

# Memoria del Proyecto

## Sistema de Gestión de Deberes en Centros de Primaria y Secundaria

Trabajo de Fin de Grado  
INGENIERÍA INFORMÁTICA



**VNiVERSiDAD  
D SALAMANCA**

**Septiembre de 2022**

**Autor:**

*Luis Martín Hernández*

**Tutores:**

*André Filipe Sales Mendes*

*Juan Francisco de Paz Santana*

*Gabriel Villarubia González*

## Memoria del Proyecto

# Certificado de los Tutores

D. André Filipe Sales Mendes, D. Gabriel Villarubia González y D. Juan Francisco de Paz Santana, profesores del Departamento de Informática y Automática de la Universidad de Salamanca.

HACEN CONSTAR:

Que el trabajo titulado “Sistema de Gestión de Deberes en Centros de Primaria y Secundaria” ha sido realizado por D. Luis Martín Hernández, con el número de documento 70918240V y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado de Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 1 de Septiembre de 2022.

D. André Filipe Sales  
Mendes

D. Gabriel Villarubia  
González

D. Juan Francisco de  
Paz Santana



## RESUMEN

---

Este proyecto consiste en el desarrollo de un sistema completo de Gestión de Deberes en Centros de Primaria y Secundaria, dando a los padres de los alumnos de dichos centros el conocimiento de los deberes a realizar por los hijos en cada una de las asignaturas.

Se puede considerar que el proyecto se divide en dos partes:

**Aplicación Web:** Es una aplicación para la administración de todo lo que interviene en el sistema, como pueden ser, alta y bajas de usuarios, alta y bajas de centros, altas y bajas de profesores, entre otros.

**Aplicación Móvil:** Es la aplicación que van a poseer tanto los profesores como los padres, madre o tutores legales de los alumnos. Desde donde los profesores podrán consultar los centros donde imparten clases, los cursos a los que imparten clases, las asignaturas que imparten, los alumnos a los que imparte clase, así como mandar y modificar los deberes de cada una de las asignaturas suyas. Los padres, madres o tutores podrán ver a que centros perteneces sus hijos, ya que, si tienen varios hijos, cada uno puede estar en uno diferente, sus asignaturas y sus deberes.

Se puede concluir que le objetivo de dicha aplicación en conjunto será desarrollar un sistema capaz de gestionar los deberes, tanto del profesor como de los hijos de los padres, para poder mejorar los resultados académicos de los alumnos, reduzca el fracaso académico haciendo que los alumnos hagan los deberes, teniendo los padres conocimientos de estos.

**Palabras Clave:** colegio, alumno, deberes, asignatura



## SUMMARY

---

This project consists of the development of a complete Homework Management system in Primary and Secondary Schools, giving the parents of the students of these centers the knowledge of the homework to be carried out by the children in each of the subjects.

The project can be considered to be divided into two parts:

**Web Application:** It is an application for the administration of everything that intervenes in the system, such as user registration and cancellation, center registration and cancellation, teacher registration and cancellation, among others.

**Mobile Application:** It is the application that both the teachers and the parents or legal guardians of the students will have. From where teachers will be able to consult the centers where they teach classes, the courses they teach, the subjects they teach, the students they teach, as well as send and modify the homework of each of their subjects. Parents or guardians will be able to see which centers their children belong to, since, if they have several children, each one can be in a different one, their subjects and their homework.

It can be concluded that the objective of this application as a whole will be to develop a system capable of managing the homework, both of the teacher and of the children of the parents, in order to improve the academic results of the students, reduce academic failure by making students do homework, having parents knowledge of these.

**Keywords:** school, student, homework, subject

## TABLA DE CONTENIDO

---

<b>RESUMEN</b> .....	5
<b>SUMMARY</b> .....	7
<b>1. INTRODUCCIÓN</b> .....	11
<b>2. ESTUDIO DEL MERCADO</b> .....	13
<b>3. OBJETIVOS</b> .....	14
3.1. Objetivos del Sistema .....	14
3.2. Objetivos Personales .....	15
<b>4. CONCEPTOS TEÓRICOS</b> .....	16
4.1. Api Rest.....	16
<b>5. TÉCNICAS Y HERRAMIENTAS</b> .....	17
5.1. Técnicas y Herramientas usadas en el servidor.....	17
5.1.1. Java .....	17
5.1.2. Framework Vaadin .....	17
5.1.3. Oracle Database.....	18
5.1.4. JSON .....	18
5.1.5. Spring Framework .....	19
5.1.6. Hibernate.....	19
5.1.7. PostMan .....	20
5.1.8. SqlDeveloper .....	21
5.1.9. Spring Tools Suite .....	21
5.1.10. Firebase .....	22
5.2. Técnicas y Herramientas usadas en la Aplicación Móvil .....	22
5.2.1. Android Studio.....	22
5.3. Herramientas CASE.....	22
5.3.1. Microsoft Project .....	22
5.3.2. Visual Paradigm.....	24
5.3.3. REM.....	24
5.3.4. EZEstimate .....	25
5.4. Herramientas de Generación de Documentación .....	25
5.5. Herramientas de Control de Versiones.....	26
5.5.1. GitHub .....	26
<b>6. ASPECTOS RELEVANTES DEL DESARROLLO</b> .....	27



## Memoria del Proyecto

6.1.	Marco de Trabajo.....	27
6.2.	Estimación de Duración del Proyecto .....	28
6.3.	Planificación Temporal .....	29
6.4.	Especificación de Requisitos .....	33
6.4.1.	Participantes.....	33
6.4.2.	Objetivos del Sistema .....	33
6.4.3.	Requisitos de Información.....	35
6.4.4.	Requisitos Funcionales.....	36
6.4.5.	Requisitos No Funcionales .....	39
6.5.	Análisis de Requisitos.....	39
6.5.1.	Modelo de Dominio .....	40
6.5.2.	Realización de Casos de Uso.....	40
6.5.3.	Clases de Análisis .....	41
6.5.4.	Vista Arquitectónica del Modelo de Análisis.....	41
6.6.	Diseño del Sistema .....	41
6.6.1.	Patrones Arquitectónicos .....	42
6.6.2.	Subsistemas de Diseño.....	43
6.6.3.	Clases de Diseño.....	44
6.6.4.	Vista Arquitectónica del Modelo de Diseño.....	45
6.6.5.	Realización de Casos de Uso de Diseño .....	45
6.6.6.	Modelo de Despliegue .....	46
6.7.	Implementación.....	47
6.8.	Pruebas.....	48
6.9.	Funcionalidades del Sistema.....	48
6.9.1.	Gestión de Usuarios.....	48
6.9.2.	Gestión de Personas.....	51
6.9.3.	Gestión de Centros .....	51
6.9.4.	Gestión de Cursos.....	53
6.9.5.	Gestión de Asignaturas .....	54
6.9.6.	Gestión de Alumnos.....	56
6.9.7.	Gestión de Deberes .....	57
6.9.8.	Gestión de Estadísticas.....	58
6.9.9.	Gestión de Profesores .....	59
6.9.10.	Gestión de Padremadretutor .....	59
<b>7.</b>	<b>LIMITACIONES DE LA APLICACIÓN.....</b>	<b>61</b>
<b>8.</b>	<b>CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO .....</b>	<b>62</b>

## Memoria del Proyecto

8.1. Conclusiones.....	62
8.2. Líneas Futuras de Trabajo.....	63
<b>9. REFERENCIAS.....</b>	<b>63</b>

# 1. INTRODUCCIÓN

---

Actualmente, vivimos en un país donde el fracaso escolar es de los más altos de Europa, concretamente el segundo país con mayor tasa de fracaso escolar, situándose con un 13,3% en 2021.

Gracias al avance de toda la tecnología y que hoy en día todo el mundo tiene un Smartphone, este problema se puede abordar mejor.

Para ello, se propone este proyecto para mejorar el rendimiento de los alumnos y así poder frenar este fracaso escolar y disminuirlo en una parte, haciendo que los padres sean conscientes de las tareas de sus hijos en el colegio y si cumplen o no con ellos.

Este proyecto tiene como objetivo desarrollar un sistema de deberes para centros de infantil e institutos de secundaria, teniendo un sistema de administración web y una aplicación móvil para el intercambio de deberes entre profesorado y padre, madre o tutores de alumnos.

Este documento tiene como fin explicar aquellos aspectos más importantes del desarrollo del proyecto, el del Trabajo de Fin de Grado.

Se va a estructurar de la siguiente manera:

- **Estado del arte:** Punto donde se mostrarán aplicaciones ya disponibles o las que más se asemejen que estén en el mercado.
- **Objetivos:** Sección donde se mostrarán los objetivos que se intentarán alcanzar con la realización del proyecto.
- **Conceptos Teóricos:** Se trata de un punto donde se hará una lista con todo detalle de conceptos que son necesarios para el entendimiento del proyecto.
- **Técnicas y Herramientas:** Documentación de las técnicas y herramientas utilizadas en la elaboración de este proyecto.
- **Aspectos relevantes del desarrollo:** Se van a recoger aquellos aspectos importantes en el desarrollo y despliegue de la aplicación.
- **Limitaciones de la aplicación:** Punto donde se explicarán las limitaciones técnicas que se han encontrado al final del proyecto.
- **Conclusiones y líneas de trabajo futuras:** Se harán comentarios acerca de las conclusiones obtenidas después de la finalización del proyecto y también, de evolutivos que se pueden llevar a cabo en un trabajo futuro de la aplicación.

Este documento está complementado con los siguientes documentos:

- **Anexo I - Planificación Temporal:** Documento donde se lleva a cabo una estimación de la duración del proyecto y una planificación temporal de las tareas a llevar a cabo.
- **Anexo II – Especificación de Requisitos:** Es el documento que tiene la especificación de requisitos del proyecto.

## Memoria del Proyecto

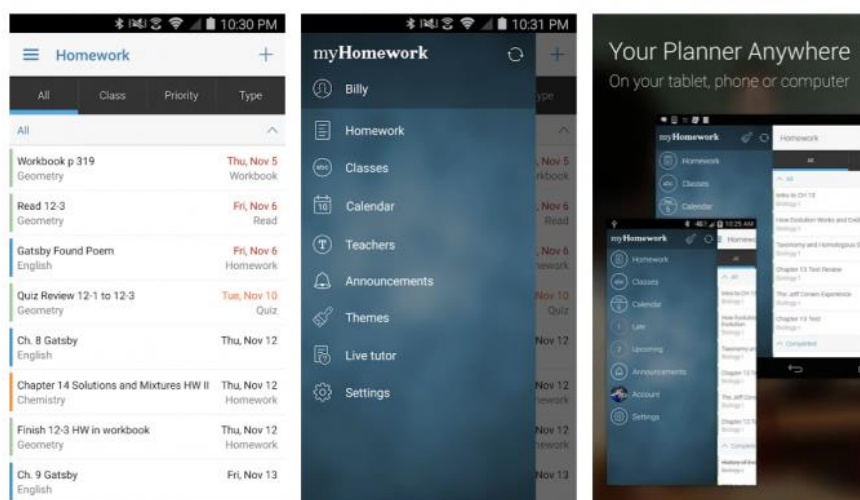
- **Anexo III – Análisis de Requisitos:** Documento que habla sobre la fase de análisis de los requisitos.
- **Anexo IV – Diseño del Sistema Software:** Contiene todo sobre la fase de diseño del sistema.
- **Anexo V – Documentación Técnica:** Documento que da al usuario facilidad para la comprensión del código desarrollado.
- **Anexo VI – Manual de Usuario:** Documento que recoge toda la funcionalidad del sistema detalladamente para que el usuario pueda utilizar la aplicación correcta y fácilmente.

## 2. ESTUDIO DEL MERCADO

Para llevar a cabo este proyecto, se ha hecho un pequeño estudio del mercado, haciendo una búsqueda de aplicaciones similares/parecidas a la que se iba a desarrollar, para así tener una visión de las necesidades de ellos usuarios y se pudiera orientar el proyecto mejor.

Aplicaciones:

- **Homework:** (instin) Esta aplicación, que se puede ver en la *Imagen 1*, ideal tanto para estudiantes como para profesores, es fantástica ya que permite llevar un registro de deberes diariamente: por materias, horarios, etc. Se puede distinguir que en nuestra aplicación se envían las notificaciones cuando los profesores mandan los deberes.



*Imagen 1: Aplicación Homework Instin*

- **HomeWork (Tarea):** ([Klwinkel.com](http://Klwinkel.com)) Esta aplicación es para dar seguimiento a las tareas de clase y a los horarios, como muestra la *Imagen 2*. Se pueden definir las materias/clases, las horas de inicio y fin de clases, permitir hacer cambio temporal de una clase en el horario. También hay que indicar que tiene que se todo hecho por el propio alumno y sin ninguna notificación.



*Imagen 2: Aplicación Homework (Tarea)*

### 3. OBJETIVOS

---

En este punto se van a describir los objetivos que debe cumplir el sistema para realizar el desarrollo del Trabajo de Fin de Grado. Se detallarán los objetivos del sistema, así como los objetivos personales.

#### 3.1. OBJETIVOS DEL SISTEMA

El principal objetivo del sistema es el diseño y desarrollo de una aplicación que permita gestionar los deberes en los centros de primaria y secundaria.

Los objetivos del sistema son:

- **Gestión de usuarios:** El sistema deberá manejar diferentes roles para cada usuario, algunos ejemplos son:
  - Usuario Administrador: Permite manejar la información del sistema, permitiendo modificarla.
  - Usuario Profesor: Permite visualizar toda la información generada por el sistema, así como crear nueva información.
  - Usuario Padre/Madre/Tutor: Permite solamente visualizar la información del sistema.
  - El sistema permitirá a los administradores a gestionar todo lo relacionado con el sistema.
- **Gestión de Centros:** El sistema deberá permitir crear, modificar o eliminar los centros. También se podrá listar los centros, tanto en la aplicación web como en la aplicación móvil. En la web se podrá hacer uso de los filtros de la pantalla index para ser más específico con la búsqueda.
- **Gestión de Cursos:** El sistema deberá permitir crear, modificar o eliminar los cursos. Se tendrá la posibilidad de listar los cursos en la aplicación web y en la aplicación móvil. En la web se tendrá filtros para poder afinar la búsqueda y en el móvil, se filtrarán según en que cursos da el profesor asignaturas o, en el caso de padre/madre/tutor, se hará un filtro automático según en que curso esté su hijo/a.
- **Gestión de Asignaturas:** El sistema deberá permitir crear, modificar o eliminar las asignaturas. Se podrán ver todas asignaturas en la web, teniendo la posibilidad de filtrar por campos de estas y en la aplicación móvil se listarán aquellas que el profesor imparta o que un alumno/a curse.
- **Gestión de Alumnos:** El sistema deberá permitir crear, modificar o eliminar alumnos. En la web se podrá filtrar por cualquiera de sus atributos y en la aplicación móvil se filtrará automáticamente por los alumnos que cursen una asignatura o por todos los alumnos a los que un profesor imparte alguna asignatura.
- **Gestión de Profesores:** El sistema deberá permitir crear, modificar o eliminar profesores. Solamente será posible de listar desde la web, usando los diversos filtros para ello o sin ninguno.
- **Gestión de Padre/Madre/Tutor:** El sistema deberá permitir crear, modificar o eliminar padres, madres o tutores.
- **Gestión de Estadísticas:** El sistema deberá poder generar automáticamente y mostrárselas a los usuarios con el rol "Profesor".
- **Gestión de Deberes:** El sistema deberá permitir crear, modificar y enviar notificaciones de deberes. Serán enviadas por los usuarios con rol "Profesor" y recibidas por usuarios con rol "Padremadretutor".

