión Online de Matrículas para un Instituto" de acuerdo con lo establecido por las normativas vigentes en la actualidad. En nuestro caso se han localizado las siguientes:

- **RN-01:** GOMI deberá cumplir la Ley Orgánica de Protección de datos en relación con la información publicada de usuarios
- **RN-02:** El usuario deberá identificarse para poder hacer uso de todas las funciones de la aplicación.
- **RN-03:** La aplicación permitirá al usuario cerrar sesión tras haberse logado.
- **RN-04:** La aplicación permitirá al usuario registrarse para poder gestionar su matrícula.
- **RN-04:** La aplicación permitirá a los administradores obtener datos sobre los alumnos matriculados.

- **RN-05:** La aplicación permitirá a los administradores generar los grupos basándose en las optativas elegidas por los alumnos.
- **RN-06:** La aplicación permitirá a los alumnos importar su matrícula por medio de un archivo en formato *csv*.
- **RN-07:** La aplicación permitirá visualizar, modificar y añadir información relativa a la matrícula.
- **RN-08:** La aplicación deberá generar un archivo en formato *pdf* con la información relativa a la matrícula.
- **RN-09:** El idioma disponible para la visualización de la aplicación es el español.

5.5. Requisitos funcionales

Caso de Uso	Identificador	Descripción
CU-01 (Iniciar sesión)	RF-01	El sistema mostrará pantalla de inicio de la aplicación
	RF-02	El sistema mostrará el formulario de login
	RF-03	El sistema mostrará errores cuando el usuario introduzca algún campo erróneo o vacío
	RF-04	El sistema mostrará un mensaje si el login es correcto
	RF-05	El sistema mostrará un mensaje si el login es erróneo
	RF-06	El sistema redirigirá al usuario a la página principal ya logado si el login es correcto
	RF-07	El sistema mostrará un link para crear una nueva cuenta debajo del formulario de inicio de sesión
CU-02 (Cerrar sesión)	RF-08	El sistema mostrará un botón para cerrar sesión si el usuario está logado
	RF-09	El sistema mostrará un “loading” mientras cierra la sesión del usuario
	RF-10	El sistema redirigirá a la página de login tras cerrar la sesión
CU-03 (Registrarse)	RF-11	El sistema mostrará el formulario de registro
	RF-12	El sistema mostrará mensajes de error en los campos con datos incorrectos o vacíos
	RF-13	El sistema mostrará un mensaje cuando la creación de cuenta es correcta
	RF-14	El sistema mostrará un mensaje cuando la creación de cuenta sea errónea
	RF-15	El sistema redirigirá al login si la creación de cuenta es correcta
CU-04 (Matricularse)	RF-16	El sistema mostrará el formulario de datos personales
	RF-17	El sistema mostrará mensajes de error en el formulario de datos personales si hay campos erróneos o vacíos
	RF-18	El sistema mostrará un botón para avanzar desactivado debajo del formulario de datos personales
	RF-19	EL sistema mostrará un botón para avanzar activado debajo del formulario de datos personales cuando este sea correcto
	RF-20	El sistema mostrará el formulario de datos parentales
	RF-21	El sistema mostrará mensajes de error en el formulario de datos parentales si hay campos erróneos o vacíos

	RF-22	El sistema mostrará un botón para avanzar desactivado debajo del formulario de datos parentales
	RF-23	El sistema mostrará un botón para volver al paso anterior debajo del formulario de datos parentales
	RF-24	EL sistema mostrará un botón para avanzar activado debajo del formulario de datos parentales cuando este sea correcto
	RF-25	El sistema mostrará el formulario de datos de dirección
	RF-26	El sistema mostrará mensajes de error en el formulario de datos de dirección si hay campos erróneos o vacíos
	RF-27	El sistema mostrará un botón para avanzar desactivado debajo del formulario de datos de dirección
	RF-28	El sistema mostrará un botón para volver al paso anterior debajo del formulario de datos de dirección
	RF-29	EL sistema mostrará un botón para avanzar activado debajo del formulario de datos de dirección cuando este sea correcto
CU-05 (Seleccionar asignaturas)	RF-30	El sistema mostrará un formulario con los distintos cursos
	RF-31	El sistema mostrará el formulario correspondiente al curso elegido
	RF-32	El sistema mostrará un botón de finalizar proceso desactivado debajo del formulario de asignaturas
	RF-33	El sistema mostrará un botón para volver al paso anterior debajo del formulario de asignaturas
	RF-34	El sistema mostrará un botón de finalizar proceso activado cuando el formulario esté completo
CU-06 (Imprimir justificante)	RF-35	El sistema mostrará un archivo pdf con los datos de la matrícula
	RF-36	El sistema mostrará un botón para imprimir la matrícula en pdf
CU-07 (Visualizar matrícula)	RF-37	El sistema mostrará el formulario de datos personales relleno
	RF-38	El sistema mostrará el formulario de datos parentales relleno
	RF-39	El sistema mostrará el formulario de datos de dirección relleno
	RF-40	El sistema mostrará el formulario de los cursos con el curso seleccionado
	RF-41	El sistema mostrará el formulario de asignaturas relleno
CU-08 (Modificar matrícula)	RF-42	El sistema mostrará el formulario de datos personales relleno y el usuario podrá modificarlo
	RF-43	El sistema mostrará el formulario de datos parentales relleno y el usuario podrá modificarlo
	RF-44	El sistema mostrará el formulario de datos de dirección relleno y el usuario podrá modificarlo
	RF-45	El sistema mostrará el formulario de los cursos con el curso seleccionado y el usuario podrá modificarlo
	RF-46	El sistema mostrará el formulario de asignaturas relleno y el usuario podrá modificarlo

CU-09 (Visualizar datos relativos a las matrículas de los alumnos)	RF-47	El sistema mostrará un botón para acceder a la zona de administradores si el usuario tiene derechos para ello
	RF-48	El sistema mostrará dos pestañas en la página de administrador
	RF-49	El sistema mostrará un formulario con las opciones para filtrar los datos en la primera pestaña
	RF-50	El sistema mostrará un formulario con los cursos para generar grupos en la segunda pestaña
CU-10 (Por asignatura)	RF-51	El sistema mostrará un formulario con las optativas del curso seleccionado
	RF-52	El sistema mostrará un gráfico con los datos de las matrículas filtradas por optativas
CU-11 (Por curso)	RF-53	El sistema mostrará un gráfico con el número de alumnos en cada curso
CU-12 (Repiten curso)	RF-54	El sistema mostrará un formulario con una opción para visualizar los alumnos que repiten curso
	RF-55	El sistema mostrará un gráfico con los datos de las matrículas filtradas por alumnos que repiten curso y no repiten curso en cada curso
CU-13 (Generar grupos)	RF-56	El sistema mostrará una tabla contraída con los grupos generados para un determinado curso
	RF-57	El sistema mostrará varias tablas contraídas con los alumnos y optativas elegidas en cada grupo para un determinado curso

Tabla 37: Requisitos funcionales

5.6. Requisitos No Funcionales

5.6.1. Requisitos de Interfaz Externa

Este tipo de requisitos definen las interfaces o componentes de carácter externo con los que contará el sistema. En nuestro caso como ya he descrito en la documentación, nuestro sistema se comunicará con los siguientes servicios:

- **RIE-01:** El sistema se comunicará con el servicio “Cloud Firestore” (de Firebase) cada vez que el usuario solicite una información, para recuperarla y posteriormente mostrársela al usuario.

5.6.2. Atributos de Calidad

Este tipo de requisitos revisten una importancia capital, ya que nos permitirán discernir si el sistema cumple con todos los estándares de calidad y también comprobar si el rendimiento del sistema es el correcto. Estos atributos serán medibles y cuantificables para determinar si el sistema cumple con la especificación.

A continuación, se agrupan los diferentes requisitos agrupados por la categoría del atributo:

- **AC-01: Seguridad:** la comunicación entre el cliente y el servidor se realiza mediante el protocolo de comunicación HTTPS.
- **AC-02: Robustez:** el sistema se mantendrá en activo ante situaciones de caída.
- **AC-03: Robustez:** el sistema se mantendrá estable ante situaciones inusuales (excesivo número de usuarios, importante tráfico de datos).
- **AC-04: Rendimiento:** el sistema responderá en un tiempo inferior a tres segundos.
- **AC-05: Disponibilidad:** el sistema deberá estar disponible 24/7 para que el usuario pueda acceder.

