



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Sistema de bajo coste para la generación automática de partes de firmas utilizando el carné de la UPV

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Adrián Ponce Martínez

**Tutor:** Juan Carlos Ruiz García

**2º Tutor:** Roberto Capilla Lladró

**Curso:** 2019-2020



# Resumen

---

Actualmente disponemos de grandes avances tecnológicos que hacen nuestra vida más sencilla. Nos permiten realizar gestiones de una manera más automatizada y con menos esfuerzo. El control de asistencia es un proceso necesario tanto en el ámbito laboral, como en el académico. Por esa razón, en este trabajo de fin de grado se emplea una de estas tecnologías con el fin de desarrollar una aplicación para la generación automática de partes de firmas mediante un dispositivo móvil con sistema operativo Android.

Con dicha aplicación, se tiene como objetivo agilizar el proceso de firma en la asistencia de un alumno mediante una interfaz intuitiva y de fácil uso. Permitirá a los usuarios contabilizar la asistencia en sus clases mediante partes de firmas generados de forma automática, visualizar información de dichos partes y almacenarla personalmente.

Para su implementación, se ha utilizado el Entorno de Desarrollo Integrado (IDE) Android Studio, tecnologías como el NFC para lecturas de carnés UPV, escaneo de códigos QR y mecanismos de extracción automatizada de datos (Web Scraping) para obtener información de alumnos y profesores.

**Palabras clave:** partes de asistencia, Android, NFC, QR, carnés, UPV.





# Resum

---

Actualment disposem de grans avanços tecnològics que fan nostra vida més senzilla. Ens permeten realitzar gestions d'una manera més automatitzada i amb menys esforç. El control d'assistència és un procés necessari tant en l'àmbit laboral, com en l'acadèmic. Per eixa raó, en aquest treball de fi de grau s'empra una d'estes tecnologies a fi de desenrotllar una aplicació per a la generació automàtica de parts de firmes per mitjà d'un dispositiu mòbil amb sistema operatiu Android.

Amb la dita aplicació, es té com a objectiu agilitzar el procés de firma en l'assistència d'un alumne per mitjà d'una interfície intuïtiva i de fàcil ús. Permetrà als usuaris comptabilitzar l'assistència en les seues classes per mitjà de parts de firmes generats de forma automàtica, visualitzar informació dels dits parts i emmagatzemar-la personalment.

Per a la seua implementació, s'ha utilitzat l'Entorn de Desenvolupament Integrat (IDE) Android Studio, tecnologies com el NFC per a lectures de carnets UPV, escaneig de codis QR i mecanismes d'extracció automatitzada de dades (Web Scraping) per a obtindre informació d'alumnes i professors.

**Keywords:** parts de firmes, Android, NFC, QR, carnets, UPV.





# Abstract

---

Currently we have great technological advances that make our lives easier. They allow us to carry out procedures in a more automated way, with less effort. Attendance control is a necessary process in the workplace and academically. For this reason, one of these technologies is used in this final project to develop an application will be developed for the automatic generation of parts of signatures using a mobile device with Android operating system.

With this application, the objective is to streamline the signing process in the attendance of a student through an intuitive and easy-to-use interface. It will allow users to record attendance in their classes through automatically generated parts of signatures, view information on those parts and store it personally.

For its implementation, the Android Studio Integrated Development Environment (IDE), technologies such as NFC for reading UPV meats, QR code scanning and automated data extraction mechanisms (Web Scraping) have been used to obtain information from students and teachers.

**Keywords:** parts of signatures, Android, NFC, QR, meats, UPV.







# Índice de contenidos

---

1.	Introducción .....	17
1.1.	Motivación.....	17
1.1	Objetivos .....	18
1.1	Estructura de la Memoria .....	19
1.1	Conclusión.....	19
2.	El estado del arte .....	21
2.1	Conclusión.....	21
3.	Especificación .....	23
3.1	Requisitos no funcionales.....	23
3.1.1	Usabilidad .....	23
3.1.2	Compatibilidad .....	23
3.1.3	Eficacia .....	24
3.1.4	Mantenibilidad.....	24
3.1.5	Consistencia .....	24
3.1.6	Integridad .....	24
3.2	Requisitos funcionales .....	24
3.3	Diagrama de casos de uso .....	25
3.4	Identificación de los casos de uso .....	26
3.5	Mockups.....	44
3.6	Modelo de dominio.....	51
3.7	Conclusión.....	52
4.	Diseño .....	53
4.1	Estructura del Proyecto .....	53
4.2	Esquema de components .....	54
4.3	Arquitectura de la aplicación .....	55
4.4	Arquitectura de la base de datos.....	56
4.5	Conclusión.....	57
5.	Tecnologías y herramientas .....	59
5.1	Android .....	59
5.2	SQLite.....	60
5.3	RFID (NFC).....	60
5.4	Código QR.....	61
5.5	Balsamic.....	62



5.6	Adobe Photoshop .....	62
5.7	BoUml.....	62
5.8	Librerías .....	62
5.9	Conclusión.....	62
6.	Implementación .....	63
6.1	Actividades .....	63
6.1.1	Lógica funcional .....	63
6.1.2	Diseñador de layouts .....	63
6.1.3	Actividades asignatura.....	63
6.1.4	Actividades professor .....	65
6.1.5	Actividades alumno.....	67
6.1.6	Actividades grupo .....	68
6.1.7	Actividades creación parte de firmas .....	69
6.1.8	Actividades búsqueda parte de firmas .....	71
6.2	Comunicación con la base de datos.....	71
6.3	Extracción de datos de la TUI.....	73
6.4	Lectura NFC .....	74
6.5	Creación del archivo PDF y XML .....	74
6.6	Carga y estructura de asignaturas y grupos .....	77
6.7	Envío de datos mediante correo electrónico.....	79
6.8	Conclusión.....	79
7.	Pruebas sobre la aplicación.....	81
7.1	Pruebas de usuario.....	81
7.2	Pruebas de carga.....	84
8.	Conclusión .....	85
8.1	Desarrollo futuro .....	86
8.2	Relación del Proyecto con los estudios cursados .....	87
9.	Bibliografía .....	89
	Lista de acrónimos .....	91

# Índice de figuras

---

Figura 1. Diagrama de casos de uso .....	25
Figura 2. Ventanas de eventos con asignaturas .....	45
Figura 3. Ventanas de eventos con profesores .....	46
Figura 4. Ventanas de eventos con alumnos .....	47
Figura 5. Ventanas de eventos con grupos .....	48
Figura 6. Ventanas de eventos generar parte de firmas.....	49
Figura 7. Ventanas de eventos para enviar correo electrónico .....	50
Figura 8. Ventanas de eventos para buscar parte de firmas .....	50
Figura 9. Modelo de dominio .....	51
Figura 10. Estructura del proyecto .....	53
Figura 11. Esquema de componentes .....	54
Figura 12. Esquema de la arquitectura de la aplicación .....	55
Figura 13. Esquema de la base de datos.....	57
Figura 14. Arquitectura Android.....	59
Figura 15. Historia versiones Android .....	60
Figura 16. Carné UPV .....	61
Figura 17. Reverso carné UPV.....	61
Figura 18. Pantallas de eventos con asignaturas .....	63
Figura 19. Pantallas de eventos con profesores.....	65
Figura 20. Pantallas de eventos con alumnos.....	67
Figura 21. Pantallas de eventos con grupos .....	68
Figura 22. Pantallas de eventos para generar un parte de firmas .....	69
Figura 23. Pantallas de eventos para enviar correo electrónico .....	71
Figura 24. Parte de firmas .....	75
Figura 25. Primera pregunta en las pruebas de usuario.....	81
Figura 26. Segunda pregunta en las pruebas de usuario.....	82
Figura 27. Tercera pregunta en las pruebas de usuario .....	82
Figura 28. Cuarta pregunta en las pruebas de usuario.....	83
Figura 29. Quinta pregunta en las pruebas de usuario .....	83





# Índice de tablas

---

Tabla 1. CU Añadir asignaturas .....	26
Tabla 2. CU Visualizar asignaturas.....	27
Tabla 3. CU Eliminar asignatura.....	27
Tabla 4. CU Visualizar datos de la asignatura .....	28
Tabla 5. CU Editar datos asignatura.....	28
Tabla 6. CU Visualizar profesores de la asignatura .....	29
Tabla 7. CU Visualizar alumnos de la asignatura .....	29
Tabla 8. CU Visualizar grupos de la asignatura.....	30
Tabla 9. CU Añadir profesor individual.....	30
Tabla 10. CU Añadir profesores por defecto .....	31
Tabla 11. CU Visualizar datos de un profesor .....	32
Tabla 12. CU Editar datos de un profesor .....	33
Tabla 13. CU Eliminar profesor.....	33
Tabla 14. CU Añadir alumno individual.....	34
Tabla 15. CU Añadir alumnos por defecto.....	35
Tabla 16. CU Visualizar datos de un alumno.....	35
Tabla 17. CU Editar datos de un alumno .....	36
Tabla 18. CU Eliminar alumno .....	37
Tabla 19. CU Añadir grupos de asignaturas.....	37
Tabla 20. CU Visualizar datos del grupo .....	38
Tabla 21. CU Editar datos de un grupo.....	39
Tabla 22. CU Eliminar grupo.....	40
Tabla 23. CU Crear PDF de parte de firmas .....	40
Tabla 24. CU Enviar PDF y datos del parte de firmas por correo.....	42
Tabla 25. CU Consultar datos de partes de firmas .....	42
Tabla 26. CU Vaciar base de datos .....	43
Tabla 27. Características de los dispositivos móviles .....	84
Tabla 28. Resultados pruebas de carga .....	84





# Índice de fragmentos de código

---

Fragmento 1. Método de borrado.....	72
Fragmento 2. Método de inserción.....	72
Fragmento 3. Método de consulta.....	72
Fragmento 4. Método de actualización.....	72
Fragmento 5. Método de acceso web mediante lectura de código QR.....	73
Fragmento 6. Código para la extracción de datos HTML.....	73
Fragmento 7. Código para la lectura del identificador por NFC.....	74
Fragmento 8. Creación y apertura del archivo PDF.....	75
Fragmento 9. Creación de cabecera y pie.....	76
Fragmento 10. Creación de título o párrafo.....	76
Fragmento 11. Creación de tabla.....	76
Fragmento 12. Código creación de archivo XML.....	76
Fragmento 13. Estructura archivo XML del parte de firmas.....	77
Fragmento 14. Método lectura archivo XML.....	78
Fragmento 15. Estructura archivo asignaturas.xml.....	78
Fragmento 16. Estructura archivo asignaturas.xml.....	79
Fragmento 17. Método para el envío de archivos por correo.....	79







# 1. Introducción

---

## 1.1. Motivación

El continuo hallazgo de nuevas tecnologías está a la orden del día y cada vez son más las diversas utilidades que presentan, de las cuales podemos hacer uso en los diferentes ámbitos de nuestra vida.

Por ello, las personas buscan continuamente comodidad y eficacia en todas las labores que desempeñan y esta búsqueda se agrava aún más, si para lograr su objetivo deben de hacer uso de un recurso intermedio.

Dichas tecnologías llevadas al uso cotidiano pueden verse integradas en los diferentes dispositivos digitales como ordenadores, smartphones y tabletas que, debido a su gran presencia entre la sociedad, promueven el uso de estas tecnologías a diario y de forma casi inintencionadamente. Centrándonos un M en las tecnologías empleadas para la implementación de este proyecto, hablamos de la tecnología de comunicación de campo cercano, más comúnmente conocida como NFC<sup>1</sup>.

Se trata de una tecnología que permite la comunicación inalámbrica y el intercambio o recogida de datos en dispositivos digitales. Esta comunicación e intercambio de datos es el mecanismo empleado a la hora conocer la información de nuestros usuarios.

Como alumno de la Universidad Politécnica de Valencia, conozco y he sido participe de la metodología empleada para reflejar las asistencias en las diferentes asignaturas que componen el grado. Basándome en ello y teniendo presente el afán de mejorar continuamente, he dedicado este proyecto a mejorar el proceso de creación de partes de asistencias para que mediante la tecnología NFC se pueda llevar a cabo de forma automática sin que el alumno deba de firmar y sin necesidad de generar e imprimir previamente un documento para ello. Además, teniendo en cuenta posibles necesidades de consulta por parte del usuario, he desarrollado una metodología de búsqueda y almacenamiento de toda la información que compone un parte de asistencia.

Encuentro en este proyecto una gran oportunidad de poder poner en práctica conceptos adquiridos, que me permitirán cumplir con los objetivos propuestos para esta aplicación e inclusive el perfeccionamiento de estos para seguir mejorando personalmente a nivel de conocimientos.

El hecho de poder usar una tecnología jamás vista personalmente a nivel de desarrollo en un procedimiento que he llevado a cabo durante los años de mi formación académica y con el fin de mejorarlo, ha influido notablemente a la hora de seleccionar este proyecto de fin de carrera.

---

<sup>1</sup> NFC: tecnología desarrollada para conectar dispositivos de manera inalámbrica, de corto alcance y a altas frecuencias.

## 1.1 Objetivos

El objetivo principal de este proyecto es el desarrollo de una aplicación móvil de Android que permita al usuario generar partes de firmas de forma automática mediante el uso de la tecnología NFC. Por otro lado, el usuario también podrá consultar la información pertinente a los partes de firmas, gestionar sus datos propios, de su asignatura y sus alumnos. Tras la creación del parte de firmas, el usuario tiene la posibilidad de almacenar y visualizar dicho parte en formato PDF<sup>1</sup> y su información en formato XML<sup>2</sup>, así como de enviarlos por correo electrónico a cualquier destinatario.

Para dotar la interfaz con un carácter atractivo, se ha diseñado siguiendo unos patrones de diseño y estilo de Google Material Design consiguiendo un carácter visualmente más formal. Además de una interfaz atractiva, otro objetivo que se ha tenido en cuenta es la eficacia y la usabilidad, convirtiendo la aplicación en un recurso fácil, intuitivo y funcional para cualquier usuario.

Además de la facilidad y el correcto funcionamiento a la hora de utilizarla, también se ha pensado en la eficiencia, lograr las metas establecidas sin llegar a saturar al usuario con tanta información. En toda la información mostrada se ha asegurado su veracidad y mantiene la coherencia para que el usuario no pueda malinterpretarla.

Entre otros objetivos secundarios cabe destacar la posibilidad de poder generar partes de firmas cuando se contabiliza la asistencia a una clase extraoficial fuera del horario lectivo. La limitación del usuario a la hora de gestionar la información o dar de alta entidades o alumnos, a la hora de escanear el carné UPV se identifica la identidad del usuario impidiendo que sea dado de alta erróneamente o notificando que su identidad no es la esperada por el usuario. Evitar también la duplicación por parte del alumno de su asistencia en el parte y reservando exclusivamente la creación de dichos partes a profesores identificados y dados de alta en la aplicación.

---

<sup>1</sup> PDF: formato para documentos digitales compuesto por imagen vectorial, mapa de bits y texto.

<sup>2</sup> XML: sistema que permite definir lenguajes que etiquetan y organizan información en un documento.

## **1.1 Estructura de la Memoria**

En este punto se va a explicar brevemente la estructura y el contenido de las distintas partes que componen este proyecto.

En primer lugar, en el capítulo 2 compararemos nuestra aplicación con las diferentes herramientas que existen en el mercado. En el capítulo 3, analizaremos los requisitos y propiedades del sistema tanto a nivel de diseño como de implementación que debe cumplir el sistema, se expondrá un esquema con el comportamiento deseado de la aplicación para más tarde poder describirlos con detalles, se representará gráficamente las diferentes actividades que componen las ventanas de la aplicación y finalmente identificaremos las diferentes relaciones entre las entidades presentes en la aplicación. En el capítulo 4, nos centraremos en el diseño de la aplicación, familiarizándonos con la estructura que sigue para ser organizada, el esquema de los componentes que la forman, las diferentes capas que la componen y la estructura que compone la base de datos. En el capítulo 5, se mencionará las diferentes Herramientas y tecnologías empleadas en la elaboración del proyecto. En el capítulo 6, nos paramos a comentar los aspectos más importantes a la hora de la implementación y como se ha llevado a cabo. En el capítulo 7, se hará énfasis en las distintas pruebas realizadas a la aplicación. Concluyendo con el capítulo 8, mencionaremos posibles desarrollos futuros en cuanto a avances, la relación entre los estudios cursados y el proyecto y finalmente, se citarán las referencias y acrónimos empleados.

## **1.1 Conclusión**

Tras hacer una introducción de nuestro proyecto mencionando los aspectos que nos han motivado para su desarrollo, los objetivos a cumplir establecidos y la estructura que va a seguir nuestra memoria, damos paso a analizar si las distintas soluciones existentes en el mercado cumplen de igual manera los objetivos planteados para nuestra aplicación.





## 2. El estado del arte

---

Actualmente, y ajustándonos a la legalidad vigente, a todas las empresas se les exige el control de asistencia a sus empleados. Debido a ello, han sido muchas las entidades que han ofrecido una solución software para ello. Hablando del mismo tema, pero en distinto ámbito en este caso el académico, en muchas instituciones se sigue el mismo control de asistencia. Esta vez, en los alumnos.

Tras una investigación sobre las posibles alternativas existentes para controlar la asistencia de los alumnos, hemos encontrado diversas herramientas empleadas para ello como son Alexia, Dinantia, Esemia, iEduca, entre otras... Estas aplicaciones cumplen con la tarea principal de registrar la asistencia del alumno a clase, pero es el alumno quien directamente accede a la aplicación y marca su asistencia rellenando un formulario con sus datos personales o bien mediante la selección en un registro [1].

Nuestra aplicación obtiene el mismo resultado con otra metodología, se sabe que cada alumno posee un carné que lo identifica como alumno de la universidad, este carné se trata de una tarjeta TUI<sup>1</sup> la cual alberga toda la información referente al alumno. Mediante este carné el alumno registra su asistencia en la aplicación únicamente colocándolo cerca del dispositivo para que el lector NFC pueda escanearlo.

Esta metodología dota a la aplicación de unicidad al emplear un recurso exclusivo de la Universidad Politécnica de Valencia ya que no existe otra herramienta que utilice dicho carné para registrar la asistencia.

Por otro lado, hemos tenido en cuenta el coste a la hora de comparar nuestra aplicación con otras posibles soluciones. Este proyecto tiene como objetivo la creación de un sistema de bajo coste, coste que con nuestra aplicación se vería reducido a nulo debido a que no necesita hacer uso de ningún recurso externo. El resto de las aplicaciones no cumplirían con este objetivo ya que hacer uso de estas conlleva un pago por sus servicios.

### 2.1 Conclusión

Tras lo expuesto anteriormente, tenemos claro que nuestra aplicación es la única que cumple los objetivos expuestos haciendo uso de una metodología en la que intervienen una serie de elementos específicos. El siguiente capítulo trata sobre la especificación de los diferentes requisitos con los que debe contar nuestra aplicación. Además, identificaremos mediante un diagrama los posibles casos de uso para posteriormente detallarlos uno a uno. Para finalizar mostraremos el mockup empleado en el diseño y el modelo de dominio de nuestra aplicación

---

<sup>1</sup> TUI: Tarjeta universitaria inteligente que acredita al protador como miembro de la comunidad universitaria.





# 3. Especificación

---

A la hora de desarrollar un software, se debe tener muy en cuenta la especificación de los requisitos. Surge como necesidad para informar y ayudar a los desarrolladores a analizar y comprender todos los requisitos y requerimientos deseados [2]. Esta labor es fundamental a la hora de definir unas correctas funcionalidades.

A continuación, vamos a diferenciar entre dos tipos de requisitos.

## 3.1 Requisitos no funcionales

Los requisitos no funcionales hacen referencia a las propiedades del sistema que este debe de cumplir [3].

Dependiendo de sus implicaciones, dichos requisitos se clasifican de la siguiente forma.

### 3.1.1 Usabilidad

Consiste en el grado de satisfacción, efectividad y eficiencia que obtiene el usuario al utilizar una aplicación.

- RNF 1. La aplicación debe de seguir los patrones de diseño establecidos por Google Material Design.
- RNF 2. El sistema debe de proporcionar continuamente soporte visual ante errores o acciones desarrolladas incorrectamente.
- RNF 3. Los usuarios familiarizados con la temática de la aplicación de comprender cualquier tipo de información mostrada-
- RNF 4. El usuario debe de saber en todo momento en que lugar de la aplicación se encuentra y que gestión está desarrollando.
- RNF 5. La aplicación debe de tener elementos fáciles de recordar
- RNF 6. La aplicación debe de ser flexible y eficiente en su uso, mostrando posibles atajos para el usuario.

