



**UNIVERSIDAD DE JAÉN**  
*Escuela Politécnica Superior de Jaén*

Trabajo Fin de Grado

# **APLICACIÓN MÓVIL CON TECNOLOGÍA NFC PARA EL INTERCAMBIO DE LA INFORMACIÓN**

**Alumno: Antonio Jesús Cruz Rodríguez**

Tutor: Dra. D<sup>a</sup>. Macarena Espinilla Estévez  
Dr. D. Luis Martínez López

Dpto: Informática

**Junio, 2016**





Universidad de Jaén  
Escuela Politecnica Superior de Jaén  
Departamento de informatica

*Dra. D<sup>a</sup>. Macarena Espinilla Estévez*  
y *Dr. D<sup>o</sup>. Luis Martínez López*, tutores  
del Trabajo Fin de Grado titulado:

**APLICACIÓN MÓVIL CON TECNOLOGÍA NFC PARA EL INTERCAMBIO DE  
LA INFORMACIÓN,**

que presenta *D. Antonio Jesús Cruz Rodríguez*, autorizan  
su presentación para defensa y evaluación en la  
Escuela Politécnica Superior de Jaén.

Jaén, junio de 2016.

El alumno:

Antonio Jesús Cruz Rodríguez

Los tutores:

Macarena Espinilla Estévez

Luis Martínez López



## *Agradecimientos*

*Quisiera expresar mi más sincero agradecimiento a todas aquellas personas que de alguna u otra forma han contribuido a que sea capaz de realizar este trabajo fin de grado.*

*En primer lugar, a mis padres, hermanos y abuelos, sin los cuales este camino hubiera sido más complicado.*

*En segundo lugar, a mi novia Nazaret, por el apoyo y el ánimo que cada día me aporta para que sea capaz de conseguir todo lo que me proponga.*

*En tercer lugar, a mis tutores, Luis Martínez López y en especial a Macarena Espinilla Estévez, por el tiempo, dedicación y la confianza que ha depositado en mi.*

*En cuarto lugar, a mi compañero Daniel Zafra, por la ayuda que me ha prestado cuando la he necesitado.*

*Y por último, a todos los profesores que he tenido suerte de poder conocer en estos años en la universidad, porque su colaboración y enseñanza me han alimentado a crecer como profesional.*

*Muchas gracias a todos.*



# ÍNDICE DE CONTENIDO

<b>ÍNDICE DE CONTENIDO</b> .....	7
<b>ÍNDICE DE FIGURAS</b> .....	10
<b>ÍNDICE DE TABLAS</b> .....	12
<b>CAPÍTULO 1</b> .....	13
1. Introducción .....	13
1.1. Motivación .....	13
1.2. Propuesta .....	14
1.3. Objetivos .....	14
1.4. Planificación temporal .....	15
1.4.1. Estimación de tiempos .....	15
1.4.2. Diagrama de Gantt .....	16
1.5. Presupuesto .....	18
1.6. Estructura de la memoria .....	19
<b>CAPÍTULO 2</b> .....	21
2. Sistemas basados en tecnología NFC .....	21
2.1. NFC .....	21
2.1.1. Introducción .....	21
2.1.2. Historia .....	25
2.1.3. Tecnologías similares a NFC .....	26
2.1.4. Motivación para usar tecnología NFC .....	29
2.1.5. El formato NDEF .....	31
2.1.6. Ventajas en el uso de tecnología NFC .....	33
2.1.7. Desventajas en el uso de tecnología NFC .....	34
2.2. Aplicaciones .....	35
2.2.1. Ejemplos de sistemas basados en tecnología NFC .....	35
2.2.2. Ejemplos orientados a manuales y guías .....	36
<b>CAPÍTULO 3</b> .....	37
3. Android y servicios web .....	37
3.1. Introducción .....	37
3.2. Android .....	38
3.2.1. Evolución de Android .....	38
3.2.2. Arquitectura .....	40
3.2.3. Componentes .....	44
3.2.4. Ciclo de vida de una actividad .....	45
3.2.5. Estructura de un proyecto Android .....	47

3.3.	Servicios web .....	49
3.3.1.	Introducción.....	49
3.3.2.	Arquitectura de los servicios web .....	51
3.3.3.	Ventajas de los servicios web.....	52
3.3.4.	Desventajas de los servicios web .....	53
3.3.5.	Tipos de servicios web.....	53
3.3.6.	Servicios REST .....	56
3.3.6.1.	Introducción a los servicios REST.....	56
3.3.6.2.	Estructura de las URLs.....	59
<b>CAPÍTULO 4</b>	.....	<b>61</b>
4.	Proceso de ingeniería del software.....	61
4.1.	Fases de la ingeniería del software .....	61
4.2.	Especificación de requerimientos.....	61
4.2.1.	Requerimientos de la aplicación web .....	62
4.2.1.1.	Funcionales.....	62
4.2.1.2.	No funcionales .....	63
4.2.2.	Requerimientos de la aplicación móvil .....	64
4.2.2.1.	Funcionales.....	64
4.2.2.2.	No funcionales .....	65
4.3.	Análisis del sistema .....	66
4.3.1.	Casos de uso. ....	66
4.3.1.1.	Casos de uso del sistema web.....	67
4.3.1.2.	Casos de uso de la aplicación móvil .....	74
4.3.2.	Escenarios.....	78
4.3.2.1.	Escenarios del sistema web .....	79
4.3.2.2.	Escenarios de la aplicación móvil .....	83
4.4.	Diseño del sistema.....	84
4.4.1.	Diseño de clases.....	84
4.4.2.	Diseño de datos .....	88
4.4.3.	Diseño de la interfaz .....	92
4.4.3.1.	Metáforas.....	92
4.4.3.2.	Estilo .....	94
4.4.3.3.	Mensajes.....	97
4.4.3.4.	Storyboard de la aplicación web .....	98
4.4.3.5.	Storyboard de la aplicación móvil.....	101
4.5.	Implementación.....	104
4.5.1.	Arquitectura del sistema .....	104