### 3.2. OBJETIVOS PERSONALES

El objetivo de este punto consiste en poder expresar aquellos objetivos personales que me han conducido y motivado a hacer este proyecto.

Uno de los objetivos personales fue el desarrollo en Android, aunque ya sé que es java y llevo trabajando varios años con este lenguaje, nunca había programado en Android y no sabía cómo funciona toda la parte de las activities y sobre todo, toda la parte que tiene que ver con el front, porque eso nunca lo había visto.

Otro de los motivos fue hacer una pequeña arquitectura persona en Vaadin, framework del que hablaré en un punto más adelante, ya que nunca lo había visto y me pareció bastante interesante para desarrollar la parte de administrador con este framework.

Este proyecto me motivo por el simple hecho de que me gusta programar, hacer los servicios y lograr una comunicación con estos. Por otra parte, en un futuro no muy probable, si esta aplicación llega a desarrollarse más y perfeccionar algunas cosas y pasa a ser usada por los centros, me sentiría super orgullos de ello.

## 4. CONCEPTOS TEÓRICOS

---

En este apartado vamos a explicar todos los conceptos teóricos que son utilizados en todo el proyecto para que así se facilite su comprensión.

### 4.1. API REST

Una API, o interfaz de programación de aplicaciones, es un conjunto de reglas, las cuáles definen como pueden las apps o los dispositivos móviles conectarse y comunicarse entre ellos.

Una API Rest es un estilo arquitectónico para una interfaz de programa de aplicación (API) que utiliza solicitudes HTTP para acceder y utilizar datos. Una API REST es una api que cumple los principios de diseño del estilo de arquitectura REST (Transferencia de estado representacional), por ello, las API REST se conocen a veces como API RESTful.

#### Características:

- **Protocolo Cliente-Servidor:** Tanto el cliente como el servidor están muy poco acoplados, así que, el cliente no necesita conocer como es la implementación en el lado del servidor y el servidor no tiene que saber como se usan sus datos que envía al cliente.
- **Sin Estado:** Su servicio no tiene estado, lo que significa que tenemos que recordarle siempre nuestros datos, tanto credenciales como otra información. Su desventaja es que puede ser una tarea tediosa repetir los datos pero por otra parte tiene como ventaja como no almacenar los datos, permite una escalabilidad mayor, etc.
- **Es más sencillo que SOAP:** La principal ventaja frente a SOAP es que el API REST es mucho más sencillo de implementar.
- **Sistema de Capas:** Da al servidor la capacidad de tener capas para la implementación para poder así mejorar su escalabilidad, seguridad, rendimiento, etc.



## 5. TÉCNICAS Y HERRAMIENTAS

---

En este punto se establecerá una descripción de las técnicas metodológicas, las herramientas utilizadas para el desarrollo, módulos software que se han usado para poder realizar este proyecto.

### 5.1. TÉCNICAS Y HERRAMIENTAS USADAS EN EL SERVIDOR

#### 5.1.1. Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos diseñado para tener pocas dependencias de implementación. Es muy usado a nivel mundial para muchas aplicaciones.

##### Algunas características del lenguaje:

- **Es Simple:** Ofrece la funcionalidad de un lenguaje potente, pero sin las características menos usadas y confusas de C yC++. Eliminar muchas características de estos lenguajes, para mantener reducidas sus especificaciones.
- **Orientado a Objetos:** Java implemente la tecnología básica de C++ y elimina algunas cosas para hacer que el lenguaje siga siendo simple. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las 3 características del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas clases y sus copias instancias.
- **Distribuido:** Java está construido con capacidades de interconexión TCP/IP. Existen librerías para poder acceder e interactuar con protocolos http y ftp, lo que permite acceder a la información a través de la red.
- **Portable:** Significa que un programa escrito en Java puede llegar a ejecutarse en cualquier tipo de hardware.

#### 5.1.2. Framework Vaadin

Vaadin es un framework de desarrollo de SPA (Single-page application) permitiendo así escribir código de aplicaciones en Java o en cualquier otro lenguaje soportado por JVM 1.6+. Esto permite la programación de la interfaz gráfica en lenguajes como Java 8, Scala, Groovy, etc.

Una de las características más importante de Vaadin es que presenta una arquitectura centrada en el servidor, lo que implica que la mayoría de lógica va a ser ejecutada en los servidores.

### 5.1.3. Oracle Database

Oracle Database es una base de datos que utiliza un lenguaje de programación estandarizado de SQL para crear estructuras de base de datos, gestionar registros, realizar acciones o recuperar datos. El lenguaje propio de Oracle es PL/SQL, que está vinculado a SQL y ofrece la posibilidad de complementar SQL con extensiones de programación de Oracle.

Para estructurar sus datos, usa tablas de filas y columnas.

Algunas de las ventajas son:

- **Alta compatibilidad con plataformas y aplicaciones.**
- **Funciones de protección de datos y seguridad.**
- **Es el sistema de gestión de bases de datos más popular.**
- **Tiene varias ediciones, desde gratuitas hasta de pago para empresas.**

Como inconvenientes tiene:

- **Hay que tener un cierto conocimiento de SQL para Oracle.**
- **Altas exigencias en cuanto a hardware se refieren.**

### 5.1.4. JSON

JSON (JavaScript Object Notation) es un formato que se utiliza para intercambiar y guardar la información de forma que cualquier persona lo pueda leer y entender. Se trata de archivos de texto con una extensión .json.

La información que almacena está estructurada y se usa para el intercambio de información entre cliente y servidor. Ejemplo de ello en la *Imagen 3*:

```
{
  "tipo": "Class",
  "nombre": "Persona",
  "operaciones": ["Consultar","Insertar","Actualizar"],
  "vistas": ["PDF","XML", "cargo; "]
}
```

*Imagen 3: Json ejemplo*

### 5.1.5. Spring Framework

Spring Framework es un framework Open Source que ayuda en la creación de aplicaciones de todo tipo en Java, Kotlin y Groovy.

Por lo más conocido es por la inyección de dependencias.

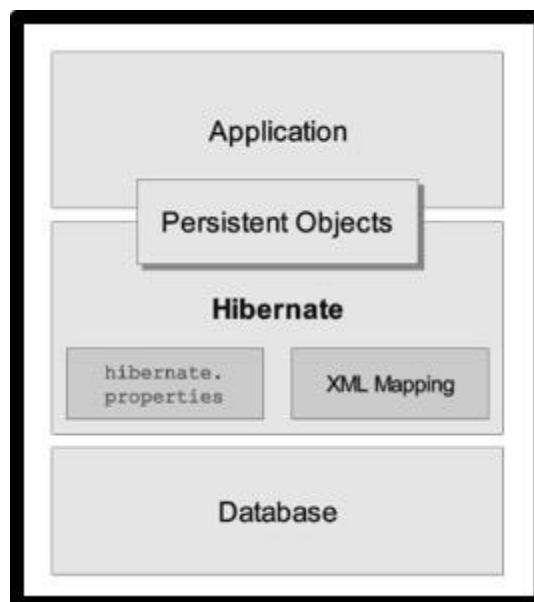
Algunas de sus funcionalidades son:

- **Core Container:** Proporciona inyección de dependencias e inversión de control.
- **Web:** Nos permite crear controladores WEB, tato de vistas MVC como aplicaciones REST.
- **Acceso a datos:** Tiene abstracciones sobre JDBC, ORMs como Hibernate, etc.
- **Programación orientada a Aspectos:** Da soporte para la programación orientada a Aspectos.

### 5.1.6. Hibernate

Hibernate es una herramienta de mapeo-objeto-relacional (Orm) para Java, facilitando el mapeo de atributos en una base de datos tradicional, como en nuestro caso Oracle.

Esto nos agiliza la relación entre la aplicación y nuestra base de datos SQL, de modo que optimiza el flujo de trabajo evitando crear código repetitivo. La forma en la que trabaja hibernate, se puede ver en la *Imagen 4*:



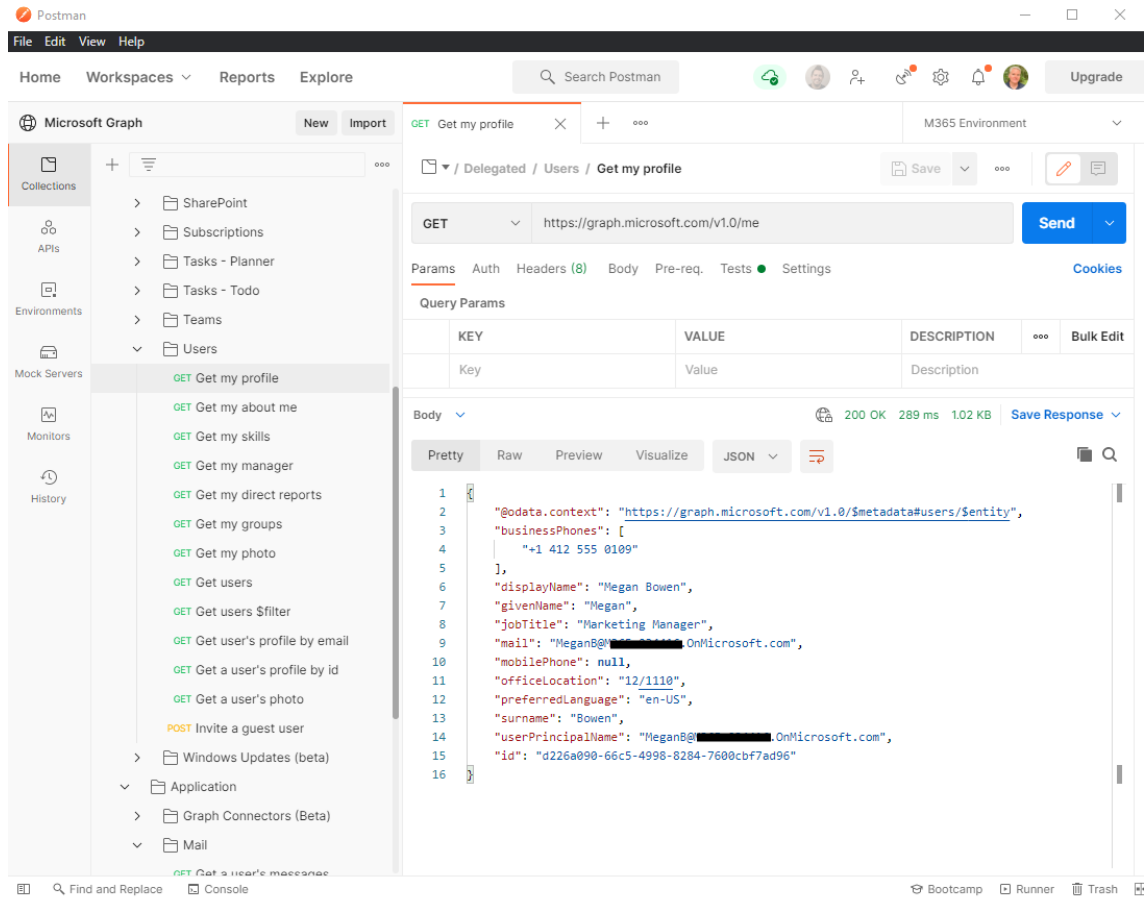
*Imagen 4: Ejemplo de Funcionalidad*

## Memoria del Proyecto

### 5.1.7. PostMan

Postman es una aplicación que nos permite realizar pruebas API. Es un cliente HTTP que nos da la posibilidad de testear 'HTTP requests' a través de una interfaz gráfica de usuario, por medio de la cual obtendremos diferentes tipos de respuesta que posteriormente deberán ser validados.

La interfaz gráfica de Postman, sería la que corresponde con la *Imagen 5*:



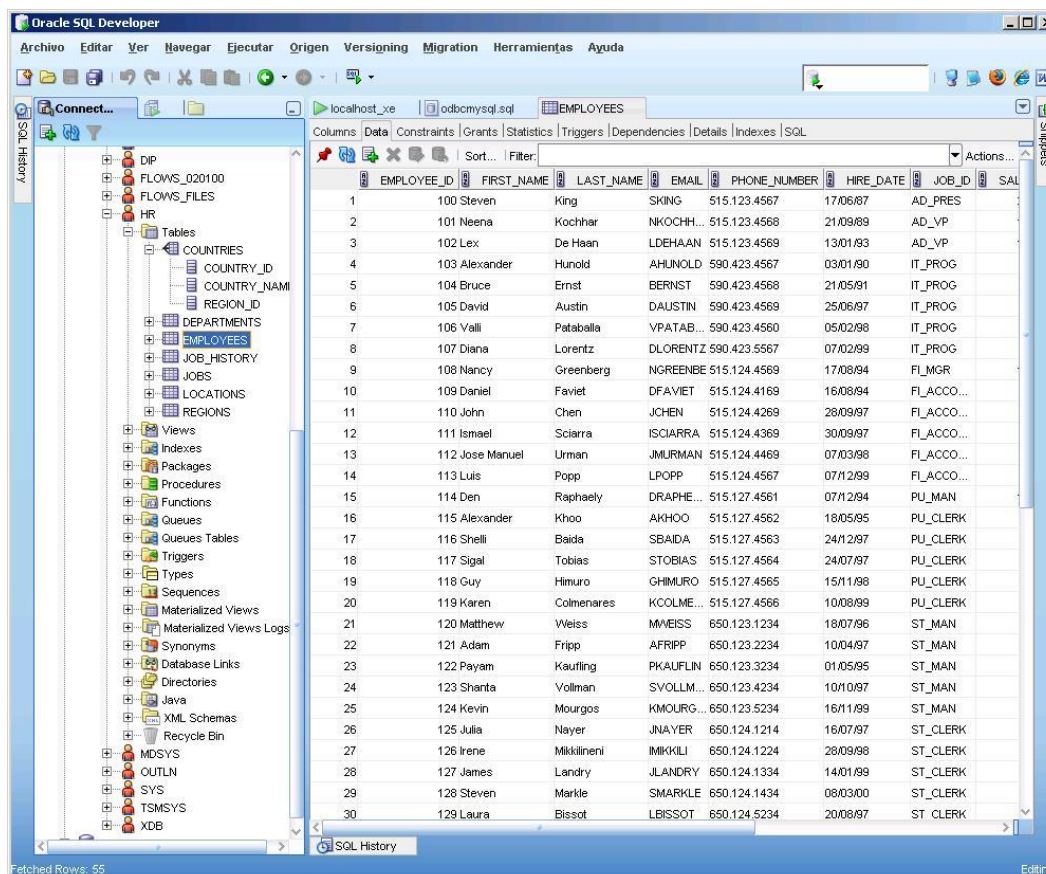
*Imagen 5: Postman*

## Memoria del Proyecto

### 5.1.8. SqlDeveloper

Oracle SQL Developer es una interfaz gráfica de usuario gratuita que permite a los usuarios y administradores de bases de datos realizar sus tareas con menos clics y pulsaciones de teclas. SQL Developer es una herramienta de productividad cuyo objetivo principal es ayudar al usuario final a ahorrar tiempo y maximizar el retorno de la inversión en el paquete de tecnología de Oracle Database.

Un ejemplo de visualización de la herramienta SqlDeveloper, se puede ver en la *Imagen 6*:



*Imagen 6: SqlDeveloper*

### 5.1.9. Spring Tools Suite

Spring Tool Suite (STS) es un IDE basado en la versión Java EE de Eclipse, pero altamente customizado para trabajar con Spring Framework. Entre las características más destacadas que STS proporciona se encuentran:

- Soporte para Spring
- Asistentes para la creación de proyectos Spring
- Herramientas para la gestión de beans
- Editores gráficos de archivos de configuración de Spring
- Herramientas de desarrollo para Spring Web Flow y Spring Batch

### 5.1.10. Firebase

Firebase es una plataforma de Google utilizada para el desarrollo de aplicaciones web y móvil. Permite usar servicios en la nube ya que está integrada con Google Cloud Platform. En nuestro caso lo hemos utilizado para el envío de notificaciones entre el servidor y el dispositivo móvil.