### 3.1.2 Compatibilidad

Cualidad de un elemento para poder interactuar con otro.

- RNF 7. La aplicación debe de ser compatible con otras versiones superiores del sistema operativo Android



### 3.1.3 Eficacia

Capacidad de lograr un objetivo haciendo uso de los mínimos recursos y el menor tiempo posible.

- RNF 8. La aplicación no debe de tardar más de 4 segundos en generar el parte de firmas en formato PDF.
- RNF 9. La aplicación no debe de tardar más de 3 segundos en iniciarse.
- RNF 10. La aplicación no debe de tardar más de 2 segundos en agregar una nueva entidad.
- RNF 11. La aplicación de debe de tardar más de 1 segundo en leer el identificador del carné UPV.

### 3.1.4 Mantenibilidad

Cualidad de un elemento para ser modificado fácilmente.

- RNF 12. El código implementado debe de tener una estructura clara e indentada que permita comprenderlo fácilmente para posibles correcciones.

### 3.1.5 Consistencia

Característica que define solidez y estabilidad.

- RNF 13. La aplicación no debe de bloquearse durante el uso de la misma.

### 3.1.6 Integridad

Cualidad que refleja realismo y veracidad.

- RNF 14. Toda la información mostrada en la aplicación debe de ser real y coherente con los datos de las diferentes entidades que forman parte de la Universidad Politécnica de Valencia.

## 3.2 Requisitos funcionales

Los requisitos funcionales hacen referencia a los servicios que prestará el sistema [3].

- RF 1. El sistema mostrará el nombre y la información de las asignaturas existentes.
- RF 2. El sistema mostrará el nombre y la información de los profesores, alumnos y grupos que forman parte de una asignatura.
- RF 3. El sistema permitirá añadir y eliminar asignaturas.
- RF 4. El sistema permitirá editar la información de las asignaturas existentes.
- RF 5. El sistema permitirá editar la información de los profesores, alumnos y grupos que forman parte de una asignatura.



- RF 6. El sistema permitirá añadir y eliminar profesores, alumnos y grupos.
- RF 7. Se podrá obtener automáticamente los datos de una clase con horario dentro del calendario lectivo para la creación del parte de firmas.
- RF 8. El sistema registrará la asistencia de los alumnos presentes en la clase.
- RF 9. Se podrá enviar por correo electrónico toda la información referente al parte de asistencia.
- RF 10. El sistema permitirá a los profesores autenticados la creación de partes de firmas.
- RF 11. Se podrá visualizar la información de los partes de firmas en formato PDF y XML.

### 3.3 Diagrama de casos de uso

El diagrama de casos de uso nos ayuda a comprender el funcionamiento esperado de la aplicación. Son empleados para describir la comunicación y el comportamiento de un sistema con el usuario a la hora de realizar cierta tarea dentro de la aplicación. Dicho diagrama ha sido diseñado mediante la herramienta de modelado BoUML.

A continuación, se muestra la **figura 1** en la que se puede observar los diagramas de casos de uso en lenguaje UML<sup>1</sup> referentes a la aplicación.

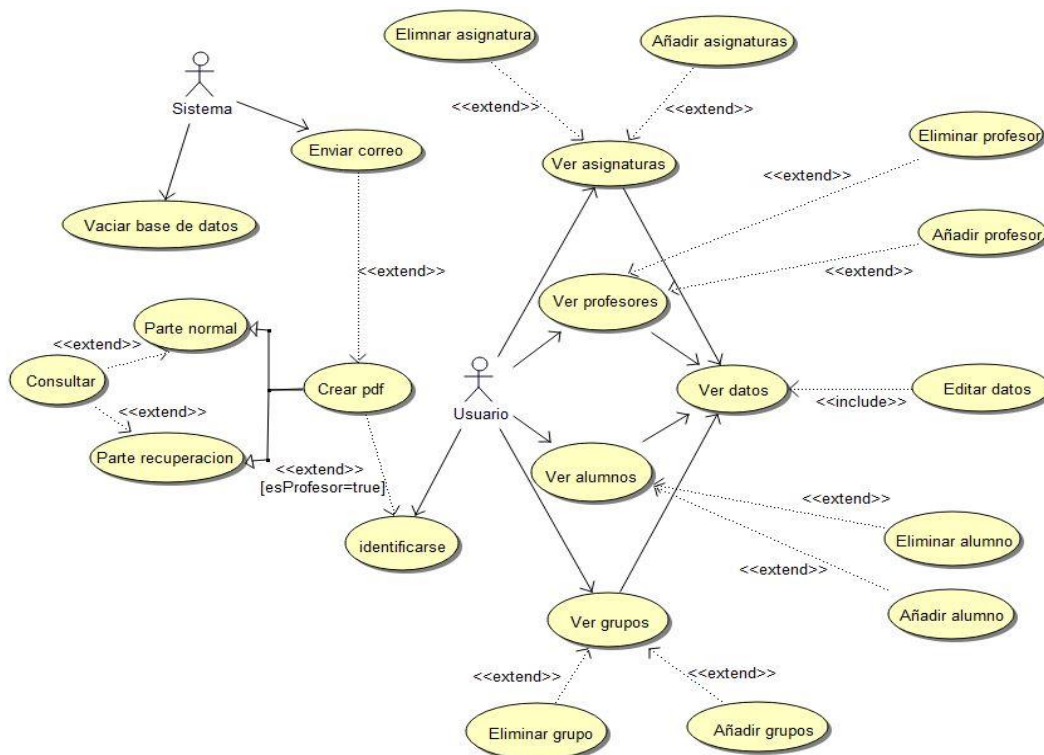


Figura 1. Diagrama de casos de uso

<sup>1</sup> UML: lenguaje de modelado de sistemas de software.



Fijándonos en la **figura 1** se pueden observar dos actores, el usuario y el sistema. En primer lugar, el usuario tiene la posibilidad de ver asignaturas, profesores, alumnos y grupos existentes en la aplicación, así como toda la información perteneciente a cada uno de ellos. Por otro lado, el usuario también es capaz de editar dicha información según sus necesidades, eliminar y añadir asignaturas, profesores, alumnos y grupos. El usuario además tiene la posibilidad de crear los partes de firmas en formato PDF tanto para la asistencia a clases dentro del horario académico, como para clases extraoficiales. Finalmente, el usuario también puede consultar la información de los partes de firmas creados con anterioridad.

Respecto al sistema, será en encargado de vaciar la base de datos por completo y de enviar mediante correo electrónico la información generada en los partes de firmas y los mismos en formato PDF a cualquier destinatario deseado

### 3.4 Identificación de los casos de uso

En este punto se van a explicar con más detalle los casos de uso mencionados anteriormente. Cada caso de uso mencionado como CU, irá acompañado de una numeración, una breve descripción, el actor que lo lleva a cabo, la precondition necesaria para su desarrollo, la metodología que el usuario debe de realizar, un flujo alternativo de respuesta frente a fallos o errores y el resultado obtenido tras su finalización.

Tabla 1. CU Añadir asignaturas

CU	Añadir asignaturas	1
<b>Descripción</b>	Poblar la Base de Datos con las asignaturas incluidas en el archivo “asignaturas.xml”.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El archivo “asignaturas.xml” debe de contener datos de asignaturas y tener una estructura correcta.</li> <li>• El archivo “asignaturas.xml” debe de ubicarse en el directorio de Descargas de nuestro dispositivo móvil.</li> <li>• No debe de existir ninguna asignatura en la Base de Datos.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes, visualizando un mensaje de alerta que le advierte de que no hay asignaturas existentes.</li> <li>2. El usuario selecciona “CARGAR ASIGNATURAS PROPIAS” y es posteriormente notificado con un diálogo informando que el proceso se ha realizado correctamente.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. Si se selecciona “CANCELAR” el sistema reubicará al usuario a la ventana principal.</li> </ol>	
<b>Resultado</b>	La Base de Datos se carga con asignaturas y se puede observar una lista con el nombre de las asignaturas existentes.	
<b>Requisitos</b>	RF 3	

Tabla 2. CU Visualizar asignaturas

CU	Visualizar asignaturas	2
<b>Descripción</b>	Mostrar las diferentes asignaturas existentes en nuestra Base de Datos.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Previamente se han tenido que dar de alta dichas asignaturas.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes, mostrando así una lista con las asignaturas existentes.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. Si no existe ninguna asignatura dada de alta, el sistema mostrará un mensaje de alerta solicitando la carga de todas las asignaturas.</li> </ol>	
<b>Resultado</b>	Se puede observar una lista con el nombre de las asignaturas existentes.	
<b>Requisitos</b>	RF 1	

Tabla 3. CU Eliminar asignatura

CU	Eliminar asignatura	3
<b>Descripción</b>	Eliminar de la Base de Datos la asignatura que el usuario ha seleccionado.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• En primer lugar, se debe de seleccionar la asignatura deseada.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona que asignatura que desea eliminar y posteriormente, selecciona el botón "BORRAR".</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de que la aplicación no pueda borrar la asignatura, se mostrará un diálogo informando que la asignatura no ha podido ser borrada.</li> </ol>	
<b>Resultado</b>	La asignatura es eliminada de la Base de Datos y con ello de la lista de asignaturas existentes.	
<b>Requisitos</b>	RF 3	



Tabla 4. CU Visualizar datos de la asignatura

<b>CU</b>	Visualizar datos de la asignatura	<b>4</b>
<b>Descripción</b>	Mostrar los diferentes datos que forman parte de la asignatura.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• En primer lugar, se debe de seleccionar la asignatura deseada.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER DATOS ASIGNATURA”, mostrando así toda la información sobre ella.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de no seleccionar ninguna asignatura, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguna.</li> </ol>	
<b>Resultado</b>	Se puede visualizar toda la información de la asignatura seleccionada.	
<b>Requisitos</b>	RF 1	

Tabla 5. CU Editar datos asignatura

<b>CU</b>	Editar datos asignatura	<b>5</b>
<b>Descripción</b>	Editar la información de la asignatura.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• En primer lugar, se debe de seleccionar la asignatura deseada.</li> <li>• El nombre de la asignatura no puede estar vacío.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER DATOS ASIGNATURA”, mostrando así toda la información sobre ella.</li> <li>4. El usuario seleccionará el botón “EDITAR” y posteriormente realizará los cambios deseados. Para finalizar el proceso, seleccionará el botón “FINALIZAR”.</li> <li>5. Los cambios serán visibles de forma automática.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de no seleccionar ninguna asignatura, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguna.</li> </ol>	

<b>Resultado</b>	La información de la asignatura se modificará según las necesidades del usuario.
<b>Requisitos</b>	RF 4

Tabla 6. CU Visualizar profesores de la asignatura

<b>CU</b>	Visualizar profesores de la asignatura	<b>6</b>
<b>Descripción</b>	Mostrar los diferentes profesores existentes en nuestra Base de Datos.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Previamente se han tenido que dar de alta dichos profesores.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER PROFESORES ASIGNATURA”, mostrando así todos los profesores que forman parte de la asignatura seleccionada.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. Si no existe ningún profesor dado de alta para esa asignatura, el sistema mostrará un mensaje de alerta solicitando el alta de uno o varios profesores.</li> </ol>	
<b>Resultado</b>	Se puede observar una lista con el nombre de los profesores existentes que pertenecen a la asignatura seleccionada.	
<b>Requisitos</b>	RF 2	

Tabla 7. CU Visualizar alumnos de la asignatura

<b>CU</b>	Visualizar alumnos de la asignatura	<b>7</b>
<b>Descripción</b>	Mostrar los diferentes alumnos existentes en nuestra Base de Datos.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Previamente se han tenido que dar de alta dichos alumnos.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> </ol>	



	3. El usuario seleccionará el botón “VER ALUMNOS ASIGNATURA”, mostrando así todos los alumnos que forman parte de la asignatura seleccionada.
<b>Flujo alternativo</b>	1. Si no existe ningún alumno dado de alta para esa asignatura, el sistema mostrará un mensaje de alerta solicitando el alta de uno o varios alumnos.
<b>Resultado</b>	Se puede observar una lista con el nombre de los alumnos existentes que pertenecen a la asignatura seleccionada.
<b>Requisitos</b>	RF 2

Tabla 8. CU Visualizar grupos de la asignatura

<b>CU</b>	Visualizar grupos de la asignatura	<b>8</b>
<b>Descripción</b>	Mostrar los diferentes grupos existentes en nuestra Base de Datos.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Previamente se han tenido que dar de alta dichos grupos.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER GRUPOS ASIGNATURA”, mostrando así todos los grupos que forman parte de la asignatura seleccionada.</li> </ol>	
<b>Flujo alternativo</b>	1. Si no existe ningún grupo dado de alta para esa asignatura, el sistema mostrará un mensaje de alerta solicitando la carga de todos los grupos.	
<b>Resultado</b>	Se puede observar una lista con el nombre de los grupos existentes que pertenecen a la asignatura seleccionada.	
<b>Requisitos</b>	RF 2	

Tabla 9. CU Añadir profesor individual

<b>CU</b>	Añadir profesor individual	<b>9</b>
<b>Descripción</b>	Añadir a la Base de Datos un profesor dado de alta por el usuario.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• No puede existir previamente otro profesor con el mismo identificador.</li> <li>• Se necesita acceso a Internet</li> </ul>	

	<ul style="list-style-type: none"> <li>• Se necesita habilitar el NFC</li> <li>• Se necesita poseer el carnet propio de la UPV</li> </ul>
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER PROFESORES ASIGNATURA”, mostrando así todos los profesores que forman parte de la asignatura seleccionada. <ol style="list-style-type: none"> <li>a. En caso de existir profesores, el usuario seleccionará en la parte superior derecha el símbolo “+” y “Añadir profesor”. Posteriormente pulsará en “PULSE PARA ESCANEAR” y leerá el código QR para obtener los datos de su carné. Tras ello, escaneará mediante NFC su carné para obtener el identificador y finalmente concluirá el proceso pulsando en “AÑADIR PROFESOR”.</li> <li>b. En caso de no existir profesores en la Base de Datos, el sistema mostrará un mensaje de alerta pidiéndole al usuario que añada algún profesor mediante dos posibilidades. <ol style="list-style-type: none"> <li>i. Pulsando en “AÑADIR PROFESOR”, será redireccionado a la pantalla para seguir el mismo procedimiento que en el apartado <b>a</b>.</li> </ol> </li> </ol> </li> <li>4. Los cambios serán visibles de forma automática.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de no tener profesores dados de alta en esa asignatura, si el usuario en el mensaje de alerta selecciona “CANCELAR”, será redireccionado al Menú de Asignaturas.</li> <li>2. Si el profesor que va a dar de alta, anteriormente ya estaba registrado, el sistema lanza un mensaje de alerta avisando de que dicho profesor ya existe .</li> </ol>
<b>Resultado</b>	A la Base de Datos se añade un nuevo profesor y se puede observar su nombre en la lista de los profesores existentes.
<b>Requisitos</b>	RF 6

Tabla 10. CU Añadir profesores por defecto

<b>CU</b>	Añadir profesores por defecto	<b>10</b>
<b>Descripción</b>	Añadir a la Base de Datos un profesor dado de alta por el usuario.	
<b>Actor</b>	Usuario	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• No puede existir ningún profesor en la Base de Datos</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> </ol>	

	<ol style="list-style-type: none"> <li>3. El usuario seleccionará el botón “VER PROFESORES ASIGNATURA”, mostrando así todos los profesores que forman parte de la asignatura seleccionada. <ol style="list-style-type: none"> <li>a. En caso de no existir profesores en la Base de Datos, el sistema mostrará un mensaje de alerta pidiéndole al usuario que añada algún profesor mediante dos posibilidades. <ol style="list-style-type: none"> <li>i. Pulsando en “CARGAR PROFESORES POR DEFECTO”, se cargarán varios profesores por defecto.</li> </ol> </li> </ol> </li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de no tener profesores dados de alta en esa asignatura, si el usuario en el mensaje de alerta selecciona “CANCELAR”, será redireccionado al Menú de Asignaturas.</li> </ol>
<b>Resultado</b>	A la Base de Datos se añaden nuevos profesores y se puede observar sus nombres en la lista de los profesores existentes.
<b>Requisitos</b>	RF 6

Tabla 11. CU Visualizar datos de un profesor

CU	Visualizar datos de un profesor	11
<b>Descripción</b>	Mostrar los diferentes datos de un profesor que forman parte de la asignatura.	
<b>Actor</b>	Usuario	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• En primer lugar, se debe de seleccionar la asignatura deseada.</li> <li>• En segundo lugar, se debe de seleccionar el profesor deseado.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER PROFESORES ASIGNATURA”, mostrando así una lista con todos los profesores existentes.</li> <li>4. El usuario seleccionara el nombre del profesor que quiere obtener información y posteriormente seleccionara el botón “VER”. Mostrando así toda la información relacionada con dicho profesor.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de no seleccionar ninguna asignatura, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguna.</li> <li>2. En caso de no seleccionar ningún profesor, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguno.</li> </ol>	
<b>Resultado</b>	Se puede visualizar toda la información del profesor seleccionado.	
<b>Requisitos</b>	RF 2	



Tabla 12. CU Editar datos de un profesor

<b>CU</b>	Editar datos de un profesor	<b>12</b>
<b>Descripción</b>	Editar información del profesor.	
<b>Actor</b>	Usuario	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• En primer lugar, se debe de seleccionar la asignatura deseada.</li> <li>• En segundo lugar, se debe de seleccionar el profesor deseado.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER PROFESORES ASIGNATURA”, mostrando así una lista con todos los profesores existentes.</li> <li>4. El usuario seleccionara el nombre del profesor que quiere editar información y posteriormente seleccionara el botón “VER”, Mostrando así toda la información relacionada con dicho profesor.</li> <li>5. El usuario seleccionará el botón “EDITAR” y posteriormente realizará los cambios deseados. Para finalizar el proceso, seleccionará el botón “FINALIZAR”.</li> <li>6. Los cambios serán visibles de forma automática.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de no seleccionar ninguna asignatura, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguna.</li> <li>2. En caso de no seleccionar ningún profesor, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguno.</li> </ol>	
<b>Resultado</b>	La información del profesor se modificará según las necesidades del usuario.	
<b>Requisitos</b>	RF 5	

Tabla 13. CU Eliminar profesor

<b>CU</b>	Eliminar profesor	<b>13</b>
<b>Descripción</b>	Eliminar de la Base de Datos el profesor que el usuario ha seleccionado.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• En primer lugar, se debe de seleccionar la asignatura deseada.</li> <li>• En segundo lugar, se debe de seleccionar el profesor deseado.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> </ol>	



	<ol style="list-style-type: none"> <li>3. El usuario seleccionará el botón “VER PROFESORES ASIGNATURA”, mostrando así una lista con todos los profesores existentes.</li> <li>4. El usuario seleccionara el nombre del profesor que desea eliminar y posteriormente seleccionara el botón “BORRAR”.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de que la aplicación no pueda borrar el profesor, se mostrará un diálogo informando que el profesor no ha podido ser borrado.</li> </ol>
<b>Resultado</b>	Eliminar de la Base de Datos el profesor que el usuario ha seleccionado.
<b>Requisitos</b>	RF 6

Tabla 14. CU Añadir alumno individual

<b>CU</b>	Añadir alumno individual	<b>14</b>
<b>Descripción</b>	Añadir a la Base de Datos un alumno dado de alta por el usuario.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• No puede existir previamente otro alumno con el mismo identificador.</li> <li>• Se necesita acceso a Internet</li> <li>• Se necesita habilitar el NFC</li> <li>• Se necesita poseer el carnet propio de la UPV</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER ALUMNOS ASIGNATURA”, mostrando así todos los alumnos que forman parte de la asignatura seleccionada. <ol style="list-style-type: none"> <li>a. En caso de existir alumnos, el usuario seleccionará en la parte superior derecha el símbolo “+” y “Añadir alumno”. Posteriormente pulsará en “PULSE PARA ESCANEAR” y leerá el código QR para obtener los datos de su carné. Tras ello, escaneará mediante NFC su carné para obtener el identificador y finalmente concluirá el proceso pulsando en “AÑADIR ALUMNO”.</li> <li>b. En caso de no existir alumnos en la Base de Datos, el sistema mostrará un mensaje de alerta pidiéndole al usuario que añada algún alumno mediante dos posibilidades. <ol style="list-style-type: none"> <li>i. Pulsando en “AÑADIR ALUMNO”, será redireccionado a la pantalla para seguir el mismo procedimiento que en el apartado <b>a</b>.</li> </ol> </li> </ol> </li> <li>4. Los cambios serán visibles de forma automática.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de no tener alumnos dados de alta en esa asignatura, si el usuario en el mensaje de alerta selecciona “CANCELAR”, será redireccionado al Menú de Asignaturas.</li> </ol>	