4.5.2. Lenguajes de programación utilizados .....	105
4.5.3. Herramientas de desarrollo.....	106
4.5.4. Servicios REST implementados .....	111
4.6. Pruebas.....	112
4.6.1. Resultados .....	117
<b>CAPÍTULO 5</b> .....	119
5. Conclusiones y líneas de trabajo futuras .....	119
<b>REFERENCIAS</b> .....	121
<b>Anexos</b> .....	124
A. Contenido del CD-ROM.....	124
B. Manual de instalación.....	124
B.1. Sistema web .....	125
B.2. Aplicación móvil.....	128
C. Manual de usuario del sistema web .....	129
C.1. Acceso.....	129
C.2. Registro del usuario .....	129
C.3. Recuperación de clave de acceso.....	130
C.4. Contacto .....	131
C.5. Gestionar vídeos-tutoriales .....	131
C.5.1. Añadir un nuevo vídeo-tutorial .....	132
C.5.2. Editar un vídeo-tutorial .....	133
C.5.3. Eliminar un vídeo-tutorial.....	134
C.6. Modificación de datos personales.....	135
D. Manual de usuario de la aplicación móvil.....	136
D.1. Enlazar vídeo-tutorial con etiqueta NFC.....	137
D.2. Visualizar vídeo-tutorial asociado a etiqueta NFC.....	142
D.3. Borrar contenido de tarjeta NFC .....	143

## ÍNDICE DE FIGURAS

Figura 1.1.-Estructura de trabajo .....	15
Figura 1.2.-Diagrama de Gantt .....	17
Figura 2.1.- Ejemplo NFC en modo activo.....	22
Figura 2.2.- Ejemplo NFC en modo pasivo.....	22
Figura 2.3.- Esquema de modos de trabajo NFC.....	23
Figura 2.4.- Ejemplo de tarjetas NFC .....	24
Figura 2.5.- Esquema utilización NFC .....	25
Figura 2.6.- Ejemplo de código de barras.....	26
Figura 2.7.- Ejemplo de código de barras lineal .....	27
Figura 2.8.- Ejemplo de código de barras en dos dimensiones .....	27
Figura 2.9.- Ejemplo de etiqueta QR .....	28
Figura 2.10.- Ejemplo de errores en etiqueta QR.....	28
Figura 2.11.- Ejemplo de etiqueta RFID .....	29
Figura 2.12.- Esquema de los campos de un registro NDEF .....	32
Figura 2.13.- Ejemplo de mensaje NDEF .....	33
Figura 3.1.- Evolución de Android.....	39
Figura 3.2.- Comparativa de uso de sistemas operativos en Smartphone .....	40
Figura 3.3.- Esquema de capas del sistema Android.....	40
Figura 3.4.- Esquema de la arquitectura Android.....	41
Figura 3.5.- Ciclo de vida de una actividad de Android .....	46
Figura 3.6.- Esquema de la estructura de un proyecto en Android Studio .....	48
Figura 3.7.- Ejemplo de funcionamiento de los servicios web .....	51
Figura 3.8.- Esquema de la arquitectura de los servicios web.....	52
Figura 3.9.- Ejemplo de servicio SOAP .....	55
Figura 3.10.- Ejemplo de servicio REST.....	55
Figura 3.11.- Métodos de los servicios REST .....	58
Figura 4.1.- Ejemplo de representación de los elementos de un caso de uso .....	67
Figura 4.2.- Diagrama frontera del sistema web .....	68
Figura 4.3.-Caso de uso: Identificación del usuario .....	69
Figura 4.4.- Caso de uso: Registro del usuario. ....	70
Figura 4.5.-Caso de uso: Recuperar contraseña. ....	71
Figura 4.6.- Caso de uso: Gestión de Vídeos.....	72
Figura 4.7.- Caso de uso: Modificación de datos personales (seguridad). ....	73
Figura 4.8.- Caso de uso: Generar clave privada.....	74
Figura 4.9.- Diagrama frontera de la aplicación móvil.....	75
Figura 4.10.- Caso de uso: Enlazar vídeo a tarjeta NFC. ....	76
Figura 4.11.- Caso de uso: Borrar contenido etiqueta NFC .....	77
Figura 4.12.- Caso de uso: Reproducir vídeo asociado a una etiqueta NFC. ....	78
Figura 4.13.- Ejemplo de clase en un diagrama UML. ....	85
Figura 4.14.- Representación de los tipos de relaciones en un diagrama UML. ....	85
Figura 4.15.- Esquema del diseño modelo-vista-controlador.....	85
Figura 4.16.- Diagrama UML controlador: Login .....	86
Figura 4.17.- Diagrama UML controlador: Admin .....	87
Figura 4.18.- Diagrama UML controlador: API .....	87
Figura 4.19.- Diagrama UML de la aplicación móvil.....	88
Figura 4.20.- Elementos de un diagrama entidad-relación .....	89
Figura 4.21.- Esquema conceptual de la base de datos. ....	90
Figura 4.22.- Metáfora icono de logo del sistema web.....	93