## 5.2. TÉCNICAS Y HERRAMIENTAS USADAS EN LA APLICACIÓN MÓVIL

### 5.2.1. Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, como las siguientes:

- Un sistema de compilación flexible basado en Gradle.
- Un emulador rápido y cargado de funciones.
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android.
- Aplicación de cambios para insertar cambios de código y recursos a la app en ejecución sin reiniciarla.
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra.

## 5.3. HERRAMIENTAS CASE

Se va a explicar en este punto cuales han sido las herramientas case utilizadas a lo largo de todo el desarrollo del proyecto. Las herramientas CASE hacen referencia a aquellos programas que han sido usados para aumentar el desarrollo software permitiendo una reducción del coste de las mismas en términos de tiempo.

### 5.3.1. Microsoft Project

Es una herramienta usada para la gestión de los proyectos permitiendo asignar tareas y recursos a un proyecto determinado. Permite realizar planes y seguimientos de todo el proceso de desarrollo del proyecto. (Documentación: <https://docs.microsoft.com/es-es/project/>)

# Memoria del Proyecto

Se ha hecho uso de esta herramienta en la fase de planificación temporal del proyecto.

A continuación, en la *Imagen 7*, se puede observar un ejemplo del diagrama de Gantt generador con Microsoft Project.

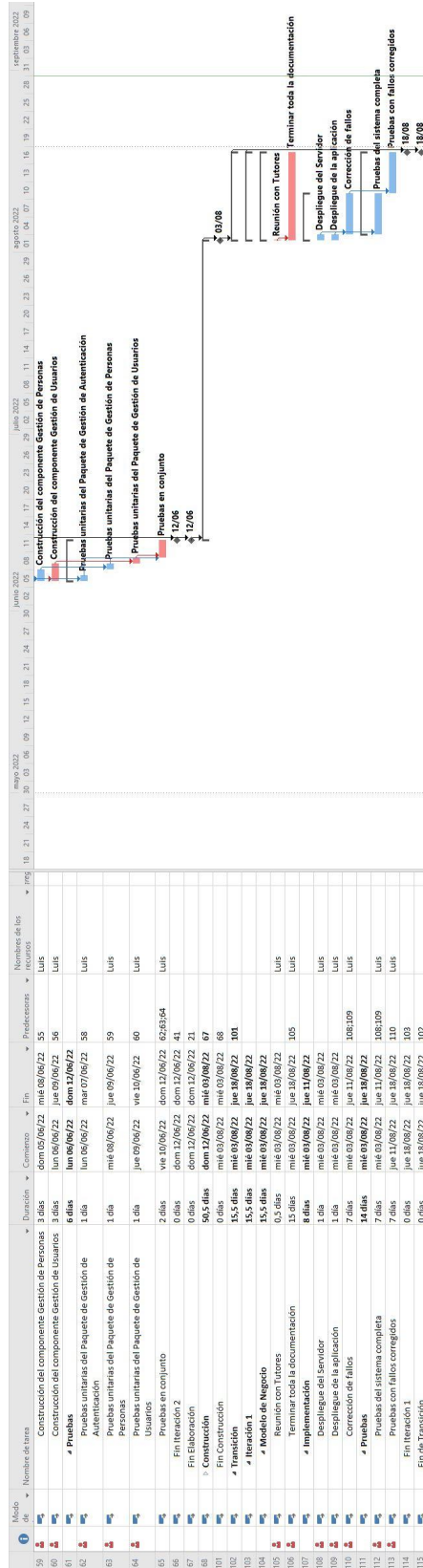


Imagen 7: Microsoft Project

## Memoria del Proyecto

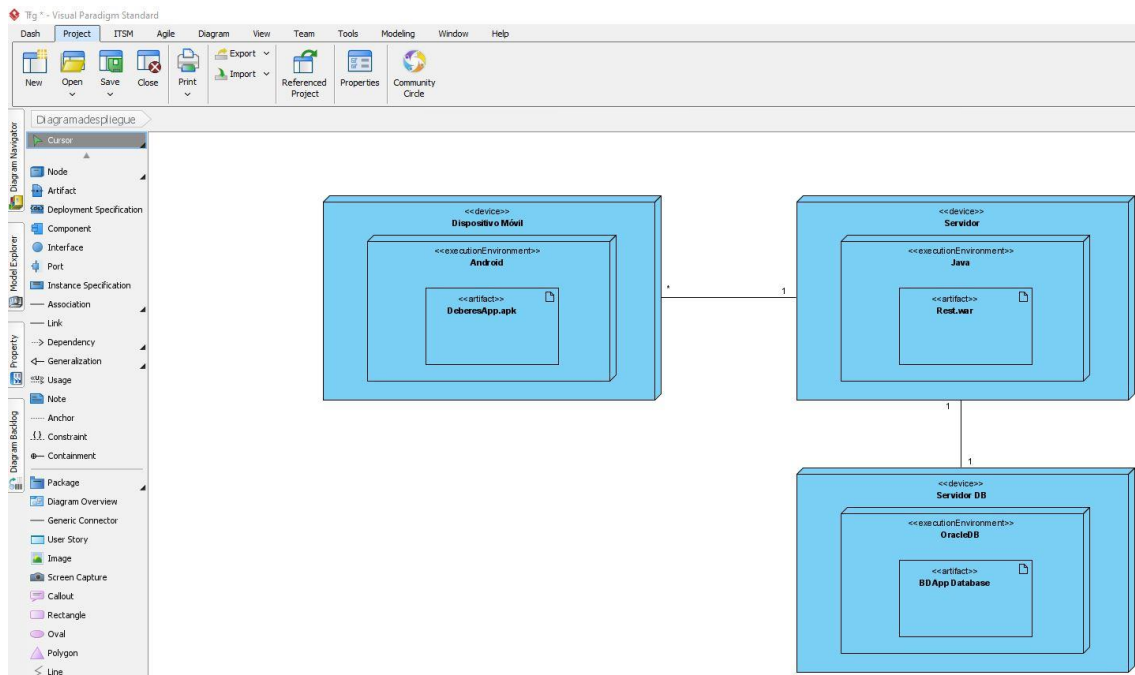
### 5.3.2. Visual Paradigm

Visual Paradigm es un programa que permite hacer el ciclo completo de vida de desarrollo del software mediante la representación en diagrama UML de cualquier tipo.

(Documentación: <https://www.visual-paradigm.com/support/documents/>)

Esta herramienta es utilizada durante las fases de especificación de requisito, análisis de requisitos y el diseño del sistema. Los diagramas que se han hecho durante todo el proceso han sido realizados con esta herramienta.

Ejemplo de uso de Visual Paradigm en la *Imagen 8*:



*Imagen 8: Visual Paradigm*

### 5.3.3. REM

REM (REquirements Manager) es una herramienta utilizada para realizar las tablas en la fase de elicitación de requisitos de un proyecto software. Utiliza la metodología de Durán y Bernárdez, proporcionando una plantilla para la documentación de objetos, participantes, actores, casos de uso.

Nos da la posibilidad de generar de forma automática las matrices de rastreabilidad permitiendo así verificar las dependencias.

La herramienta REM es utilizada en la fase de especificación de requisitos.



## Memoria del Proyecto

### 5.3.4. EZEstimate

EZEstimate es una herramienta utilizada para la gestión de proyectos en cuanto a la estimación de la duración de este se refiere, en función de la complejidad de sus casos de uso y los actores que existen e interaccionan con el sistema. También la herramienta va a tener en cuenta factores de complejidad y de ambiente.

EZEstimate se ha utilizado durante la fase de planificación temporal del proyecto.

Un ejemplo de esta herramienta se puede ver en la *Imagen 9*, que corresponde con la estimación del proyecto desarrollado:

The screenshot shows the EZEstimate application window. The title bar indicates the file path: C:\Users\Luisitopana\Desktop\Tfg Memoria\Anexo 1\Tfg.ezp. The interface is divided into several sections:

- Module:** A dropdown menu shows 'Gestión de Autenticación'. Below it are 'Add Module' and 'Delete' buttons.
- Summary:** Displays 'Total Modules' as 12. It includes an 'Excel Report' button and a 'Generate Report' button. Below this, it shows 'Use cases' counts: Simple (43), Average (0), and Complex (0). 'Actors' counts are: Simple (1), Average (0), and Complex (4).
- Add Actor / Use case:** A form with 'Actor / Use case Name', 'Select Type', and 'Complexity' dropdowns, and an 'Add' button.
- Tech / Env Factors:** Two buttons: 'Set Tech Factor' and 'Set Env Factors'.
- Estimation Summary:** A series of input fields for calculations:
  - UAW: 13
  - UUCW: 215
  - UUPC = UAW + UUCW: 228
  - TFactor: 16
  - EFactor: 22
  - TCF = 0.6 + (.01\*TFactor): 0.76
  - EF = 1.4 + (.03\*EFactor): 0.74
  - UCP = UUCP\*TCF\*EF: 128.2272
  - Total Effort@ 7 Hrs/UCP: 897.5904
- Use case / Actor List:** A table with columns: Id, Module, Type, Name, and complexity. It lists 25 items with their respective IDs, modules, types (Actor or Usecase), names, and complexity levels (Simple or Complex).

Imagen 9: EZEstimate

## 5.4. HERRAMIENTAS DE GENERACIÓN DE DOCUMENTACIÓN

Para la generación de la documentación se han utilizado los dos IDEs correspondientes con los que se ha desarrollado el proyecto completo: Eclipse y Android Studio.

**En Eclipse:** Project -> Generate Javadoc -> Y elegir el destino donde se va a generar toda la documentación.

**En Android:** Tools -> Generate JavaDoc -> Elegir ruta donde se va a guardar la documentación.

## **5.5. HERRAMIENTAS DE CONTROL DE VERSIONES**

### **5.5.1. GitHub**

GitHub es una herramienta basada en la nube que aloja un sistema de control de versiones llamado Git, el cuál permita a los desarrolladores colaborar y realizar cambios en proyectos compartidos, manteniendo así un seguimiento detallado de su progreso.

## 6. ASPECTOS RELEVANTES DEL DESARROLLO

---

Este punto se desarrollará especificando las partes más importantes y representativas de todas las fases de desarrollo del proyecto.

### 6.1. MARCO DE TRABAJO

Para el marco del trabajo se ha utilizado el Proceso Unificado, que permite definir un marco de desarrollo de software dirigido por casos de uso, el cuál permite un desarrollo iterativo e incremental. Algunas de las características del Proceso Unificado son:

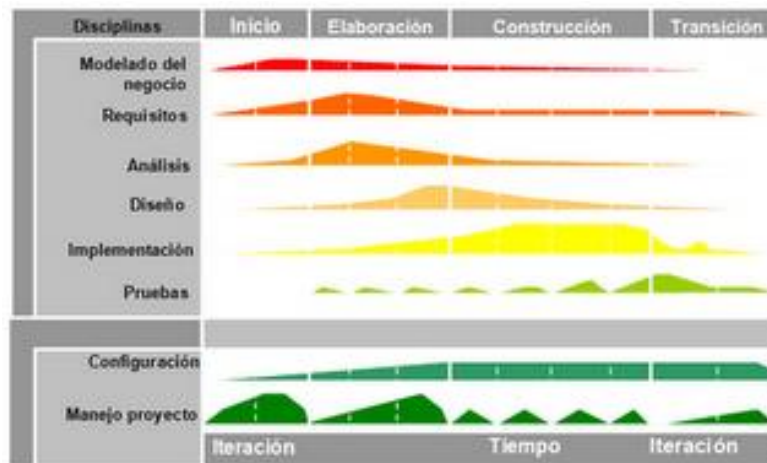
- **Iterativo e Incremental:** Este marco de desarrollo es iterativo e incremental, que está compuesto por cuatro fases: Inicio, elaboración, construcción y transición, las cuales se explicarán más adelante. Estas fases, a su vez, pueden estar divididas en iteraciones. Estas iteraciones dan como resultado un incremento del producto añadiendo o mejorando sus funcionalidades.
- **Centrado en la arquitectura:** El proceso unificado asume que no existe un modelo único para cubrir los aspectos del sistema, por lo tanto, hay múltiples modelos y vistas que definen la arquitectura de software de un sistema.
- **Dirigido por casos de uso:** Los casos de uso se utilizan para capturar los requisitos funcionales y definir el contenido de las iteraciones.

En cuanto a las fases que hemos citado antes, el proceso unificado se divide en:

- **Inicio:** Es una fase donde se define el alcance del proyecto y se define el modelo de negocio, es decir, deseos del usuario, metas, costes...
- **Elaboración:** Fase donde se planifica el proyecto, especificando el detalle la mayoría de los casos de uso y se diseña la arquitectura del sistema.
- **Construcción:** Fase donde se construye el proyecto, hasta llegar a un prototipo donde se incluyen los requisitos mínimos haciendo una revisión de los aspectos anteriormente citados.
- **Transición:** Fase durante la cual el sistema debe estar listo para ser probado, instalado y ser utilizado por el usuario final.

## Memoria del Proyecto

En la *Imagen 10*, puede verse las distintas fases del proceso de desarrollo, en conjunto con el solapamiento de tareas a lo largo de todas las fases:



*Imagen 10: Proceso Unificado*

## 6.2. ESTIMACIÓN DE DURACIÓN DEL PROYECTO

Una de las fases primeras fue la estimación de duración del desarrollo del proyecto. El objetivo de esta fase se trataba de obtener una estimación de duración en tiempo del desarrollo completo del sistema. Con dicha estimación podremos repartir el tiempo en cada una de las tareas definidas en la planificación temporal de forma más acertada.

Para tener más información acerca de todo este proceso de estimación, se podrá acceder al Anexo I - Planificación Temporal.

En la *Imagen 11* podemos observar los parámetros que se han introducido para la estimación de tiempo y el resultado de esta misma:

La imagen muestra la interfaz de usuario del software EZEstimate. El título de la ventana es "EZEstimate - C:\Users\Luisitopana\Desktop\Tfg Memoria\Anexo I\Tfg.ezp".