- **AC-06: Escalabilidad:** la aplicación será compatible con cualquier dispositivo que disponga de un navegador web.
- **AC-07: Usabilidad:** la aplicación será compatible con los dispositivos conectados a internet
- **AC-08: Usabilidad:** los usuarios disponen de permiso de lectura y escritura de sus datos almacenados en la base de datos.
- **AC-09: Usabilidad:** el sistema mostrará al usuario los mensajes pertinentes en caso de no aparecer los datos o por necesidad de alguna confirmación.
- **AC-10: Usabilidad:** el usuario será capaz de acceder de manera sencilla e intuitiva a los datos y opciones que le interesen.
- **AC-11: Accesibilidad:** la aplicación será adaptable a los diferentes tamaños de las pantallas presentes en el mercado para su correcta visualización.
- **AC-12: Accesibilidad:** la aplicación será compatible con el idioma español.
- **AC-13: Eficiencia:** el sistema deberá ser capaz de soportar la presencia de numerosos usuarios al mismo tiempo.
- **AC-14: Implementación:** el desarrollo de la aplicación se realizará a través del entorno de programación Visual Studio Code y con los lenguajes de programación Javascript, sobre el framework Node Js y Typescript, sobre el framework Angular en su versión 8.2.0.

5.7. Requisitos de Información

Los requisitos de información describen la información que debe almacenar y gestionar el sistema para dar soporte a los procesos de negocio. En el caso de nuestro proyecto son:

- **RI-01:** El sistema se encargará de gestionar los datos de usuario de los alumnos con cuenta creada en la aplicación
- **RI-02:** El sistema se encargará de almacenar y gestionar las matrículas de los alumnos registrados en la aplicación.

Capítulo 6.

Diseño.

6. Diseño

En este apartado se explican cuestiones referentes al diseño que se ha seguido a la hora de desarrollar la aplicación, tratando temas como la arquitectura, lógica y física, los modelos usados en el diseño y las interfaces de las que consta la aplicación.

Los subapartados de los que consta esta sección son:

- **Arquitectura:** se explica la arquitectura que se ha seguido a la hora de desarrollar el proyecto, tanto física como lógica.
 - Arquitectura lógica
 - Arquitectura física
- **Modelos de diseño:** se encuentran los distintos diagramas y representaciones que ayudarán a comprender el diseño final de nuestra aplicación.
 - **Diagramas de secuencia**
 - **Modelo lógico de datos**
 - **Diccionario de datos**
- **Diseño de interfaces:** se exponen y explican las distintas interfaces de las que consta la aplicación.

6.1. Arquitectura

6.1.1. Arquitectura lógica

La arquitectura lógica consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software.

Para elaborar el diseño de la arquitectura lógica de la aplicación hay que plantearse diferentes cuestiones, como por ejemplo elegir el número de capas de las que va a constar la aplicación. En nuestro caso hemos elegido una arquitectura de 2 capas (presentación y lógica y persistencia) que se adapte a las necesidades de nuestro proyecto.

La capa de presentación se corresponde con la parte del cliente de nuestra aplicación web, puesto que se encarga de pedir los datos a una Web Api y mostrarlos al usuario. Se corresponde con la parte hecha en el lenguaje de programación Angular.

La capa de lógica y persistencia se corresponde con la parte del servidor de nuestra aplicación y está escrita en Node Js. Se encarga de realizar todas las operaciones sobre los datos, para servirlos y que la capa de presentación los muestre. Además, se encarga de almacenar y obtener los datos desde la base de datos, y es el único responsable de interactuar con los datos almacenados, puesto que es una parte compleja y peligrosa, puesto que puede dañar o incluso eliminar datos de la aplicación.

Por último, la capa de lógica y presentación se comunica con Firebase, y más en concreto con el servicio Cloud Firestore, que es el encargado de almacenar los datos de la aplicación.

A continuación, se expone la arquitectura lógica asociada a nuestro sistema:

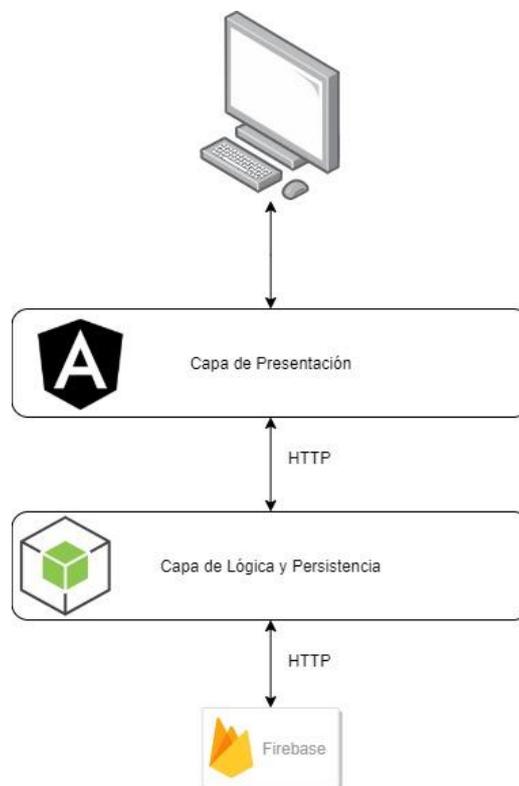


Figura 7: Arquitectura lógica

6.1.2. Arquitectura física

Se encarga de representar por qué medios interactúa nuestra aplicación con los usuarios finales que la usan.

Lo que hay que tener en cuenta para determinar esta arquitectura son los atributos de calidad que hay que satisfacer, para ello se ha decidido llevar a cabo un modelo de cliente-servidor, el modelo típico de una aplicación web.

Las ventajas principales de esta arquitectura son:

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de empleo.
- En las redes C/S los demás clientes no tienen acceso a las IP's por lo que se dificulta el rastreo y/o hackeo de los usuarios

Para nuestro proyecto, el cliente será el navegador web de un dispositivo con conexión a internet y la parte de servidor será un servidor con nuestro código desplegado que conecte con la base de datos y sirva contenido al cliente por medio de llamadas HTTP.

Por lo que la arquitectura física se representaría con:

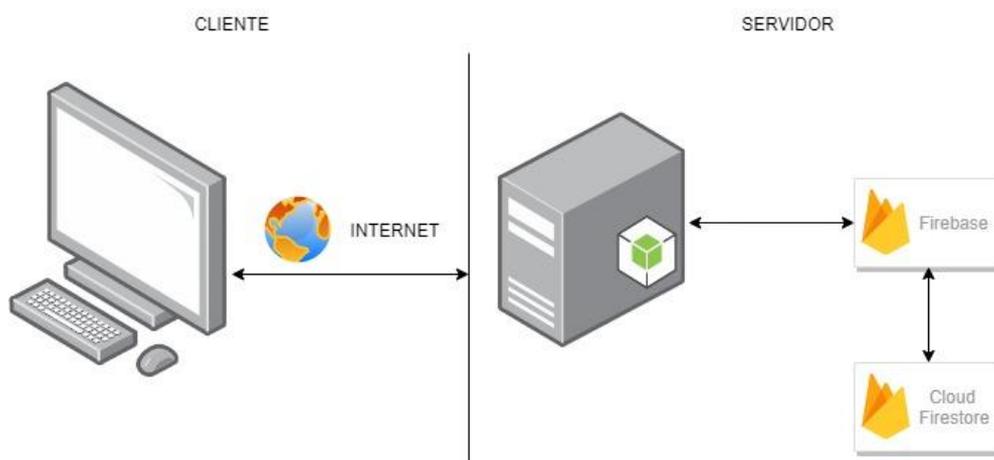


Figura 8: Arquitectura física

6.2. Modelos de Diseño

6.2.1. Diagrama de secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario.

En nuestro caso hemos modelado tres diagramas de secuencia que se centran en tres funcionalidades de nuestro proyecto que representan muchas otras, bastante similares.