	2. Si el alumno que va a dar de alta, anteriormente ya estaba registrado, el sistema lanza un mensaje de alerta avisando de que dicho alumno ya existe.
<b>Resultado</b>	A la Base de Datos se añade un nuevo alumno y se puede observar su nombre en la lista de los alumnos existentes.
<b>Requisitos</b>	RF 6

Tabla 15. CU Añadir alumnos por defecto

<b>CU</b>	Añadir alumnos por defecto	<b>15</b>
<b>Descripción</b>	Añadir a la Base de Datos un alumno dado de alta por el usuario.	
<b>Actor</b>	Usuario	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>No puede existir ningún alumno en la Base de Datos</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>El usuario seleccionará el botón “VER ALUMNOS ASIGNATURA”, mostrando así todos los alumnos que forman parte de la asignatura seleccionada. <ol style="list-style-type: none"> <li>En caso de no existir alumnos en la Base de Datos, el sistema mostrará un mensaje de alerta pidiéndole al usuario que añada algún alumno mediante dos posibilidades. <ol style="list-style-type: none"> <li>Pulsando en “CARGAR ALUMNOS POR DEFECTO”, se cargarán varios alumnos por defecto.</li> </ol> </li> </ol> </li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>En caso de no tener alumnos dados de alta en esa asignatura, si el usuario en el mensaje de alerta selecciona “CANCELAR”, será redireccionado al Menú de Asignaturas.</li> </ol>	
<b>Resultado</b>	A la Base de Datos se añaden nuevos alumnos y se puede observar sus nombres en la lista de los profesores existentes.	
<b>Requisitos</b>	RF 6	

Tabla 16. CU Visualizar datos de un alumno

<b>CU</b>	Visualizar datos de un alumno	<b>16</b>
<b>Descripción</b>	Mostrar los diferentes datos de un alumno que forman parte de la asignatura.	
<b>Actor</b>	Usuario	



<b>Precondición</b>	<ul style="list-style-type: none"> <li>• En primer lugar, se debe de seleccionar la asignatura deseada.</li> <li>• En segundo lugar, se debe de seleccionar el alumno deseado.</li> </ul>
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER ALUMNOS ASIGNATURA”, mostrando así una lista con todos los alumnos existentes.</li> <li>4. El usuario seleccionara el nombre del alumno que quiere obtener información y posteriormente seleccionara el botón “VER”. Mostrando así toda la información relacionada con dicho alumno.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de no seleccionar ninguna asignatura, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguna.</li> <li>2. En caso de no seleccionar ningún alumno, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguno.</li> </ol>
<b>Resultado</b>	Se puede visualizar toda la información del alumno seleccionado.
<b>Requisitos</b>	RF 2

Tabla 17. CU Editar datos de un alumno

<b>CU</b>	Editar datos de un alumno	<b>17</b>
<b>Descripción</b>	Editar información del alumno.	
<b>Actor</b>	Usuario	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• En primer lugar, se debe de seleccionar la asignatura deseada.</li> <li>• En segundo lugar, se debe de seleccionar el alumno deseado.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER ALUMNOS ASIGNATURA”, mostrando así una lista con todos los alumnos existentes.</li> <li>4. El usuario seleccionara el nombre del alumno que quiere editar información y posteriormente seleccionara el botón “VER”, Mostrando así toda la información relacionada con dicho alumno.</li> <li>5. El usuario seleccionará el botón “EDITAR” y posteriormente realizará los cambios deseados. Para finalizar el proceso, seleccionará el botón “FINALIZAR”.</li> <li>6. Los cambios serán visibles de forma automática.</li> </ol>	

<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de no seleccionar ninguna asignatura, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguna.</li> <li>2. En caso de no seleccionar ningún alumno, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguno.</li> </ol>
<b>Resultado</b>	La información del alumno se modificará según las necesidades del usuario.
<b>Requisitos</b>	RF 2

Tabla 18. CU Eliminar alumno

<b>CU</b>	Eliminar alumno	<b>18</b>
<b>Descripción</b>	Eliminar de la Base de Datos el alumno que el usuario ha seleccionado.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• En primer lugar, se debe de seleccionar la asignatura deseada.</li> <li>• En segundo lugar, se debe de seleccionar el alumno deseado.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER ALUMNOS ASIGNATURA”, mostrando así una lista con todos los alumnos existentes.</li> <li>4. El usuario seleccionara el nombre del alumno que desea eliminar y posteriormente seleccionara el botón “BORRAR”.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. En caso de que la aplicación no pueda borrar el alumno, se mostrará un diálogo informando que el alumno no ha podido ser borrado.</li> </ol>	
<b>Resultado</b>	Eliminar de la Base de Datos el profesor que el usuario ha seleccionado.	
<b>Requisitos</b>	RF 6	

Tabla 19. CU Añadir grupos de asignaturas

<b>CU</b>	Añadir grupos de asignaturas	<b>19</b>
<b>Descripción</b>	Poblar la Base de Datos con las asignaturas incluidas en el archivo “grupos.xml”.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El archivo “grupos.xml” debe de contener datos de grupos de asignaturas y tener una estructura correcta.</li> </ul>	



	<ul style="list-style-type: none"> <li>El archivo “grupos.xml” debe de ubicarse en el directorio de Descargas de nuestro dispositivo móvil.</li> </ul>
<b>Metodología</b>	<ol style="list-style-type: none"> <li>Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>El usuario seleccionará el botón “VER GRUPOS ASIGNATURA”. <ol style="list-style-type: none"> <li>En caso de no existir ningún grupo de asignaturas, el usuario visualizará un mensaje de alerta que le advierte de que no hay grupos de asignaturas existentes. <ol style="list-style-type: none"> <li>El usuario selecciona “CARGAR GRUPOS PROPIOS” y es posteriormente notificado con un diálogo informando que el proceso se ha realizado correctamente.</li> </ol> </li> <li>En caso de que no existan grupos en la asignatura deseada pero sí en otras asignaturas, el usuario visualizará un mensaje de alerta que le advierte de que no hay grupos existentes para esa asignatura. <ol style="list-style-type: none"> <li>El usuario selecciona “CARGAR GRUPOS PROPIOS” y es posteriormente notificado con un diálogo informando que el conjunto de asignaturas ha sido actualizado correctamente.</li> </ol> </li> </ol> </li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>Si se selecciona “CANCELAR” el sistema reubicará al usuario a la ventana principal.</li> </ol>
<b>Resultado</b>	La Base de Datos se carga con grupos de asignaturas y se puede observar una lista con el nombre de los grupos existentes.
<b>Requisitos</b>	RF 6

Tabla 20. CU Visualizar datos del grupo

<b>CU</b>	Visualizar datos del grupo	<b>20</b>
<b>Descripción</b>	Mostrar los diferentes datos de un grupo que forman parte de la asignatura.	
<b>Actor</b>	Usuario	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>En primer lugar, se debe de seleccionar la asignatura deseada.</li> <li>En segundo lugar, se debe de seleccionar el grupo deseado.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>El usuario seleccionará el botón “VER GRUPOS ASIGNATURA”, mostrando así una lista con todos los grupos existentes.</li> </ol>	

	4. El usuario seleccionará el nombre del grupo que quiere obtener información y posteriormente seleccionará el botón “VER”. Mostrando así toda la información relacionada con dicho grupo.
<b>Flujo alternativo</b>	3. En caso de no seleccionar ninguna asignatura, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguna. 4. En caso de no seleccionar ningún grupo, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguno.
<b>Resultado</b>	Se puede visualizar toda la información del grupo seleccionado.
<b>Requisitos</b>	RF 2

Tabla 21. CU Editar datos de un grupo

<b>CU</b>	Editar datos de un grupo	<b>21</b>
<b>Descripción</b>	Editar información del grupo.	
<b>Actor</b>	Usuario	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• En primer lugar, se debe de seleccionar la asignatura deseada.</li> <li>• En segundo lugar, se debe de seleccionar el grupo deseado.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER GRUPOS ASIGNATURA”, mostrando así una lista con todos los grupos existentes.</li> <li>4. El usuario seleccionara el nombre del grupo que quiere editar información y posteriormente seleccionara el botón “VER”, Mostrando así toda la información relacionada con dicho grupo.</li> <li>5. El usuario seleccionará el botón “EDITAR” y posteriormente realizará los cambios deseados. Para finalizar el proceso, seleccionará el botón “FINALIZAR”.</li> <li>6. Los cambios serán visibles de forma automática.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>3. En caso de no seleccionar ninguna asignatura, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguna.</li> <li>4. En caso de no seleccionar ningún grupo, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguno.</li> </ol>	
<b>Resultado</b>	La información del grupo se modificará según las necesidades del usuario.	
<b>Requisitos</b>	RF 5	



Tabla 22. CU Eliminar grupo

<b>CU</b>	Eliminar grupo	<b>22</b>
<b>Descripción</b>	Eliminar de la Base de Datos el grupo que el usuario ha seleccionado.	
<b>Actor</b>	Usuario.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• En primer lugar, se debe de seleccionar la asignatura deseada.</li> <li>• En segundo lugar, se debe de seleccionar el grupo deseado.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de ajustes.</li> <li>2. Una vez visualiza la lista de asignaturas existentes, selecciona de que asignatura desea ver información y posteriormente, selecciona el botón “VER”.</li> <li>3. El usuario seleccionará el botón “VER GRUPOS ASIGNATURA”, mostrando así una lista con todos los grupos existentes.</li> <li>4. El usuario seleccionara el nombre del grupo que desea eliminar y posteriormente seleccionara el botón “BORRAR”.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>3. En caso de que la aplicación no pueda borrar el grupo, se mostrará un diálogo informando que el grupo no ha podido ser borrado.</li> </ol>	
<b>Resultado</b>	Eliminar de la Base de Datos el profesor que el usuario ha seleccionado.	
<b>Requisitos</b>	RF 6	

Tabla 23. CU Crear PDF de parte de firmas

<b>CU</b>	Crear PDF de parte de firmas	<b>23</b>
<b>Descripción</b>	Mediante el registro de la asistencia de los alumnos, se genera un archivo Pdf que contiene el parte de firmas de una clase con eventualidad regular.	
<b>Actor</b>	Usuario	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El profesor que va a registrar la asistencia de una asignatura debe de estar dado de alta como profesor de dicha asignatura.</li> <li>• Deben de estar dados de alta alumnos que cursen la asignatura.</li> <li>• Se debe de seleccionar la asignatura de la cual se quiere registrar la asistencia.</li> <li>• Se necesita acceso a Internet</li> <li>• Se necesita habilitar el NFC</li> <li>• Se necesita poseer el carnet propio de la UPV</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de registro de asistencia.</li> <li>2. Posteriormente el usuario debe de escanear por NFC su carné UPV para obtener su identificador y leer su código QR para obtener sus datos. Tras la identificación, el usuario seleccionará el botón “SIGUIENTE”.</li> </ol>	



	<p>3. El usuario visualizará en una lista el nombre de las asignaturas de las que es profesor y deberá de seleccionar aquella de la cual quiere registrar la asistencia. Tras seleccionarla, el usuario seleccionará uno de los botones inferiores dependiendo del tipo de asistencia:</p> <ol style="list-style-type: none"> <li>a. Si el usuario selecciona el botón “ASISTENCIA EXTRAOFICIAL” será redirigido a una ventana en la que podrá observar los datos de la asignatura. Al tratarse de una clase fuera del horario estipulado, el usuario tendrá que rellenar: <ol style="list-style-type: none"> <li>i. Grupo: Nombre del grupo de la asignatura.</li> <li>ii. Espacio: Lugar donde se imparte la clase.</li> <li>iii. Hora Entrada: Hora a la que comienza la clase.</li> <li>iv. Hora Salida: Hora a la que finaliza la clase.</li> </ol> <p>Y finalmente seleccionar el botón “INICIAR ASISTENCIA”.</p> </li> <li>b. Si el usuario selecciona el botón “ASISTENCIA NORMAL” será redirigido a una ventana en la que podrá observar todos los datos de la asignatura rellenados automáticamente ya que se está registrando la asistencia de una clase dentro del horario normal. Y finalmente seleccionar el botón “INICIAR ASISTENCIA”.</li> </ol> <p>4. Tras una de las dos opciones anteriores, el usuario es redirigido a una nueva ventana en la que todos los alumnos deben de escanear sus carnés UPV mediante NFC para registrar su asistencia.</p> <p>5. Finalmente, tras terminar la lectura de todas los carnés UPV de los alumnos, el usuario seleccionará “CREAR PDF” para obtener un archivo Pdf con todos los datos del parte de asistencia que ha realizado.</p> <p>6. El archivo Pdf podrá visualizarse tras finalizar el proceso o accediendo al directorio Descargas/ParteFirmasUPV</p>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. Si el usuario selecciona algún botón sin previamente escanear su carné UPV, el sistema lanzará un mensaje de alerta indicando que el primer paso a realizar es dicho escaneo.</li> <li>2. Si la persona a identificar no está registrada como profesor, al seleccionar el botón “VER DATOS” o “SIGUIENTE” el sistema lanzará un mensaje de alerta avisando de que el usuario a identificar no es un profesor</li> <li>3. En caso de no seleccionar ninguna asignatura a la que pertenece el usuario, el sistema mostrará un mensaje de alerta indicando que debe de seleccionar alguna.</li> <li>4. Si el usuario selecciona “ASISTENCIA NORMAL” y el sistema no detecta una asignatura con ese horario, se notificará al usuario con un mensaje de alerta de que no existe ningún grupo con este horario.</li> </ol>
<b>Resultado</b>	Se genera un parte de firmas en formato Pdf en el que se puede observar todos los datos del profesor, clase, grupo y alumnos que han asistido a dicha clase.
<b>Requisitos</b>	RF 7, RF 8 y RF 10

Tabla 24. CU Enviar PDF y datos del parte de firmas por correo


<b>CU</b>	Enviar PDF y datos del parte de firmas por correo	<b>24</b>
<b>Descripción</b>	Envío mediante correo electrónico del parte de asistencia en formato Pdf y de los datos de dicho parte en formato XML.	
<b>Actor</b>	Sistema.	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Previamente se debe de generar un parte de asistencias.</li> <li>• Se necesita acceso a internet.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Tras finalizar el Caso de uso 23, el usuario seleccionará el botón “ENVIAR PDF” y aparecerá una ventana emergente en la que el usuario debe de introducir el correo electrónico donde quiere recibir la documentación. Finalmente, el usuario debe de seleccionar el botón “ENVIAR”.</li> </ol>	
<b>Flujo alternativo</b>	Si el sistema detecta algún problema, se notifica al usuario mediante un mensaje de alerta de que el correo no ha sido enviado.	
<b>Resultado</b>	El usuario recibe un correo electrónico con el parte de firmas en formato Pdf y los datos de dicho parte en formato XML.	
<b>Requisitos</b>	RF 9	

Tabla 25. CU Consultar datos de partes de firmas

<b>CU</b>	Consultar datos de parte de firmas	<b>25</b>
<b>Descripción</b>	Búsqueda de información de los partes de firmas realizados con anterioridad mediante el nombre de la asignatura, grupo y fecha en el que se realizó.	
<b>Actor</b>	Usuario	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>▪ Previamente se debe de haber generado el Pdf del parte de asistencia que se está buscando.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de búsqueda de partes de firmas.</li> <li>2. El usuario deberá de seleccionar: <ol style="list-style-type: none"> <li>a. Asignatura: Nombre de la asignatura del parte de asistencias.</li> <li>b. Grupo: nombre del grupo de la asignatura.</li> <li>c. Fecha: fecha en la que se realizó el parte de asistencia.</li> </ol> </li> <li>3. Finalmente, el usuario selecciona el botón “SIGUIENTE” para finalizar el proceso y así poder visualizar en otra pantalla, todos los datos del parte de firmas que estaba buscando.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. Si el usuario selecciona el botón “SIGUIENTE” sin previamente rellenar los criterios de búsqueda, el sistema notifica al usuario mediante un mensaje de alerta diciendo que debe de rellenar todos los criterios.</li> <li>2. En caso de que el sistema no detecte el parte de firmas solicitado, se notificará al usuario con un mensaje de alera de que el parte de</li> </ol>	

	firmas solicitado no ha sido generado y permite una nueva búsqueda.
<b>Resultado</b>	El usuario puede visualizar todos los datos que componen el parte de asistencias.
<b>Requisitos</b>	RF 11

Tabla 26. CU Vaciar base de datos

<b>CU</b>	Vaciar base de datos	<b>26</b>
<b>Descripción</b>	Borrado de toda la información de la Base de Datos.	
<b>Actor</b>	Sistema	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>▪ La Base de Datos debe de tener información para poder ser borrada.</li> </ul>	
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Desde la pantalla de inicio, el usuario selecciona el botón de Ajustes y en la pantalla que ha sido redireccionado, selecciona el botón derecho del menú superior  y posteriormente en “Vaciar BD”.</li> </ol>	
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de que el sistema detecte algún fallo durante el borrado, notificara al usuario con un mensaje de alerta de que la Base de Datos no ha podido ser borrada.</li> </ol>	
<b>Resultado</b>	Se eliminan de la Base de Datos todos los registros existentes.	
<b>Requisitos</b>	RF 12	

## 3.5 Mockups

Una técnica muy empleada a la hora de diseñar una interfaz es la utilización de los denominados mockups<sup>1</sup> ya que les permite hacerse una idea previa del resultado final para identificar futuras correcciones o mejoras en el diseño. En ocasiones puede ser únicamente una maqueta sin funcionalidad y en otros casos pueden contener funcionalidades básicas para así evaluarlas en su uso.

Los mockups se emplean en la fase inicial de desarrollo para presentaciones y control de calidad. Mediante ellos, el cliente y el diseñador coordinan ideas y establecen cuáles serán los requisitos [4]. Se realizan con ellos pruebas de usabilidad de modo que permitan detectar cualquier problema o error antes de alcanzar una mayor fase de desarrollo.

A continuación, se van a mostrar los diferentes mockups de las ventanas de la aplicación y se ha identificado en ellos los elementos que suscitan los casos de uso comentados en el punto anterior.

---

<sup>1</sup> Mockup: diseño inicial de una aplicación.

Figura 2. Ventanas de eventos con asignaturas



Tal y como podemos observar en la escena anterior, partimos de la ventana de Inicio y al seleccionar el botón de configuración podemos visualizar las asignaturas existentes cumpliendo así el CU 2. En caso de que no existiera ninguna asignatura el sistema lanzaría un mensaje de alerta pidiendo al usuario que cargase las asignaturas como se puede ver en el CU 1.

Si el usuario selecciona el botón borrar, podrá eliminar una asignatura efectuando el CU 3. En caso de que el usuario seleccione el botón ver, podrá visualizar los datos de una asignatura en concreto tal y como especifica el CU 4. En caso de que el usuario seleccione el botón Editar, podrá modificar los datos de la asignatura realizando el CU 5. Finalmente, si el usuario selecciona el botón del menú superior eliminará todos los datos de la base de datos.



Figura 3. Ventanas de eventos con profesores



Para llegar a la escena anterior, desde la ventana de configuración esta vez, seleccionamos el botón de ver profesores asignatura para así visualizar los profesores existentes de una asignatura cumpliendo así el CU 6. En caso de que no existiera ningún profesor, el sistema lanzaría un mensaje de alerta pidiendo al usuario que añadiese uno o varios profesores referenciando el CU 9 y el CU 10. El usuario puede añadir un nuevo profesor seleccionando el icono del menú superior cumpliendo así el CU 9. Si el usuario selecciona el botón borrar, podrá eliminar un profesor efectuando el CU 13. En caso de que el usuario seleccione el botón ver, podrá visualizar los datos de un profesor en concreto tal y como especifica el CU 11. Finalmente, si el usuario selecciona el botón Editar, podrá modificar los datos del profesor realizando el CU 12.