**Module:** Gestión de Autenticación

**Summary:**

- Total Modules: 12
- Excel Report:  Generate Report
- Use cases: Simple 43, Average 0, Complex 0
- Actors: Simple 1, Average 0, Complex 4

**Add Actor / Use case:**

Actor / Use case Name:  Select Type:  Complexity:  Add

**Tech / Env Factors:**

Set Tech Factor:  Set Env Factors:

**Estimation Summary:**

- UAW: 13
- UUCW: 215
- UUPC = UAW + UUCW: 228
- TFactor: 16
- EFactor: 22
- TCF = 0.6 + (.01\*TFactor): 0.76
- EF = 1.4 + (.03\*EFactor): 0.74
- UCP = UUPC\*TCF\*EF: 128.2272
- Total Effort@ 7 Hrs/UCP: 897.5904

**Use case / Actor List ( Double click to delete )**

Id	Module	Type	Name	complexity
1	Gestión de Aut...	Actor	Usuario No Ide...	Complex
10	Gestión de Usu...	Usecase	UC0009	Simple
11	Gestión de Usu...	Usecase	UC0010	Simple
12	Gestión de Usu...	Usecase	UC0011	Simple
13	Gestión de Roles	Actor	Usuario Web A...	Complex
14	Gestión de Roles	Usecase	UC0012	Simple
15	Gestión de Roles	Usecase	UC0013	Simple
16	Gestión de Roles	Usecase	UC0014	Simple
17	Gestión de Cen...	Usecase	UC0015	Simple
18	Gestión de Cen...	Usecase	UC0016	Simple
19	Gestión de Cen...	Usecase	UC0017	Simple
2	Gestión de Aut...	Usecase	UC0001	Simple
20	Gestión de Cen...	Usecase	UC0018	Simple
21	Gestión de Curs...	Usecase	UC0019	Simple
22	Gestión de Curs...	Usecase	UC0020	Simple
23	Gestión de Curs...	Usecase	UC0021	Simple
24	Gestión de Curs...	Usecase	UC0022	Simple
25	Gestión de Asin...	Usecase	UC0023	Simple

*Imagen 11: Casos de Uso y Resultados*

### 6.3. PLANIFICACIÓN TEMPORAL

Después de realizar una estimación con la herramienta EZEstimate y conocerse su resultado, se ha procedido a realizar una planificación temporal para tener conocimiento de las tareas que son necesarias para llevar a cabo el desarrollo del proyecto. Para la realización de planificación temporal se ha seguido el Proceso Unificado, siguiendo todas las fases anteriormente explicadas, dentro de las cuales tienen las siguientes etapas:

- **Modelado de Negocio:** Se analiza los aspectos asociados al proyecto que estamos desarrollando como pueden ser la investigación, búsqueda de recursos que se relacionen con el proyecto.
- **Requisitos:** Etapa en la que se van a fijar objetivos y requisitos que van a tener el proyecto.
- **Análisis:** En esta etapa se va a realizar un análisis más minucioso de los requisitos recogidos en la etapa anterior.
- **Diseño:** Etapa donde se determina el funcionamiento y la forma de los componentes del sistema.
- **Implementación:** Es donde se va a realizar toda la programación del sistema.
- **Pruebas:** Etapa que corresponde con las pruebas unitarias y completas del sistema para comprobar su correcto funcionamiento y la posible corrección de errores.

Para tener más información acerca de todo este proceso de estimación, se podrá acceder al Anexo I - Planificación Temporal.

Las tareas que se deben realizar para el desarrollo del proyecto junto al diagrama de Gantt, se pueden observar en las imágenes *Imagen 12*, *Imagen 13* e *Imagen 14*:

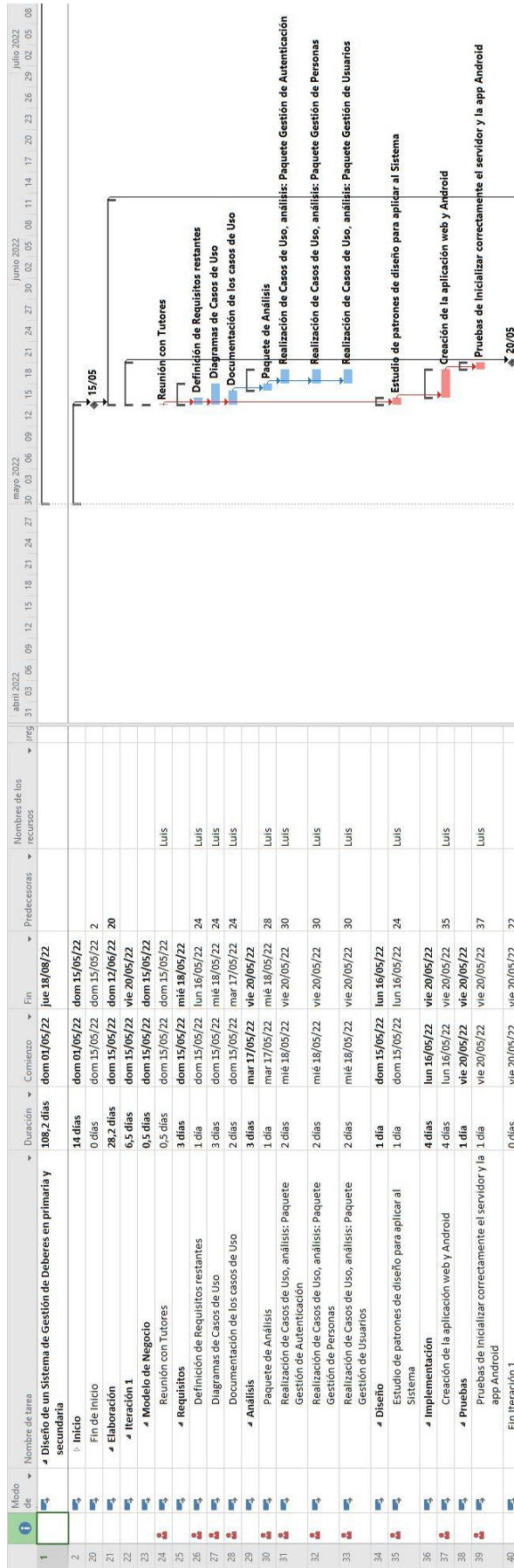


Imagen 12: Diagrama de Gantt 1



Imagen 13: Diagrama de Gantt 2

# Memoria del Proyecto

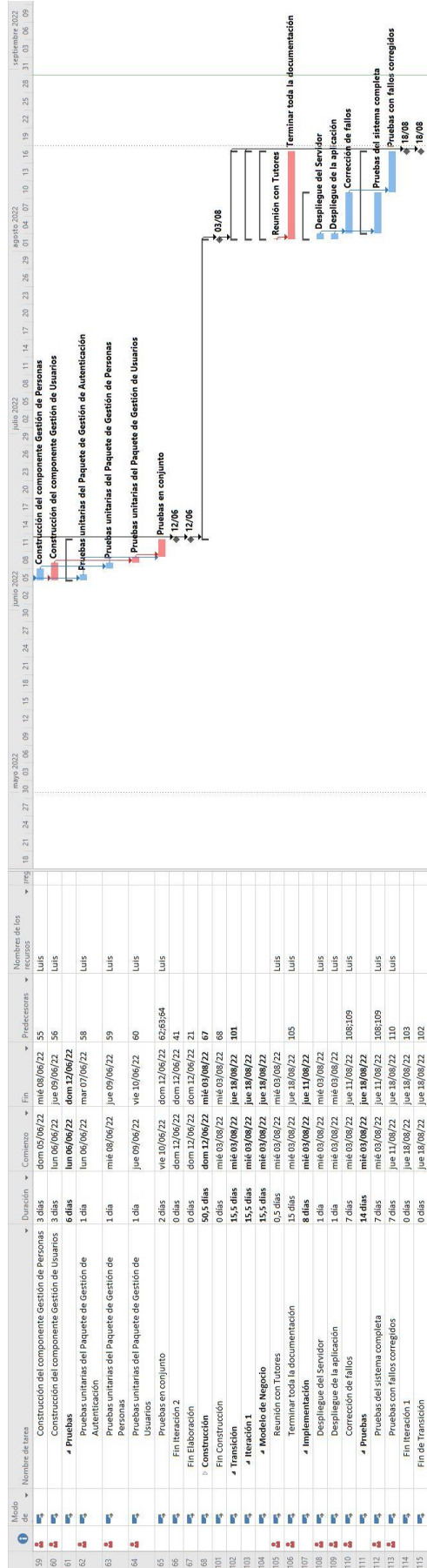


Imagen 14: Diagrama de Gantt 3



## 6.4. ESPECIFICACIÓN DE REQUISITOS

En este punto se ha realizado la fase de especificación de requisitos, durante la cuál se han ido recogiendo las especificaciones de los requisitos software del sistema. Para la elicitación de requisitos del sistema se ha seguido la metodología de Durán y Bernárdez. Esto va a servir de base para el desarrollo de las siguientes fases del proyecto.

Si se requiere de más información de este apartado, se podrá acceder al Anexo II – Especificación de requisitos del sistema.

A continuación, se mostrarán algunos ejemplos de la fase de especificación de requisitos:

### 6.4.1. Participantes

Estos son los participantes que han estado durante todo el desarrollo del proyecto:

- Luis Martín Hernández
- André Filipe Sales Mendes
- Gabriel Villarubia González
- Juan Francisco de Paz Santana

<b>Participante</b>	<b>Luis Martín Hernández</b>
<b>Organización</b>	Universidad de Salamanca
<b>Rol</b>	Diseñador / desarrollador
<b>Es desarrollador</b>	Sí
<b>Es cliente</b>	No
<b>Es usuario</b>	No
<b>Comentarios</b>	Ninguno

*Tabla 1: Participante Luis Martín Hernández*

### 6.4.2. Objetivos del Sistema

Los objetivos que se han planteado para lograr desarrollar el sistema son:

1. **Gestión de Usuarios:** El sistema deberá manejar diferentes roles para cada usuario, algunos ejemplos son:
  - Usuario Administrador: Permite manejar la información del sistema, permitiendo modificarla.
  - Usuario Profesor: Permite visualizar toda la información generada por el sistema, así como crear nueva información.
  - Usuario Padre/Madre/Tutor: Permite solamente visualizar la información del sistema.

Para acceder al sistema, se requiere autenticarse, permitiendo a los administradores crear nuevos usuarios, pero no se permitirá que cualquier usuario se registre sin el permiso del administrador.

## Memoria del Proyecto

2. **Gestión de Centros:** El sistema deberá permitir crear, modificar o eliminar los centros.
3. **Gestión de Cursos:** El sistema deberá permitir crear, modificar o eliminar los cursos.
4. **Gestión de Asignaturas:** El sistema deberá permitir crear, modificar o eliminar las asignaturas.
5. **Gestión de Alumnos:** El sistema deberá permitir crear, modificar o eliminar los alumnos.
6. **Gestión de Deberes:** El sistema deberá permitir la creación, modificación y eliminación de deberes por parte los profesores y de visualización por parte de ellos padres.
7. **Gestión de Estadísticas:** El sistema deberá poder generar estadísticas automáticamente y mostrárselas a los usuarios con rol "Profesor".
8. **Gestión de Personas:** El sistema deberá permitir crear, modificar o eliminar las personas.
9. **Gestión de Roles:** El sistema deberá permitir crear, modificar o eliminar los roles
10. **Gestión de Profesores:** El sistema deberá permitir crear, modificar o eliminar los alumnos.
11. **Gestión de Padremadretutor:** El sistema deberá permitir crear, modificar o eliminar los padres, madres o tutores de alumnos.

Ejemplo de Objetivo en la *Tabla 2*:

<b>OBJ-0008</b>	<b>Gestión de Personas</b>
<b>Versión</b>	1.0 (09/07/2022)
<b>Autores</b>	<ul style="list-style-type: none"><li>• <a href="#">Luis Martin Hernández</a></li></ul>
<b>Fuentes</b>	<ul style="list-style-type: none"><li>• <a href="#">André Filipe Sales Mendes</a></li><li>• <a href="#">Gabriel Villarrubia González</a></li><li>• <a href="#">Juan Francisco De la Paz Santana</a></li></ul>
<b>Descripción</b>	El sistema deberá <i>poder crear, modificar y eliminar personas.</i>
<b>Subobjetivos</b>	Ninguno
<b>Importancia</b>	Vital
<b>Urgencia</b>	Inmediatamente
<b>Estado</b>	Validado
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

*Tabla 2: OBJ-0008 Gestión de Personas*

### 6.4.3. Requisitos de Información

Estos requisitos son los que indican los datos que se van a poder guardar en el sistema. Los requisitos de información son los siguientes:

1. Información sobre Personas
2. Información sobre Usuarios
3. Información sobre Roles
4. Información sobre RolUsuario
5. Información sobre Centros
6. Información sobre Cursos
7. Información sobre Asignaturas
8. Información sobre Alumnos
9. Información sobre Padremadretutor
10. Información sobre Profesores
11. Información sobre Deberes
12. Información sobre Personas

Un ejemplo de requisito de información se puede observar en la *Tabla 3*:

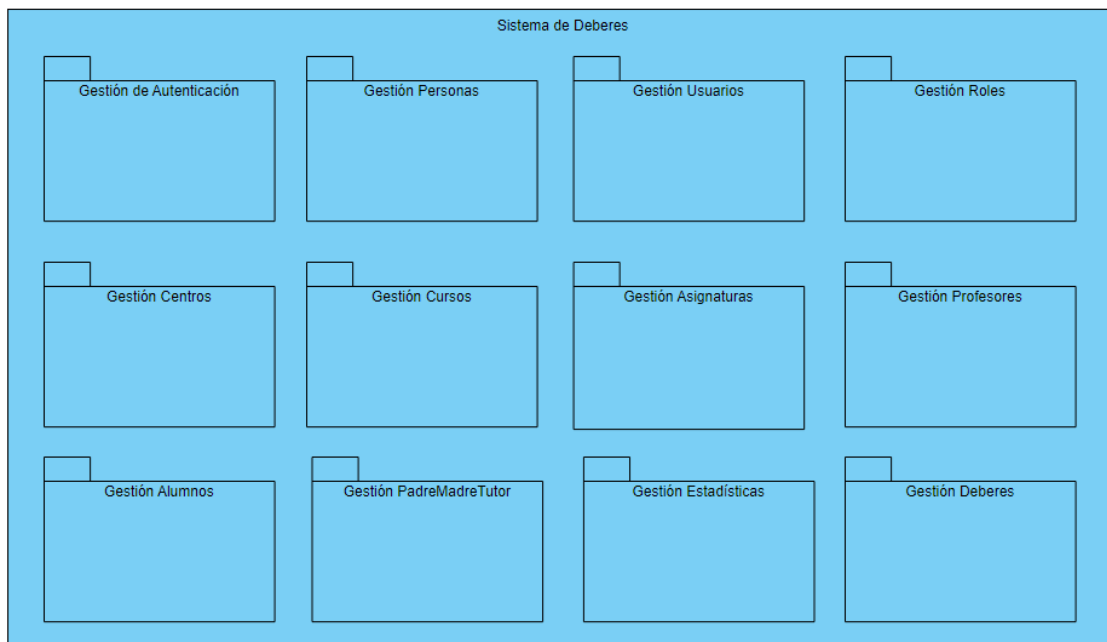
<b>IRQ-0001</b>	<b>Información sobre Personas</b>
<b>Versión</b>	1.0 (09/07/2022)
<b>Autores</b>	<ul style="list-style-type: none"> <li>• <a href="#">Luis Martin Hernández</a></li> </ul>
<b>Fuentes</b>	<ul style="list-style-type: none"> <li>• <a href="#">André Filipe Sales Mendes</a></li> <li>• <a href="#">Gabriel Villarrubia González</a></li> <li>• <a href="#">Juan Francisco De la Paz Santana</a></li> </ul>
<b>Dependencias</b>	<ul style="list-style-type: none"> <li>• <a href="#">[OBJ-0008] Gestión de Personas</a></li> </ul>
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a <i>cada una de las personas</i>
<b>Datos específicos</b>	<i>Idpersona (Único)</i> <i>Nombre</i> <i>Apellido 1</i> <i>Apellido 2</i> <i>Dni</i> <i>Fecha Nacimiento</i> <i>Dirección</i> <i>Teléfono</i> <i>Email</i>
<b>Importancia</b>	Vital
<b>Urgencia</b>	Inmediatamente
<b>Estado</b>	Validado
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

*Tabla 3: IRQ-0001 Información sobre Personas*

### 6.4.4. Requisitos Funcionales

Los requisitos funcionales son los requisitos que van a definir los escenarios y los casos de uso y de cómo se debe comportar el sistema en cada caso.

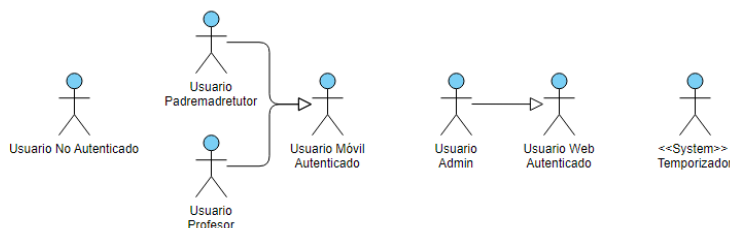
En este punto, la primera tarea que se ha realizado es el diagrama de paquetes para así tener una visión general del proyecto. El diagrama de paquetes se puede ver en la *Imagen 15*.