Figura 4.23.- Metáfora icono de mi cuenta .....	93
Figura 4.24.- Metáfora de icono de ayuda.....	93
Figura 4.25.- Metáfora de icono de información.....	94
Figura 4.26.- Metáforas de iconos para estado del proceso de escritura en NFC.	94
Figura 4.27.- Paleta de colores.....	95
Figura 4.28.- Icono de Bootstrap. ....	96
Figura 4.29.- Iconos de Google Desing. ....	96
Figura 4.30.- Botones de Bootstrap. ....	96
Figura 4.31.- Ejemplo de mensajes en Bootstrap. ....	97
Figura 4.32.- Storyboard pantalla principal del sistema web.....	99
Figura 4.33.- Storyboard Gestión de Vídeos .....	100
Figura 4.34.- Storyboard Modificación de datos personales .....	101
Figura 4.35.- Storyboard enlazar vídeo con tarjeta NFC.....	102
Figura 4.36.- Storyboard Visualizar vídeo. ....	103
Figura 4.37.- Storyboard borrar contenido tarjeta NFC.....	103
Figura 4.38.- Arquitectura del sistema .....	104
Figura 4.39.- CodeIgniter .....	107
Figura 4.40.- Herramienta NetBeans .....	108
Figura 4.41.- Herramienta phpMyAdmin.....	109
Figura 4.42.- Herramienta Android Studio .....	110
Figura 4.43.- Herramienta SmartGit.....	111
Figura Manual 1.- Panel de administración de phpMyAdmin .....	125
Figura Manual 2.- Panel de administración de nuestra base de datos. ....	126
Figura Manual 3.- Panel de administración (importación) .....	127
Figura Manual 4.- Formulario de acceso .....	129
Figura Manual 5.- Formulario de registro .....	130
Figura Manual 6.- Formulario de recuperación de contraseña .....	130
Figura Manual 7.- Formulario de contacto.....	131
Figura Manual 8.- Panel de administración de nuestro sistema web.....	132
Figura Manual 9.- Formulario de subir un nuevo vídeo.....	133
Figura Manual 10.- Formulario para editar los datos de un vídeo.....	134
Figura Manual 11.- Mensaje de confirmación de borrado.....	135
Figura Manual 12.- Formulario para editar los datos personales del usuario .....	136
Figura Manual 13.- Pantalla principal de nuestra aplicación Android .....	137
Figura Manual 14.- Buscar vídeos públicos.....	138
Figura Manual 15.- Pantalla de enlace con tarjeta NFC.....	139
Figura Manual 16.- Panel de administración de nuestro sistema web.....	140
Figura Manual 17.- Pantalla de estado de escritura en etiqueta NFC .....	141
Figura Manual 18.- Demostración de escritura en NFC .....	141
Figura Manual 19.- Estados en el proceso de escritura .....	142
Figura Manual 20.- Pantalla de visualización de vídeo .....	143
Figura Manual 21. Demostración de visualización de un vídeo-tutorial.....	143
Figura Manual 22.- Pantalla de estado de borrado .....	144

## ÍNDICE DE TABLAS

Tabla 1.1.- Estimación de tiempos.....	16
Tabla 1.2.- Estimación de presupuesto. ....	19
Tabla 3.1.- Códigos de estado HTTP. ....	57
Tabla 4.1.- Servicios REST Implementados.....	112
Tabla 4.2.- Pruebas. Caso: Login del usuario (correcto).....	113
Tabla 4.3.-Pruebas. Caso: Login del usuario (incorrecto).....	113
Tabla 4.4.-Pruebas. Caso: Registro del usuario (correcto).....	113
Tabla 4.5.-Pruebas. Caso: Registro del usuario (incorrecto).....	113
Tabla 4.6.- Pruebas. Caso: Gestión de vídeos: Editar (correcto).....	113
Tabla 4.7.- Pruebas. Caso: Gestión de vídeos: Editar (incorrecto).....	114
Tabla 4.8.- Pruebas. Caso: Gestión de vídeos: Borrar (correcto).....	114
Tabla 4.9.- Pruebas. Caso: Gestión de vídeos: Borrar (incorrecto).....	114
Tabla 4.10.- Pruebas. Caso: Gestión de vídeos: Añadir (correcto).....	114
Tabla 4.11.- Pruebas. Caso: Gestión de vídeos: Añadir (incorrecto).....	114
Tabla 4.12.- Pruebas. Gestión de vídeos: Añadir sin URL del vídeo (incorrecto)	115
Tabla 4.13.- Pruebas. Caso: Modificación de datos personales (correcto).....	115
Tabla 4.14.- Pruebas. Caso: Modificación de datos personales (incorrecto).....	115
Tabla 4.15.- Pruebas. Caso: Contacto (correcto).....	115
Tabla 4.16.- Pruebas. Caso: Contacto (incorrecto).....	115
Tabla 4.17.-Pruebas. Caso: Cambiar privacidad (correcto).....	115
Tabla 4.18.- Pruebas. Caso: Cambiar privacidad (incorrecto).....	116
Tabla 4.19.- Pruebas. Caso: Visualizar vídeo de etiqueta NFC (correcto).....	116
Tabla 4.20.- Pruebas. Caso: Visualizar vídeo de etiqueta NFC (incorrecto).....	116
Tabla 4.21.- Pruebas. Caso: Borrar contenido etiqueta NFC (correcto).....	116
Tabla 4.22.- Pruebas. Caso: Borrar contenido etiqueta NFC (incorrecto).....	116
Tabla 4.23.- Pruebas. Caso: Enlazar vídeo con etiqueta NFC (correcto).....	117
Tabla 4.24.- Pruebas. Caso: Enlazar vídeo con etiqueta NFC (incorrecto).....	117
Tabla 4.25.- Pruebas. Caso: vídeo usando clave privada (correcto).....	117
Tabla 4.26.- Pruebas. Caso: vídeo usando clave privada (incorrecto).....	117
Tabla 4.27.- Resultados obtenidos en el sistema web.....	118
Tabla 4.28.- Resultados obtenidos en la aplicación Android.....	118

# Capítulo 1

---

## 1. Introducción

### 1.1. Motivación

Hoy en día existen millones de teléfonos inteligentes (Smartphone) con grandes capacidades de procesamiento, conexión a internet y una multitud de aplicaciones que han conseguido que estos dispositivos sean imprescindibles en nuestras vidas. Gracias a las capacidades computacionales de estos dispositivos junto con la multitud de tecnologías de las que disponen, permiten la creación de nuevas funcionalidades para los usuarios. Estas funcionalidades suponen una oportunidad de negocio para las empresas o desarrolladores.

Por otro lado, existe una tecnología inalámbrica denominada NFC que permite la comunicación a corta distancia, apenas 4 o 5 centímetros. A pesar de que esta tecnología no es reciente ya que lleva existiendo desde el 2003, son en los últimos años cuando se están proliferando numerosas aplicaciones que utilizan NFC que evidencia su gran utilidad práctica. Así, en la actualidad cada vez son más las aplicaciones y dispositivos que hacen uso de esta tecnología para realizar tareas como pueden ser: realizar pagos a través del dispositivo móvil o identificar personas.

Además, la tecnología NFC permite un grado de seguridad alto ya que es improbable que en primer lugar se produzca la comunicación sin intencionalidad y en segundo lugar que existan intrusos que intercepten los datos transmitidos en el proceso de comunicación. Es por ello por lo que hoy en día, se utiliza esta tecnología para realizar tareas de comunicación que transportan datos muy sensibles como pueden ser: datos bancarios, datos identificativos...etc.