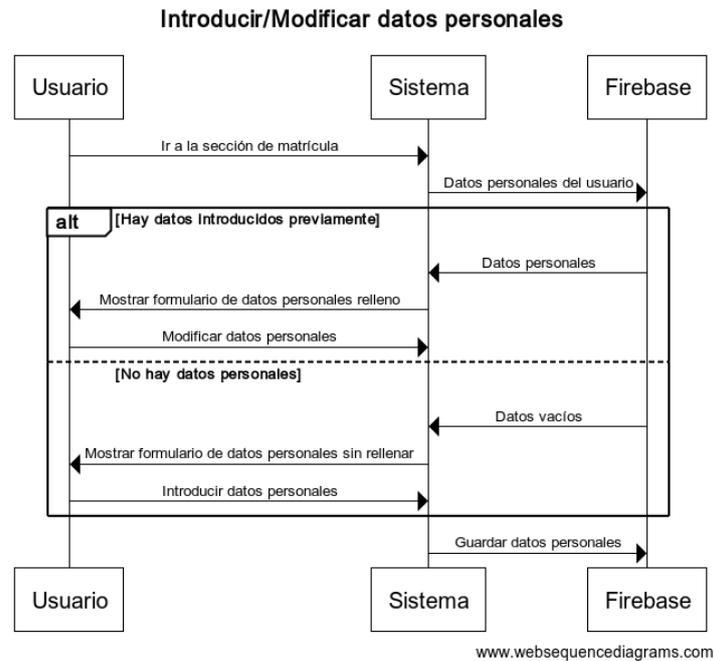


Figura 9: Diagrama de secuencia "Introducir/Modificar datos personales"

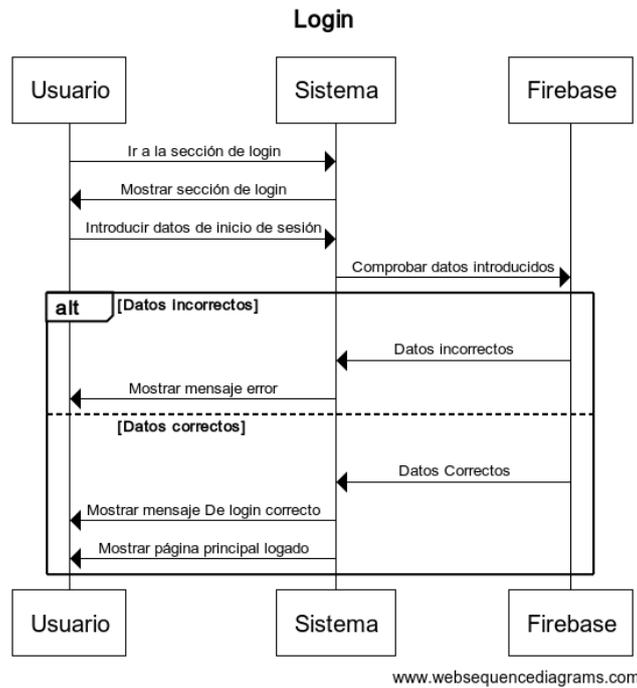


Figura 10: Diagrama de secuencia "Login"

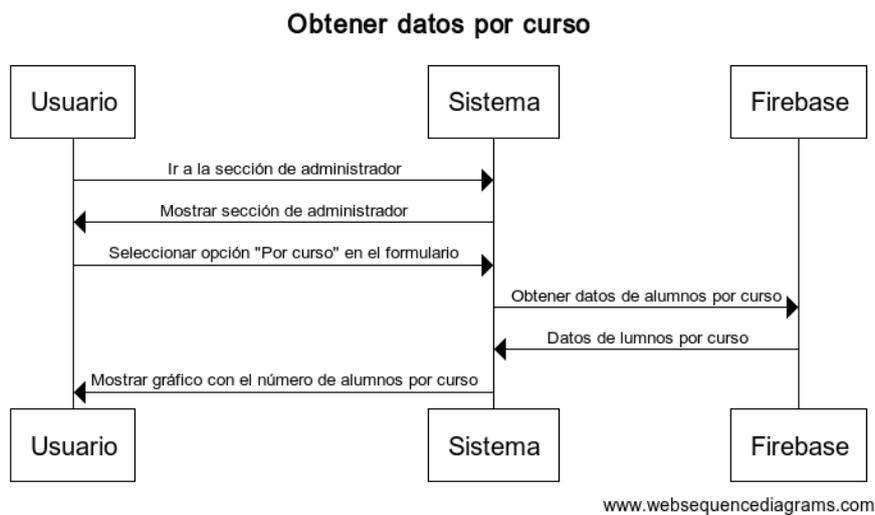


Figura 11: Diagrama de secuencia "Obtener datos por curso"

6.2.2. Modelo Lógico de Datos

En primer lugar, antes de describir la estructura de la base de datos seguida en nuestro proyecto, es necesario precisar la tipología de base de datos utilizada:

- NoSQL: Son los mecanismos para el almacenamiento y recuperación de datos que difieren, en aspectos importantes, del modelo clásico de relaciones entre entidades (tablas) existente en los sistemas de gestión de bases de datos relacionales, siendo el más destacado el que no usan SQL como lenguaje principal de consulta.

Los motivos para decantarnos por este modelo son la posibilidad de disponer de un esquema de base de datos no homogéneo, y la capacidad de soportar picos elevados de uso del sistema por parte de los usuarios.

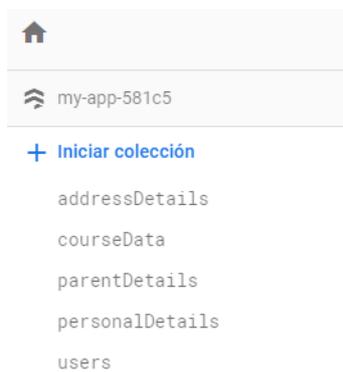
Dentro de este modelo existen diferentes tipos, pero nosotros hemos utilizado uno basado en los documentos debido a sus características principales:

- Almacenan la información como un documento.
- Utilizan estructuras simples como JSON.
- Permite realizar búsquedas por clave-valor y consultas más complejas sobre el contenido almacenado.
- La naturaleza flexible, jerárquica y semiestructurada permite que evolucionen atendiendo a las necesidades de las distintas aplicaciones.

El servicio utilizado por nuestro proyecto para proporcionarnos dicho esquema de base de datos es “Cloud Firestore” perteneciente a la plataforma Firebase. A diferencia de las SQL, no hay ni tablas ni filas, simplemente almacena los diferentes campos en documentos (pares clave-valor a modo de archivo Json), que se organizan en colecciones.

Los detalles de implementación relativos a este servicio se describen en el apartado “Servidores de bases de datos utilizados” dentro de esta misma documentación.

Tras esta explicación, se expone la estructura de la base de datos utilizada para este proyecto:



- Observamos las diferentes colecciones de datos que almacenan los datos de la aplicación.
- Cada colección cuenta con una serie de documentos que se identifican con un identificador único y almacenan los datos de la aplicación.
- En nuestro proyecto, no hay ningún documento con subcolecciones anidadas, algo que es posible en Cloud Firestore, pero no necesario para este proyecto.

Figura 12: Estructura de base de datos "Cloud firestore"

Para acabar esta sección, hay que describir el procedimiento seguido para almacenar y tratar estos datos desde la implementación de la aplicación. Una de las mayores ventajas de Cloud Firestore es que interpreta los datos de manera automática, lo que hace que simplemente haya que enviarle los datos con un modelo predefinido desde nuestra aplicación.