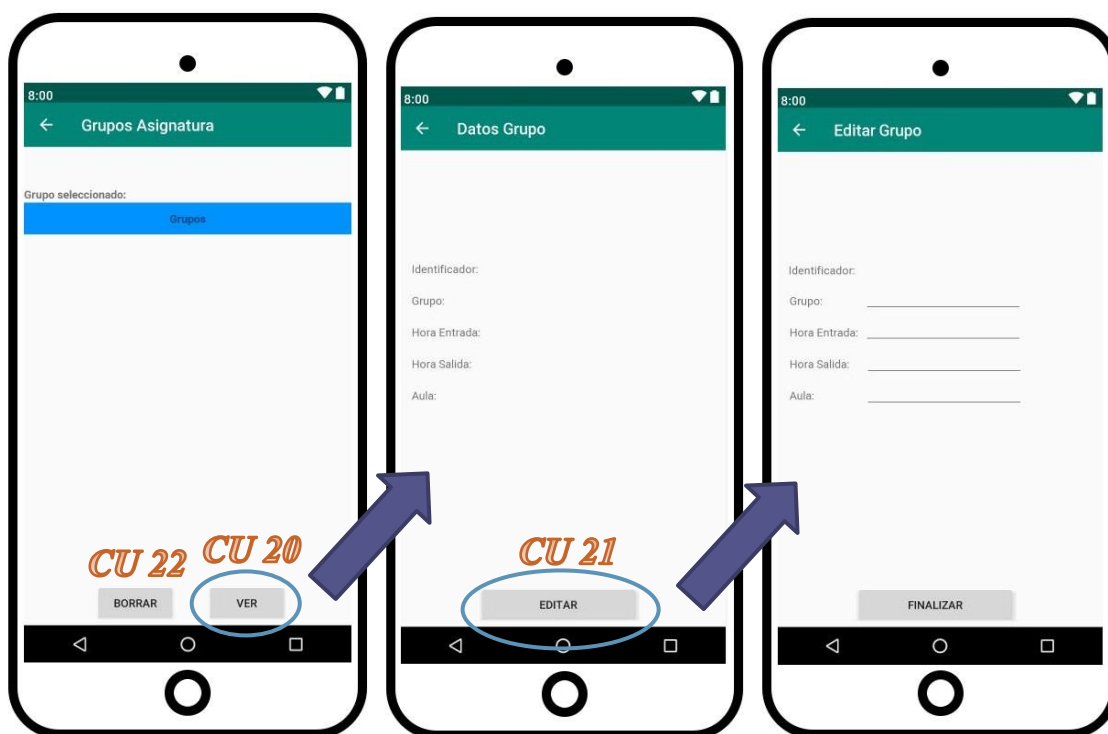
Figura 4. Ventanas de eventos con alumnos



Para llegar a la escena anterior, desde la ventana de configuración esta vez, seleccionamos el botón de ver alumnos asignatura para así visualizar los alumnos existentes de una asignatura cumpliendo así el CU 7. En caso de que no existiera ningún alumno, el sistema lanzaría un mensaje de alerta pidiendo al usuario que añadiese uno o varios alumnos referenciando el CU 14 y el CU 15. El usuario puede añadir un nuevo alumno seleccionando el icono del menú superior cumpliendo así el CU 14. Si el usuario selecciona el botón borrar, podrá eliminar un alumno efectuando el CU 18. En caso de que el usuario seleccione el botón ver, podrá visualizar los datos de un alumno en concreto tal y como especifica el CU 16. Finalmente, si el usuario selecciona el botón Editar, podrá modificar los datos del alumno realizando el CU 17.



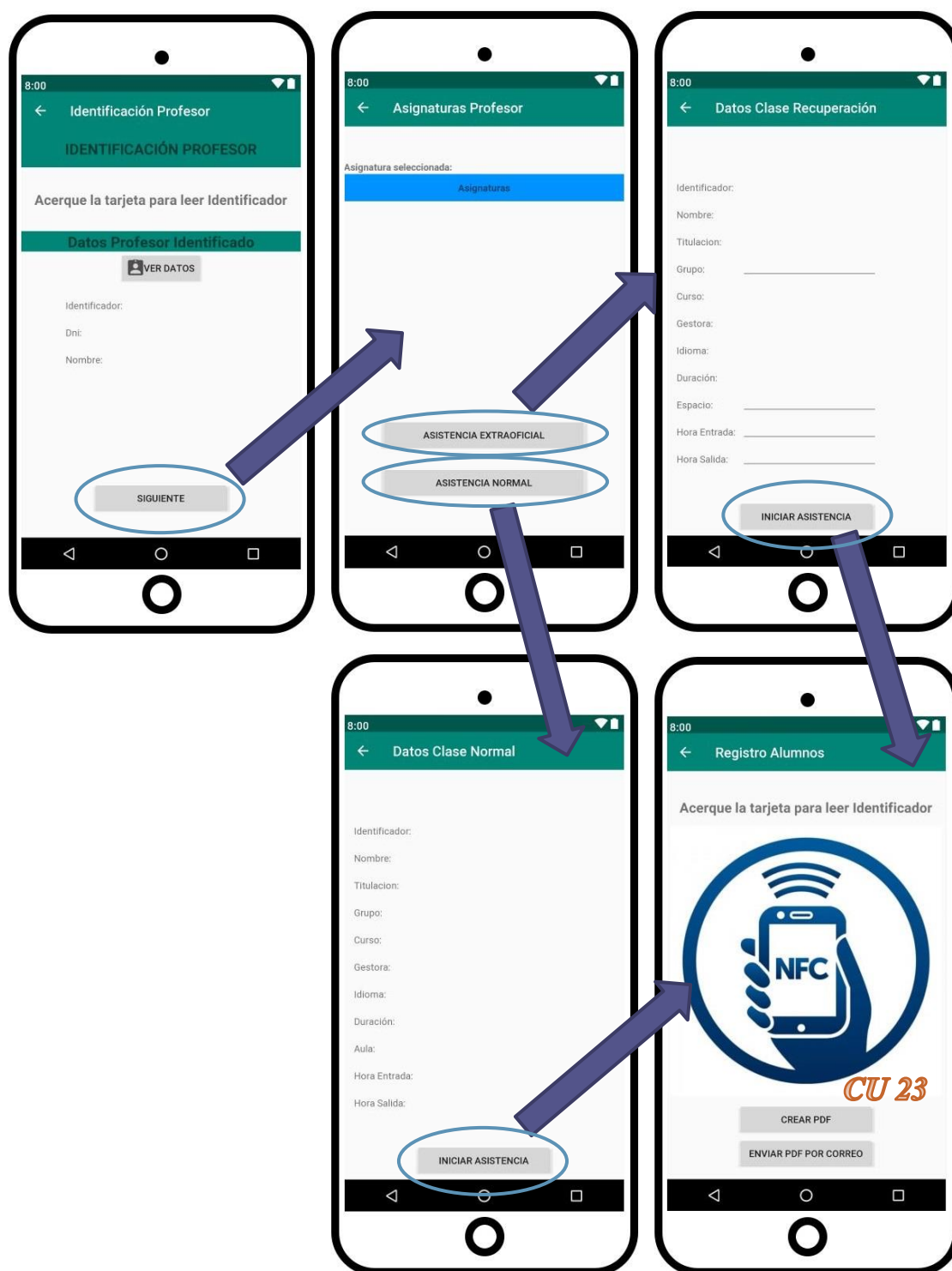
Figura 5. Ventanas de eventos con grupos



Para llegar a la escena anterior, desde la ventana de configuración esta vez, seleccionamos el botón de ver grupos asignatura para así visualizar los grupos existentes de una asignatura cumpliendo así el CU 7. En caso de que no existiera ningún grupo, el sistema lanzaría un mensaje de alerta pidiendo al usuario que cargase los grupos referenciando el CU 19. Si el usuario selecciona el botón borrar, podrá eliminar un grupo efectuando el CU 22. En caso de que el usuario seleccione el botón ver, podrá visualizar los datos de un grupo en concreto tal y como especifica el CU 20. Finalmente, si el usuario selecciona el botón Editar, podrá modificar los datos del grupo realizando el CU 21.



Figura 6. Ventanas de eventos generar parte de firmas



Para llegar a la escena anterior, desde la ventana de configuración esta vez, seleccionamos el botón de partes de firmas para que el usuario pueda identificarse como. Tras identificarse, el usuario debe de elegir el tipo de clase de la cual se registrará la. Una vez el usuario haya revisado/rellenado los datos de la clase, selecciona el botón de iniciar asistencia y comienza a registrar la asistencia de los alumnos. Para concluir el proceso tras el registro, el usuario pulsará el botón de crear PDF para cumplir así con el CU 23.

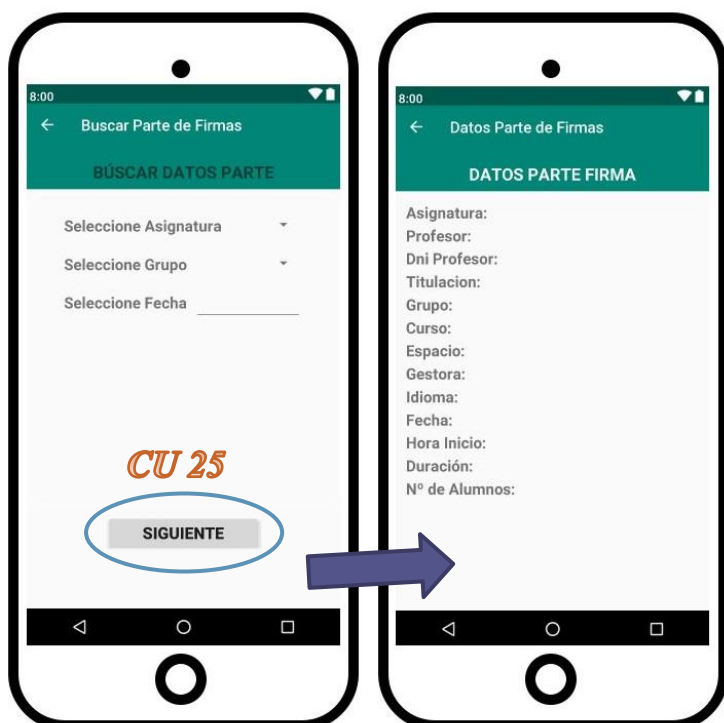


Figura 7. Ventanas de eventos para enviar correo electrónico



Para cumplir con el CU 24, tras el registro de los alumnos que han asistido a clase mediante la lectura por NFC de sus carnés UPV y la creación del parte de firmas en formato DFP, el usuario debería de seleccionar el botón enviar PDF por correo.

Figura 8. Ventanas de eventos para buscar parte de firmas



El CU 25 comienza desde la Ventana principal de la aplicación cuando el usuario selecciona el botón buscar parte, si se selecciona la asignatura a la que pertenece dicho parte de firmas. Cuando vaya a seleccionar el grupo, solo podrá seleccionar los grupos que pertenezcan a la asignatura anteriormente seleccionada. Esto sirve de atajo para el usuario a la hora de la búsqueda. Tras introducir todas las credenciales de búsqueda el usuario debe de seleccionar el botón de siguiente para así poder visualizar todos los datos que conforman el parte de firmas.

### 3.6 Modelo de dominio

Gracias al modelo de dominio podemos hacernos una idea de las distintas clases de entidades que componen nuestra aplicación [5]. Mediante el esquema representado en la **figura 9** podemos hacernos una fiel idea de su estructura.

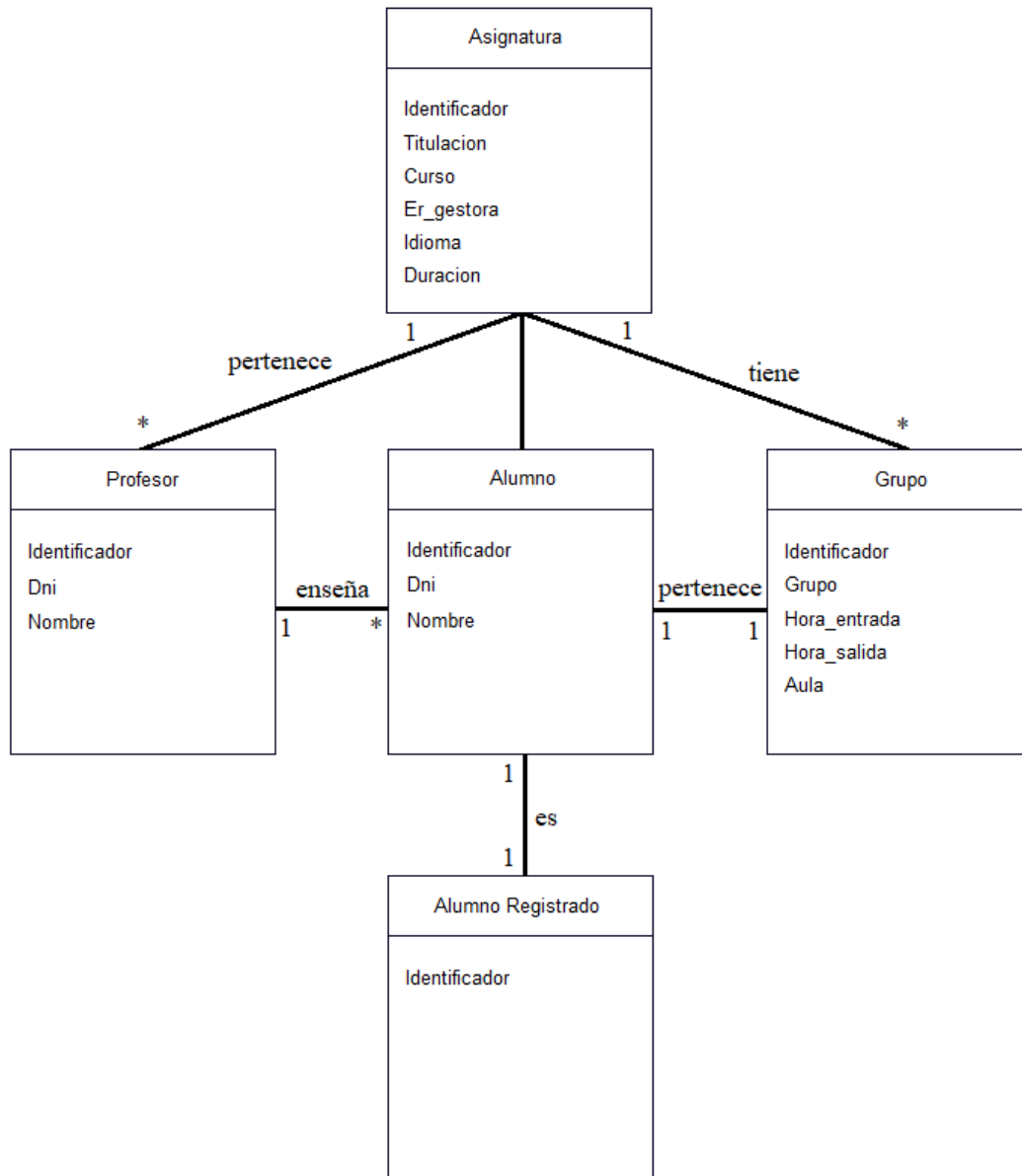


Figura 9. Modelo de dominio

Respecto al modelo de dominio, se pueden diferenciar 4 entidades:

- **Asignatura**. Contiene los atributos (Id, titulación, curso, er\_gestora, idioma y duracion) necesarios para realizar las consultas a la base desde la interfaz.
- **Profesor y Alumno**. Contiene los atributos (Id, dni y nombre) necesarios para su identificación.



- Grupo. Contiene los atributos (Id, grupo, hora\_entrada, hora\_salida y aula) utilizados para distinguir el tipo de clase en la que se registrará la asistencia.
- Alumno Registrado. Contiene el atributo (Id) necesario para mostrar que el alumno ha registrado asistencia.

### **3.7 Conclusión**

Tras exponer los diferentes requisitos con los que debe contar nuestra aplicación, identificar los posibles casos de uso y detallarlos uno a uno, mostrar el mockup empleado en el diseño y el modelo de dominio de nuestra aplicación, podemos dar paso a su diseño. En el siguiente capítulo, mostraremos la estructura que va a seguir nuestro proyecto, un esquema de los diferentes componentes que lo forman y la arquitectura de nuestra aplicación y de la base de datos siguiendo las pautas establecidas para cumplir con los requisitos expuestos anteriormente.

# 4. Diseño

---

En este capítulo nos centraremos en el diseño de nuestra aplicación y en todos los componentes que intervienen en él.

## 4.1 Estructura del Proyecto

Además de un correcto funcionamiento por parte de la aplicación, es necesario un claro y organizado esquema interno que facilite su comprensión y acceso a todos los recursos que conforman el proyecto. A continuación, en la **figura 10** se puede observar la estructura elegida para nuestro proyecto.

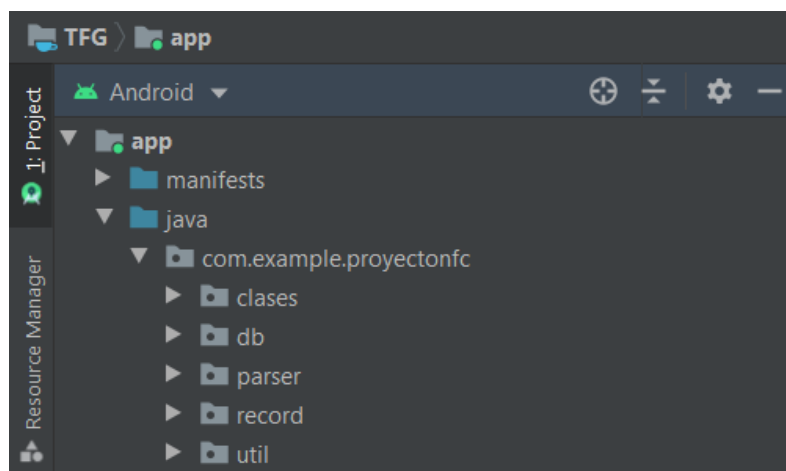


Figura 10. Estructura del proyecto

La estructura consta de 5 carpetas. La carpeta “clase”, contiene todas las clases que forman parte de la lógica funcional. Para cada una de las entidades o procesos que componen los casos de uso, estas clases definen lo que ocurre dentro del sistema a través de eventos definidos en ellas. Las acciones realizadas por el usuario son controladas y resueltas por estos eventos permitiendo así la interacción con la interfaz. Estas clases permiten manejar y moldear datos solicitados a la base de datos o por lo contrario reflejarlos en ella, sirven de instancia para conversores y métodos que nos permiten realizar esas modificaciones y transformar los datos según nuestras necesidades.

Por otro lado, podemos observar la carpeta “db”. En ella está alojada la gestión de la base de datos, tiene como objetivo, ser el emisor y receptor de todo el flujo de datos que circula por la aplicación. En esta carpeta, aparecen implementadas todas las consultas y acciones sobre la base de datos necesarias para que el usuario pueda cumplir con los casos de uso, así como también todas las claves y restricciones para garantizar el correcto funcionamiento de esta. Más adelante nos centraremos en su estructura.



Otra carpeta que podemos ver es la llamada “parser”, esta carpeta aloja un transformador empleado a la hora de escanear mediante NFC los carnés UPV.

La carpeta “record” aloja distintas clases empleadas para garantizar una lectura correcta por parte del NFC y sirve como apoyo al transformador, el objetivo de esta carpeta junto con la anterior es mantener el lector NFC activo a la espera del contacto con un carné UPV y una correcta lectura del identificador que alberga dicho carné.

Finalmente, la carpeta “util” contiene todas las entidades que conforman el modelo de dominio, en cada una de estas entidades, se declaran todos sus atributos para que puedan ser utilizados en los distintos eventos que se generen.

## 4.2 Esquema de components

En este punto vamos a identificar los distintos componentes que forman parte del Proyecto y son los encargados de obtener, gestionar y emitir la información que circula por toda la aplicación.

En la **figura 11** podemos observar el esquema de los componentes.