Este trabajo fin de grado nace de la premisa que la tecnología NFC puede ser utilizada para ayudar a las personas mayores o con algún tipo de limitación en las tareas cotidianas en el hogar.

En muchas ocasiones nos encontramos con personas con ciertas discapacidades que no les permite recordar o aprender a realizar determinadas tareas cotidianas con dispositivos electrónicos o eléctricos y que necesitan

apoyo de otra persona para realizar dichas acciones. Dichas tareas se pueden explicar realizando un vídeo-tutorial y gracias a un sistema de gestión que permita su almacenamiento y su posterior búsqueda por parte una aplicación móvil que haga uso de la tecnología NFC, podemos conseguir que todas estas personas puedan realizar sus tareas con el simple gesto de pasar su dispositivo móvil o tablet por una tarjeta NFC. En definitiva, podemos conseguir que todas estas personas con cierto grado de dependencia sean un poco más independientes y se sientan mejor al saber que son capaces de realizar su vida sin la necesidad de contar con el apoyo de otras personas.

## 1.2. Propuesta

El propósito de este trabajo fin de grado es desarrollar un sistema web que permita gestionar vídeos explicativos junto con un prototipo de aplicación Android que permita encontrar el vídeo que se adapte a nuestra necesidad y haciendo uso de la tecnología NFC poder asociarlo a una tarjeta NFC para su posterior reproducción automática.

## 1.3. Objetivos

Los objetivos derivados de nuestra propuesta de trabajo fin de grado son los siguientes:

1. Estudio de las tecnologías necesarias para el desarrollo de este proyecto.
2. Desarrollar una aplicación web que permita gestión de los vídeos-tutoriales de un usuario y poder dar la posibilidad de que el usuario pueda registrar su vídeo.
3. Desarrollar un prototipo de aplicación Android que permita buscar vídeos explicativos, poder asociarlos a una tarjeta NFC y finalmente ser capaz reproducir el vídeo-tutorial asociado a una tarjeta NFC.
4. Desarrollar de manera correcta la comunicación entre la aplicación Android y la aplicación web mediante servicios REST.
5. Realizar los manuales necesarios para el sistema web y la aplicación móvil. Grabar y editar algunos vídeo-tutoriales explicativos.

6. Crear una memoria que describa el trabajo desarrollado para la realización de este trabajo fin de grado.

## 1.4. Planificación temporal

Para entender la planificación temporal de este proyecto, se muestra la estructura de trabajo propuesta en la figura 1.1 junto con la estimación inicial de tiempos de cada tarea que aparece reflejada en la tabla 1.1 y, finalmente, se adjunta el diagrama de Gantt.

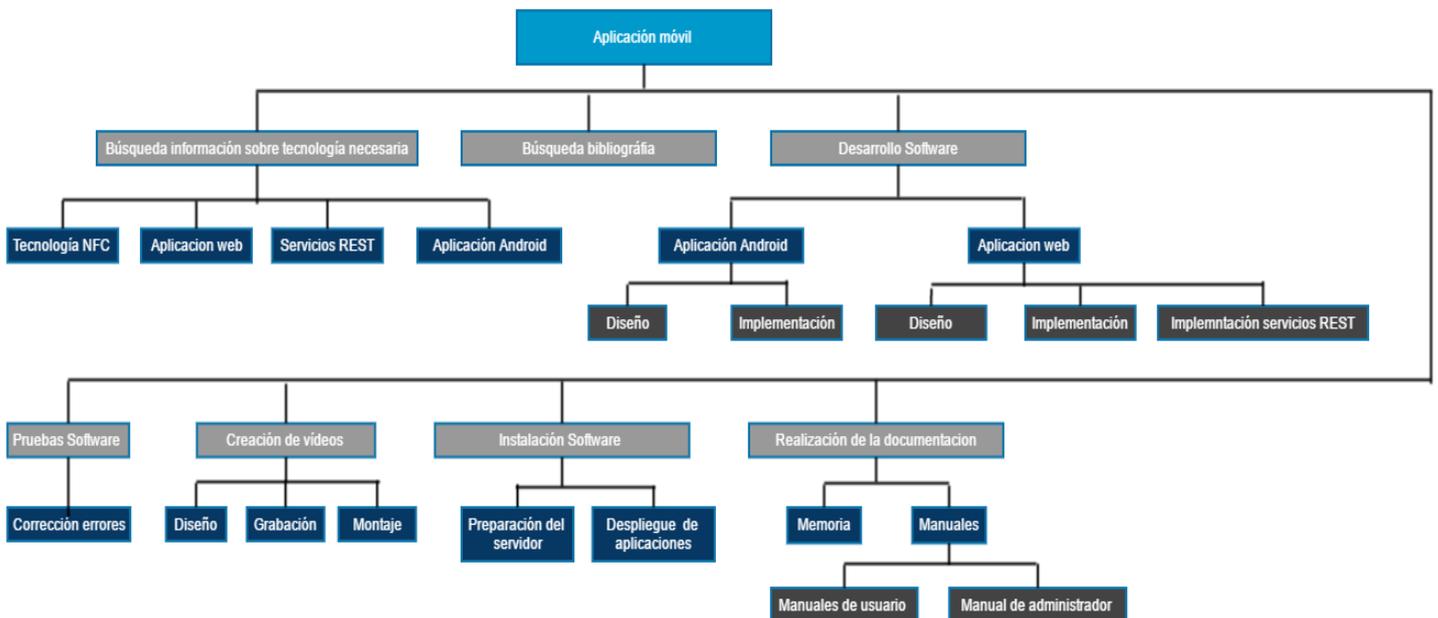


Figura 1.1.-Estructura de trabajo

### 1.4.1. Estimación de tiempos

La estimación de tiempos es una estimación optimista de la duración inicial para cada una de las tareas a desempeñar para la realización de este proyecto. En la tabla 1.1 se detalla cada tarea acompañada de su correspondiente duración. Nuestra estimación nos indica que necesitaremos un total de 128 días para realizar la propuesta que se plantea en este trabajo fin de grado.