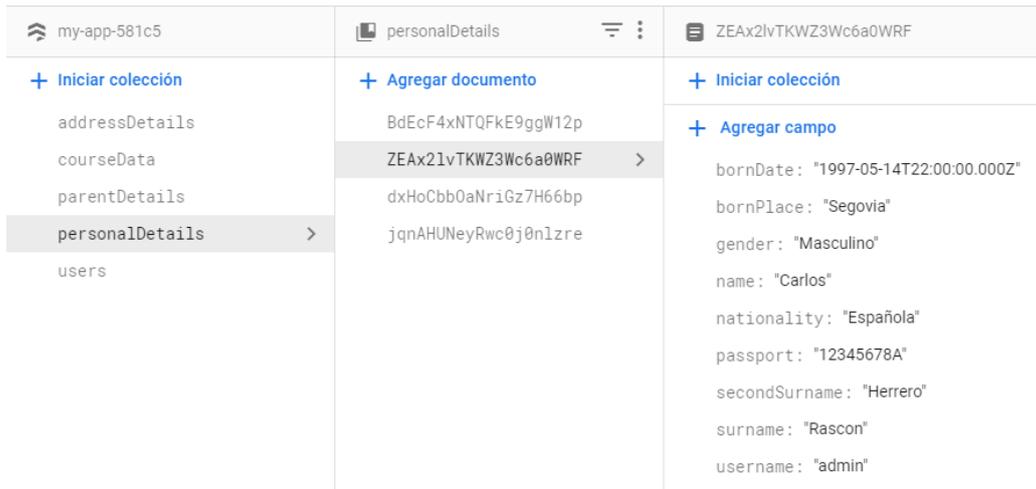


Figura 14: Estructura de documentos en base de datos

Observamos que todos los campos del documento se corresponden con los del modelo en Typescript exceptuando el campo "username" puesto que es el usamos para asociar el documento con el usuario, pero no lo necesitamos en la parte de cliente.

```

TS personal-details.ts X
src > app > register > classes > TS personal-details.ts > ...
1  export interface PersonalDetails {
2      name: string;
3      surname: string;
4      secondSurname: string;
5      bornDate: Date;
6      gender: ['Masculino', 'Femenino'];
7      bornPlace: string;
8      nationality: string;
9      passport: string;
10 }

```

Figura 13: Equivalencia de modelo de datos en Typescript

6.2.3. Diccionario de datos

En este apartado se van a especificar los requisitos de información definidos en otro punto de esta documentación. Cada uno de ellos está asociado a una colección o subcolección de las descritas en el modelo lógico de datos.

Identificador	Nombre
RI-C1	Usuarios
RI-C2	Datos personales
RI-C3	Datos parentales
RI-C4	Datos de dirección
RI-C5	Datos de matrícula

Tabla 38: Colecciones de datos

RI-C1	Usuarios					
Definición	Datos de los usuarios del sistema					
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C1.1	Email	Dirección de correo electrónico del usuario	String	Sí	No	

RI-C1.2	Name	Nombre del usuario	String	No	No	
RI-C1.3	Password	Contraseña de usuario	String	Sí	No	Guardada hasheada en base 64
RI-C1.4	Role	Rol del usuario	String	No	No	Puede ser administrador o usuario básico
RI-C1.5	Username	Nombre de usuario del usuario	String	Sí	No	

Tabla 39: Diccionario Datos-Colección "Usuarios"

RI-C2	Datos personales					
Definición	Datos personales del usuario relacionados a la matrícula					
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C2.1	Username	Nombre de usuario de usuario	String	Sí	No	FK
RI-C2.2	Name	Nombre del usuario	String	No	No	
RI-C2.3	Surname	Primer apellido	String	No	No	
RI-C2.4	SecondSurname	Segundo apellido	String	No	No	
RI-C2.5	BornDate	Fecha de nacimiento	String	No	No	
RI-C2.6	Gender	Sexo	String	No	No	Puede ser hombre o mujer
RI-C2.7	BornPlace	Lugar de nacimiento	String	No	No	
RI-C2.8	Passport	Dni o pasaporte	String	No	No	
RI-C2.9	Nationality	Nacionalidad	String	No	No	

Tabla 40: Diccionario Datos-Colección "Datos personales"

RI-C3	Datos parentales					
Definición	Datos de los progenitores del usuario relacionados con la matrícula					
ID	Nombre	Descripción	Dominio	Único	Null	Notas
RI-C3.1	Username	Nombre de usuario de usuario	String	Sí	No	FK
RI-C3.2	Parent1	Nombre y apellidos del primer progenitor	String	No	No	
RI-C3.3	Parent2	Nombre y apellidos del segundo progenitor	String	No	No	
RI-C3.4	Passport1	Dni o pasaporte del primer progenitor	String	Sí	No	
RI-C3.5	Passport2	Dni o pasaporte del segundo progenitor	String	Sí	No	
RI-C3.6	LiveInHouse1	Vive en el domicilio familiar el primer progenitor	String	No	No	Puede ser sí o no
RI-C3.7	LiveInHouse2	Vive en el domicilio familiar el segundo progenitor	String	No	No	Puede ser sí o no
RI-C3.8	PhoneNumber1	Número de teléfono del primer progenitor	String	No	No	
RI-C3.9	PhoneNumber2	Número de teléfono del segundo progenitor	String	No	No	
RI-C3.10	MobilePhoneNumber1	Número de teléfono móvil del primer progenitor	String	No	No	
RI-C3.11	MobilePhoneNumber2	Número de teléfono móvil del segundo progenitor	String	No	No	

RI-C3.12	Email1	Correo electrónico del primer progenitor	String	Sí	No	
RI-C3.13	Email2	Correo electrónico del segundo progenitor	String	Sí	No	

Tabla 41: Diccionario Datos-Colección "Datos parentales"

RI-C4	Datos de dirección					
Definición	Datos de la dirección del usuario relacionados con la matrícula					
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C4.1	Username	Nombre de usuario de usuario	String	Sí	No	FK
RI-C4.2	Street	Calle de la dirección	String	No	No	
RI-C4.3	Number	Número de portal	String	No	No	
RI-C4.4	Stair	Escalera	String	No	Sí	
RI-C4.5	Floor	Piso	String	No	No	
RI-C4.6	Letter	Letra	String	No	No	
RI-C4.7	City	Municipio	String	No	No	
RI-C4.8	Cp	Código postal	String	No	No	
RI-C4.9	Province	Provincia	String	No	No	
RI-C4.10	PhoneNumber	Número de teléfono	String	No	No	
RI-C4.11	EmergencyNumber	Número de teléfono de emergencias	String	No	No	

Tabla 42: Diccionario Datos-Colección "Datos de dirección"

RI-C5	Datos de matrícula					
Definición	Datos de la matrícula del usuario					
Consideraciones	Se usan distintos modelos de datos para cada curso, con algunos campos en común					
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C5.1	Username	Nombre de usuario de usuario	String	Sí	No	FK
RI-C5.2	Course	Curso seleccionado	String	No	No	
RI-C5.3	PreviousCollege	Instituto previo	String	No	No	Puede ser el que se ofrece por defecto o el que escriba el usuario
RI-C5.4	RepeatCourse	Repite curso	String	No	No	Puede ser sí o no

Tabla 43: Diccionario Datos-Colección "Datos de matrícula"

A continuación, se expone el modelo de datos para cada uno de los cursos:

RI-C5.5	Primer curso					
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C5.5.1	FirstOption	Primera optativa	String	No	No	
RI-C5.5.2	SecondOption	Segunda optativa	String	No	No	
RI-C5.5.3	ThirdOption	Tercera optativa	String	No	No	

Tabla 44: Diccionario Datos-Colección "Primer curso"

RI-C5.5	Segundo curso					
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C5.5.1	FirstOption	Primera optativa	String	No	No	
RI-C5.5.2	SecondOption	Segunda optativa	String	No	No	

Tabla 45: Diccionario Datos-Colección "Segundo curso"

RI-C5.5	Tercer curso					
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C5.5.1	FirstOption	Primera optativa	String	No	No	
RI-C5.5.2	SecondOption	Segunda optativa	String	No	No	
RI-C5.5.3	ThirdOption	Tercera optativa	Array de string	No	No	Contiene las optativas seleccionadas por orden de preferencia

Tabla 46: Diccionario Datos-Colección "Cuarto curso"

RI-C5.5	Cuarto curso					
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C5.5.1	Branch	Rama	String	No	No	
RI-C5.5.2	BranchOptions	Optativas de la rama seleccionada	Array de string	No	No	
RI-C5.5.3	Optionals	Optativas	Array de string	No	No	Contiene las optativas seleccionadas por orden de preferencia

Tabla 47: Diccionario Datos-Colección "Cuarto curso"

RI-C5.5	Quinto curso					
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C5.5.1	Branch	Rama	String	No	No	
RI-C5.5.2	BranchOptions	Optativas de la rama seleccionada	Array de string	No	No	
RI-C5.5.3	Optionals	Optativas	Array de string	No	No	Contiene las optativas seleccionadas por orden de preferencia

Tabla 48: Diccionario Datos-Colección "Quinto curso"

RI-C5.5	Sexto curso					
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C5.5.1	Branch	Rama	String	No	No	
RI-C5.5.2	BranchOptions	Optativas de la rama seleccionada	Array de string	No	No	
RI-C5.5.3	FirstOptional	Primera optativa	Array de string	No	No	Contiene las optativas seleccionadas por orden de preferencia
RI-C5.5.4	SecondOptional	Segunda optativa	Array de string	No	No	Contiene las optativas seleccionadas por orden de preferencia