*Imagen 15: Diagrama de Paquetes*

Después de realizar este diagrama, se ha procedido a definir los actores que representan a los usuarios que van a interactuar con el sistema, viéndose la relación en la *Imagen 16*. Estos actores son:

- Usuario No autenticado
- Usuario Padremadretutor
- Usuario Profesor
- Usuario Móvil Autenticado
- Usuario Admin
- Usuario Web Autenticado
- <<System>> Temporizador



*Imagen 16: Diagrama de Actores*

Especificación de Actor en la *Tabla 4*:

<b>ACT-0004</b>	<b>Usuario Profesor</b>
<b>Autores</b>	<ul style="list-style-type: none"> <li>• <a href="#">Luis Martin Hernández</a></li> </ul>
<b>Fuentes</b>	<ul style="list-style-type: none"> <li>• <a href="#">André Filipe Sales Mendes</a></li> <li>• <a href="#">Gabriel Villarrubia González</a></li> <li>• <a href="#">Juan Francisco De la Paz Santana</a></li> </ul>
<b>Descripción</b>	Este actor representa <i>al usuario que ha iniciado sesión con rol de profesor</i>
<b>Comentarios</b>	Ninguno

*Tabla 4: ACT-0004 Usuario Profesor*

Después de todo este proceso, a continuación, se han realizado todos los casos de uso del sistema. Se han hecho los diagramas de casos de uso por paquetes y después, se ha especificado uno por uno.

Ejemplo de diagrama de Caso de Uso en la *Tabla 5*:

<b>UC-0005</b>	<b>Listar Personas</b>	
<b>Versión</b>	1.0 (09/07/2022)	
<b>Autores</b>	<ul style="list-style-type: none"> <li>• <a href="#">Luis Martin Hernández</a></li> </ul>	
<b>Fuentes</b>	<ul style="list-style-type: none"> <li>• <a href="#">André Filipe Sales Mendes</a></li> <li>• <a href="#">Gabriel Villarrubia González</a></li> <li>• <a href="#">Juan Francisco De la Paz Santana</a></li> </ul>	
<b>Dependencias</b>	<ul style="list-style-type: none"> <li>• <a href="#">[OBJ-0008] Gestión Personas</a></li> <li>• <a href="#">[IRQ-0001] Información Persona</a></li> </ul>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuándo <i>el Usuario Web Autenticado solicita listar las personas</i>	
<b>Precondición</b>		
<b>Secuencianormal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor <a href="#">Usuario Web Autenticado (ACT-0007)</a> <i>solicita listar las personas.</i>
	2	El Sistema <i>recupera y devuelve la lista de personas.</i>
	3	Si solicita <i>editar una persona</i> , se realiza el caso de uso <a href="#">[UC-0006] Editar Persona</a> .
	4	Si solicita <i>eliminar una persona</i> , se realiza el caso de uso <a href="#">[UC-0007] Eliminar Persona</a> .
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	-	-

## Memoria del Proyecto

<b>Rendimiento</b>	<b>Paso</b>	<b>Tiempo máximo</b>
	2	4 segundos
<b>Frecuencia esperada</b>	2 veces por hora(s)	
<b>Importancia</b>	vital	
<b>Urgencia</b>	hay presión	
<b>Estado</b>	validado	
<b>Estabilidad</b>	alta	
<b>Comentarios</b>	Ninguno	

*Tabla 5: UC-0005 Listar Personas*

#### **6.4.5. Requisitos No Funcionales**

Aparte de los requisitos funcionales, tenemos que establecer también los requisitos no funcionales, que son aquellos que definen las restricciones que debe cumplir el sistema para garantizar un diseño seguro y con estándares de calidad.

Los requisitos no funcionales son los siguientes:

- 1. Usabilidad**
- 2. Compatibilidad**
- 3. Portabilidad**
- 4. Fiabilidad**
- 5. Rendimiento**
- 6. Privacidad**

#### **6.5. ANÁLISIS DE REQUISITOS**

A partir de los requisitos que se han alcanzado en la fase de especificación de requisitos, se pasa a la fase de análisis de requisitos, en la cual van a refinarse con un análisis más exhaustivo.

Para obtener más información del análisis de requisitos, se podrá hacer uso del Anexo III – Análisis de Requisitos.

6.5.1. Modelo de Dominio

El modelo de dominio es una representación de clases conceptuales. Recoge las necesidades de gestión y almacenamiento de la información que maneja el sistema.

Para poder realizarlo, se deberá determinar todas las entidades del modelo de negocio necesarias, los atributos de estas y las relaciones entre ellas.

En la figura Imagen 25, podemos ver el modelo de dominio:

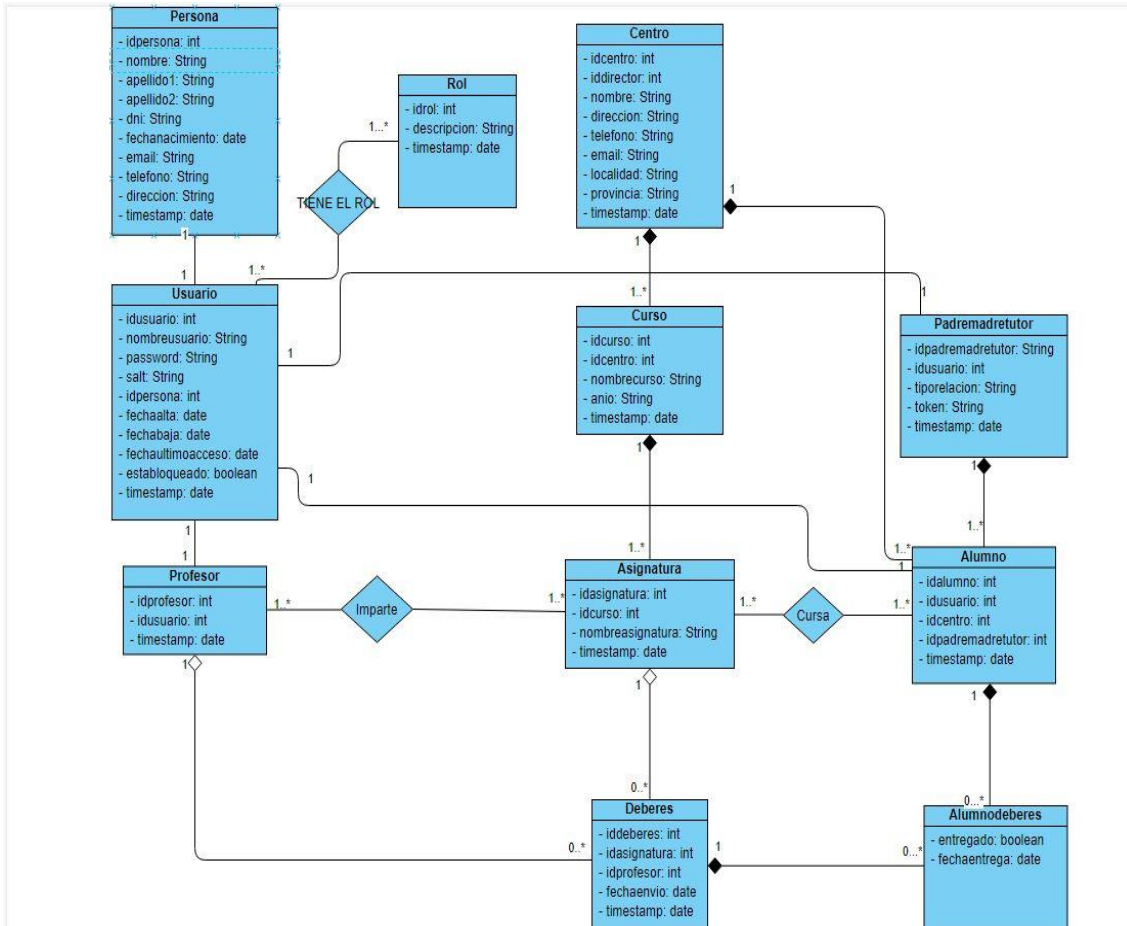


Imagen 17: Modelo de Dominio

6.5.2. Realización de Casos de Uso

En este punto se van a incluir todos los diagramas de secuencias de los casos de uso que hemos definido en el Anexo II, para así obtener una aproximación del funcionamiento real del sistema en cada uno de los casos. Mediante estos diagramas se van a mostrar la comunicación existente entre los elementos que contendrán cada uno de los casos de uso. Un ejemplo de ello, lo podemos encontrar en la Imagen 18:



## Memoria del Proyecto

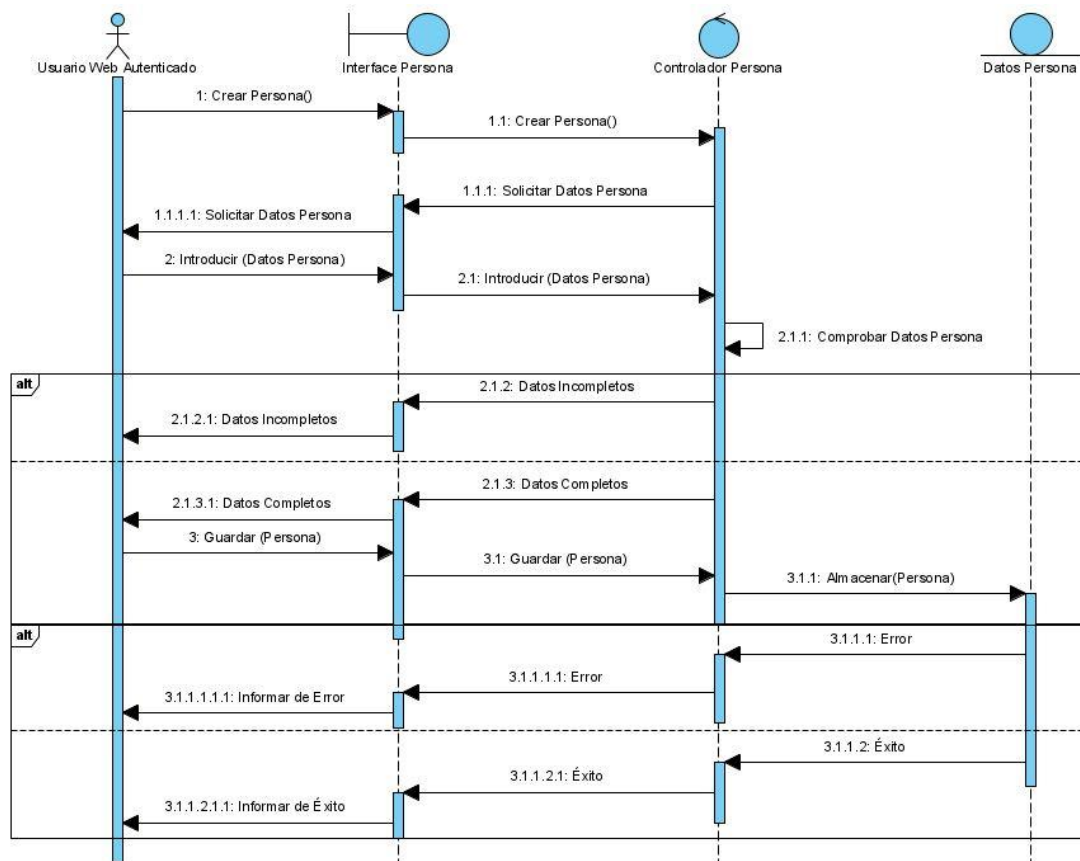


Imagen 18: Diagrama de Secuencia Crear Persona

### 6.5.3. Clases de Análisis

En este punto se mostrarán los diagramas de comunicación para poder observar las comunicaciones y la distribución de clases de análisis de los diagramas descritos en el punto anterior. Dichos diagramas, se agruparán por paquetes, por los cuales se divide el sistema.

### 6.5.4. Vista Arquitectónica del Modelo de Análisis

La vista arquitectónica se ha utilizado para poder situar a cada una de las clases de análisis dentro de la capa propia del patrón Modelo-Vista-Controlador.

El resultado de la arquitectura se puede ver a continuación:

## 6.6. DISEÑO DEL SISTEMA

La fase de diseño del sistema, la cual, a diferencia de la fase de análisis, que describía el dominio del problema, esta se va a centrar en el dominio de la solución. Al tratarse

## Memoria del Proyecto

del dominio de la solución, nos aproximaremos a la implementación, por lo que nombres de clases, métodos, atributos, etc, Serán parecidos a los nombres reales.

Si se desea expandir toda la información que se va a explicar en este punto, proceso de análisis de requisitos, consultar el Anexo IV – Diseño del Sistema Software.

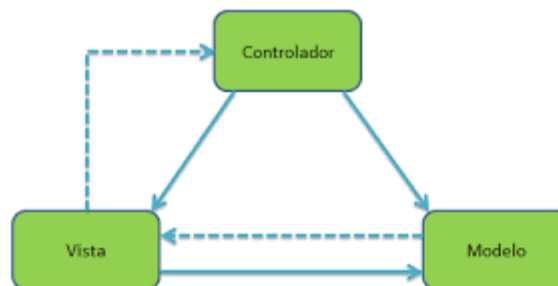
### 6.6.1. Patrones Arquitectónicos

#### Patrón Modelo-Vista-Controlador (MVC)

El MVC, modelo-vista-controlador, es un patrón de arquitectura de software, el cuál separa los datos del sistema, la interfaz del usuario como toda la parte de lógica de control en tres componentes distintos:

- **Modelo:** Es el que contiene la representación de los datos que utiliza el sistema, su lógica de negocio y su persistencia.
- **Vista:** Vista o interfaz de usuario, que forma la información enviada al cliente y la interacción con éste.
- **Controlador:** Es el que actúa como intermediario entre los dos descritos anteriormente, modelo y vista, y el que gestiona todos los flujos de información entre ellos y las transformaciones oportunas para adaptar los datos a las necesidades de cada uno.