Tarea	Estimación Tiempo (días)
<b>Búsqueda de información sobre tecnología necesaria</b>	
Tecnología NFC	5
Aplicación web	4
Servicios REST	3
Aplicación Android	4
<b>Búsqueda de Bibliografía</b>	5
<b>Desarrollo Software</b>	
Aplicación Android	
Diseño	10
Implementación	15
Aplicación web	
Diseño	10
Implementación	25
Implementación servicios REST	3
<b>Pruebas Software</b>	
Pruebas y corrección errores	5
<b>Creación de vídeos</b>	
Diseño	1
Grabación	2
Montaje	3
<b>Instalación software</b>	
Preparación del servidor	2
Despliegue de aplicaciones	2
<b>Realización de la documentación</b>	
Memoria	25
Manuales	
Manuales de usuario	2
Manuales de administrador	2
<b>TOTAL</b>	<b>128</b>

Tabla 1.1.- Estimación de tiempos.

### 1.4.2. Diagrama de Gantt

El diagrama de Gantt es una herramienta que se emplea para planificar y programar tareas a lo largo de un periodo determinado de tiempo. Además, permite realizar un seguimiento y control del progreso de cada una de las etapas de un proyecto.



## 1.5. Presupuesto

El presupuesto de un proyecto es una de las herramientas fundamentales de las que dispone un profesional que se dedique al Desarrollo software. En muchos casos constituye nuestra primera toma de contacto y nuestra carta de presentación de cara al cliente. El presupuesto es la valorización económica de un software, ya sea de sistema o de aplicación. Intenta determinar cuánto dinero, esfuerzo, recursos y tiempo supondrá constituir un sistema o producto específico de software. A continuación, se detalla el presupuesto para este proyecto. En él se tienen en cuenta el coste de los materiales informáticos, dispositivos grabación y software necesarios. Aunque no se incluyen el resto de software utilizado para la realización del proyecto, ya que este software será gratuito, con lo que no supondrá ningún gasto extra. Por último, se incluye el coste de la mano de obra empleada para el desarrollo del proyecto.

<i>Descripción</i>	<i>Precio unidad (IVA incluido)</i>	<i>unidades</i>	<i>Coste Total (euros)</i>
<b>Costes Software</b>			
<i>Alojamiento Web anual</i>	50	1	50
<i>Tarjetas NFC</i>	0.90	10	9
<b>Costes de la mano de obra</b>	<b>Sueldo mensual</b>	<b>Numero de meses</b>	
<i>Análisis</i>	1539.39	1	1539.39
<i>Diseño</i>	1539.39	1	1539.39
<i>Implementación</i>	1103.04	2	2206.08
<i>Memoria</i>	1103.04	1	1103.04
<i>Coste total de la mano de obra sin IVA</i>	6387.9		
<i>Coste total sin beneficio</i>	6446.9		
<i>Beneficio 5%</i>	3223.45		

<i>Coste Total</i>	<b>9670.35</b> 1
--------------------	---------------------

Tabla 1.2.- Estimación de presupuesto.

Como podemos observar, la estimación del presupuesto necesario para realizar este trabajo fin de grado nos da como resultado el valor de 9670,35 euros.

## 1.6. Estructura de la memoria

A continuación, se describe la estructura de la memoria perteneciente a este trabajo fin de grado. La memoria consta de 7 capítulos:

- **Capítulo 1:** En este capítulo, se ha explicado la motivación que nos mueve a desarrollar este proyecto, los objetivos a cumplir, la propuesta concreta a la que tenemos que perseguir y finalmente, la planificación temporal de este trabajo.
- **Capítulo 2:** En este capítulo, se va describir en qué consiste la tecnología NFC y cuáles son sus ventajas e inconvenientes frente a otras tecnologías. Además, se describirán las principales motivaciones que esta tecnología brinda a nuestro dominio de aplicación centrado en las personas mayores y con algún tipo de limitación. Finalmente, se describirán algunos ejemplos de sistemas basados en NFC.
- **Capítulo 3:** Este capítulo está dedicado al estudio de la tecnología Android, haciendo un estudio de su evolución, su arquitectura y la estructura de un proyecto Android. Además, estudiaremos los servicios web, en concreto los servicios REST que son unas piezas fundamentales de este trabajo.
- **Capítulo 4:** Este capítulo es fundamental para esta memoria, al ser el capítulo de mayor extensión y estar dedicado al proceso completo de desarrollo de este proyecto. En primer lugar, se hará un repaso de

---

Tabla 1.2.- "Curiosidades Varias".<http://curiosidadesvarias.portalfree.net/categorias-profesionales-y-sueldos-de-los-trabajadores-de-informatica-como-programadores-y-analistas/>

diferentes etapas de la Ingeniería Software, seguido de la especificación de requisitos funcionales y no funcionales de cada una de las aplicaciones de las que consta este proyecto. En segundo lugar, se realizará el diseño de las aplicaciones y por último, describiremos el proceso implementación necesario para desarrollar cada una de las aplicaciones y las pruebas realizadas.

- **Capítulo 5:** Este es el último capítulo de esta memoria, en donde se expondrán las conclusiones que nos ha aportado la realización de este trabajo fin de grado y las posibles líneas de trabajo para continuar con el desarrollo de nuestra propuesta.
- **Referencias:** En este apartado se indica la bibliografía utilizada a lo largo de la presente memoria.
- **Anexos:** En el Anexo A, se visualizará el contenido del CD-ROM adjunto a esta memoria, explicando con detalle el contenido de cada una de las carpetas y archivos presentes en él. Además, este CD-ROM incluirá una copia de esta memoria. En los Anexos B, C y D se describen los manuales para la instalación, administración y utilización del software desarrollado en este trabajo fin de grado, respectivamente.

## Capítulo 2

---

### 2. Sistemas basados en tecnología NFC

#### 2.1. NFC

##### *2.1.1. Introducción*

NFC significa Near Field Communication. Se trata de una tecnología inalámbrica que funciona en la banda de los 13.56 MHz (en esa banda no hace falta de licencia) y es capaz de transmitir a velocidades de 106Kbit/s, 212 Kbits/s o 424 Kbits/s [2]. Además, es compatible con otras tecnologías e incluso se puede utilizar para configurar e iniciar otras conexiones Wireless como por ejemplo Wifi y fue aprobada como estándar ISO en 2003 (ISO 18092). Esta tecnología deriva de las etiquetas RFID (identificación por radio frecuencia), que están presentes en abonos de transporte o incluso en sistemas de seguridad de tiendas físicas. La principal característica de esta tecnología es que utiliza un radio de comunicación de alta frecuencia y muy corto alcance, no más de 5 cm.