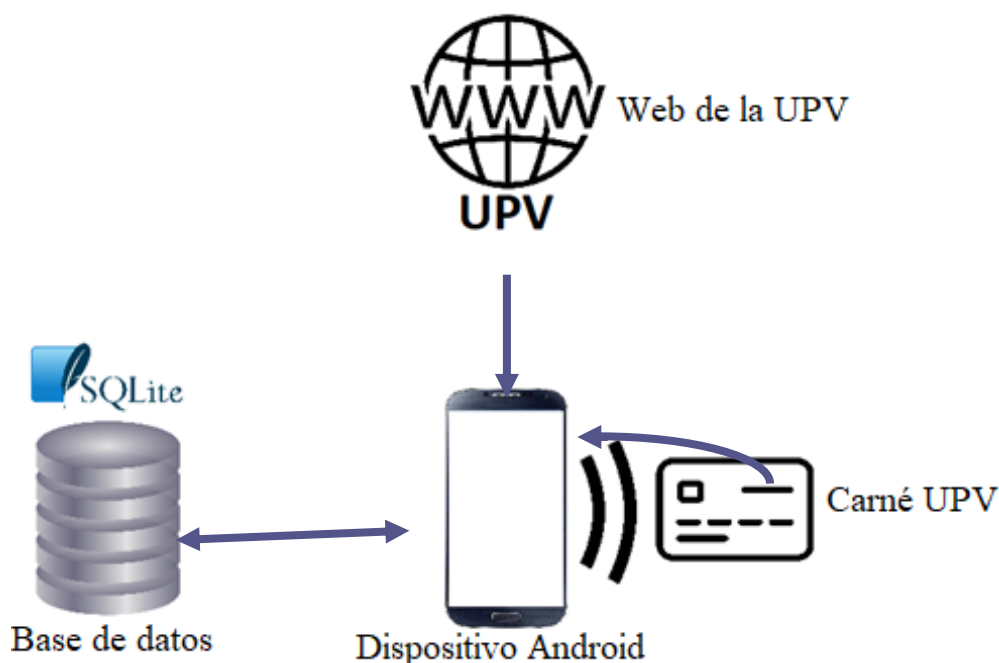


Figura 11. Esquema de componentes

La arquitectura consta de un acceso a la web de la UPV mediante la lectura del código QR<sup>1</sup> del carné UPV para poder extraer automáticamente los datos (Web Scraping) de la persona escaneada. Del mismo modo también se escanea el carné UPV, pero esta vez mediante NFC para obtener el identificador de dicho carné.

<sup>1</sup> Código QR: código de barras bidimensional de forma cuadrada que almacena datos codificados.

Por otro lado, también encontramos un base de datos SQLite que aloja toda la información de las distintas entidades. El flujo de información a la base de datos será tanto de entrada como de salida (lectura y escritura).

Finalmente concluimos la arquitectura con la lectura de 2 archivos en formato XML “asignaturas.xml” y “grupos.xml” que poblarán la base de datos y la generación de un archivo PDF del parte de firmas generado que podrá ser enviado por correo electrónico a cualquier destinatario.

### 4.3 Arquitectura de la aplicación

Nuestra aplicación consta de una arquitectura de tres capas dividiendo el Sistema en tres partes diferentes y cada capa solo puede comunicarse con su capa inferior.

A la hora de desarrollar nuestra aplicación, hemos tenido en cuenta una base de datos para almacenar toda la información tratada que fluye por ella, una interfaz para que los usuarios interactúen con la aplicación y una parte del sistema que será el encargado de procesar datos y gestionar todas las operaciones que se hagan con ellos. La arquitectura en tres capas tiene como resultado la división del sistema en tres partes como muestra la siguiente figura.

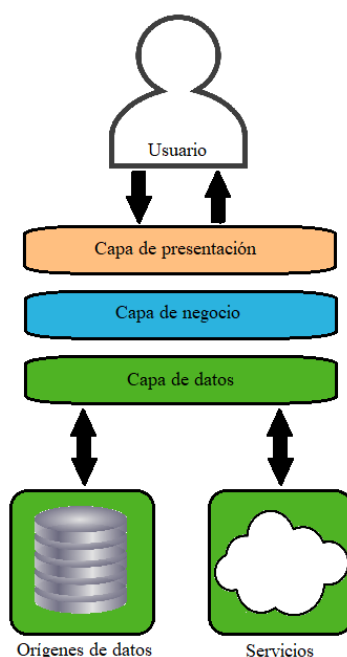


Figura 12. Esquema de la arquitectura de la aplicación

- La capa de presentación responde a la parte visual de la aplicación, en ella se crea la interfaz del usuario y se visualizan los resultados.
- La capa de negocio es la capa intermedia donde se gestiona la lógica de la aplicación, se escoge el propósito de los datos. Las solicitudes propuestas por el usuario son recibidas desde la capa de presentación hasta ella y se relaciona con la capa de persistencia para solicitar o guardar información.
- La capa de persistencia es el lugar donde se almacenan los datos y se encuentran las conexiones necesarias para comunicarse con la base de datos.

Tras lo visto anteriormente, llegamos a la conclusión de que el usuario establece el primer contacto con la capa de presentación mediante su interfaz. Esta capa envía solicitudes a su inferior, la capa de negocio en la que se recogen los datos obtenidos por la capa de persistencia mediante una base de datos o un servicio determinado.

Al emplear la arquitectura en tres capas, logramos la ventaja de dividir el sistema en partes claramente diferenciadas, logrando con ello aumentar la velocidad y la facilidad a la hora de localizar un fallo. Por otro lado, nos resulta más sencillo incluir nuevas funcionalidades o modificar las existentes ya que una mejora o un añadido no provocará la participación de todo el sistema, sino únicamente de la capa afectada, dotando a la aplicación de mantenibilidad.

## 4.4 Arquitectura de la base de datos

SQLite es la base de datos empleada para la aplicación, está compuesta por varias tablas:

- La tabla Asignatura que corresponde a las asignaturas impartidas por los profesores y cursadas por los alumnos y formadas por grupos, consta de las siguientes columnas.
  - Id. Nomenclatura empleada para identificar la asignatura.
  - Titulación. Corresponde a numeración de la licenciatura a la que pertenece.
  - Curso. Año en el que se cursa la asignatura.
  - Er\_gestora. Hace referencia al departamento al que pertenece
  - Idioma. Idioma en el que se cursa la asignatura.
  - Duración. Minutos que dura una clase de esa asignatura
- La tabla Profesor que corresponde a los profesores que imparten la docencia de las asignaturas e imparten clase a los alumnos, consta de las siguientes columnas.
  - Id. Nomenclatura empleada para identificar al profesor.
  - Dni. Número del documento nacional de identificación del profesor.
  - Nombre. Nombre del profesor.
- La tabla Alumno que corresponde a los alumnos que cursan asignaturas impartidas por profesores y pertenecen a un grupo de la asignatura, consta de las siguientes columnas.
  - Id. Nomenclatura empleada para identificar al alumno.
  - Dni. Número del documento nacional de identificación del alumno.
  - Nombre. Nombre del alumno.
- La tabla Grupo que corresponde a los grupos formados por los alumnos que cursan las asignaturas, consta de las siguientes columnas.
  - Id. Nomenclatura empleada para identificar al grupo.
  - Grupo. Nombre del grupo.
  - Hora\_entrada. Hora inicio de la clase según el calendario docente.
  - Hora\_salida. Hora fin de la clase según el calendario docente.
  - Aula. Nombre del aula donde se imparte la clase.
- La tabla Alumnotemporal que corresponde a los alumnos cuya asistencia a clase ha sido contabilizada. Consta de la siguiente columna.
  - Id. Nomenclatura empleada para identificar a los alumnos con asistencia registrada.



Todas las tablas tienen como clave primaria el atributo “id” lo que impide que haya dos registros iguales en cualquiera de las tablas.

La base de datos es generada al iniciar la aplicación por primera vez, está implementada para permitir generar versiones y puede vaciarse desde la aplicación.

A continuación, se muestra en la siguiente figura un esquema de la base de datos.

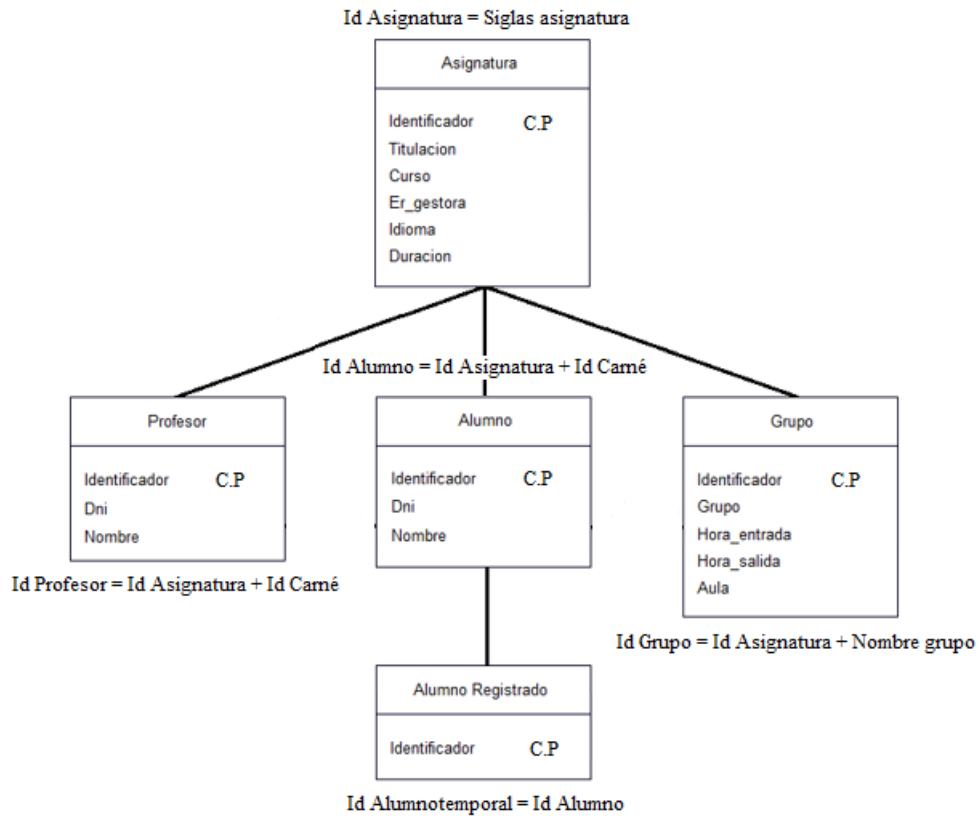


Figura 13. Esquema de la base de datos

## 4.5 Conclusión

Una vez establecido el diseño de nuestra aplicación y de todos los componentes relacionados con su arquitectura, pasaremos a mencionar en el siguiente capítulo la serie de tecnologías y herramientas que han sido clave a la hora de realizar dicho diseño.





# 5. Tecnologías y herramientas

---

En este apartado trataremos en detalle las diferentes tecnologías y herramientas empleadas a la hora de implementar nuestra aplicación

## 5.1 Android

Android es el sistema operativo utilizado para desarrollar nuestra aplicación, está basado en Linux, un software libre que agrada a los desarrolladores por su amplia posibilidad de implementación. Nuestra aplicación permite su funcionamiento en dispositivos cuyo sistema operativo Android es igual o superior a la API<sup>1</sup> 22, conocida como “Lollipop” corresponde a la versión 5 de Android. De este modo conseguimos que dispositivos de hasta 7 años de antigüedad puedan utilizar la aplicación sin ningún problema.

La arquitectura de Android está basada en un sistema de capas sobre un kernel<sup>2</sup> de Linux [6]. En la siguiente figura podemos observar las diferentes capas que la conforman.

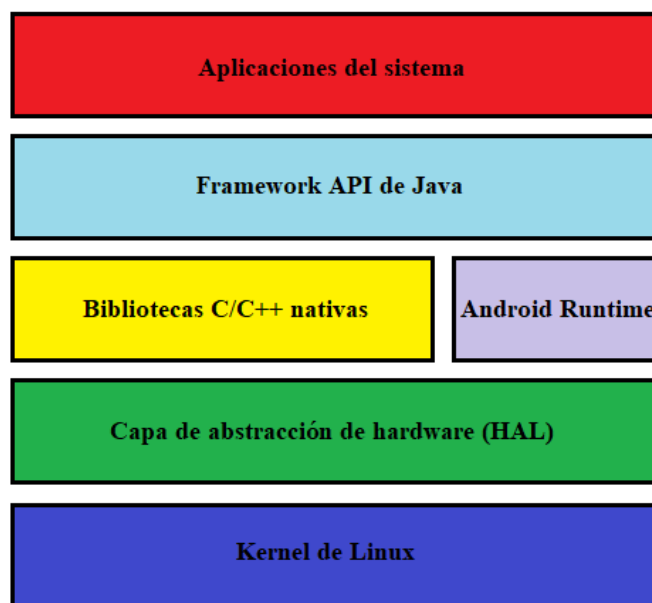


Figura 14. Arquitectura Android

A lo largo de los años Android ha ido lanzando al mercado nuevas versiones de su sistema operativo cada vez incluyendo más y mejores implementaciones que permitan al usuario todo tipo de funcionalidades. En la **figura 14** podemos observar el nombre de todas las versiones desarrolladas por Android y su año de lanzamiento.

---

<sup>1</sup> API: interfaz que permite acceder a las funcionalidades de una aplicación software.

<sup>2</sup> Kernel: parte central del sistema operativo que realiza la comunicación segura entre hardware y software.





Figura 15. Historia versiones Android

Centrándonos en el desarrollo de nuestra aplicación, el entorno de desarrollo (IDE) empleado ha sido Android Studio. Se trata de un software para el desarrollo de aplicaciones. En su interior cabe destacar opciones de funcionalidad como depuración, compilación y ejecución de aplicaciones tanto con un emulador propio que incluye como en el dispositivo del mismo desarrollador [7].

Se trata de un entorno de desarrollo que ya había utilizado anteriormente y por esa misma razón he decidido hacer uso de él para esta aplicación.

## 5.2 SQLite

SQLite se trata de una base de datos de tipo relacional escrita en lenguaje C, totalmente gratuita y de uso público sin necesidad de instalación previa, la librería implementa gran parte del estándar SQL-92 [8]. Su código fuente es sencillo y accesible para todo nivel de programadores lo que justifica su gran popularidad. La base de datos se almacena en un único fichero que utiliza únicamente el espacio necesario en disco.

## 5.3 RFID (NFC)

NFC son las siglas de Near Field Communication, significa comunicación de campo cercano. Se trata de una tecnología que permite comunicarse de forma inalámbrica y el intercambio de datos

entre dispositivos colocados uno cerca del otro de forma instantánea sin necesidad de enlace previo.

Tiene dos modos de funcionamiento, en el modo activo los dispositivos con chip NFC mediante un campo electromagnético intercambian datos y en el modo pasivo, únicamente un dispositivo genera ese campo y el otro aprovecha para transmitir la información.

El chip NFC mencionado anteriormente se trata en realidad de una etiqueta o tarjeta RFID la cual contiene todos los datos y genera una señal que mediante un lector puede ser capturada para poder ser leídos en formato digital [9].

Gracias a esta tecnología hemos podido obtener mediante el lector NFC del dispositivo móvil, el número de identificación del carné UPV empleado para la identificación de los profesores y alumnos. En la siguiente figura, se puede observar un ejemplo de carné de la UPV con etiqueta RFID válida para nuestra aplicación.



Figura 16. Carné UPV

## 5.4 Código QR

El código QR es la evolución del código de barras con una respuesta rápida, permite almacenar información en una matriz de punto. Para poder acceder a dicha información el usuario debe de escanearlo mediante la cámara de su dispositivo una vez hecho, el software interpretará el esquema mostrando el texto almacenado o redirigiendo al usuario a una página web específica.

En relación con la aplicación, hemos utilizado el escaneo del código QR para poder obtener los datos personales del usuario del carné UPV ya que, tras la lectura el usuario es redireccionado a la web de la universidad de donde se extrae de forma automática dicha información mediante Web Scraping..

En la siguiente figura, se puede observar un ejemplo del reverso de un carné de la UPV donde se encuentra el código QR necesario para nuestra aplicación.



Figura 17. Reverso carné UPV

## 5.5 Balsamic

Balsamiq es una herramienta para la elaboración de prototipos. Permite diseñar de manera fluida y cómoda los mockups utilizados para un primer punto de vista de nuestra interfaz durante el proceso de diseño [10]. No es un software libre, pero existe la posibilidad de utilizarlo de forma gratuita mediante la versión de prueba durante un mes.

## 5.6 Adobe Photoshop

Adobe Photoshop es una herramienta que permite la edición y el tratamiento de imágenes con un gran conjunto de funcionalidades, obteniendo así un resultado profesional. Hemos utilizado esta herramienta tanto para la elaboración de las figuras la memoria como en el desarrollo de la aplicación a la hora de diseñar logos e imágenes.

## 5.7 BoUml

BoUml es una herramienta de modelado UML que permite diseñar casos de uso, diagramas de estado, clases, modelos de dominio y esquemas [11]. Además de ser multiplataforma, permite programar en diferentes lenguajes. En nuestro caso lo hemos empleado para la realización de los casos de uso y el modelo de dominio.

## 5.8 Librerías

A continuación, vamos a comentar las distintas librerías empleadas en el desarrollo de nuestra aplicación.

- ZXing [12]: es una librería que ofrece soporte para la lectura y decodificación de la mayoría de los códigos de barras, códigos BIDI o QR en múltiples plataformas.
- JavaMail [13]: es una librería destinada al envío de correos electrónicos directamente desde la aplicación haciendo uso del protocolo SMTP.
- DroidText [14]: es una librería que nos permite en varias plataformas crear, analizar, modificar y mantener documentos en formato PDF.
- Jsoup [15]: es una librería que proporciona operaciones para trabajar con HTML. Permite extraer y manipular datos de cualquier directorio web.

## 5.9 Conclusión

Una vez se han expuesto las diferentes tecnologías empleadas en el proyecto, procedemos en el siguiente capítulo a comentar como han sido utilizadas a la hora de implementar todas las posibles actividades que presenta la aplicación, haciendo uso para ello de la lógica funcional y del diseñador de layouts.

# 6. Implementación

En este punto vamos a hacer hincapié en las funcionalidades de la aplicación y vamos a comentar los pasos que debe de realizar el usuario para llevarlas a cabo.

## 6.1 Actividades

Cuando hablamos de actividad, estamos haciendo referencia a las distintas ventanas que componen nuestra aplicación y que permiten al usuario realizar tareas en ella. Todo ello gracias a la unión de dos elementos clave de la programación como son la lógica funcional y el diseñador de layouts.

### 6.1.1 Lógica funcional

La lógica de negocio engloba a los componentes lógicos del código de programación. En ella residen todas las funciones que se ejecutan, recibe solicitudes del usuario y envía respuestas una vez finalizado en proceso [16]. Define lo que sucede dentro del sistema en consecuencia a eventos.

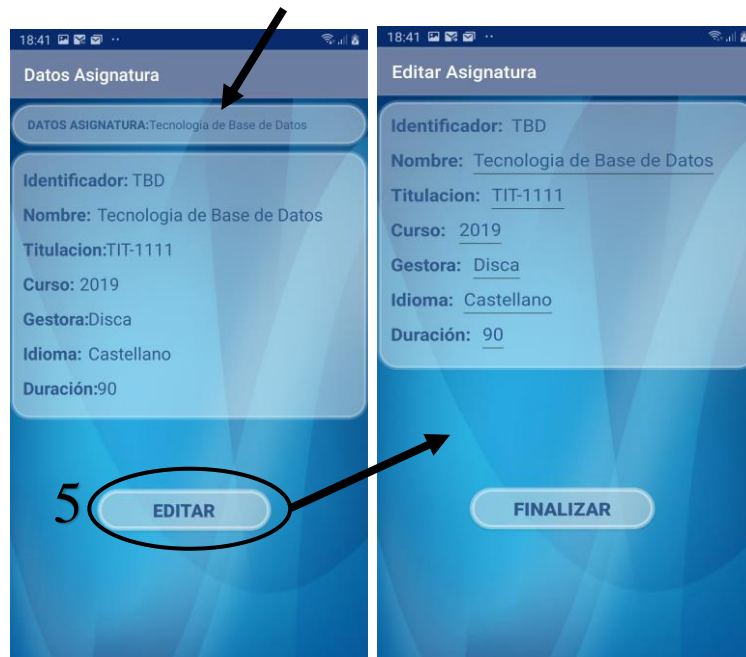
### 6.1.2 Diseñador de layouts

Android Studio permite mediante lenguaje XML la creación de interfaces que permiten separar la presentación de la aplicación del código que controla el comportamiento [17]. También facilita la creación de diseños para diferentes tamaños de pantalla y orientación del dispositivo.

### 6.1.3 Actividades asignatura

Figura 18. Pantallas de eventos con asignaturas





**1.** En esta secuencia el Usuario desde la pantalla de inicio, selecciona el botón de configuración y es redirigido a la ventana de asignaturas, aquí puede observar todas las asignaturas existentes en la base de datos. Seleccionando en una asignatura de la lista, el usuario podrá obtener información de las entidades que la forman y de ella misma. Para ello, tras seleccionar el botón ver, el usuario será redirigido a la pantalla de menú de asignaturas donde tendrá las posibilidades de:

- Ver datos de la asignatura seleccionada.
- Ver los profesores que imparten docencia de la asignatura seleccionada.
- Ver los alumnos que cursan la asignatura seleccionada.
- Ver los grupos que pertenecen a la asignatura seleccionada.