Tabla 49: Diccionario Datos-Colección "Sexto curso"

6.3. Diseño de interfaces

En este apartado se van a explicar cada una de las interfaces de la aplicación:

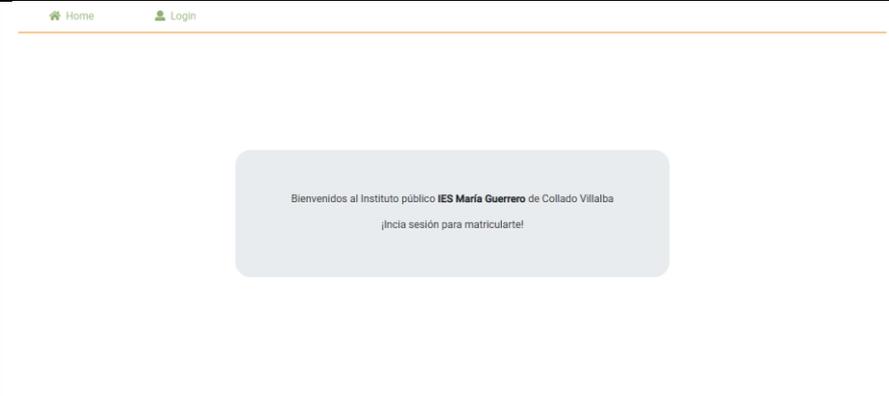
Pantalla principal	
Descripción	Pantalla inicial de la aplicación
Activación	Al entrar a la aplicación
Eventos	<ul style="list-style-type: none"> • Ir a la pantalla de login • Cerrar sesión (si estás logado) • Ir a la página de datos personales (si estás logado) • Ir a la página de administrador (si eres administrador)
Diseño	

Tabla 50: Interfaz "Pantalla principal"

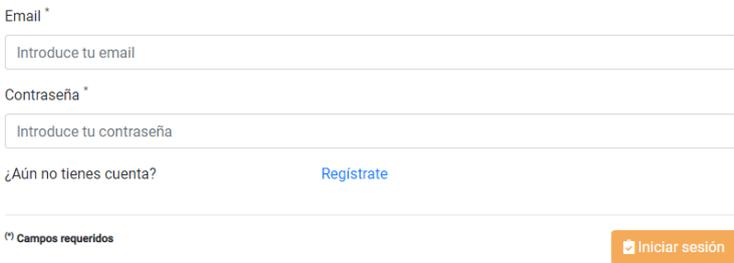
Pantalla de login	
Descripción	Pantalla para iniciar sesión en la aplicación
Activación	Pulsando el botón de "Login" en el menú superior
Eventos	<ul style="list-style-type: none"> • Acceder a la aplicación logado • Ir a la página de registro • Volver a la pantalla inicial
Diseño	 <p>The screenshot shows a login form with the following elements: <ul style="list-style-type: none"> An 'Email' field with a placeholder 'Introduce tu email'. A 'Contraseña' field with a placeholder 'Introduce tu contraseña'. A link '¿Aún no tienes cuenta? Regístrate'. A footer note '(*) Campos requeridos'. An orange button labeled 'Iniciar sesión'. </p>

Tabla 51: Interfaz "Pantalla de login"

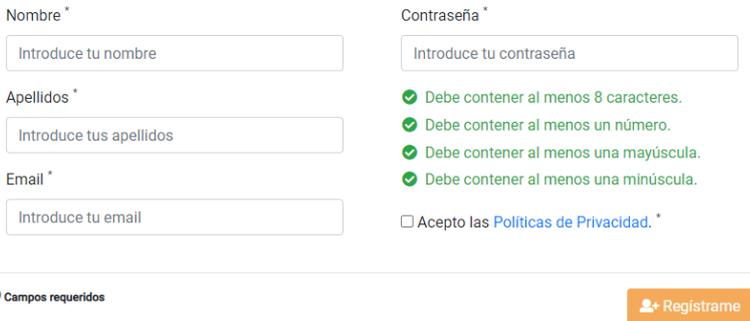
Pantalla de registro	
Descripción	Pantalla para crear una nueva cuenta en la aplicación
Activación	Pulsando el botón de "Regístrate" en la pantalla de login
Eventos	<ul style="list-style-type: none"> • Ir a la pantalla de inicio con un nuevo usuario creado • Volver a la pantalla inicial • Ir al login
Diseño	 <p>The screenshot shows a registration form with the following elements: <ul style="list-style-type: none"> 'Nombre' field with placeholder 'Introduce tu nombre'. 'Apellidos' field with placeholder 'Introduce tus apellidos'. 'Email' field with placeholder 'Introduce tu email'. 'Contraseña' field with placeholder 'Introduce tu contraseña'. Three password requirements: <ul style="list-style-type: none"> ✓ Debe contener al menos 8 caracteres. ✓ Debe contener al menos un número. ✓ Debe contener al menos una mayúscula. ✓ Debe contener al menos una minúscula. A checkbox 'Acepto las Políticas de Privacidad'. A footer note '(*) Campos requeridos'. An orange button labeled 'Regístrate'. </p>

Tabla 52: Interfaz "Pantalla de registro"

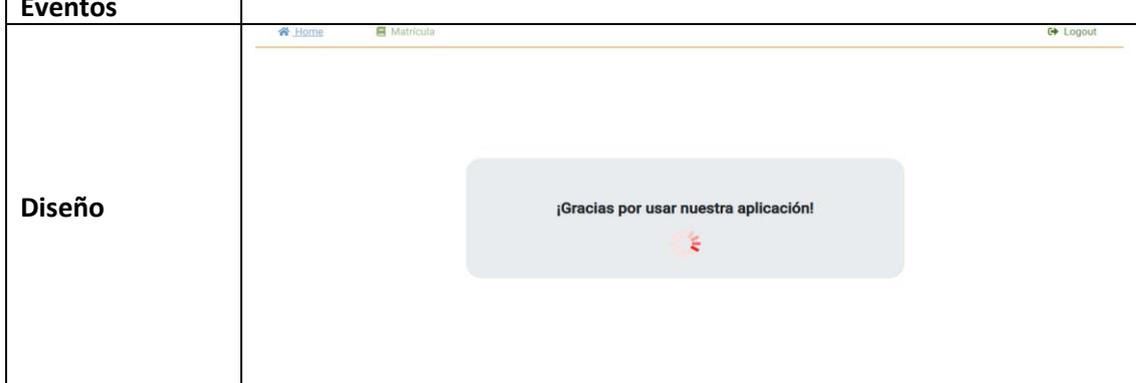
Pantalla de logout	
Descripción	Pantalla para cerrar sesión en la aplicación
Activación	Pulsando el botón de “Logout” en el menú superior
Eventos	
Diseño	

Tabla 53: Interfaz "Pantalla de logout"

Pantalla del formulario personal	
Descripción	Pantalla para rellenar o modificar los datos personales
Activación	Pulsando el botón de “Matrícula” en el menú superior
Eventos	<ul style="list-style-type: none"> • Avanzar a la página de datos de tutores legales del alumno • Volver a la página principal • Ir a la página de cerrar sesión
Diseño	

Tabla 54: Interfaz "Pantalla del formulario personal"