Aunque en un principio la característica de que la tecnología NFC solo permite la comunicación a una distancia corta puede parecer un inconveniente frente a tecnologías similares como pueden ser Bluetooth o RFID, hay que destacar que es esta característica lo que diferencia a esta tecnología de otras y lo que permite que sea de un gran valor. La condición de que solo se puede realizar el intercambio de información entre dos dispositivos que se encuentren a poca distancia permite dotar a esa comunicación de seguridad ya que se considera prácticamente imposible interceptar la señal sin que las personas que intervienen el proceso de comunicación se den cuenta de ello. Además, esta característica identificativa de esta tecnología permite que sea adecuada para el intercambio de información de carácter personal o sensible, como por ejemplo realizar pagos por móvil, identificación de personas...etc.

En el protocolo NFC siempre hay un dispositivo que inicia la conversación y a la vez es el encargado de monitorizarla, este rol se puede intercambiar en cualquier momento entre los dispositivos implicados en la comunicación [2]. Existen dos modos de funcionamiento y todos los dispositivos estándar NFCIP-1 deben ser capaces de soportarlos:

- **Modo activo:** Requiere que los dispositivos que intervienen en la comunicación creen su propio campo magnético para poder realizar la comunicación. Para este tipo de funcionamiento es necesario que ambos dispositivos cuenten con energía para funcionar. Este modo es característico de las comunidades Peer to Peer (P2P) entre dispositivos NFC.



Figura 2.1.- Ejemplo NFC en modo activo

- **Modo pasivo:** En este caso solo uno de los dispositivos que intervienen en la comunicación debe crear el campo magnético y el otro se aprovecha de la modulación de la carga para poder transmitir sus datos. El inicializador de la comunicación es el encargado de crear el campo magnético.



Figura 2.2.- Ejemplo NFC en modo pasivo

Dentro de esos modos de funcionamiento, se pueden encontrar tres modos de comunicación:

- **Modo Lectura/Escritura:** Las aplicaciones tienen permiso para transferir datos en el formato definido. Este modo no está considerado seguro y es soportado por la API de comunicación “Contactless”.
- **Modo emulación NFC Card:** Este modo permite al dispositivo a actuar como un estándar Smartcard. En este modo, la transferencia de datos es segura, y también es soportado por el API de comunicación “Contactless”.
- **Modo Peer to Peer:** Modo de comunicación directa entre dispositivos, a nivel de enlace. No está soportado por el API de comunicación “Contactless”.

Modo de trabajo	¿Activo o pasivo?	Lectura/escritura	Formato
NFC Card emulation (o Target NFC)	Pasivo, se comporta como un <u>tag</u> RFID	No, sólo puede ser leído	Típicamente es un tag, aunque también hay dispositivos activos que se pueden comportar como tal (emulación tarjeta) si así lo deseamos
NFC Read/Write (o iniciador NFC)	Activo, tiene capacidad de leer/grabar un tag	Sí, puede hacer ambas cosas	En general, cualquier dispositivo NFC es read/write
NFC peer-to-peer	Activo	Comunicación bidireccional con otro dispositivo	No todos los dispositivos NFC son peer-to-peer. Es una funcionalidad todavía poco extendida, sobre todo por la existencia del Bluetooth

Figura 2.3.- Esquema de modos de trabajo NFC

Por otro lado, toda comunicación NFC consta de 5 fases:

- Descubrimiento.
- Autenticación
- Negociación (en la comunicación)
- Transferencia
- Reconocimiento

Además, tag NFC es un transponder contactless compatible con una de las 4 tipologías definidas por el NFC Forum, o bien con el estándar ISO/IEC

---

Figura 2.3. “FQ ingeniería electrónica”.

<http://www.fqingenieria.com/es/conocimiento/fundamentos-de-la-tecnologia-nfc-45>

18902 [3]. Un tag NFC es un elemento pasivo, en el sentido de que no tiene la capacidad de iniciar la comunicación, pero utiliza la energía de la señal enviada por un dispositivo para responder.

Las 4 tipologías de tag NFC según NFC Forum son:

- **NFC Forum Tag type 1** (compatible con tags Innovision Topaz/Jewel) Standard ISO/IEC 14443-3 Tipo A (sin anticolisión).
- **NFC Forum Tag type 2** (compatible con tags NXP MIFARE Ultralight®) Standard ISO/IEC 14443-3 Tipo A.
- **NFC Forum Tag type 3** (compatible con tags Sony FeliCa) Standard JISX6319-4.
- **NFC Forum Tag type 4** (compatible con SmartCards que soportan comandos APDU) Standard ISO/IEC 14443-4 e ISO/IEC7816-4.



Figura 2.4.- Ejemplo de tarjetas NFC

Por último, en la actualidad, esta tecnología es utilizada en una gran variedad de situaciones, pero la tecnología NFC suele asociarse a pagos por móvil. Esto es debido al gran empeño de grandes compañías por hacerse con ese mercado, ya que supone una gran revolución en el método de pago. Así pues, la tecnología NFC está presente en multitud de tareas, para entender un poco mejor los campos en los que se encuentra implantada se puede observar en la figura un esquema de ello.