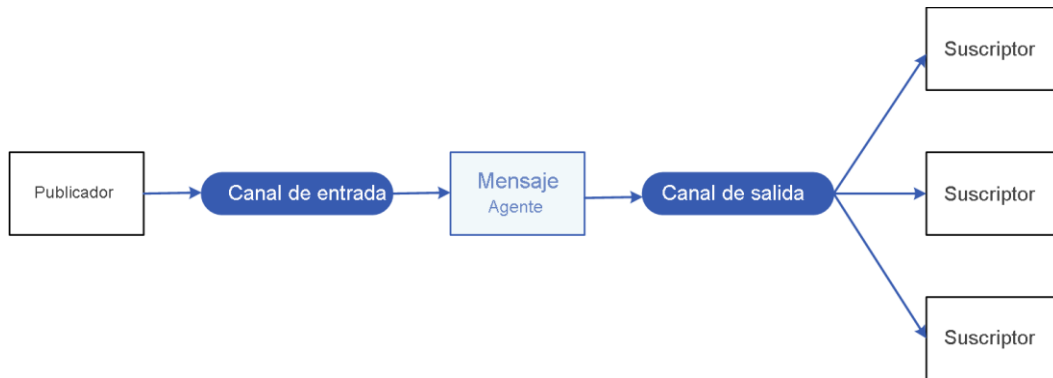
Para entender el patrón, en la *Imagen 19*, se puede observa una representación de MVC:



*Imagen 19: Representación MVC*

### Publicador-Suscriptor

Este patrón es un patrón de mensajería en el que los remitentes de mensajes no programan los mensajes para que se envíen directamente a receptores específicos, sino que se clasifican los mensajes publicados en clases sin saber que suscriptores puede haber, como muestra el esquema de la *Imagen 20*:

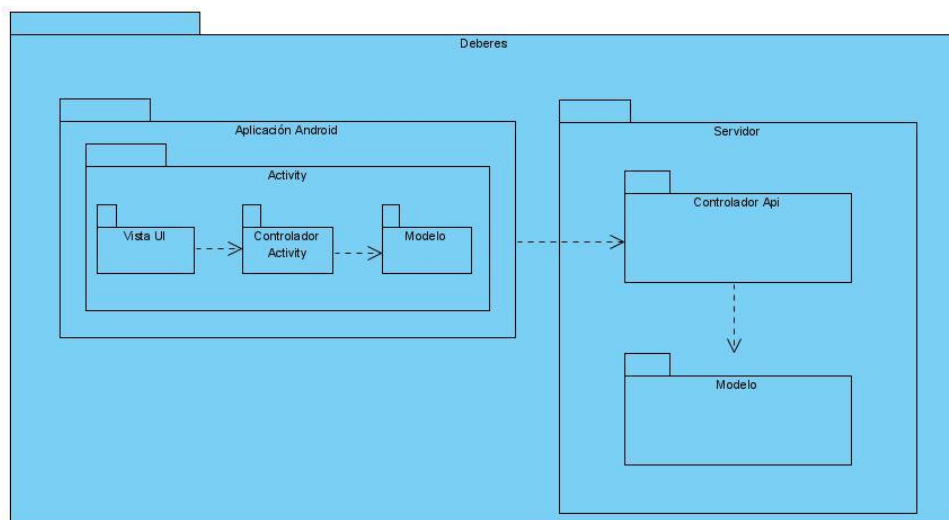


*Imagen 20: Publicador-Suscriptor*

### 6.6.2. Subsistemas de Diseño

En este punto se va a hacer el reparto del sistema en subpaquetes más específicos y se va a hacer la relación existente entre ellos. La *Imagen 21* muestra cual sería el sus

Para más información de estos paquetes se puede ver el apartado 3 del Anexo IV – Diseño del Sistema Software.



*Imagen 21: Subsistema de Diseño*

### 6.6.3. Clases de Diseño

En este punto se han detallado todos los contenidos en cada una de las capas del patrón Modelo-Vista-Controlador de los diferentes paquetes que se han definido en los subsistemas de diseño, donde, dentro de estos están definidos los métodos y atributos que se van a utilizar. En la *Imagen 22*, la *Imagen 23* y la *Imagen 24* podremos ver un ejemplo de las clases de diseño.

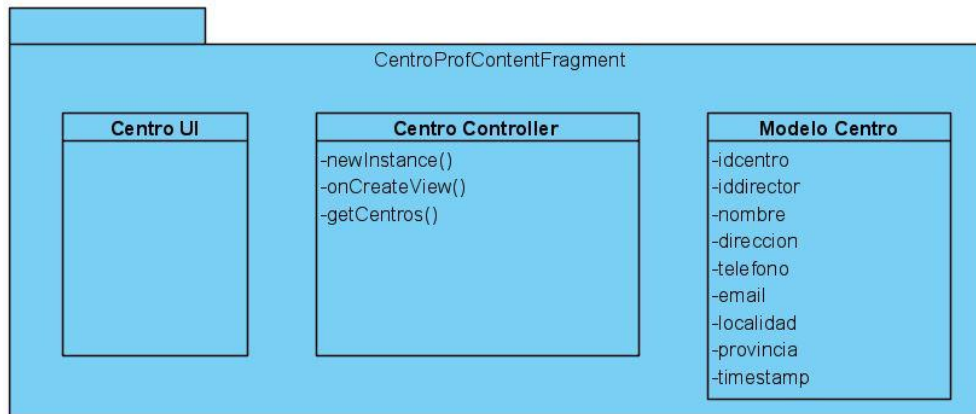


Imagen 22: Centro Profesor Content Fragment

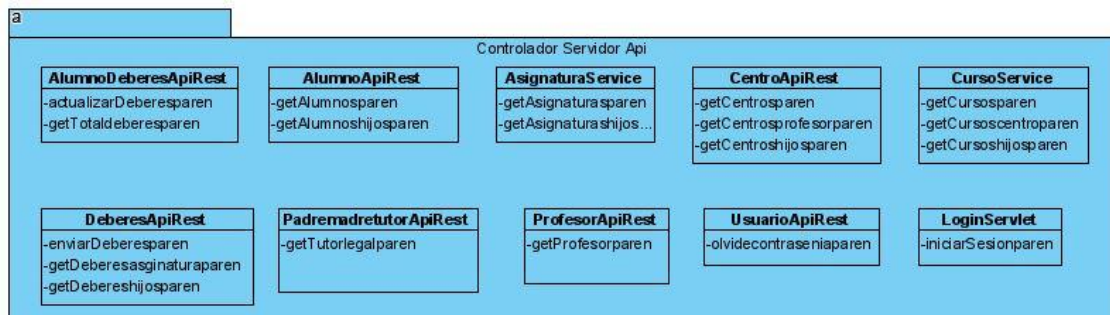


Imagen 23: Controlador Servidor API

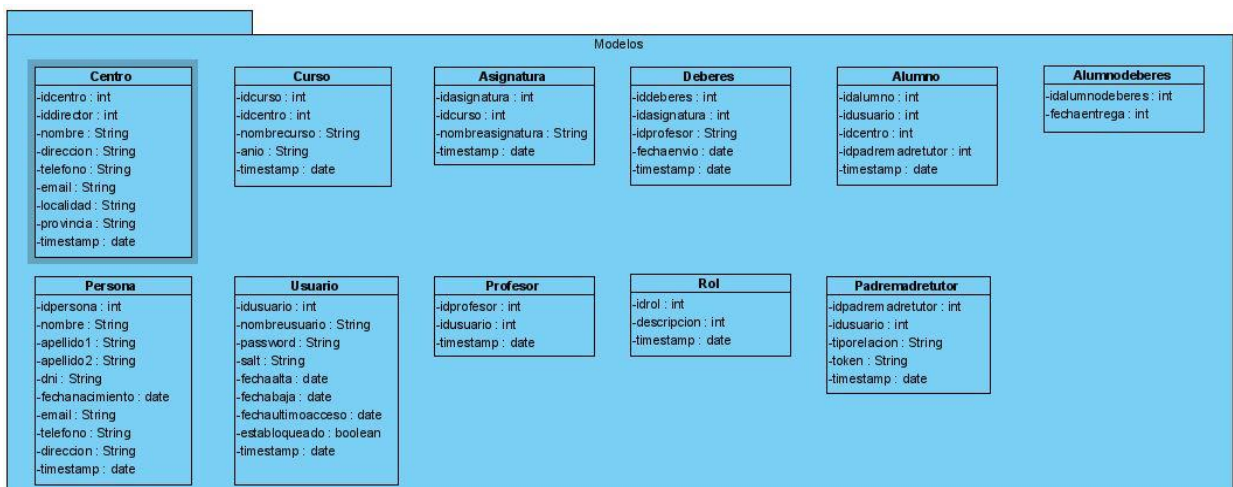
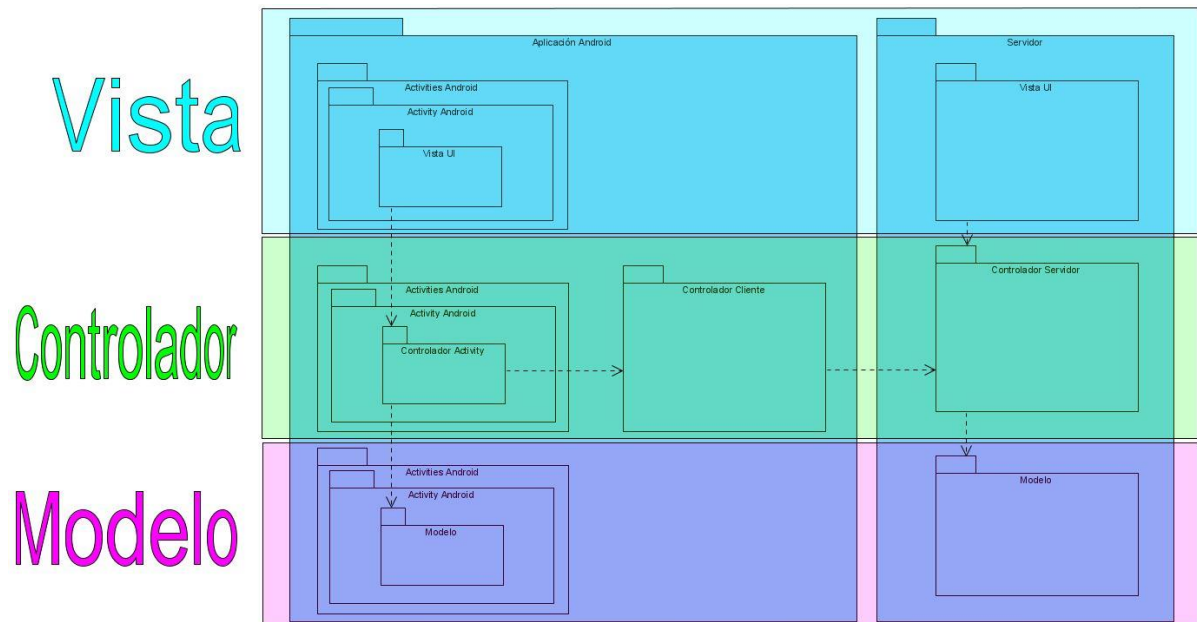


Imagen 24: Modelo Servidor API

### 6.6.4. Vista Arquitectónica del Modelo de Diseño

En este punto se podrá detallar el patrón MVC adaptado para poder así determinar mejor las relaciones y divisiones que presenta el Sistema. La *Imagen 25* se corresponde con nuestra vuestra vista arquitectónica del proyecto.



*Imagen 25: Vista Arquitectónica*

### 6.6.5. Realización de Casos de Uso de Diseño

Para poder realizar los casos de uso del diseño del Software, se van a coger los diagramas de secuencias generados en el Anexo III y se van a refinar para que tenga una aproximación final a los métodos que harán la comunicación existente entre los diferentes componentes. Como ejemplo a la realización de un caso de uso, tenemos la *Imagen 26*, que se muestra a continuación:

## Memoria del Proyecto

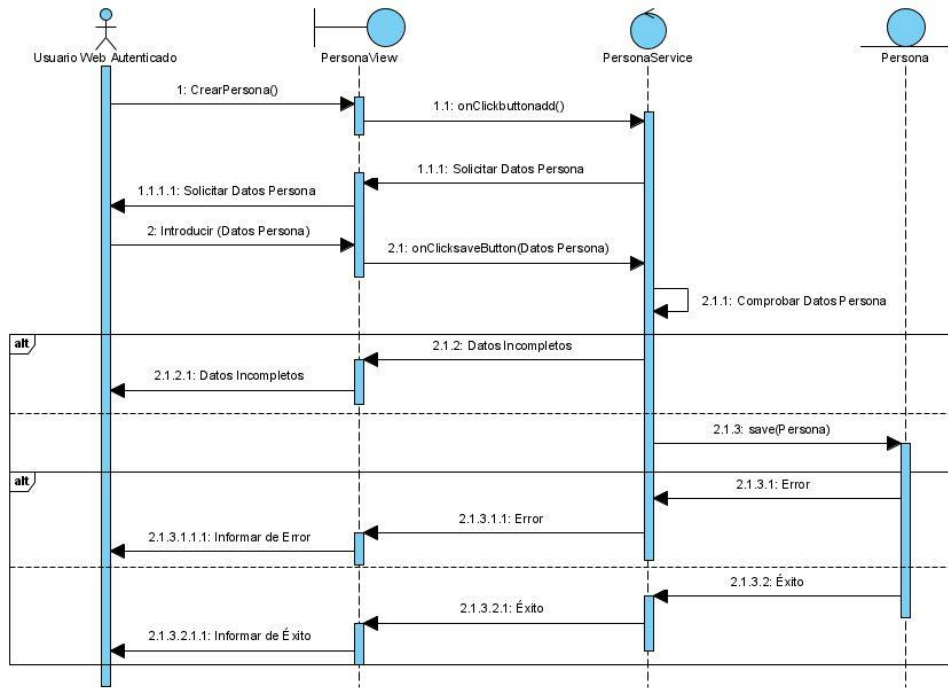


Imagen 26: Diagrama de Secuencia de Crear Persona

### 6.6.6. Modelo de Despliegue

En este punto se adjunta el diagrama de despliegue del sistema, como se puede observar en la *Imagen 27*:

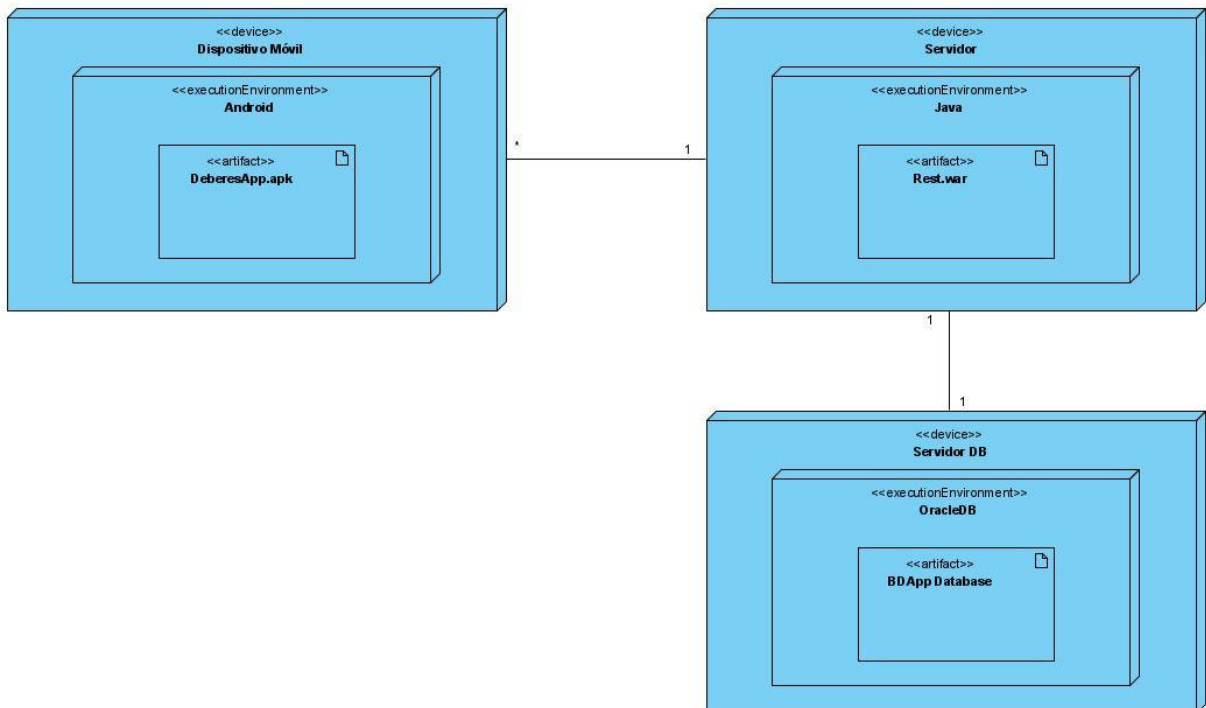


Imagen 27: Diagrama de Despliegue

## Memoria del Proyecto

El sistema se compone de tres nodos:

- **Dispositivo Móvil:** Es un dispositivo donde el entorno de ejecución es Android y donde se encuentra la aplicación a desarrollar. Mediante esta app, los usuarios podrán entrar al sistema y donde se permitirá realizar las acciones detalladas en el documento.
- **Servidor API:** Servidor donde se guardará toda la información manejada por el sistema y cuya Api ayudará a los dispositivos móviles para manejar información. El entorno de ejecución es Java a través de un servidor de Tomcat.
- **Servidor DB:** Es el servidor donde persistirá todos los datos utilizados por la aplicación y el sistema. La base de datos será Oracle.