Pantalla del formulario de datos de tutores legales																																	
Descripción	Pantalla para rellenar o modificar los datos de los tutores legales del alumno																																
Activación	Rellenando los datos personales y pulsando el botón de “Siguiente”																																
Eventos	<ul style="list-style-type: none"> • Avanzar a la página de datos de dirección del alumno • Volver a la página de datos personales • Volver a la página principal • Ir a la página de cerrar sesión 																																
Diseño	<p>Importar matrícula: Seleccionar archivo Ningún archivo seleccionado</p> <p>Datos Tutores legales</p> <table border="1"> <tr> <td colspan="2">Nombre apellidos del TUTOR 1</td> <td colspan="2">Nombre apellidos del TUTOR 2</td> </tr> <tr> <td colspan="2">Parent1</td> <td colspan="2">Parent2</td> </tr> <tr> <td>DNI/NIE/Pasaporte</td> <td>¿Vive en el domicilio familiar?</td> <td>DNI/NIE/Pasaporte</td> <td>¿Vive en el domicilio familiar?</td> </tr> <tr> <td>123456789</td> <td><input checked="" type="radio"/> Sí <input type="radio"/> No</td> <td>123456789</td> <td><input checked="" type="radio"/> Sí <input type="radio"/> No</td> </tr> <tr> <td>Teléfono TUTOR 1</td> <td>Móvil TUTOR 1</td> <td>Teléfono TUTOR 2</td> <td>Móvil TUTOR 2</td> </tr> <tr> <td>987654321</td> <td>123456789</td> <td>987654321</td> <td>123456789</td> </tr> <tr> <td colspan="2">Email TUTOR 1</td> <td colspan="2">Email TUTOR 2</td> </tr> <tr> <td colspan="2">parent1@mail.com</td> <td colspan="2">parent2@mail.com</td> </tr> </table> <p style="text-align: right;"> ← Atrás Siguiente → </p>	Nombre apellidos del TUTOR 1		Nombre apellidos del TUTOR 2		Parent1		Parent2		DNI/NIE/Pasaporte	¿Vive en el domicilio familiar?	DNI/NIE/Pasaporte	¿Vive en el domicilio familiar?	123456789	<input checked="" type="radio"/> Sí <input type="radio"/> No	123456789	<input checked="" type="radio"/> Sí <input type="radio"/> No	Teléfono TUTOR 1	Móvil TUTOR 1	Teléfono TUTOR 2	Móvil TUTOR 2	987654321	123456789	987654321	123456789	Email TUTOR 1		Email TUTOR 2		parent1@mail.com		parent2@mail.com	
Nombre apellidos del TUTOR 1		Nombre apellidos del TUTOR 2																															
Parent1		Parent2																															
DNI/NIE/Pasaporte	¿Vive en el domicilio familiar?	DNI/NIE/Pasaporte	¿Vive en el domicilio familiar?																														
123456789	<input checked="" type="radio"/> Sí <input type="radio"/> No	123456789	<input checked="" type="radio"/> Sí <input type="radio"/> No																														
Teléfono TUTOR 1	Móvil TUTOR 1	Teléfono TUTOR 2	Móvil TUTOR 2																														
987654321	123456789	987654321	123456789																														
Email TUTOR 1		Email TUTOR 2																															
parent1@mail.com		parent2@mail.com																															

Tabla 55: Interfaz "Pantalla del formulario de datos de tutores legales"

Pantalla del formulario de datos de dirección																					
Descripción	Pantalla para rellenar o modificar los datos de la dirección del alumno																				
Activación	Rellenando los datos de los tutores legales y pulsando el botón de “Siguiente”																				
Eventos	<ul style="list-style-type: none"> • Avanzar a la página de datos de matrícula del alumno • Volver a la página de datos de tutores legales • Volver a la página principal • Ir a la página de cerrar sesión 																				
Diseño	<p>Importar matrícula: Seleccionar archivo Ningún archi... seleccionado</p> <p>Domicilio del alumno</p> <table border="1"> <tr> <td>Calle</td> <td>Nº</td> <td>Esc.</td> <td>Piso</td> <td>Letra</td> </tr> <tr> <td>Street</td> <td>10</td> <td>2</td> <td>5</td> <td>D</td> </tr> <tr> <td>Municipio</td> <td>C.P.</td> <td>Provincia</td> <td>Teléfono</td> <td>Teléfono urgencias</td> </tr> <tr> <td>Collado Villalba</td> <td>28062</td> <td>Madrid</td> <td>123456789</td> <td>987654321</td> </tr> </table> <p style="text-align: right;"> ← Atrás Siguiente → </p>	Calle	Nº	Esc.	Piso	Letra	Street	10	2	5	D	Municipio	C.P.	Provincia	Teléfono	Teléfono urgencias	Collado Villalba	28062	Madrid	123456789	987654321
Calle	Nº	Esc.	Piso	Letra																	
Street	10	2	5	D																	
Municipio	C.P.	Provincia	Teléfono	Teléfono urgencias																	
Collado Villalba	28062	Madrid	123456789	987654321																	

Tabla 56: Interfaz "Pantalla del formulario de datos de dirección"

Pantalla del formulario de datos de matrícula	
Descripción	Pantalla para rellenar o modificar los datos de la matrícula del alumno
Activación	Rellenando los datos de dirección y pulsando el botón de “Siguiete”
Eventos	<ul style="list-style-type: none"> • Finalizar matrícula y generar pdf • Volver a la página de datos de dirección • Volver a la página principal • Ir a la página de cerrar sesión
Diseño	

Tabla 57: Interfaz "Pantalla del formulario de datos de matrícula"

Página de datos de optativas	
Descripción	Página inicial de la sección de administrador en la que se pueden generar datos relativas a los cursos, optativas y repetidores
Activación	Pulsando el botón de “Admin” en el menú superior
Eventos	<ul style="list-style-type: none"> • Volver a la pantalla principal • Ir a la página de datos personales • Cerrar sesión • Generar datos de optativas • Ir a la página de generación de grupos
Diseño	

Tabla 58: Interfaz "Página de datos de optativas"

Página de generación de grupos	
Descripción	Pantalla para generar los grupos por curso
Activación	Pulsando la opción "generación de grupos" en el menú superior de la página de administrador
Eventos	<ul style="list-style-type: none"> • Generar grupos de un curso • Ir a la página de datos de optativas • Ir a la página de datos personales • Volver a la página principal • Ir a la página de cerrar sesión
Diseño	

Tabla 59: Interfaz "Página de generación de grupos"

Capítulo 7.

Implementación.

7. Implementación

Este apartado recoge distintos detalles propios de la implementación que son importantes para comprender mejor el proyecto. Incluye detalles concretos de alguna parte de la aplicación, como el servidor de base de datos utilizado, o la distribución de las carpetas del proyecto, para facilitar la búsqueda dentro de este.

Los subapartados en los que se divide son:

- **Consideraciones de implementación:** se tratan cuestiones propias de la implementación que se consideran importantes para entender la aplicación.
- **Servidores de bases de datos:** se explica el modelo de base de datos que se ha utilizado.
- **Organización interna del proyecto:** se explica la estructura de carpetas que se ha seguido en el proyecto que se ha desarrollado.

7.1. Consideraciones de implementación

En esta sección se va a explicar algunas cuestiones importantes referentes a la implementación de la aplicación.

Lo primero a tener en cuenta es la autenticación, una de las partes más importantes y potencialmente atacables de una aplicación web. En nuestro caso el método usado ha sido JWT (Json Web Token) que es un estándar abierto basado en JSON para la creación de tokens de acceso que permiten la propagación de identidad y privilegios o claims en inglés.

Estos tokens son generados desde el Back-End cuando el usuario inicia sesión y son almacenados en el local storage del navegador para autenticar las llamadas a la Web Api desde el cliente. Si alguna de las llamadas que llegan al BackEnd no lleva token o lleva un token inválido o caducado, el usuario es desautenticado de la aplicación.

Otra de las funcionalidades que cabe destacar es la importación de la matrícula mediante un SVG. Con un archivo SVG con el formato correcto, leemos los datos y los asignamos a la propiedad de los distintos modelos que componen el total de la matrícula y que son guardados en la base de datos por separado. Esta funcionalidad está hecha totalmente desde la parte de front end.

La última funcionalidad que cabe destacar es la generación de grupos para cada curso. En primer lugar, esta funcionalidad es puramente del back end. El proceso que se lleva a cabo es tan simple como calcular el número aproximado de grupos que vamos a generar partiendo del número total de alumnos y calculando grupos de 30.

Tras esto, se recorren cada una de las matrículas de los alumnos de ese curso y se incluyen en el grupo con el que más optativas tengan en común que no esté lleno.

7.2. Servidores de bases de datos utilizados

En nuestro caso, como ya se ha mencionado antes, se ha decidido utilizar el servicio **“Cloud Firestore”** (Firebase).

Se ha decidido utilizar esta opción por tener una estructura de base de datos no SQL, puesto que nos permite almacenar los datos sin tener que establecer una estructura marcada para cada tipo de dato. Esto nos proporciona una gran flexibilidad e independencia a la hora de gestionar los datos.

En este servicio los datos se almacenan en estructuras llamadas **“colecciones”**, que a su vez contienen registros llamados **“documento”**, que es dónde almacenan realmente los datos.