**2.** Seleccionando en una asignatura de la lista, el usuario podrá borrarla seleccionando el botón borrar.

**3.** Otra alternativa posible de evento en esta secuencia es vaciar la base de datos, para ello el usuario debe de seleccionar el botón derecho que hay en el menú superior de la ventana.

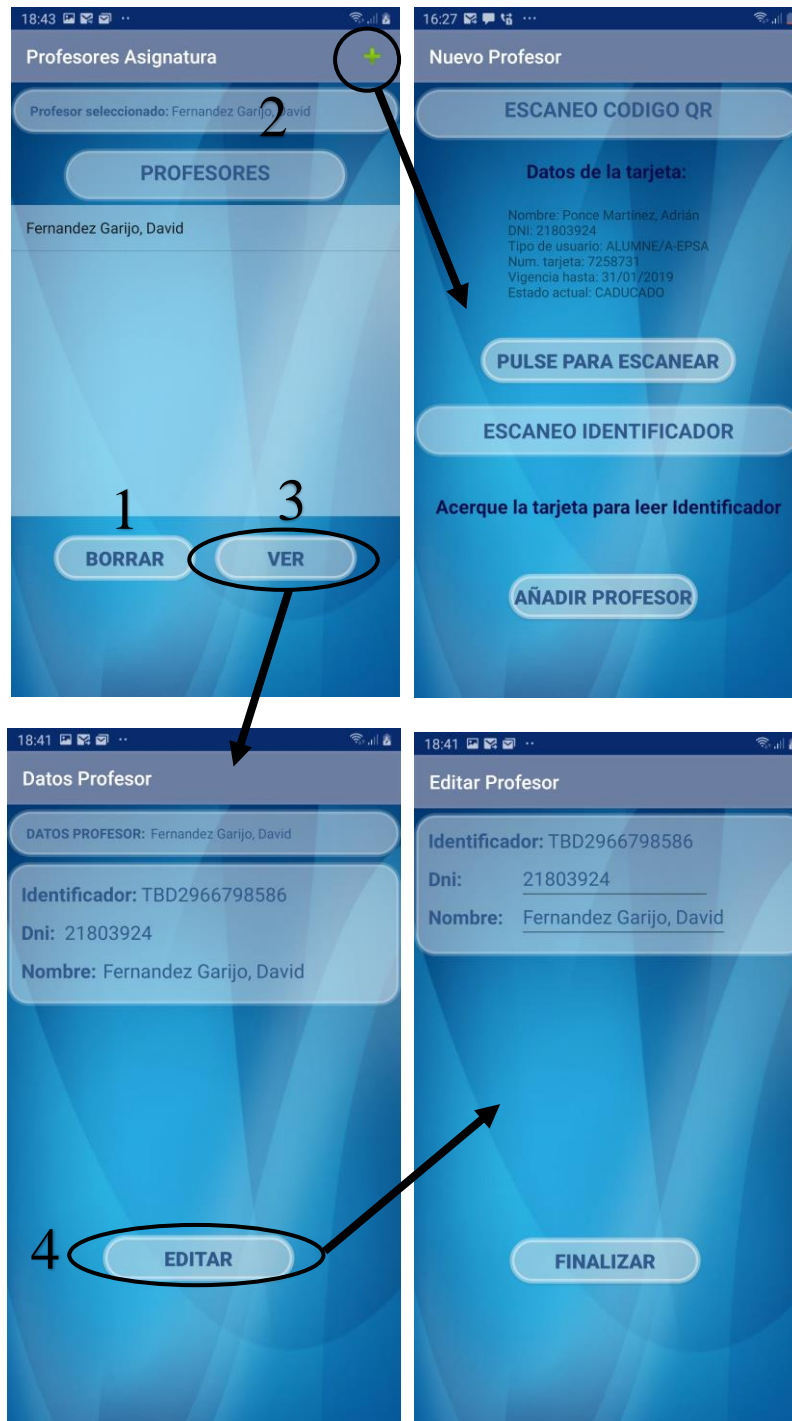
**4.** Si el usuario selecciona el botón de ver datos asignatura será redirigido a la pantalla de datos de la asignatura y ahí podrá observar todos los datos de la asignatura que ha seleccionado.

**5.** En caso de que el usuario quiera modificar los datos de la asignatura, deberá de seleccionar el botón editar para ser redireccionado a la pantalla de editar asignatura y de este modo modificar los registros deseados.



## 6.1.4 Actividades professor

Figura 19. Pantallas de eventos con profesores

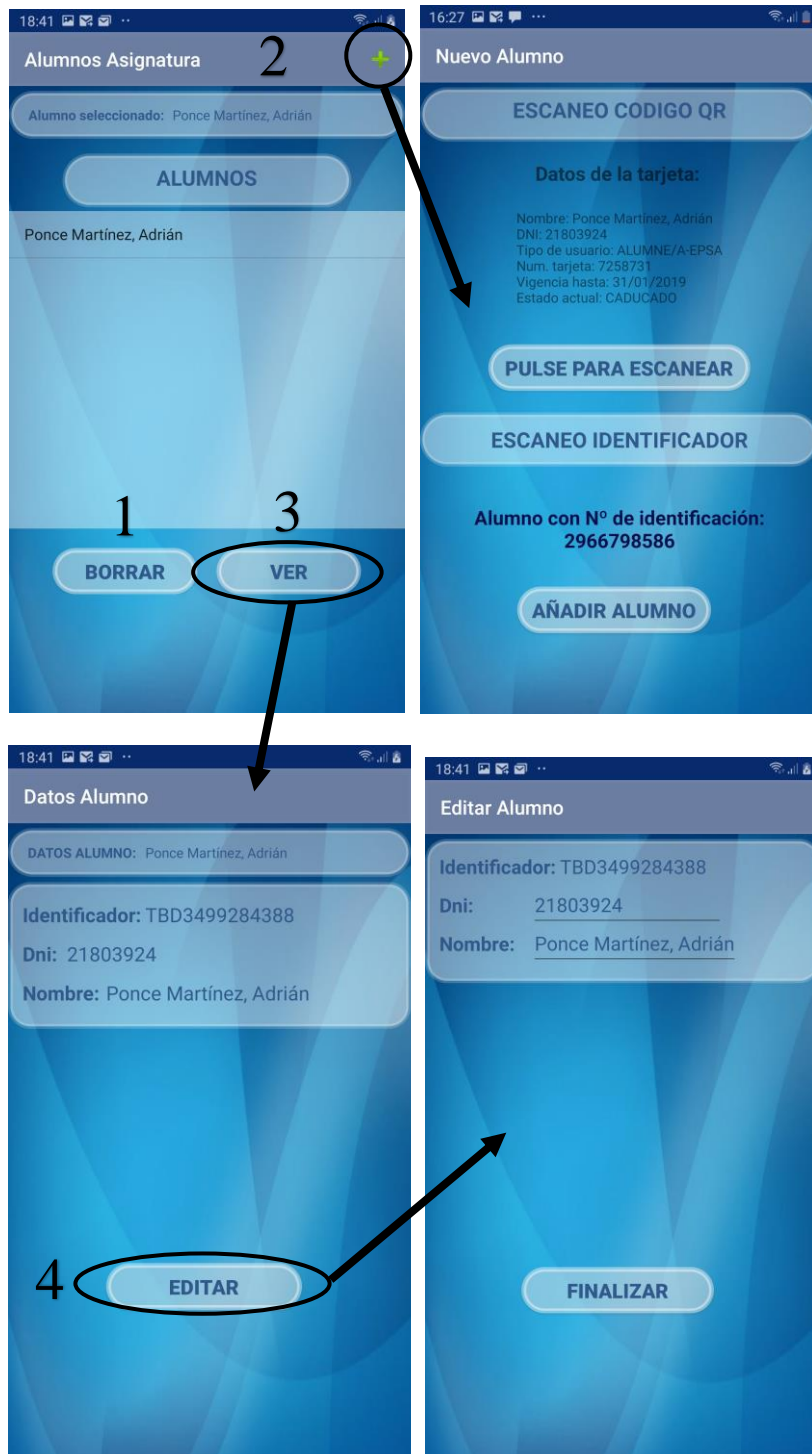


En caso de que el usuario en la primera pantalla de la **figura 16** haya seleccionado el botón ver profesores asignatura será redireccionado a la pantalla de profesores y ahí podrá visualizar todos los profesores existentes en la base de datos.

- 1.** Seleccionando en un profesor de la lista, el usuario podrá borrarlo seleccionando el botón borrar.
- 2.** Otra alternativa posible de evento en esta secuencia es añadir un nuevo profesor, para ello el usuario debe de seleccionar el botón derecho que hay en el menú superior de la ventana y será redirigido a la pantalla de nuevo profesor donde deberá de leer el código QR de su carné UPV y a continuación mediante NFC escanear dicho carné para obtener su identificador. Finalizará el evento pulsando en el botón añadir profesor.
- 3.** Seleccionando en un profesor de la lista, el usuario podrá ver sus datos seleccionando el botón de ver, será redirigido a la pantalla de datos de profesores y ahí podrá observar todos los datos del profesor que ha seleccionado anteriormente.
- 4.** En caso de que el usuario quiera modificar los datos del profesor, deberá de seleccionar el botón editar para ser redireccionado a la pantalla de editar profesor y de este modo modificar los registros deseados. Finalizará el evento pulsando en el botón finalizar.

## 6.1.5 Actividades alumno

Figura 20. Pantallas de eventos con alumnos



En caso de que el usuario en la primera pantalla de la **figura 16** haya seleccionado el botón ver alumnos asignatura será redireccionado a la pantalla de alumnos y ahí podrá visualizar todos los alumnos existentes en la base de datos.

1. Seleccionando en un alumno de la lista, el usuario podrá borrarlo seleccionando el botón borrar.
2. Otra alternativa posible de evento en esta secuencia es añadir un nuevo alumno, para ello el usuario debe de seleccionar el botón derecho que hay en el menú superior de la ventana y será redirigido a la pantalla de nuevo alumno donde deberá de leer el código QR de su carné UPV y a continuación mediante NFC escanear dicho carné para obtener su identificador. Finalizará el evento pulsando en el botón añadir alumno.
3. Seleccionando en un alumno de la lista, el usuario podrá ver sus datos seleccionando el botón de ver, será redirigido a la pantalla de datos de alumnos y ahí podrá observar todos los datos del alumno que ha seleccionado anteriormente.
4. En caso de que el usuario quiera modificar los datos del alumno, deberá de seleccionar el botón editar para ser redireccionado a la pantalla de editar alumno y de este modo modificar los registros deseados. Finalizará el evento pulsando en el botón finalizar.

## 6.1.6 Actividades grupo

Figura 21. Pantallas de eventos con grupos



En caso de que el usuario en la primera pantalla de la **figura 16** haya seleccionado el botón ver grupos asignatura será redireccionado a la pantalla de grupos y ahí podrá visualizar todos los grupos existentes en la base de datos.

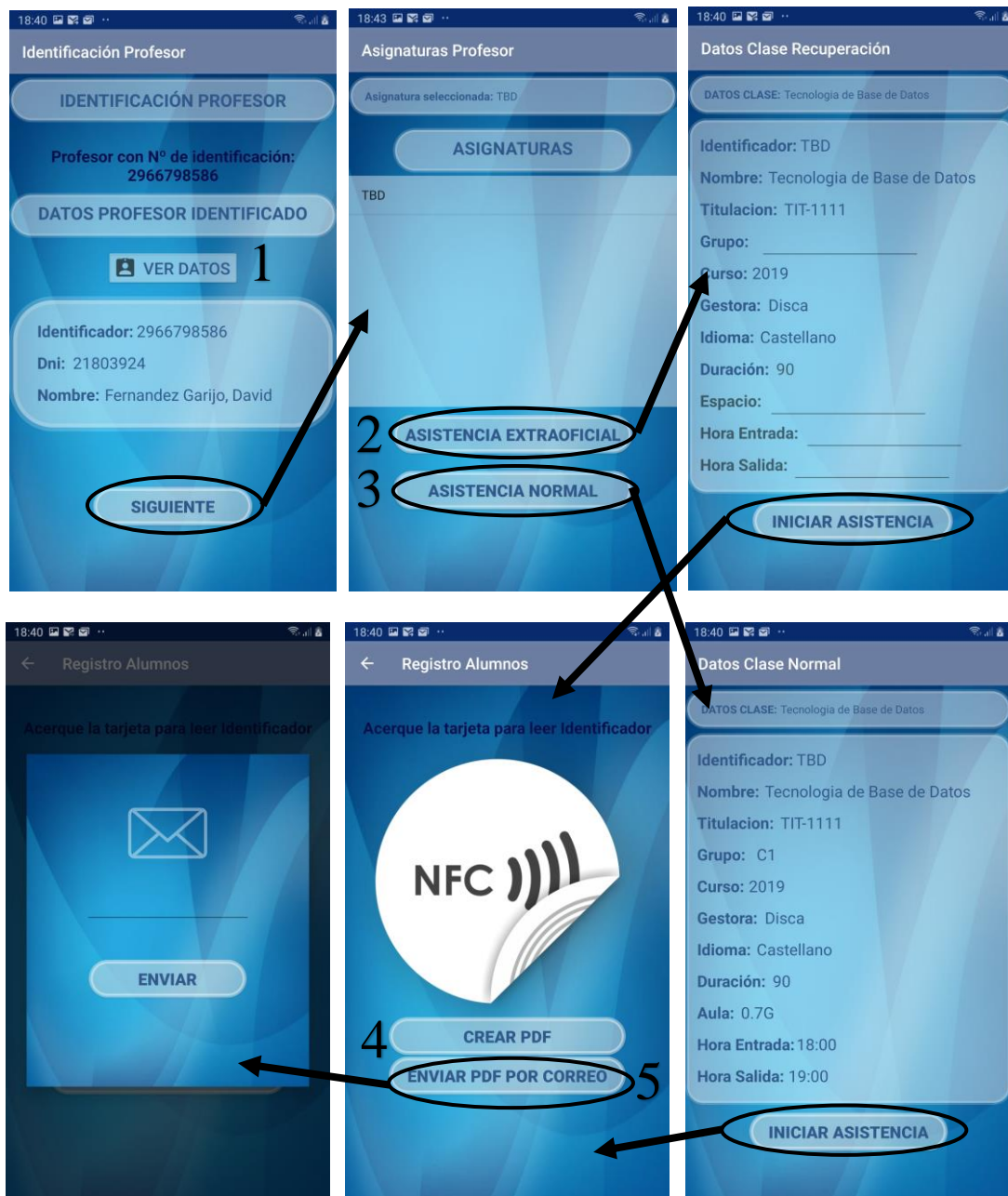
1. Seleccionando en un grupo de la lista, el usuario podrá borrarlo seleccionando el botón borrar.

2. Seleccionando en un grupo de la lista, el usuario podrá ver sus datos seleccionando el botón de ver, será redirigido a la pantalla de datos de grupos y ahí podrá observar todos los datos del grupo que ha seleccionado anteriormente.

3. En caso de que el usuario quiera modificar los datos del grupo, deberá de seleccionar el botón editar para ser redireccionado a la pantalla de editar grupo y de este modo modificar los registros deseados. Finalizará el evento pulsando en el botón finalizar.

### 6.1.7 Actividades creación parte de firmas

Figura 22. Pantallas de eventos para generar un parte de firmas



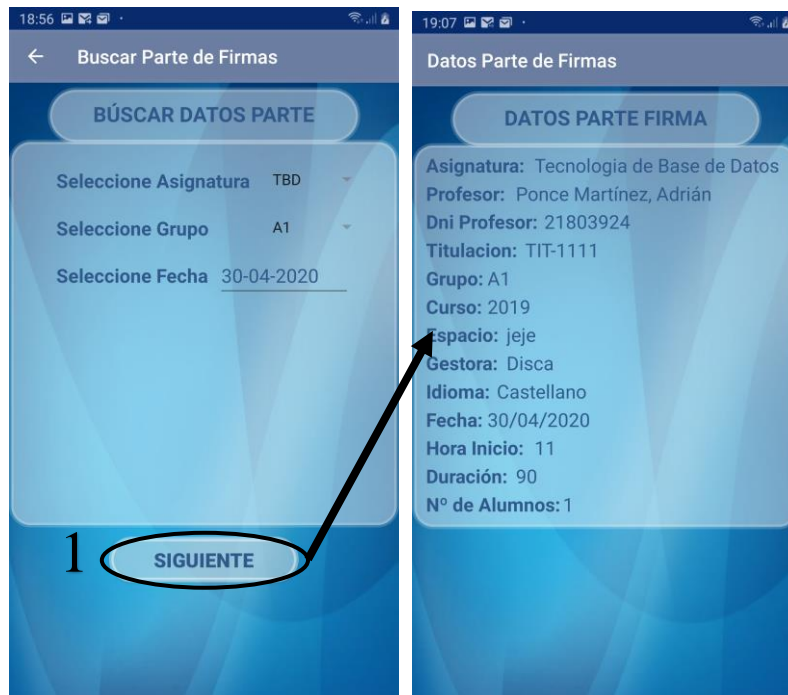
En caso de que el usuario en la primera pantalla de la **figura 16** haya seleccionado el botón de identificar profesor será redireccionado a la pantalla de asignaturas del profesor y ahí podrá visualizar todas las asignaturas en las que imparte docencia el profesor autenticado.

- 1.** En primer lugar el usuario debe de identificarse escaneando mediante NFC su carne de la UPV. Tras este paso, si el usuario selecciona el botón ver datos podrá visualizar los datos del profesor autenticado. Para finalizar el evento el usuario seleccionará el botón siguiente y será redireccionado a la pantalla de asignaturas profesor donde podrá visualizar todas las asignaturas en las que imparte docencia el profesor autenticado, para posteriormente seleccionar la que desea generar su parte de firmas.
- 2.** Si el usuario selecciona el botón de asistencia extraoficial, será redirigido la pantalla de datos clase de recuperación donde de forma automática se le rellenarán todos los registros, excepto Grupo, Espacio, Hora entrada y Hora salida ya que se trata de una case fuera del horario docente. Finalizará seleccionando el botón iniciar asistencia.
- 3.** Si el usuario selecciona el botón de asistencia normal, será redirigido la pantalla de datos clase normal donde de forma automática se le rellenarán todos los registros ya que se trata de una case dentro del horario docente. Finalizará seleccionando el botón iniciar asistencia.
- 4.** Tras iniciar la asistencia, el usuario será redirigido a la ventana de registro de los alumnos donde cada alumno debe de reflejar su asistencia escaneando mediante NFC su carné UPV. Finalmente, para obtener el parte de firmas, el usuario seleccionará el botón crear PDF y podrá posteriormente visualizarlo con otra aplicación externa.
- 3.** Tras la creación del parte, el usuario tiene la posibilidad de enviarlo por correo electrónico junto con sus datos en formato XML a cualquier destinatario. Para ello, debe de seleccionar el botón enviar PDF por correo y se abrirá una ventana emergente donde debe de escribir la dirección de correo del destinatario y finalmente seleccionar el botón enviar para terminar el proceso.



## 6.1.8 Actividades búsqueda parte de firmas

Figura 23. Pantallas de eventos para enviar correo electrónico



En caso de que el usuario en la primera pantalla de la **figura 16** haya seleccionado el botón de buscar parte, será redireccionado a la pantalla de buscar parte de firmas.

1. En primer lugar el usuario debe de rellenar todas las credenciales de búsqueda (Asignatura, Grupo y Fecha del parte), tras ello seleccionará el botón siguiente y será redireccionado a la pantalla de datos parte de firmas donde podrá observar toda la información perteneciente al parte.

## 6.2 Comunicación con la base de datos

En la aplicación se generan una serie de eventos que provocan el intercambio de información y para garantizarlo el sistema necesita una base de datos donde poder almacenarla.

Para poder implementar una base de datos en nuestra aplicación, en primer lugar, debemos de establecer su esquema y declarar todas las tablas junto con sus atributos que formaran parte de ella.

A la hora de realizar operaciones de lectura y escritura se deben de crear una serie de accesos a la base de datos, para ello hemos implementado una conexión mediante la ejecución de consultas. A continuación, vamos a exponer algunas de ellas para conocer mejor su origen y funcionamiento.