## 6.7. IMPLEMENTACIÓN

En este punto se van a desarrollar el software teniendo como base todos los resultados que se obtuvieron en la fase de diseño. Es una fase donde se usan y se tienen en cuenta las técnicas y herramientas que se han especificado en el punto 5 de este documento.

Para realizar la implementación, se puede dividir en:

- **Servidor:** Para llevar a cabo el proceso de implementación del servidor web, donde se encuentra toda la parte de administración del sistema, primero se ha llevado a cabo la construcción de la base de datos para posteriormente empezar a generar el modelo de la aplicación en el Servidor.  
Una vez que se ha construido todo el modelo en base de datos y en el servidor, se empieza a construir lo que sería el API Rest, es decir, toda la funcionalidad relacionada con el almacenamiento, la recuperación, la modificación y la eliminación de la información con la base de datos.  
Para hacer todo tipo de pruebas para comprobar el funcionamiento de los servicios REST, se ha utilizado la herramienta de postman, explicada con anterioridad.  
Una vez que comprobamos con esta herramienta que funciona, se ha procedido a desarrollar toda la parte del frontal con el framework de Vaadin, para poder realizar las operaciones de alta, modificación y eliminación de los elementos que intervienen en el sistema, como pueden ser: centros, profesores, etc.  
También en este punto se ha configurado Firebase para las notificaciones a los móviles correspondientes.
- **Aplicación Móvil:** Una vez que se ha desarrollado toda la parte de los servicios rest en el servidor, se ha empezado a implementar toda la parte de la aplicación móvil, empezando por las llamadas a los servicios rest del servidor. Una vez que se comprueba que ambos tienen conexión se ha procedido a desarrollar toda la parte visual de la aplicación, usando todo el código que conecta la aplicación con el servidor, para así poder realizar las peticiones de búsqueda, de creación de deberes y de petición de envío de notificaciones.

## Memoria del Proyecto

Esta parte de implementación se ha procedido a explicarla en el Anexo V – Documentación Técnica, junto con la documentación técnica generada en formato HTML de todo el código, para que pueda ser consultada y que pueda ser entendida para aquel que la consulte.

### 6.8. PRUEBAS

La fase de pruebas es muy importante en el desarrollo de cualquier proyecto, ya que nos va a permitir saber si la funcionalidad del sistema es la correcta y si cumple con los objetivos que se han marcado.

Se ha procedido a hacer pruebas unitarias en cada una de las funcionalidades del sistema durante todo el proceso de desarrollo y también en la finalización de este. Así se va a lograr que dicha funcionalidad va a realizar la acción correctamente.

Una vez implementado todo el sistema, se procede a realizar una prueba global para saber si el sistema va a comportarse de manera correcta después de unificar todos los componentes.

Las pruebas de este proyecto se han realizado con parte de familia y amigos asignándoles a cada uno de ellos alumnos diferentes, con diferentes asignaturas y comprobando que le llegaban las notificaciones correctamente que yo mismo enviaba como Profesor en cada una de estas asignaturas.

### 6.9. FUNCIONALIDADES DEL SISTEMA

Este punto va a ser para resumir los aspectos que son más relevantes de la funcionalidad del sistema. Se va a explicar todo el contenido en función de los diagramas de paquetes de la parte donde se ha hecho la elicitación de requisitos.

Para tener una información completa sobre las funcionalidades del sistema, se recomienda visualizar el Anexo VI – Manual de Usuario.

#### 6.9.1. Gestión de Usuarios

##### Servidor Web

Para acceder a la aplicación podemos acceder a la siguiente URL: <http://localhost:8080/login> o sin la ruta login <http://localhost:8080>. Cabe destacar que actualmente la aplicación se ejecuta en local, pero está preparada para poder ser usada a través de un DNS gratuito. En la *Imagen 28* podemos observar la pantalla de Inicio de Sesión en la web:



## Administración de App Deberes

### Inicio de Sesión

Nombre de Usuario  
admin

Contraseña  
.....













Iniciar Sesión

Imagen 28: Inicio de Sesión

En la *Imagen 29*, se puede observar la Index de Usuarios, donde podremos dar de alta, visualizar y borrar usuarios.

Usuarios

+

Acciones	Nombre Usuario	Nombre	Bloqueado
 			SI No
	admin	Admin	No
	u71958630	Emilia	No
	u72869836	Noelia	No
	u76287068	Moisés	No
	u70687721	Cecilia	No
	u74723848	Antonio	No
	u74174485	Consuelo	No
	u79680192	Urbano	No
	u73066672	Arturo	No
	u77916481	Fabián	No

Total Resultados: 408

« < Página 1 2 de 41 > »

Imagen 29: Index Usuario

En la master de usuarios, correspondiente con la *Imagen 30*, se pueden ver y editar usuarios:

Usuarios [Editar Usuario](#)



Nombre Usuario: admin

Persona: Admin Apellido1 Apellido2

Bloqueado

Roles Asignados

- ROLE\_ADMIN
- ROLE\_DIRECTOR
- ROLE\_PROFESOR
- ROLE\_PMTUTOR
- ROLE\_ALUMNO

Imagen 30: Master Usuario

## Memoria del Proyecto

### Aplicación Móvil

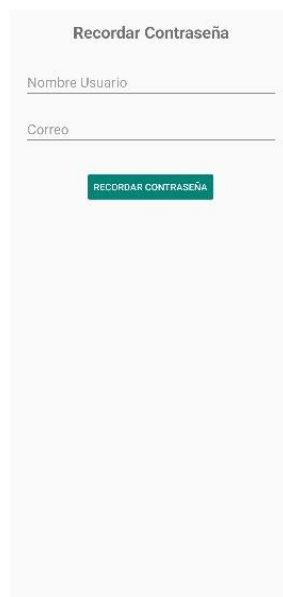
Esta es la pantalla principal de la aplicación, donde introduciremos el usuario y la contraseña correspondiente para iniciar sesión. También para recordar la contraseña.

La *Imagen 31*, se corresponde con el inicio de sesión desde la aplicación móvil:



*Imagen 31: Pantalla Login*

Esta pantalla sirve para recordar la contraseña, solamente debemos introducir nuestro nombre de usuario y nuestro correo, al cual nos llegará la contraseña para poder iniciar correctamente sesión. Como se muestra en la *Imagen 32*, se necesitan el nombre de usuario y el correo asociado a ese usuario:



*Imagen 32: Pantalla Recordar Contraseña*

### 6.9.2. Gestión de Personas

En la *Imagen 33*, se puede observar la Index de Persona, donde podremos dar de alta, visualizar y borrar personas.



Acciones	Nombre	Apellido 1	Apellido 2	Fecha Nacimiento	Email	Teléfono	Dirección
[icon]	Admin	Apellido1	Apellido2	01/01/1900	deberesonline20@gmail...	678978697	
[icon]	Emilia	Álvarez	Parra	03/03/1988	u71958630@hotmail.com	688832141	Calle Ficticia Nº181, 37658
[icon]	Noelia	Vega	Gutiérrez	12/01/1987	u72869836@hotmail.com	612803220	Calle Ficticia Nº161, 37510

*Imagen 33: Index Persona*

La master de personas, mostrada en la *Imagen 34*, nos permite ver y modificar una persona:



Nombre	Apellido 1	Apellido 2	Fecha Nacimiento	Dni
Admin	Apellido1	Apellido2	01/01/1900	00000000V
Email	Teléfono	Dirección		
deberesonline20@gmail	678978697			

*Imagen 34: Master Persona*

### 6.9.3. Gestión de Centros

#### Aplicación Web

En este apartado del menú se crean, modifican o eliminan los distintos centros que estarán disponibles en nuestra aplicación. Como se puede ver en la *Imagen 35*, en la index se muestran los resultados de la búsqueda:



Acciones	Nombre Centro	Dirección	Provincia	Nombre Director	Ape1 Director	Ape2 Director	telefono	email
[icon]	Colegio Público C.R.A. ...	C. Unamuno, 1, 37881	Valdecarros	Fidel	Blanco	Álvarez	923383037	crallanogrande@hotm...
[icon]	IES Leonardo Da Vinci	C. San Francisco, s/n, 3...	Alba de Tormes	Ignacio	Sánchez	Sánchez	923300269	iesleonardodavinci@h...
[icon]	Colegio Santa Isabel	C/ Benitas, 17, 37800	Alba de Tormes	Encarnación	Castro	Reyes	923301908	colegiosantisabel@h...

Total Resultados: 3

*Imagen 35: Index Centro*

## Memoria del Proyecto

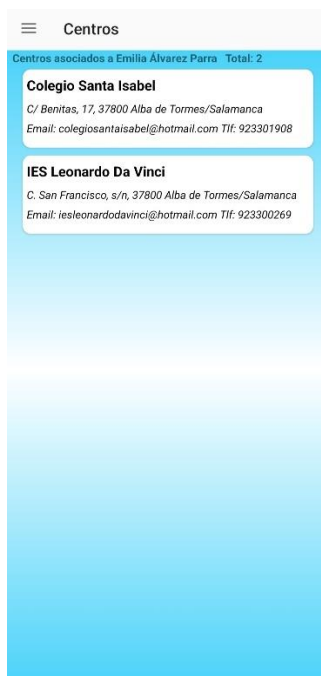
En esta pantalla, mostrada en la *Imagen 36*, se van a poder modificar los centros.

Nombre Curso	Año Curso
1º Primaria	2021/2022
2º Primaria	2021/2022
3º Primaria	2021/2022
4º Primaria	2021/2022
5º Primaria	2021/2022
6º Primaria	2021/2022

*Imagen 36: Master Centro*

## Aplicación Móvil

En esta pantalla tendremos todos los centros, para el usuario que tenga rol de profesor tendrá los centros en los que imparte alguna asignatura y para los padres/madres/tutores los centros donde su hijo cursa alguna asignatura. se puede observar en la *Imagen 37*, donde vemos los centros de un profesor:



*Imagen 37: Pantalla Centros*

## Memoria del Proyecto

Desde los centros podemos acceder directamente a los cursos, asignaturas y a los alumnos, solamente tenemos que dejar pulsado el centro que queramos y nos saldrá el siguiente popup mostrado en la *Imagen 38*:



*Imagen 38: Pantalla Centros Popup*

### 6.9.4. Gestión de Cursos

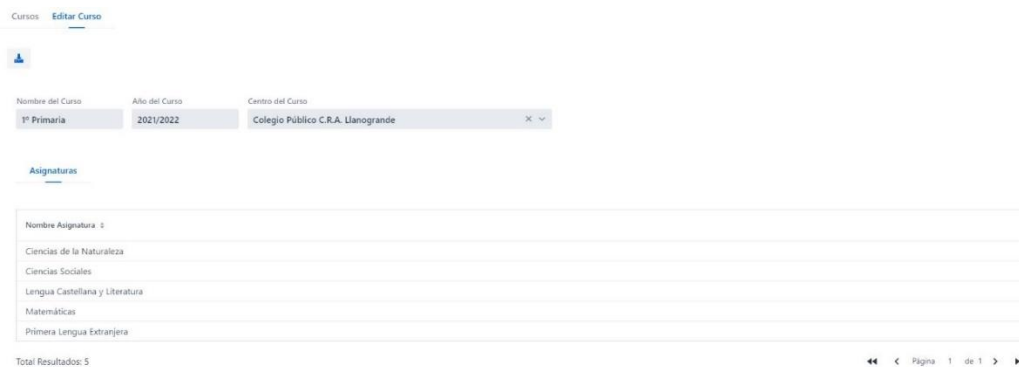
#### Aplicación Web

En este apartado del menú se crean, modifican o eliminan los distintos cursos de la aplicación, como se puede apreciar en la *Imagen 38 e Imagen 39*.



Acciones	Nombre del Curso	Año del Curso	Centro del Curso
	1º Primaria	2021/2022	Colegio Público C.R.A. Llanogrande
	2º Primaria	2021/2022	Colegio Público C.R.A. Llanogrande
	3º Primaria	2021/2022	Colegio Público C.R.A. Llanogrande
	4º Primaria	2021/2022	Colegio Público C.R.A. Llanogrande

*Imagen 38: Index Curso*



Cursos [Editar Curso](#)

Nombre del Curso: 1º Primaria Año del Curso: 2021/2022 Centro del Curso: Colegio Público C.R.A. Llanogrande

Asignaturas

Nombre Asignatura:

- Ciencias de la Naturaleza
- Ciencias Sociales
- Lengua Castellana y Literatura
- Matemáticas
- Primera Lengua Extranjera

Total Resultados: 5

«« < Página 1 de 1 > »»

*Imagen 39: Master Curso*

## Memoria del Proyecto

### Aplicación Móvil

En esta pantalla tendremos todos los cursos, para el usuario que tenga rol de profesor tendrá los cursos en los que imparte alguna asignatura y para los padres/madres/tutores los cursos donde su hijo tiene alguna asignatura. En la *Imagen 40* se puede observar los cursos a los que imparte alguna asignatura el profesor:



*Imagen 40: Pantalla Cursos*

### 6.9.5. Gestión de Asignaturas

### Aplicación Web

En este apartado del menú se crean, modifican o eliminan las distintas asignaturas de la aplicación, que serán asociadas a los distintos cursos elegido en cada una de ellas en la master, pudiéndose ver en la *Imagen 41* e *Imagen 42*.