Figura 2.5.- Esquema utilización NFC

### 2.1.2. Historia

La tecnología NFC es una de las tecnologías más recientes de nuestra época, por lo tanto, su historia es bastante resumida, pero resulta interesante conocer su evolución a lo largo de los pocos años que lleva existiendo. La historia a grandes rasgos de esta tecnología es:

- **1983:** Charles Walton registra la primera patente asociada con RFID
- **2003:** Se lleva a cabo el primer estándar 18092 entorno a NFC, como caso particular de la tecnología RFID
- **2004:** Nokia, Philips y Sony crean el NFC Forum, organismo encargado de su estandarización, implantación y certificación
- **2006:** Primeras especificaciones para tags NFC
- **2006:** Primer móvil con tecnología NFC (Nokia 6131)
- **2009:** El NFC Forum publica el primer estándar peer-to-peer para poder transferir contactos, URLs, etc.
- **2010:** Sale el primer Android con NFC (Samsung Nexus S)

- **2010:** Google demuestra cómo usar NFC en Android en el Google IO
- **2010:** Symbian da soporte a NFC (Anna)
- **2010:** Android da soporte a NFC (API 9). Sólo tags.
- **2011:** Android soporta NFC entre dispositivos (Beamming)
- **2011:** RIM es la primera compañía en ser certificada por MasterCard para el pago por móvil

### *2.1.3. Tecnologías similares a NFC*

Existen diversas tecnologías que se asemejan a la tecnología NFC y que podrían haberse usado para la realización de este proyecto.

En primer lugar, hablaremos de los códigos de barras, son una técnica de entrada de datos, con imágenes formadas por combinaciones de barras y espacios paralelos, de anchos variables. Representan números que a su vez pueden ser leídos y descifrados por lectores ópticos o scanner [4].



Figura 2.6.- Ejemplo de código de barras

El código de barras permite identificar un elemento de forma única, a través de una asociación con una base de datos. Esta tecnología es muy usada en la producción y distribución de artículos. Además, existen dos tipos de código de barras:

- Los lineales (1-D) como los que se usan en productos y permiten incluir mensajes cortos.



Figura 2.7.- Ejemplo de código de barras lineal

- Los de dos dimensiones (2-D) que han empezado a usarse en documentos para controlar su envío o en seguros médicos y, en general, en documentos que requieren la inserción de mensajes más grandes (de hasta 2 725 dígitos) como un expediente clínico completo.



Figura 2.8.- Ejemplo de código de barras en dos dimensiones

Para utilizar esta tecnología, el usuario solo tendrá que disponer de un lector de código de barras o de un móvil con cámara y una aplicación que le permita leer este tipo de códigos. Esta tecnología no permite leer un código de barras deteriorado. Esto supone una gran desventaja con respecto a las etiquetas NFC.

En segundo lugar, hablaremos de la tecnología QR que proviene del inglés Quick Response ya que el objetivo principal de esta tecnología es proporcionar un acceso rápido a la información [5]. Es una tecnología muy similar a la tecnología de códigos de barras, pero tiene algunas diferencias. Esta tecnología cuenta con mucho éxito, ya que es un estándar abierto y su uso asociado a teléfonos móviles, que pueden funcionar como lectores.



Figura 2.9.- Ejemplo de etiqueta QR

Un código QR contiene información tanto en sentido horizontal como vertical. Además, la información que pueden contener es muy variada: desde un simple texto, hasta tarjetas de visitas electrónicas. Los códigos QR tienen la capacidad de corregir errores en la lectura de manera que en el supuesto de que se manche o se deteriore parcialmente el código, este se podrá continuar leyendo sin ningún problema. Tienen, en concreto, cuatro niveles de corrección de errores.

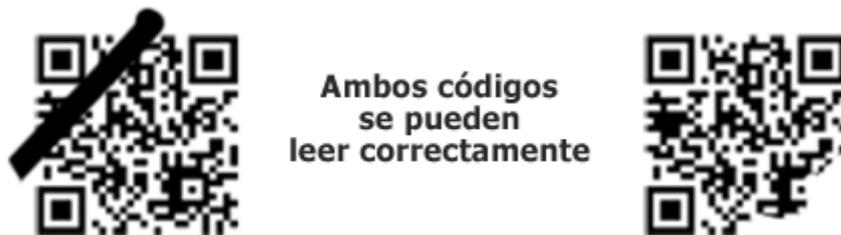


Figura 2.10.- Ejemplo de errores en etiqueta QR

Los códigos pueden ser leídos desde cualquier ángulo. Eso es debido a los patrones de alineamiento (los tres cuadrados situados en las esquinas superiores e inferior izquierda) sirven para detectar la posición del código. Para poder leer un código QR el usuario solo necesitará un móvil con cámara y de una aplicación que sea capaz de decodificar este código. A diferencia de la tecnología NFC, esta tecnología implica un proceso menos intuitivo a la hora de leer etiquetas ya que se debe poner el dispositivo móvil preparado para la realización de la lectura del código, en lugar del simple gesto de pasar el móvil por una etiqueta NFC.

En tercer lugar, otra tecnología es RFID (Radio Frequency Identification), tal y como hemos indicado anteriormente esta tecnología es la fundamentación de la tecnología NFC [6]. Los sistemas de identificación por radio frecuencia están destinados a la identificación de objetos a distancia sin necesidad de estar en contacto, ni si quiera en contacto visual. Para ello se requiere etiquetas o tags RFID que contienen un microchip y una antena de tamaño reducido para identificar al elemento portador de estas.

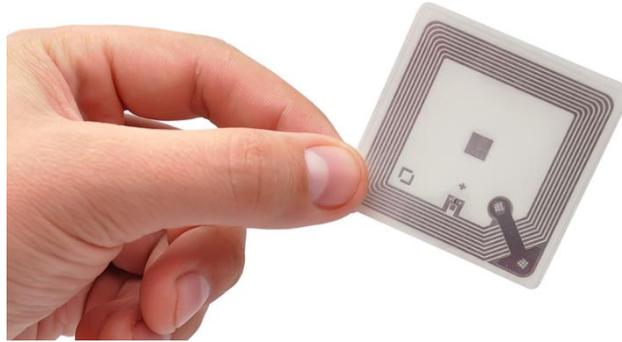


Figura 2.11.- Ejemplo de etiqueta RFID

La información almacenada en etiquetas RFID puede ir desde 1 bit hasta varios kilobytes, siendo su capacidad máxima 4KB. Para poder usar esta tecnología es necesario un lector capaz de obtener los datos almacenados en estas etiquetas. Lo normal es tener un dispositivo que tenga una o varias antenas que emitan continuamente ondas de radio y que reciban las señales devueltas por las etiquetas RFID. Esta tecnología tendría un problema a la hora de utilizarla en este proyecto y es que la tecnología RFID puede generar errores de colisión cuando en un espacio pequeño se encuentra múltiples etiquetas.