En primer lugar, vamos a ver el método utilizado para la eliminación de un profesor en particular, se eliminará de la tabla profesor en base a su identificador.



```

public void borrarTodoProfesores(String identificador) {
    SQLiteDatabase db = getWritableDatabase();
    if(db!=null) {
        db.execSQL("DELETE FROM PROFESOR WHERE id LIKE '"+identificador+"%");
        db.close();
    }
}

```

*Fragmento 1. Método de borrado*

En este caso, el siguiente fragmento de código mostrado corresponde al método para la inserción de un profesor en particular, se añadirá a la tabla profesores con el valor de sus atributos correspondientes.

```

public void agregarProfesor(String identificador, String dni, String nombre) {
    SQLiteDatabase db = getWritableDatabase();

    if(db!=null) {
        db.execSQL("INSERT INTO PROFESOR VALUES('"+identificador+"','"+dni+"','"+nombre+"') ");
        db.close();
    }
}

```

*Fragmento 2. Método de inserción*

Por otro lado, se realizan una serie de consultas como por el ejemplo el siguiente método que tiene como resultado obtener de forma ordenada atributos de un alumno con cierto identificador.

```

Cursor cursor=db.rawQuery( sql: "SELECT * FROM ALUMNO WHERE id LIKE"+"'+asignatura+'% " +
    "ORDER BY nombre", selectionArgs: null);

```

*Fragmento 3. Método de consulta*

Otro tipo de método de consulta empleado es aquel utilizado para actualizar la información de la base de datos. En el siguiente fragmento podemos observar un método empleado para actualizar de la tabla profesores, el nombre y el DNI de un profesor en base a su identificador.

```

public void ActualizarProfesor(String identificador, String nombre, String dni) {
    SQLiteDatabase db = getWritableDatabase();
    if(db!=null) {
        db.execSQL("UPDATE PROFESOR SET nombre='"+nombre+"', dni='"+dni+" ' +
            "WHERE id = '"+identificador+'");
        db.close();
    }
}

```

*Fragmento 4. Método de actualización*



## 6.3 Extracción de datos de la TUI

En la aplicación ocurren ciertos eventos como el alta de profesores y alumno que necesitan cierta información personal para poder llevarse a cabo. Para ello empleamos un método de extracción automática de datos conocido como “WebScraping”. Este método se basa en obtener ciertos datos de una página web.

Para poder acceder a la página objetivo de la cual queremos recoger información, hacemos una lectura del código QR situado en la parte trasera del carné UPV. Una vez hemos accedido a la página comienza el barrido de los datos necesario para nuestra aplicación.

A continuación, se muestra el método empleado para recoger los datos leídos del código QR y con ellos, acceder a una dirección web en particular.

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    IntentResult result = IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
    if (result != null) {
        if (result.getContents() == null) {
            Toast.makeText( context: this, text: "Cancelaste el escaneo", Toast.LENGTH_LONG).show();
        } else {
            String[] strParts = result.getContents().split( regex: "/" );
            getWebsiteInfo( url: "https://www.upv.es/pls/oalu/sic_tui.inicio?p_tui="+
                strParts[strParts.length-1]);
        }
    } else {
        super.onActivityResult(requestCode, resultCode, data);
    }
}
```

Fragmento 5. Método de acceso web mediante lectura de código QR

Otro método importante, es aquel en el que una vez tenemos la información de la página web, nos quedamos únicamente con los datos que sean de nuestro interés. En este caso el estado de alta, el número, la fecha de vigencia del carné, el nombre y la entidad del dueño del carné.

```
for (int i=0; i< links.size(); i++){
    org.jsoup.nodes.Element e = links.get(i);
    String token = null;
    if (e!=null) token = e.text();
    if ((token!=null) && (e.text().length()>0)) {
        if (ci.isDNI(token)){ ci.setDNI(links.get(++i).text());
            dni = links.get(i).text();
        }
        else if (ci.isEstado(token)) ci.setEstado(links.get(++i).text());
        else if (ci.isTarjeta(token)) ci.setTarjeta(links.get(++i).text());
        else if (ci.isVigencia(token)) ci.setVigencia(links.get(++i).text());
        else if (ci.isNombre(token)){
            ci.setNombre(links.get(++i).text());
            nombre = links.get(i).text();
            ci.setFuncion(links.get(++i).text());
            funcion = links.get(i).text();
        }
    }
}
builder.append(ci.toString());
```

Fragmento 6. Código para la extracción de datos HTML

## 6.4 Lectura NFC

La lectura mediante NFC es el objetivo fundamental de la aplicación ya que permite obtener el número de identificación del carné empleado para el alta de entidades y el registro de la asistencia del alumno durante el proceso de creación de los partes de firmas.

Para poder obtener el identificador, cuando el usuario acerque su carné al lector NFC de su dispositivo este detectará el campo electromagnético recogiendo la información del carné y localizará su identificador que estará expresado en bytes tal y como muestra la siguiente figura.

Más tarde para su uso, será tratado convirtiéndolo a decimal.

```
if (NfcAdapter.ACTION_TAG_DISCOVERED.equals(action)
    || NfcAdapter.ACTION_TECH_DISCOVERED.equals(action)
    || NfcAdapter.ACTION_NDEF_DISCOVERED.equals(action)) {
    Parcelable[] rawMsgs = intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);
    NdefMessage[] msgs;
    if(rawMsgs != null) {
        msgs = new NdefMessage[rawMsgs.length];
        for (int i = 0; i < rawMsgs.length; i++) {
            msgs[i] = (NdefMessage) rawMsgs[i];
        }
    }else{
        byte[] empty = new byte[0];
        byte[] id = intent.getBytesExtra(NfcAdapter.EXTRA_ID);
        identificador = id;
        Tag tag = (Tag) intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
        byte[] payload = dumpTagData(tag).getBytes();
        NdefRecord record = new NdefRecord(NdefRecord.TNF_UNKNOWN, empty, id, payload);
        NdefMessage msg = new NdefMessage(new NdefRecord[] {record});
        msgs = new NdefMessage[] {msg};
    }
    displayMsgs(msgs);
}
```

*Fragmento 7. Código para la lectura del identificador por NFC*

## 6.5 Creación del archivo PDF y XML

Una vez que el profesor ha contabilizado la asistencia de todos los alumnos, puede generar el parte de firmas que será creado en formato PDF, el aparecerán a modo de cabecera todos los datos referentes a la asignatura y su grupo, además de los datos del profesor que ha impartido la clase. Por otro lado, se verá reflejado a modo de tabla los datos de los alumnos que han registrado su asistencia. La siguiente figura muestra un ejemplo de parte de firmas generado mediante la aplicación.

Parte de firmas Universidad Politécnica de Valencia

---

## SEGUIMIENTO DE LAS ACTIVIDADES DOCENTES

Espacio: 0.3G		
Asignatura: TBD-Tecnología de Base de Datos Titulación: TIT-1211 Grupo: A3		Nombre Profesor: Fernandez Garjio, David DNI: 21803924
Curso/Sem.: 2019 ER Gestora: Disca Idioma: Castellano	Fecha: 12/05/2020 Hora: 12:00 Duración: 90	Firma:
Observaciones:		

IDENTIFICADOR	DNI	NOMBRE
2966798586	21803924	Ponce Martínez, Adrián
3499284388	24763286S	Romero Navarro, Paula

---

Parte de firmas Universidad Politécnica de Valencia

Figura 24. Parte de firmas

Para la creación del PDF se han llevado a cabo ciertos procesos que requieren de los fragmentos de código siguientes.

En primer lugar, es necesario crear el archivo y posteriormente lanzarlo a ejecución.

```
//Creación archivo pdf
File f = crearFichero(nombre_documento);
FileOutputStream ficheroPdf = new FileOutputStream(f.getAbsolutePath());
PdfWriter writer = PdfWriter.getInstance(documento, ficheroPdf);
//Apertura
documento.open();
```

Fragmento 8. Creación y apertura del archivo PDF

Después, incluiremos elementos dentro del PDF como pueden ser una cabecera, un pie de página, títulos, párrafos y una tabla como se muestra los siguientes fragmentos de código.

```
HeaderFooter cabecera = new HeaderFooter(new Phrase(
    string: "Parte de firmas Universidad Politécnica de Valencia"), numbered: false);
cabecera.setAlignment(Element.ALIGN_CENTER);
HeaderFooter pie = new HeaderFooter(new Phrase(
    string: "Parte de firmas Universidad Politécnica de Valencia"), numbered: false);
pie.setAlignment(Element.ALIGN_CENTER);
```

*Fragmento 9. Creación de cabecera y pie*

```
Font font = FontFactory.getFont(FontFactory.HELVETICA_BOLD, size: 20,
    Font.BOLD, Color.BLACK);
documento.add(new Paragraph( string: "SEGUIMIENTO DE LAS ACTIVIDADES DOCENTES\n", font));
```

*Fragmento 10. Creación de título o párrafo*

```
PdfPTable tablaA = new PdfPTable( numColumns: 1);
tablaA.setWidthPercentage(100.00f);
tablaA.addCell( text: "\nEspacio: "+aula+"\n\n");
documento.add(tablaA);
```

*Fragmento 11. Creación de tabla*

Por otro lado, toda la información referente al parte de firmas es guardada en un archivo XML mediante la ejecución de las siguientes líneas de código.

```
XmlSerializer serializer = Xml.newSerializer();
String documento = "ParteFirmas-" + asignatura + "-" + grupo + "-" +
    fecha.replace( oldChar: '/', newChar: '-');
nombre_documentoXml = "ParteFirmas-" + asignatura + "-" + grupo + "-" +
    fecha.replace( oldChar: '/', newChar: '-') + ".xml";
File f = crearFichero2(nombre_documentoXml);
FileOutputStream ficheroXml = new FileOutputStream(f.getAbsolutePath());
OutputStreamWriter fout =
    new OutputStreamWriter(ficheroXml);
try {
    serializer.setOutput(fout);
    serializer.startDocument( encoding: "UTF-8", standalone: true);
    serializer.startTag( namespace: "", name: "ParteFirmas");

    serializer.startTag( namespace: "", documento);

    serializer.startTag( namespace: "", name: "espacio");
    serializer.text(aula);
    serializer.endTag( namespace: "", name: "espacio");
}
```

*Fragmento 12. Código creación de archivo XML*

Tras la ejecución del código anterior obtendremos un archivo XML con la siguiente estructura.

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<ParteFirmas>
  <ParteFirmas-TBD-A3-12-05-2020>
    <espacio>0.3G</espacio>
    <asignatura>TBD-Tecnología de Base de Datos</asignatura>
    <titulacion>TIT-1211</titulacion>
    <grupo>A3</grupo>
    <nombreProfesor>Fernandez Garijo, David</nombreProfesor>
    <dni>21803924</dni>
    <curso>2019</curso>
    <gestora>Disca</gestora>
    <idioma>Castellano</idioma>
    <fecha>12/05/2020</fecha>
    <horaInicio>12:00</horaInicio>
    <duracion>90</duracion>
    <numeroAlumnos>2</numeroAlumnos>
  </ParteFirmas-TBD-A3-12-05-2020>
</ParteFirmas>
```

*Fragmento 13. Estructura archivo XML del parte de firmas*

Tanto el archivo PDF como el XML siguen la misma estructura para su nomenclatura ParteFirmas-”Nombre de la asignatura”-“Nombre del grupo”-“Fecha creación en formato dd-mm-aaaa”.

## 6.6 Carga y estructura de asignaturas y grupos

Otro tipo de implementación que hemos tenido en cuenta ha sido poblar la base de datos, más específicamente la inserción de datos en las tablas asignaturas y grupos. Para ello hemos desarrollado una lectura de archivos en formato XML donde se recogen todos los datos.

Para poder leer un archivo XML hemos implementado un método donde creamos un transformador que permita la lectura de un documento señalando el directorio donde reside, posteriormente indicamos que dicha lectura debe de comenzar en una parte específica de nuestro archivo, en concreto donde aparece cierta palabra.

Esta búsqueda resulta efectiva gracias a la estructura del archivo ya que está compuesto por nodos que vamos recorriendo. Una vez localizado el nodo deseado, recogemos todos sus atributos y los guardamos en variables para su posterior uso. Todo lo mencionado anteriormente aparece reflejado en las siguientes líneas de código.

```

private void leerXML1() {
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    DocumentBuilder dobu;
    try {
        dobu = dbf.newDocumentBuilder();
        Context context = getApplicationContext();
        String filePath = "/storage/emulated/0/Download/Asignaturas.xml";
        InputStream in = new BufferedInputStream(new FileInputStream(filePath));
        Document doc = dobu.parse(in);
        NodeList num = doc.getElementsByTagName("NumeroAsignaturas");
        Node valor = num.item( index: 0);
        String numero = valor.getTextContent();
        numfinal = Integer.parseInt(numero);
        NodeList asignaturas = doc.getElementsByTagName("Asignatura" + count);
        for (int i = 0; i < asignaturas.getLength(); i++) {
            Node asignatura = asignaturas.item(i);
            NodeList asignaturaInfo = asignatura.getChildNodes();
            for (int j = 0; j < asignaturaInfo.getLength(); j++) {
                Node info = asignaturaInfo.item(j);
                listaAsignaturas2[j] = info.getTextContent();
            }
            identificadorAsignatura = listaAsignaturas2[1];
            nombre = listaAsignaturas2[3];
            titulacion = listaAsignaturas2[5];
            curso = listaAsignaturas2[7];
            gestora = listaAsignaturas2[9];
            idioma = listaAsignaturas2[11];
            duracion = listaAsignaturas2[13];
        }
    } catch (ParserConfigurationException | IOException | SAXException e) {
        e.printStackTrace();
    }
}
}

```

Fragmento 14. Método lectura archivo XML

Este proceso se realiza con los archivos “asignaturas.xml” y “grupos.xml” cuya estructura veremos a continuación.

```

<Asignaturas>
  <NumeroAsignaturas>4</NumeroAsignaturas>
  <Asignatural>
    <Identificador>TBD</Identificador>
    <Nombre>Tecnologia de Base de Datos</Nombre>
    <Titulacion>TIT-1111</Titulacion>
    <Curso>2019</Curso>
    <Gestora>Disca</Gestora>
    <Idioma>Castellano</Idioma>
    <Duracion>90</Duracion>
  </Asignatural>

```

Fragmento 15. Estructura archivo asignaturas.xml



```

<Grupos>
  <NumeroGrupos>9</NumeroGrupos>
  <Grupol>
    <Identificador>TBD-A1</Identificador>
    <grupo>A1</grupo>
    <hEntrada>10:00</hEntrada>
    <hSalida>11:00</hSalida>
    <Aula>0.1G</Aula>
  </Grupol>

```

*Fragmento 16. Estructura archivo asignaturas.xml*

## 6.7 Envío de datos mediante correo electrónico

Tras la creación del parte de firmas, el usuario tiene la posibilidad de enviar los archivos generados por correo electrónico al destinatario que desee.

Para ello se ha implementado un método que permite introducir el destinatario del correo, un asunto, un mensaje aclarativo y datos adjuntos que son añadidos indicando el directorio donde residen. Todo este proceso podemos visualizarlo en las siguientes líneas de código.

```

private void enviarEmail() {
    StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
    StrictMode.setVmPolicy(builder.build());
    Log.e( tag: "Test email:", msg: "enviando email");
    nombre_directorioPdf = "ParteFirmasUPV/" + nombre_documento;
    nombre_directorioXml = "DatosParteFirmasUPV/" + nombre_documentoXml;
    direccion = textDireccion.getText().toString();
    String[] TO = {String.valueOf(direccion)};
    String[] CC = {};
    Intent emailIntent = new Intent(Intent.ACTION_SEND_MULTIPLE);
    emailIntent.putExtra(Intent.EXTRA_EMAIL, TO);
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, value: "Parte de Firmas");
    emailIntent.putExtra(Intent.EXTRA_TEXT, value: "Se adjunta Pdf de Parte de firmas");
    emailIntent.setType("*/*");
    ArrayList<Uri> uris = new ArrayList<Uri>();
    String[] filePaths = new String[]{"storage/emulated/0/Download/" + nombre_directorioPdf,
        "storage/emulated/0/Download/" + nombre_directorioXml};
    for (String file : filePaths) {
        File fileIn = new File(file);
        Uri u = Uri.fromFile(fileIn);
        uris.add(u);
    }
    emailIntent.putParcelableArrayListExtra(Intent.EXTRA_STREAM, uris);
    startActivity(emailIntent);
}

```

*Fragmento 17. Método para el envío de archivos por correo*

## 6.8 Conclusión

Una vez se han implementado las diversas actividades que permiten llevarse a cabo, daríamos por finalizado el proceso de diseño e implementación. El siguiente capítulo contiene las diversas pruebas a realizar tras finalizar el proceso anterior para conocer así la opinión del usuario ante diversos aspectos de nuestra aplicación.







# 7. Pruebas sobre la aplicación

---

En este punto vamos a comprobar que la aplicación cumple con los objetivos planteados mediante la realización una serie de pruebas.

## 7.1 Pruebas de usuario

En primer lugar, vamos a comprobar el correcto funcionamiento de nuestra aplicación con la ayuda de varios usuarios. Hemos instalado la aplicación en sus dispositivos móviles, les hemos pedido que realizaran ciertas acciones y acto seguido hemos realizado un breve cuestionario para conocer su opinión.

Los usuarios que han participado en esta prueba comprenden edades entre 20 y 25 años, esto quiere decir que están bastante familiarizados con la tecnología y haces uso de ella a diario.

Para la realización de los cuestionarios nos hemos basado en 4 preguntas clave.

- ¿Se ha sentido cómodo utilizando la aplicación?
- ¿La aplicación ha conseguido identificar sus datos personales?
- ¿Mientras que usaba la aplicación le han servido de apoyo los mensajes de ayuda?
- ¿Considera que la aplicación tiene un diseño atractivo?
- ¿En algún momento la aplicación se ha bloqueado o finalizado?

Tras un análisis de todas las respuestas, estos han sido los resultados.

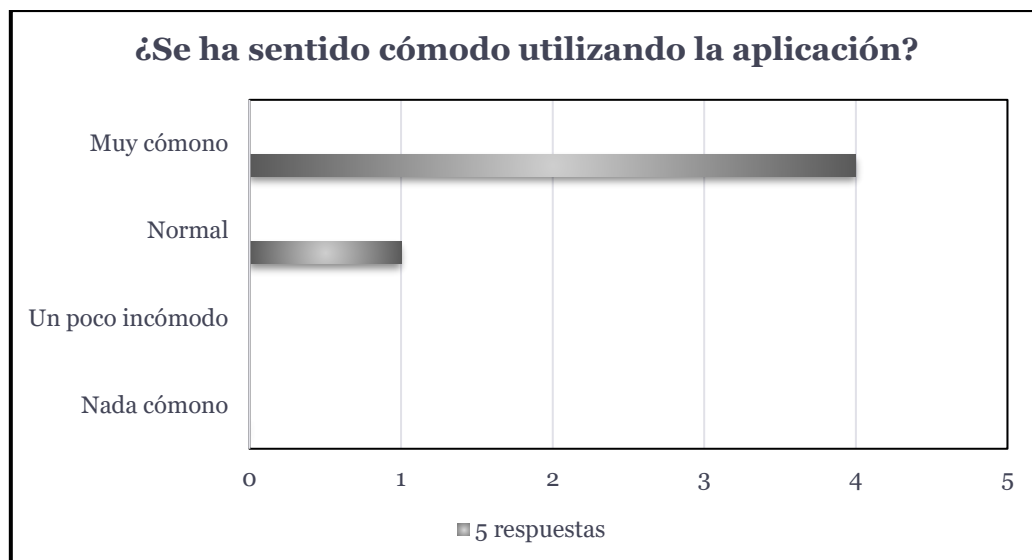


Figura 25. Primera pregunta en las pruebas de usuario

Como podemos comprobar los usuarios están bastante contentos a la hora de utilizarla. Esto indica que se trata de una aplicación intuitiva y fácil de utilizar.