Asignaturas

+

Acciones	Nombre Asignatura	Nombre del Curso	Año	Nombre del Centro
 	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	Ciencias de la Naturaleza	1º Primaria	2021/2022	Colegio Público C.R.A. Llanogrande
	Ciencias Sociales	1º Primaria	2021/2022	Colegio Público C.R.A. Llanogrande
	Lengua Castellana y Literatura	1º Primaria	2021/2022	Colegio Público C.R.A. Llanogrande
	Matemáticas	1º Primaria	2021/2022	Colegio Público C.R.A. Llanogrande

*Imagen 41: Index Asignatura*

## Memoria del Proyecto

Asignaturas [Editar Asignatura](#)

Nombre Asignatura Curso

Ciencias de la Naturaleza 1º Primaria 2021/2022 Colegio Público C.R.A. Llanogrande

Alumnos

Nombre Alumno	Nombre Padre/Madre/Tutor	Tipo Relación	TIF Padre/Madre/Tutor
Dolores Ibáñez Soto	Joel Bravo Ramos	Padre/Madre/Tutor	663863729
Virginia Domínguez Santos	Sixto Ruiz Delgado	Padre/Madre/Tutor	626880530
Rogelio Cano Carmona	Adela Rey Calvo	Padre/Madre/Tutor	643251723
Augusto Méndez Esteban	Miriam Herrero Herrera	Padre/Madre/Tutor	618226001
Irene Guerrero Herrera	Elena Moreno Ferrer	Padre/Madre/Tutor	698323067
Casiano Herrera Rey	Erico Díaz Molina	Padre/Madre/Tutor	652993021
Benigno Blanco Prieto	Adolfo Benítez Hernández	Padre/Madre/Tutor	635556626
Luciano Vega Ibáñez	Baldomeo Bravo Benítez	Padre/Madre/Tutor	663817778

Total Resultados: 9

« < Página 1 de 2 > »

Imagen 42: Master Asignatura

## Aplicación Móvil

En esta pantalla tendremos todas las asignaturas, para el usuario que tenga rol de profesor tendrá las asignaturas que imparte y para los padres/madres/tutores las asignaturas que cursan sus hijos. En la *Imagen 43*, se pueden ver las diferentes asignaturas que imparte un profesor:



Imagen 43: Pantalla Asignaturas

## Memoria del Proyecto

### 6.9.6. Gestión de Alumnos

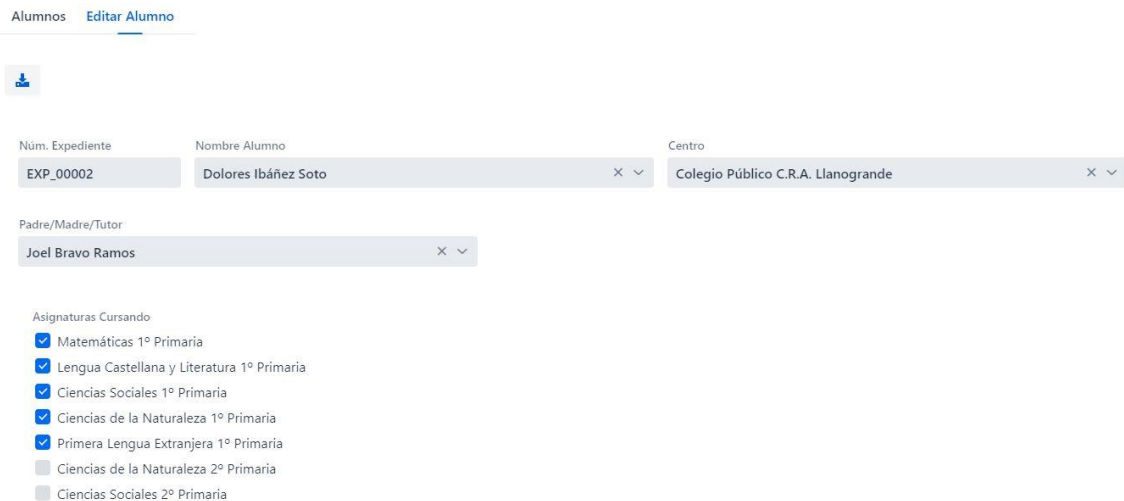
#### Aplicación Web

En este apartado del menú se crean, modifican o eliminan los distintos usuarios de la aplicación. El listbox de la master se cargará con las personas que ya están creadas y que no tienen asociado ningún usuario a ellas. Como ejemplo de ello, tenemos la *Imagen 44* y la *Imagen 45*:



Acciones	Núm. Expediente	Nombre	Apellido 1	Apellido 2	Dni	Centro
	EXP_00002	Dolores	Ibáñez	Soto	73864443N	Colegio Público C.R.A. Llanogrande...
	EXP_00003	Virginia	Domínguez	Santos	72481555E	Colegio Público C.R.A. Llanogrande...
	EXP_00004	Rogelio	Cano	Carmona	78280347P	Colegio Público C.R.A. Llanogrande...
	EXP_00005	Augusto	Méndez	Esteban	78728366X	Colegio Público C.R.A. Llanogrande...

*Imagen 44: Index Alumno*



Núm. Expediente: EXP\_00002

Nombre Alumno: Dolores Ibáñez Soto

Centro: Colegio Público C.R.A. Llanogrande

Padre/Madre/Tutor: Joel Bravo Ramos

Asignaturas Cursando:

- Matemáticas 1º Primaria
- Lengua Castellana y Literatura 1º Primaria
- Ciencias Sociales 1º Primaria
- Ciencias de la Naturaleza 1º Primaria
- Primera Lengua Extranjera 1º Primaria
- Ciencias de la Naturaleza 2º Primaria
- Ciencias Sociales 2º Primaria

*Imagen 45: Master Alumno*



### Aplicación Móvil

En esta pantalla tendremos todos los alumnos, para el usuario que tenga rol de profesor tendrá los alumnos a los que imparte alguna asignatura y para los padres/madres/tutores los hijos que tienen. En la *Imagen 46*, se pueden ver los alumnos de una única asignatura que imparte un profesor, ya que vemos el botón de “*Enviar Deberes*”



*Imagen 46: Pantalla Alumnos*

#### 6.9.7. Gestión de Deberes

En esta pantalla tendremos todos los deberes, para el usuario que tenga rol de profesor tendrá los deberes mandados de cualquier asignatura y para los padres/madres/tutores los deberes que tienen que entregar sus hijos.

En esta pantalla podremos diferenciar 3 colores de los deberes:

- **Color Verde:** Quiere decir que aún hay tiempo para entregarlo.
- **Color naranja:** Quiere decir que los deberes tienen como fecha de entrega el mismo día que se están viendo.
- **Color Rojo:** Quiere decir que el plazo de entrega de los deberes ha pasado.

A continuación, en la *Imagen 47 y 48*, observamos un ejemplo de la parte de gestión de deberes.

## Memoria del Proyecto



Imagen 47: Pantalla Deberes



Imagen 48: Pantalla Modificar Deberes

### 6.9.8. Gestión de Estadísticas

En esta pantalla podremos observar las estadísticas de deberes entregados, de deberes entregados en plazo y de deberes entregados fuera de plazo, como pueden verse en la *Imagen 49*.

The screenshot shows a mobile application interface titled 'Estadísticas' displaying statistics for four subjects:

Asignatura	Alumnos	Deberes Entregados	Porcentaje	Deberes No Entregados	Porcentaje	Deberes Entr. Fuera Plazo	Porcentaje
Inglés	9	1	11,11%	8	88,89%	0	0,00%
Educación Plástica, Visual y Audiovisual	9	2	22,22%	7	77,78%	0	0,00%
Educación Física	9	7	77,78%	2	22,22%	0	0,00%
Ciencias de la Naturaleza	9	1	11,11%	3	33,33%	5	55,56%

Imagen 49: Pantalla Estadísticas

## Memoria del Proyecto

### 6.9.9. Gestión de Profesores

En este apartado del menú se crean, modifican o eliminan los distintos profesores de la aplicación. El listbox de la master se cargará con las personas que ya están creadas y que no tienen asociado ningún profesor a ellas. Tenemos unos ejemplos en la *Imagen 50* y en la *Imagen 51*:



Acciones	Nombre Profesor	Apellido Profesor	Apellido Profesor
	Emilia	Álvarez	Parra
	Noelia	Vega	Gutiérrez
	Moisés	Méndez	Esteban
	Cecilia	Pascual	Velasco

*Imagen 50: Index Profesor*



Profesors [Editar Profesor](#)

Nombre Profesor

Noelia Vega Gutiérrez

Asignaturas Asociadas

- Ciencias de la Naturaleza 1º Primaria Colegio Público C.R.A. Llanogrande
- Ciencias Sociales 1º Primaria Colegio Público C.R.A. Llanogrande
- Lengua Castellana y Literatura 1º Primaria Colegio Público C.R.A. Llanogrande
- Matemáticas 1º Primaria Colegio Público C.R.A. Llanogrande
- Primera Lengua Extranjera 1º Primaria Colegio Público C.R.A. Llanogrande

*Imagen 51: Master Profesor*

### 6.9.10. Gestión de Padremadretutor

En este apartado del menú se crean, modifican o eliminan los distintos usuarios de la aplicación. El listbox de la master se cargará con las personas que ya están creadas y que no tienen asociado ningún usuario a ellas. Tanto la master como la index las podemos observar en la *Imagen 52* e *Imagen 53*:



Acciones	Nombre Padre/Madre/Tutor	Apellido Padre/Madre/Tutor	Apellido Padre/Madre/Tutor	Tipo Relacion
	Joel	Bravo	Ramos	Padre/Madre/Tutor
	Sibto	Ruiz	Delgado	Padre/Madre/Tutor
	Adela	Rey	Calvo	Padre/Madre/Tutor

*Imagen 52: Index Padremadretutor*

# Memoria del Proyecto

Padremadretutores [Editar Padremadretutor](#)



Tipo Relación: Padre/Madre/Tutor  
Nombre Padre/Madre/Tutor: Joel Bravo Ramos

[Hijos](#)

Nombre Hijo	Centro
Dolores Ibáñez Soto	Colegio Público C.R.A. Llanogrande

Total Resultados: 1

« « Página 1 de 1 » »

*Imagen 53: Master Padremadretutor*

## 7. LIMITACIONES DE LA APLICACIÓN

---

Al fin y al cabo, la aplicación que se ha desarrollado es un prototipo que tiene una serie de limitaciones, las cuales pueden ser solucionadas con nuevas versiones del prototipo.

Algunas de ellas pueden ser:

- La aplicación móvil solo tiene compatibilidad para móviles Android, por lo tanto, se tendría que hacer un nuevo desarrollo, creando una aplicación que sea compatible para los móviles Iphone, ya que utilizan un sistema operativo completamente distinto, el IOS.
- Otra de las limitaciones que actualmente tiene el sistema, es que el servidor está pensado en este prototipo para ser desplegado en local, es decir, que no está visible desde fuera, por lo tanto, tanto el servidor como la aplicación móvil debe tener la misma conexión de red. Para solucionar esto, habría que hacer un nuevo evolutivo para poder hacer que se comuniquen estando en redes diferentes.
- La limitación para crear usuarios, personas nuevas y tal, ya que solamente puede ser, por ahora, para usuario administradores y tiene que ser desde la web.

## 8. CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO

---

Este punto va a recoger todas las conclusiones que se han obtenido una vez que se han dado por finalizadas todas las fases de desarrollo del proyecto “*Sistema de Gestión de Deberes en Centros de Primaria y Secundaria*”. También se comentarán los trabajos que se pueden hacer en un futuro para mejorar la aplicación.

### 8.1. CONCLUSIONES

Una vez que se han podido satisfacer los objetivos que se habían redactado acerca del sistema, podemos establecer las siguientes conclusiones:

- **Gestión de Usuarios:** El sistema nos va a permitir la gestión de usuarios mediante el alta, la baja o modificación de los usuarios.
- **Gestión de Centros:** El sistema cumple con el objetivo planteado para la gestión de los centros, permitiendo así la creación, modificación o eliminación de los centros.
- **Gestión de Asignaturas:** La gestión de las asignaturas se ve satisfecha en el sistema, ya que se pueden dar de alta, borrar o modificar las asignaturas.
- **Gestión de Alumnos:** Se pueden dar de alta, modificar o eliminar alumnos satisfaciendo así un objetivo del sistema.
- **Gestión de Profesores:** El objetivo de dar de alta, modificar o eliminar profesores en el sistema se ha cumplido, pudiendo así, administrar esta parte.
- **Gestión de Padre/Madre/Tutor:** Dar de alta, baja o modificar padres, madre o tutores legales en la administración, es posible, por lo tanto, este objetivo queda alcanzado.
- **Gestión de Estadísticas:** Se pueden generar estadísticas de deberes, lo que hace que se alcance el objetivo del sistema.
- **Gestión de Deberes:** Se permite el envío y modificación de los deberes por parte de los profesores.

Se ha desarrollado un sistema logrando que sea estable y que cumple con todas las funcionalidades requeridas y que se establecieron en los objetivos del sistema. A parte, se ha diseñado y desarrollado una aplicación web y una aplicación móvil con la finalidad de gestionar el sistema, fortaleciendo el desarrollo web y el de interfaces de usuario.

Para finalizar, se concluye que se ha logrado el proyecto hasta conseguir un prototipo funcional de la aplicación.

Otra de las conclusiones que se obtiene con todo el desarrollo del proyecto, es la de establecer un calendario y de tener las dificultades de establecer el tiempo necesario para cada una de las tareas en las que se divide el desarrollo del proyecto software. Como apunte aparte, esto va a ser muy necesario a la hora de tener un trabajo

## Memoria del Proyecto

estable en la que te pidan estimar cuanto te va a llevar una tarea (Incidencia o evolutivo, en mi caso), para que puedas cumplir con dicho objetivo.

Como conclusión, se ha puesto en práctica todo lo adquirido en conocimiento de I grado de Ingeniería Informática, ayudado con los conocimientos adquiridos con la experiencia laboral.

## 8.2. LÍNEAS FUTURAS DE TRABAJO

Las líneas futuras de trabajo que se podría seguir en este proyecto, según mi punto de vistas, podría ser:

- La creación de una imagen Docker, para poder ser desplegada en la nube, ya sea gratis o de pago, y que así la aplicación web pudiera desplegarse en la nube y la aplicación fuera funcional desde cualquier punto del mapa.
- Otra línea sería dotar al usuario Administrador la capacidad de poder modificar datos desde la aplicación móvil.
- Poder extender la aplicación más allá de secundaria, cómo, por ejemplo, a bachillerato o a ciclos formativos.
- También vería como línea futura la utilización, solo como parte informativa, a los alumnos para que así pudieran tener un calendario de entrega de deberes y este pudiera servir como agenda personal, ya que hoy en día cualquier chico de secundaria tiene un móvil.

El sistema permite el alta, baja y modificación de cursos, por lo tanto, el objetivo se ha cumplido.

## 9. REFERENCIAS

---

- *Documentación Postman* <https://www.postman.com/>
- *Documentación Visual Studio* <https://visualstudio.microsoft.com/es/vs/>
- *Documentación Visual Studio Code.* <https://code.visualstudio.com/>
- Durán Toro, A., & Bernárdez Jiménez. *Metodología para la Elicitación de Requisitos de Sistemas Software.*
- García Peñalvo, F. J., & García Holgado, A. *Apuntes de la Asignatura Ingeniería del Software I del Grado en Ingeniería Informática.*
- *Introducción a REM 1.2.2.*  
<https://www.infor.uva.es/~mlaguna/is1/materiales/Intro-REM.pdf>

## Memoria del Proyecto

- Larman, C. (2002). *Applying UML and Patterns: Introduction to Object-Oriented Analysis and Design and the Unified Process*. Second Ed. Prentice Hall.
- App Homework (Tarea):  
<https://play.google.com/store/apps/details?id=klwinkel.huiswerk&hl=es&gl=US>  
(Visitado 31/08/2022)
- App Homework: <https://spartanhack.com/12-mejores-aplicaciones-para-tareas-escolares/> (Visitado en 31/08/2022)
- Api Rest: <https://www.ibm.com/es-es/cloud/learn/rest-apis> (Visitado en 31/08/2022)
- Java: <http://www.itlp.edu.mx/web/java/Tutorial%20de%20Java/Intro/carac.html>  
(Visitado en 31/08/2022)
- Vaadin: <https://www.adictosaltrabajo.com/2015/11/30/introduccion-a-vaadin/>  
(Visitado en 31/08/2022)
- Oracle: <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/oracle-database/> (Visitado en 31/08/2022)
- Spring: <https://openwebinars.net/blog/conoce-que-es-spring-framework-y-por-que-usarlo/> (Visitado en 31/08/2022) <https://www.campusmvp.es/recursos/post/que-son-spring-framework-y-spring-boot-tu-primer-programa-java-con-este-framework.aspx> (Visitado en 31/08/2022)
- Spring Tools: <http://www.davidmarco.es/articulo/integracion-de-eclipse-y-springsource-tool-suite> (Visitado en 31/08/2022)
- Andorid Studio: <https://developer.android.com/studio/intro?hl=es-419> (Visitado en 31/08/2022)
- Proceso unificado:  
[https://es.wikipedia.org/wiki/Proceso\\_unificado#:~:text=El%20Proceso%20Unificado%20de%20Desarrollo,por%20ser%20iterativo%20e%20incremental.](https://es.wikipedia.org/wiki/Proceso_unificado#:~:text=El%20Proceso%20Unificado%20de%20Desarrollo,por%20ser%20iterativo%20e%20incremental.) (Visitado en 31/08/2022)