Los documentos de una colección pueden tener distintos campos, lo que nos permite, en el caso de este proyecto, almacenar las matrículas de diferentes cursos en una misma colección, teniendo cada curso diferente número de optativas o ramas.

Otro de los motivos por los que se ha decidido utilizar este servicio es la facilidad para integrarlo con nuestro código de servidor, en Node Js.

7.3. Organización interna del proyecto

En esta sección se describe la estructura de nuestro proyecto, es decir, la distribución de los archivos en carpetas, para facilitar la comprensión y localización de cada una de las funcionalidades.

La estructura se compone de:

- **/Gomi:** Este directorio contiene archivos necesarios para ejecutar la aplicación, los paquetes de librerías usadas en la aplicación y los dos subdirectorios que contienen las partes de cliente y servidor del proyecto.
- **/Gomi/server:** Directorio principal de la parte de servidor. Se compone de varios subdirectorios:
 - **/Gomi/server/routes:** Define las rutas de la Web Api y a que controlador deben apuntar. Además, en esta sección se definen los *middlewares* que se usan para cada una de las rutas.
 - **/Gomi/server/api:** En esta sección están los controladores de la Web Api, que contienen el código que se ejecuta cuando llamamos a alguna de las rutas.
 - **/Gomi/server/middlewares:** Contiene los *middlewares* que se usan para las distintas rutas.
 - **/Gomi/server/model:** Contiene los modelos usados en la Web Api.
 - **/Gomi/server/database:** Contiene el código que se usa para conectar e interactuar con los datos de la base de datos.
 - **/Gomi/server/services:** Contiene servicios que usamos como apoyo para los controladores con acciones necesarias para la Web Api.
- **/Gomi/src:** Contiene la parte de cliente de nuestro proyecto. Se compone de varios subdirectorios:
 - **/Gomi/src/assets:** Contiene las imágenes y otros archivos que se usan a lo largo de la aplicación web.
 - **/Gomi/src/api:** Contiene los servicios Angular que realizan las llamadas *Http* a la Web Api.
 - **/Gomi/src/app:** Contiene el resto de los elementos de la aplicación web. Este directorio se divide en:
 - **/Gomi/src/app/common:** Contiene elementos comunes de la aplicación, tanto modelos, como servicios y otros elementos.
 - **/Gomi/src/app/admin:** Contiene todo lo relacionado con la parte de administrador, tanto vistas en *html* como lógica necesaria para el correcto funcionamiento.
 - **/Gomi/src/app/login:** Contiene todo lo relacionado a la parte del formulario de login de la aplicación.
 - **/Gomi/src/app/logout:** Contiene todo lo relacionado a la parte del logout de la aplicación.
 - **/Gomi/src/app/register-user:** Contiene todo lo relacionado con la parte del formulario de registro de la aplicación.
 - **/Gomi/src/app/welcome:** Contiene la página principal de la aplicación.
 - **/Gomi/src/app/register:** Es la parte más extensa de este directorio. Contiene todos los formularios de la matrícula de los alumnos. Además de servicios que se usan a lo largo del proceso de matriculación.

Capítulo 8.

Pruebas.

8. Pruebas

Esta sección incluye las pruebas que se han realizado para garantizar el perfecto funcionamiento de la aplicación, tanto las de caja blanca, usadas para validar las funciones internas de los módulos, como las de caja negra, dirigidas a validar las salidas producidas por el sistema.

Los subapartados en los que se divide esta sección son:

- **Pruebas de caja blanca:** se enumeran y explican las pruebas de caja blanca realizadas durante el desarrollo de la aplicación.
- **Pruebas de caja negra:** se enumeran y explican algunas de las pruebas de caja negra que se han utilizado para asegurar el correcto funcionamiento de la aplicación.

8.1. Pruebas de Caja Blanca

El objetivo es validar las funciones internas de los módulos o subprogramas que se van a probar.

El funcionamiento de estas pruebas consiste en coger porciones de código y dividirlos en bloques, es decir cada bloque se trata como un proceso independiente y que se realiza de manera secuencial. Con esto se consigue rastrear todos los flujos de actuación que puede adoptar una aplicación web para asegurarnos de que realiza las acciones de la forma esperada.

Para este proyecto, estas pruebas se han llevado a cabo durante todo el proceso de implementación de este. Algunas de las más importantes son:

- Comprobación de las rutas de la Web Api.
- Comprobación del funcionamiento del sistema en pantallas de tamaño reducido.
- Comprobación de casos de error controlados.
- Comprobación de las llamadas a base de datos.
- Comprobación de las distintas funcionalidades de la aplicación.
- Comprobación de imposibilidad de acceder a datos que requieran de sesión sin haberla iniciado.

8.2. Pruebas de Caja Negra

La intención de estas pruebas es validar respuestas o salidas que el sistema produce al introducirle unas determinadas entradas. No atienden al funcionamiento interno del código con lo que pueden ser realizadas por usuarios finales.

Para nuestro proyecto las entradas de usuario son la parte clave, por lo que las pruebas de caja negra han sido una de las herramientas más útil para comprobar el correcto funcionamiento del proyecto.

PCN-01: Inicio de sesión	
Propósito	Comprobar que el formulario de login funciona correctamente
Prerequisitos	Tener una cuenta en la aplicación web GOMI
Datos de entrada	Nombre de usuario y contraseña
Resultado esperado	Al pulsar el botón "Iniciar sesión" el sistema muestra un mensaje de éxito y el usuario tiene una sesión en la aplicación
Resultado obtenido	Correcto

Tabla 60: Prueba Caja Negra "Inicio de sesión"

PCN-02: Registro	
Propósito	Comprobar que el formulario de registro funciona
Prerequisitos	Ninguno
Datos de entrada	Datos del nuevo usuario
Resultado esperado	El usuario tendrá una nueva cuenta creada y podrá acceder con los datos introducidos
Resultado obtenido	Correcto

Tabla 61: Prueba Caja Negra "Registro"

PCN-03: Filtrado de matrículas por curso	
--	--

Propósito	Comprobar que los datos ofrecidos a los administradores para el filtrado por curso funcionan correctamente
Prerequisitos	Tener cuenta de administrado en la aplicación
Datos de entrada	Selección de la opción "Por curso" en el formulario de filtrado
Resultado esperado	Se muestra un gráfico con el número de alumnos por curso
Resultado obtenido	Correcto

Tabla 62: Prueba Caja Negra "Filtrado de matrículas por curso"

PCN-04: Filtrado de matrículas por optativa de un determinado curso	
Propósito	Comprobar que los datos ofrecidos a los administradores para el filtrado por optativa de un curso funcionan correctamente
Prerequisitos	Tener cuenta de administrado en la aplicación
Datos de entrada	Curso y optativa seleccionado para el filtrado
Resultado esperado	Se muestra un gráfico con el número de alumnos de cada opción para ese curso y optativa
Resultado obtenido	Correcto

Tabla 63: Prueba Caja Negra "Filtrado de matrículas por optativa de un determinado curso"

PCN-05: Filtrado de matrículas que repiten curso	
Propósito	Comprobar que los datos ofrecidos a los administradores para el filtrado por alumnos que repiten curso funcionan correctamente
Prerequisitos	Tener cuenta de administrado en la aplicación
Datos de entrada	Curso y opción "Repetidores"
Resultado esperado	Se muestra un gráfico con el número de alumnos que repiten curso
Resultado obtenido	Correcto

Tabla 64: Prueba Caja Negra "Filtrado de matrículas que repiten curso"

PCN-06: Generación de grupos por curso	
Propósito	Comprobar que la generación de grupos funciona correctamente
Prerequisitos	Tener cuenta de administrado en la aplicación
Datos de entrada	El grupo para el que se quiere generar los grupos
Resultado esperado	Se muestra una tabla con los alumnos que pertenecen a cada grupo y sus optativas seleccionadas
Resultado obtenido	Correcto

Tabla 65: Prueba Caja Negra "Generación de grupos por curso"

PCN-07: Formulario de datos personales	
--	--

Propósito	Comprobar que el formulario de datos personales funciona correctamente
Prerequisitos	Tener la sesión iniciada en la aplicación
Datos de entrada	Datos personales del alumno
Resultado esperado	Los datos se almacenan y se muestran rellenos la próxima vez que se acceda al formulario
Resultado obtenido	Correcto