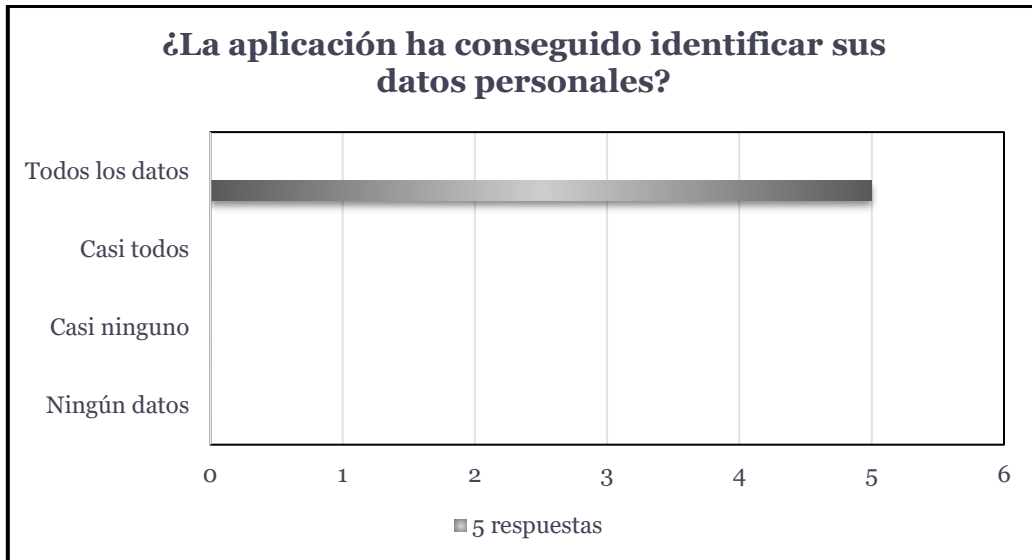


Figura 26. Segunda pregunta en las pruebas de usuario

En este caso hay una opinión común en todos los usuarios. La aplicación ha logrado obtener e identificar todos los datos del usuario, logrando así cumplir su principal objetivo. Este proceso es crucial que funcione perfectamente ya que gracias a esta identificación logramos que el resto de los eventos sean posibles

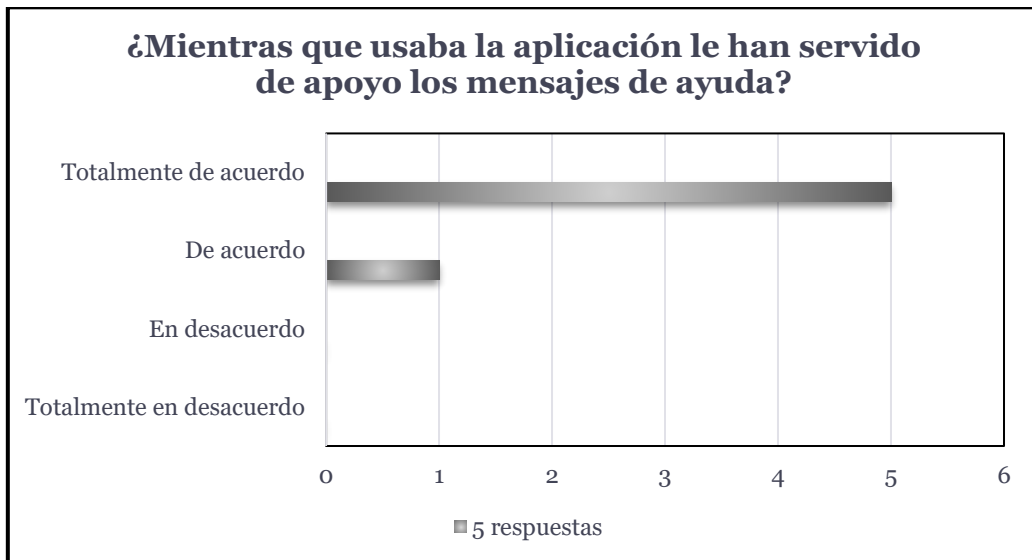


Figura 27. Tercera pregunta en las pruebas de usuario

Gracias a esta pregunta comprobamos que, ante cualquier fallo, la aplicación siempre mantiene al usuario al tanto del problema y le proporciona soporte inmediato. Consultando al usuario que presentaba cierto desagrado, hemos constatado que no todos los dispositivos tienen el lector NFC en el mismo lugar, por lo que la posición del carné puede ser diferente. Este problema es externo a nuestra aplicación y no podemos darle soporte.

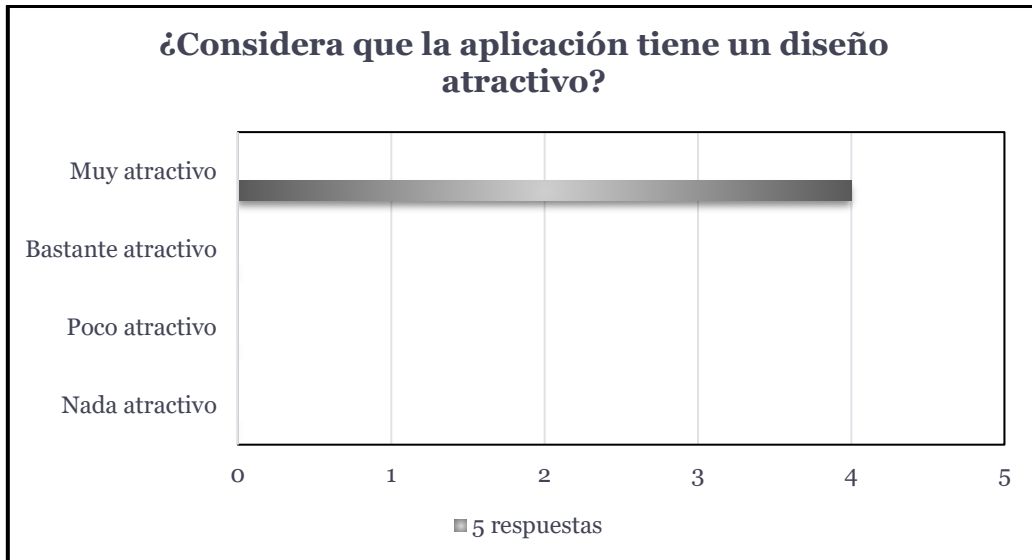


Figura 28. Cuarta pregunta en las pruebas de usuario

Mediante esta pregunta, conocemos como de atractiva es la interfaz que hemos diseñado. Todos los usuarios han reconocido que la encuentran muy atractiva dotándola así de una aplicación bastante agradable visualmente.

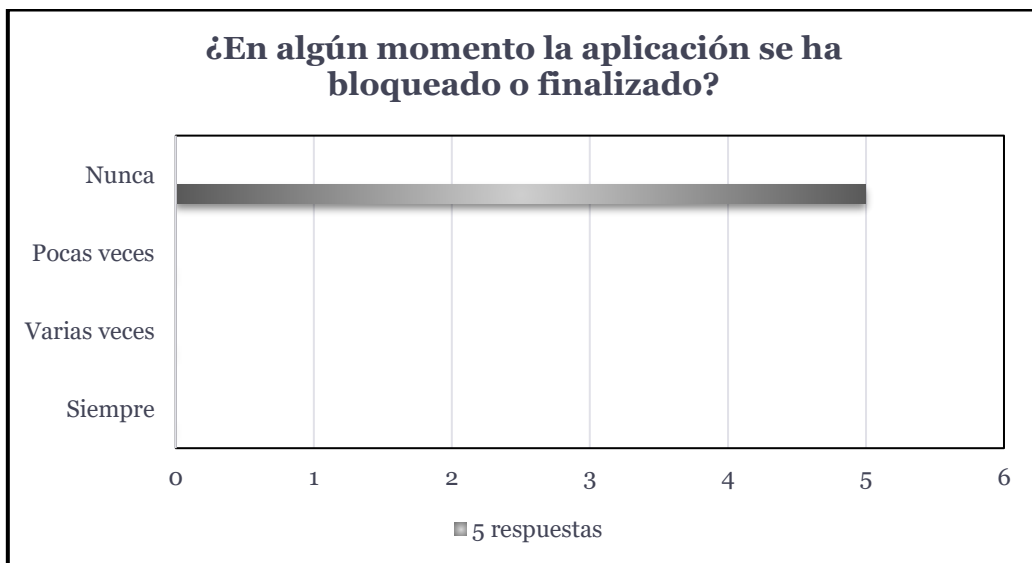


Figura 29. Quinta pregunta en las pruebas de usuario

La respuesta del usuario ante esta cuestión es bastante importante ya que fallos inesperados pueden provocar problemas y descontentos con el usuario. En este caso ningún usuario ha experimentado ningún tipo de error de este tipo lo que convierte a la aplicación en un sistema sólido y estable.

## 7.2 Pruebas de carga

Otro tipo de pruebas realizadas son las pruebas de carga, están relacionadas con el tiempo que tarda la aplicación en ejecutar un proceso en concreto dependiendo de la carga de procesamiento.

Para poder realizar esta prueba, se deben de medir los tiempos en distintos sistemas operativos con distintas características para así comprobar el rango de diferencia entre ellos.

Tabla 27. Características de los dispositivos móviles

	Dispositivo 1	Dispositivo 2	Dispositivo 3
Nombre	Huaewi p10	Samsung Galaxy s9	Samsung Note 10
OS	Android 7.1 Nougat	Android 8.1 Oreo	Android 9.0 Pie
Procesador	Kirin 970 2.36Ghz	Exynos 9820 2.73Ghz	Snapdragon 855 2,84Ghz
Almacenamiento	4GB RAM 64GB M.I	6GB RAM 128GB M.I	8GB RAM 256GB M.I

Los procesos que se han tenido en cuenta para realizar esta prueba son los siguientes.

- Arranque de la aplicación. En este caso podemos hacer dos lecturas ya que la primera vez que se inicie la aplicación, el tiempo será mayor que en un posterior inicio debido a la creación de la base de datos.
- Lectura por NFC.
- Lectura por código QR.

Los resultados se pueden observar en la siguiente tabla.

Tabla 28. Resultados pruebas de carga

	Arranque inicial	Arranque normal	Lectura NFC	Lectura QR
Dispositivo 1	2,95s	2,18s	0,90s	0,80s
Dispositivo 2	2,72s	1,86s	0,60s	0,70s
Dispositivo 3	2,49s	1,51s	0,40s	0,50s

Observando la tabla, comprobamos que nunca se ha sobrepasado el tiempo límite establecido en los requisitos no funcionales 9 y 11.

## 8. Conclusión

---

Durante la realización del Proyecto, han surgido diversas complicaciones que tras exhaustivos análisis e investigaciones he conseguido paliar. Sin duda alguna, la mayor incógnita a la que me he enfrentado es la de implementar acciones con la tecnología NFC cuando nunca he hecho uso de ella a nivel de desarrollo.

Tras investigar sobre esta tecnología y ponerla en práctica, he aprendido mucho sobre ella y me he dado cuenta de las infinitas utilidades que puede tener. Pienso que su uso en este proyecto es ideal a la hora de identificar a un usuario ya que se trata de un proceso rápido y sencillo, basta con acercar cualquier elemento con una etiqueta RFID a un lector NFC.

Por ejemplo, cuando un alumno registra su asistencia, solo bastará con acercar su carné UPV al dispositivo del profesor que tiene instalada la aplicación. Esa simple acción podría sustituir al tradicional gesto de firmar en una hoja de papel que va circulando por todos los alumnos. Cualquier idea o desarrollo que sirva para mejorar una tarea, significa avance y ese es el principal objetivo de la sociedad, avanzar para garantizar unas condiciones de vida más cómodas. Y que mejor forma de avanzar que con la tecnología ya que hoy en día está al alcance de todos y ofrece todo tipo de posibilidades.

Otra tarea a tener en cuenta es cumplir con los objetivos que debe de tener una aplicación. No basta con crear una aplicación determinada, debe de cumplir una serie de estándares para garantizar su usabilidad, su funcionamiento y la aceptación del usuario. Para ello, he realizado una búsqueda de los modelos correctos que debería de seguir la aplicación, para implementarla según sus normas.

Gracias a ello las pruebas realizadas a usuarios han tenido resultados satisfactorios, se ha conseguido desarrollar una aplicación capaz de cumplir con sus funciones de una forma eficiente, con una interfaz óptima y visualmente atractiva. Con un sistema sólido y tolerante a fallos que permita dar soporte al usuario en cualquier momento.

Por otro lado, ha sido necesaria la investigación de los distintos elementos que intervienen en el proceso de creación de un parte de firmas. Asignaturas, alumnos, profesores y grupos, cada uno dispone de sus propios atributos que deben de ser lo más fieles a la realidad. No todas las clases son iguales, es posible que se dé la circunstancia de que la clase en la que se va a realizar el parte de asistencias esté fuera del calendario docente que se ha establecido. Debido a que se trata de una clase de recuperación y su horario y lugar dependen del profesor, de este modo el proceso adquiere una metodología diferente que se debe de tener en cuenta.

Al tratarse de una aplicación con un uso exclusivamente para la Universidad Politécnica de Valencia, la competencia con otras aplicaciones es muy reducida, aunque existan otras alternativas para registrar la asistencia, que hagan uso del carné UPV para ello, no se da el caso. Además, hay que tener cuenta que estas alternativas conllevan un gasto monetario por utilizar esos servicios y el objetivo de este proyecto es cubrir la creación de los partes de firmas sin dicho coste.



Otro punto a tener en cuenta es la implicación con el medio ambiente reduciendo así el uso de papel, ya que generar el parte de firmas con esta metodología implica no hacer uso de la típica hoja de asistencias en formato impreso que cada día es utilizada para ello.

Este proyecto me ha servido para poner en práctica y afianzar todos los conocimientos que he ido adquiriendo a lo largo de mi formación académica y de esto modo, ampliarlos con nuevos aspectos que anteriormente desconocía. Con él, ha mejorado mi capacidad de organización, gestión del tiempo y respuesta ante posibles problemas u obstáculos presentes a la hora de cualquier implementación. Claramente un escenario que me ayudará de cara a mi vida laboral.

## 8.1 Desarrollo futuro

Este proyecto forma parte de la primera versión de la aplicación, lo cual quiere decir que puede estar sujeta a modificaciones futuras que doten a la aplicación de más y mejores funcionalidades.

En primer lugar, al tratarse de un proyecto desarrollado externamente con la Universidad Politécnica de Valencia hay ciertos datos a los que no hemos tenido acceso, como son los datos de los profesores y alumnos. Prescindir de estos datos nos lleva a implementar un método para las altas de profesores y alumnos de forma manual, mediante el ingreso de uno en uno. Tarea que en un futuro podría verse automatizada si la universidad dispusiera de un archivo con todos los datos necesarios y listos para cargarlos en la base de datos. Podría darse incluso el caso de que la aplicación accediese directamente a la API de la universidad donde se recogiese toda la información necesaria.

Por otro lado, existe la posibilidad de implementar en la aplicación un apartado de firmas y observaciones para que el profesor lo rellene desde su dispositivo y se añadiese al parte de firmas. Esta función no ha podido verse desarrollada debido a que, por proximidad a la fecha de entrega carecemos de más tiempo.

Otra opción de mejora sería la compatibilidad con otros sistemas operativos distintos a Android, el uso de la aplicación podría verse ampliado a un mayor número de dispositivos.

Existe también la posibilidad de incluir como futuro objetivo, el lanzamiento de la aplicación a una plataforma de descarga dentro de la propia Universidad para que todos los profesores pudiesen hacer uso de ella.

Además, sería interesante desarrollar un almacenamiento de los partes de firmas y sus datos en un servidor web para que, desde cualquier otro dispositivo, el usuario pueda hacer uno de ellos.

Finalmente, para concluir las posibles opciones de mejoras futuras, sería la mejora del diseño de la interfaz siguiendo estándares de diseño empleados por la Universidad Politécnica de Valencia.

## 8.2 Relación del Proyecto con los estudios cursados

Durante la elaboración de este proyecto, me he encontrado con situaciones en cierto modo familiares, debido a que he tenido que lidiar con problemas o llevar a cabo implementaciones similares a las realizadas durante mi formación académica.

A la hora de implementar la base de datos, ha sido de gran ayuda las nociones adquiridas gracias a asignaturas como Tecnología de bases de datos o Bases de datos y sistemas de información.

Para la realización de los casos de uso, el diagrama de casos de uso y modelo de dominio, he tenido cierta comodidad ya que anteriormente he desarrollado estos escenarios en asignaturas como Ingeniería del software.

A la hora de escoger el entorno de desarrollo, claramente me he decantado por Android Studio porque permite la implementación en lenguaje Java, un lenguaje que llevo desarrollando durante toda mi formación.

Otro añadido a la hora de escoger Android Studio es debido a que anteriormente durante mi formación realicé un curso de desarrollo de aplicaciones en Android impartido por el CFP, Centro de Formación Permanente de la Universidad Politécnica de Valencia.







## 9. Bibliografía

---

- [1] «Herramientas alternativas» 28 Agosto 2019. [En línea]. Disponible:  
<https://www.educaciontrespuntocero.com/recursos/herramientas-control-asistencia/>.  
[Último acceso: 26 Abril 2020]
- [2] «Junta de andalucía» 2012. [En línea]. Disponible:  
<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/407>.  
[Último acceso: 27 Abril 2020].
- [3] «Requerimientos funcionales y no funcionales» 20 abril 2018. [En línea]. Disponible:  
<https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a>.  
[Último acceso: 28 Abril 2020].
- [4] «Mockup» 2014. [En línea]. Disponible:  
[https://es.ryte.com/wiki/Mockup#Usos\\_y\\_beneficios\\_de\\_los\\_mockups](https://es.ryte.com/wiki/Mockup#Usos_y_beneficios_de_los_mockups).  
[Último acceso: 29 Abril 2020].
- [5] Iván Garcerant, «Modelo de Dominio» 10 Julio 2008. [En línea]. Disponible:  
<https://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.  
[Último acceso: 30 Abril 2020].
- [6] «Kernel de Linux» 25 Agosto 2018. [En línea]. Disponible:  
<https://maslinux.es/entendiendo-el-kernel-de-linux-2/>.  
[Último acceso: 1 Mayo 2020].
- [7] «Android Studio» 18 Febrero 2020. [En línea]. Disponible:  
<https://developer.android.com/studio/features?hl=es-419>.  
[Último acceso: 2 Mayo 2020].
- [8] «SQLite» 2012. [En línea]. Disponible:  
<https://www.ecured.cu/SQLite#Caracter.C3.ADsticas>.  
[Último acceso: 3 Mayo 2020].
- [9] «RFID» 2015. [En línea]. Disponible:  
<https://www.actum.es/preguntas-frecuentes/%C2%BFqu%C3%A9-significa-rfid>.  
[Último acceso: 4 Mayo 2020].



- [10] «Balsamiq» 2015. [En línea]. Disponible:  
<https://balsamiq.com/learn/articles/what-are-wireframes/>.  
[Último acceso: 5 Mayo 2020]
- [11] Emersson Mendoza, «BoUML» 24 Octubre 2017. [En línea]. Disponible:  
<https://es.slideshare.net/emerssonmendozagarcia/bouml-caracteristicas>.  
[Último acceso: 6 Mayo 2020].
- [12] «ZXing» 4 Marzo 2013. [En línea]. Disponible:  
<https://biljetapp.wordpress.com/2013/03/04/presentacion/>.  
[Último acceso: 7 Mayo 2020].
- [13] Rubén Gómez López, «Java Mail» 1 Diciembre 2008. [En línea]. Disponible:  
<https://www.adictosaltrabajo.com/2008/12/01/javamail/>.  
[Último acceso: 8 Mayo 2020].
- [14] Julio Yáñez Novo, «iText» 24 Febrero 2016. [En línea]. Disponible:  
<http://codigoxules.org/java-itext-pdf-creando-pdf-java-itext/>.  
[Último acceso: 9 Mayo 2020].
- [15] Urbano Villanueva Rodríguez, «JSoup» 14 Mayo 2019. [En línea]. Disponible:  
<https://www.adictosaltrabajo.com/2019/05/14/implementando-un-crawler-sencillo-con-jsoup/>.  
[Último acceso: 10 Mayo 2020].
- [16] Manuel Jesús Morales, «Lógica funcional» 2017. [En línea]. Disponible:  
<https://www.modusoperantic.com/es/logica-de-negocio-vs-logica-funcional/>.  
[Último acceso: 11 Mayo 2020].
- [17] «Diseñador de Layout» 27 Diciembre 2019. [En línea]. Disponible:  
<https://developer.android.com/guide/topics/ui/declaring-layout?hl=es-419>.  
[Último acceso: 12 Mayo 2020].

# Lista de acrónimos

---

UPV: Universidad Politécnica de Valencia

TUI: Tarjeta Universitaria Inteligente

RNF: Requisito No Funcional

RF: Requisito Funcional

M.I: Memoria interna

XML: del inglés, Extensible Markup Language

PDF: del inglés, Portable Document Format

NFC: del inglés, Near Field Communication

RFID: del inglés, Radio Frequency Identification

API: del inglés, Application Programming Interface

OS: del inglés, Operative System

RAM: del inglés, Random Access Memory

BIDI: diminutive del inglés, Bidimensional

QR: del inglés, Quick Response

HTML: del inglés, Hyper Text Markup Language

DB: del inglés, Data Base

UML: del inglés Unified Modeling Language