Tabla 66: Prueba Caja Negra "Formulario de datos personales"

PCN-08: Formulario de datos parentales	
Propósito	Comprobar que el formulario de datos parentales funciona correctamente
Prerequisitos	Tener la sesión iniciada en la aplicación
Datos de entrada	Datos de los progenitores del alumno
Resultado esperado	Los datos se almacenan y se muestran rellenos la próxima vez que se acceda al formulario
Resultado obtenido	Correcto

Tabla 67: Prueba Caja Negra "Formulario de datos parentales"

PCN-09: Formulario de datos de dirección	
Propósito	Comprobar que el formulario de datos de dirección funciona correctamente
Prerequisitos	Tener la sesión iniciada en la aplicación
Datos de entrada	Datos de la dirección del alumno
Resultado esperado	Los datos se almacenan y se muestran rellenos la próxima vez que se acceda al formulario
Resultado obtenido	Correcto

Tabla 68: Prueba Caja Negra "Formulario de datos de dirección"

PCN-10: Formulario de matrícula	
Propósito	Comprobar que el formulario de matrícula funciona correctamente
Prerequisitos	Tener la sesión iniciada en la aplicación
Datos de entrada	El curso y las opciones que ha elegido el alumno
Resultado esperado	Los datos se almacenan y se muestran rellenos la próxima vez que se acceda al formulario
Resultado obtenido	Correcto

Tabla 69: Prueba Caja Negra "Formulario de matrícula"

PCN-11: El formulario de registro no acepta un email incorrecto	
Propósito	Comprobar que el formulario de registro no acepta un formato incorrecto
Prerequisitos	
Datos de entrada	Un email invalido
Resultado esperado	El sistema muestra un mensaje diciendo que el formato es incorrecto
Resultado obtenido	Correcto

Tabla 70: Prueba Caja Negra "El formulario de registro no acepta un email incorrecto"

PCN-12: El usuario no puede acceder con claves incorrectas	
Propósito	El usuario no puede acceder con claves incorrectas
Prerequisitos	
Datos de entrada	El usuario o contraseña incorrectos
Resultado esperado	El sistema muestra un mensaje diciendo que el login es incorrecto y no deja acceder a la aplicación
Resultado obtenido	Correcto

Tabla 71: Prueba Caja Negra "El usuario no puede acceder con claves incorrectas"

PCN-13: El sistema no puede avanzar en el registro sin todos los datos rellenos	
Propósito	Comprobar que hay que rellenar todos los datos para avanzar en el formulario
Prerequisitos	Tener la sesión iniciada en la aplicación
Datos de entrada	Todos los datos de uno de los formularios menos uno
Resultado esperado	El botón de avanzar está desactivado y no se puede acceder al siguiente formulario
Resultado obtenido	Correcto

Tabla 72: Prueba Caja Negra "El sistema no puede avanzar en el registro sin todos los datos rellenos"

Cómo ya se ha mencionado anteriormente, las pruebas descritas representan sólo una parte de la totalidad de las realizadas, ya que el funcionamiento de las características no incluidas es similar, por lo que las pruebas y resultados obtenidos no aportan nada nuevo.

Capítulo 9.

Conclusiones.

9. Conclusiones y posibles mejoras

Este apartado recoge las conclusiones sacadas tras la realización de este proyecto, además de una serie de posibles mejoras sobre la aplicación para abordar, si es posible, en el futuro.

Los subapartados de los que consta esta sección son:

- **Conclusiones:** se explican las conclusiones que se han sacado al término del proyecto.
- **Posibles mejoras:** se recogen una serie de posibles mejoras que se podrían llevar a cabo en un futuro partiendo de la aplicación desarrollada.

9.1. Conclusiones

Para mí, este trabajo ha significado el primer proyecto completo que he realizado de manera individual, puesto que, aunque llevo tiempo trabajando, siempre lo he hecho en proyectos ya empezados o en los que trabajábamos en equipo.

Por otra parte, me ha servido para afianzar conocimientos sobre lenguajes de programación y *frameworks* que ya había utilizado anteriormente, pero, sobre todo, para conocer nuevas herramientas y tecnologías. La tecnología más novedosa y de la que más he aprendido es *Firebase*, puesto que nunca había utilizado bases de datos no relacionales y creo que son una herramienta muy útil de la que en un futuro podré sacar mucho provecho y me ayudará a desarrollar otros proyectos a nivel personal y laboral. Otra de las cosas positivas que me ha aportado ha sido el tener que trabajar más con la parte de diseño de estilos en *CSS*, puesto que siempre me había resultado muy complicado y había profundizado poco en esta tecnología.

Tengo que destacar que la realización de este proyecto ha sido muy compleja ya que se ha desarrollado al mismo tiempo que trabajaba, la compatibilidad de horarios ha sido muy complicada y no he dispuesto de todo el tiempo que me hubiese gustado para desarrollarlo en menos tiempo. A todo esto, también tengo que añadir la realización de varios proyectos a nivel personal, lo que ha supuesto un gran esfuerzo y ha hecho que me sintiera sobrepasado en algunos momentos por la demora en el tiempo de la finalización y la entrega de la aplicación.

Por último, he de añadir que llevar a cabo el desarrollo de este proyecto me ha traído muchas sensaciones y experiencias positivas por lo que a nivel general me siento satisfecho del resultado final y de haber sido capaz de realizar algo de estas características de principio a fin por mis propios medios.

9.2. Posibles mejoras

El proyecto en un principio podía parecer simple, pero a medida que se ha ido avanzando se han pedido nuevas funcionalidades que han aumentado la complejidad de este. Algunas de ellas han tenido que ser rechazadas por falta de tiempo o por tener una dificultad demasiado grande y han sido recogidas en esta sección.

Algunas de las posibles mejoras de este proyecto son:

- Coordinación con la base de datos del centro de enseñanza peticionario para evitar a los alumnos tener que crear una cuenta. Esto no se ha hecho por falta de medios en el centro.
- Añadir histórico de matrículas de los alumnos, coordinándose de nuevo con la base de datos del centro.
- Posibilidad de poder incorporar a los profesores del centro y poder asignar optativas o cursos a cada uno de ellos.
- La más ambiciosa y difícil es la generación de horarios partiendo de las matrículas y los profesores del centro, lo que me parecía demasiado ambicioso para este proyecto.
- Envío del archivo *PDF* con la matrícula por correo electrónico a los alumnos al finalizar el peticionario proceso.

10. Webgrafía

- **Angular.io:** sitio oficial del *framework* en *typescript* conocido como *Angular*. Recoge toda la documentación del *framework* además de tener un blog con ejemplos y artículos muy útiles a la hora de desarrollar. Último acceso: 20/01/2022
Disponible en: <https://angular.io/>
- **Angular Google charts:** sitio oficial de la librería de *Angular* “Google charts”, contiene la documentación de esta librería. Último acceso: 19/11/2021
Disponible en: <https://www.npmjs.com/package/angular-google-charts>
- **Angular material:** sitio oficial de la librería “Angula material” utilizada en el proyecto. Último acceso: 06/12/2021
Disponible en: <https://material.angular.io/>
- **Cloud firestore (Firebase):** documentación relativa a la base de datos usada en nuestro sistema. Último acceso: 06/12/2021
Disponible en: <https://firebase.google.com/docs/firestore>
- **Express Js:** sitio oficial del *framework* “Express Js” que recoge toda la documentación y ejemplos. Último acceso: 10/06/2021
Disponible en: <https://expressjs.com/>
- **Stack overflow:** sitio web de preguntas y respuestas para programadores profesionales y aficionados. Último acceso: 10/01/2022
Disponible en: <https://stackoverflow.com/>
- **W3Schools:** sitio web que recoge documentación de todo tipo de lenguajes de programación y *frameworks*. Último acceso: 25/09/2021
Disponible en: <https://www.w3schools.com/>
- **Desarrollo iterativo e incremental:** blog de Jmbeas. Último acceso 11/11/2021
Disponible en: <https://blog.jmbeas.es/2019/09/22/desarrollo-iterativo-e-incremental/>
- **Método Albrecht para el Análisis de los Puntos Función:** Manuel Cillero. Último acceso 21/11/2021
Disponible en: <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/tecnicas-de-estimacion/metodo-albrecht/>