#### *2.1.4. Motivación para usar tecnología NFC*

La tecnología NFC es relativamente nueva, pero sus características le permiten que sea una buena opción para resolver el problema que se plantea en este proyecto. Además, es una tecnología muy barata ya que para utilizarla solo se necesitan dos elementos fundamentales como son: un dispositivo móvil o tablet que soporte NFC. Sin embargo, este hecho no es una limitación, ya que cada vez son más los dispositivos que soportan este tipo de tecnología. Adicionalmente, es necesario un conjunto de tarjetas NFC para grabar la información necesaria, siendo el coste de este tipo de tarjetas muy económico.

Por las razones expuestas se evidencia que la tecnología NFC es la más adecuada para realizar la tarea que hay que desempeñar en este proyecto. Además, tal y como lo hemos indicado anteriormente, hay que tener en cuenta que el objetivo principal de este trabajo fin de grado es ayudar a personas mayores o dependientes, por lo que se requiere de una tecnología que permita realizar esta tarea de una forma fácil e intuitiva. Para ello la tecnología NFC gana al resto de tecnologías que se podrían usar para la realización de este objetivo. A continuación, se indican algunos de los argumentos por los que se ha rechazado el uso de otras tecnologías:

- Los códigos de barras podrían ser utilizados para realización de este proyecto, pero esta tecnología implica que el usuario deba mantenerse quieto durante unos instantes para que la lectura del código se realice correctamente, ya que esta tecnología no soporta la corrección de errores. Esto puede suponer un gran problema para personas que debido a una enfermedad son incapaces de mantenerse durante unos segundos en la misma posición (enfermedad de Parkinson).
- La tecnología QR podría ser apta para la realización de este trabajo fin de grado. Sin embargo, su utilización no es tan sencilla e intuitiva como la tecnología NFC, ya que implique la necesidad de poner la cámara del dispositivo móvil o tablet en estado de lectura y esperar hasta que el lector consiga leer el contenido de la etiqueta QR. Esto puede suponer un problema para personas con enfermedades como el Parkinson que serían incapaces de poder leer el contenido de una etiqueta QR de modo correcto.

Además, las tarjetas NFC nos aportan una gran ventaja sobre las tarjetas QR y es que nos permiten poder imprimir una imagen o logo que sea intuitivo para las personas y las etiquetas QR se podrían realizar esto, pero supondría tener unas etiquetas muy grandes y que las personas se equivocaran e intentaran leer la imagen en lugar del código QR.

### *2.1.5. El formato NDEF*

El formato NDEF o NFC Data Exchange Format es un formato registrado por el NFC Forum para poder compartir datos entre los dispositivos NFC o entre dispositivos NFC y etiquetas. NDEF es conceptualmente muy similar al estándar MIME. NDEF define un formato de encapsulación de mensaje para el intercambio de datos entre dispositivos NFC.

NDEF se caracteriza por poseer una cabecera de datos, denominada cabecera NDEF a partir de la cual se encuentran los bloques de información. Cada bloque de información se agrupa en registros que a su vez contienen los datos agrupados en mensajes NDEF y caracterizados por un tipo MIME.

NDEF no proporciona soporte para el manejo de errores. Además, el formato NDEF tiene una desventaja al permitir que la totalidad de los bloques de información son accesibles por todos los dispositivos que soporten la tecnología NFC ya que la clave empleada para acceder a estos datos es la definida por defecto (en hexadecimal FF FF FF FF FF FF). Esto permite que datos escritos en una etiqueta RFID por un dispositivo NFC puedan ser borrados o sobrescritos con otra información en cualquier momento y por cualquier dispositivo. Si esta desventaja supusiera un inconveniente, podría ser solucionado combinando las ventajas de los formatos NDEF y MIFARE en el que se pone de manifiesto la existencia de claves de acceso [7].

A continuación, veamos una breve descripción de los campos que forman un registro NDEF.

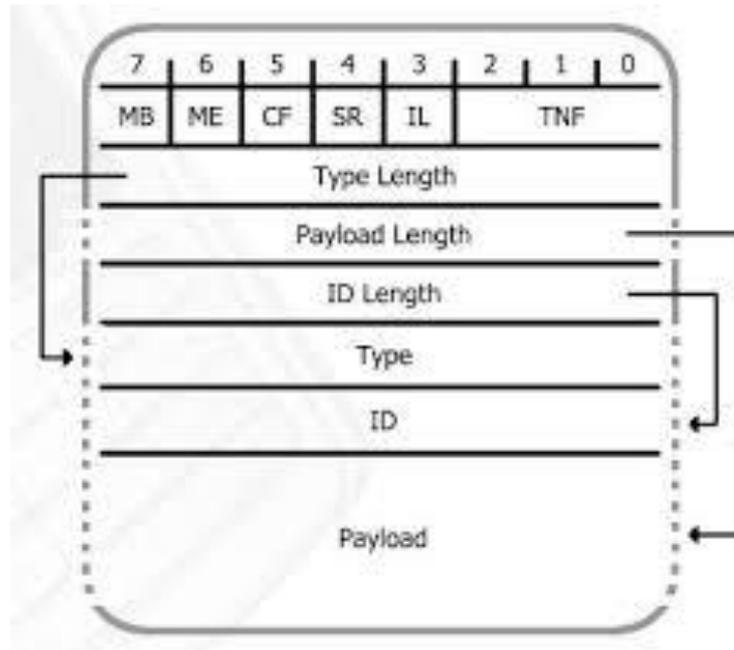


Figura 2.12.- Esquema de los campos de un registro NDEF

- **MB:** Es un campo de 1bit que indica el inicio de un mensaje NDEF.
- **ME:** Es un campo de 1bit que indica el final de un mensaje NDEF.
- **CF:** Es un campo de 1bit que indica si es el primer trozo del registro o un trozo de la mitad del registro de un payload fragmentado.
- **TNF:** Es un campo de 3 bits que indica la estructura del valor de campo TYPE.
- **TYPE\_LENGTH:** Este campo es un entero de 8 bits que especifica la longitud del campo type.
- **ID\_LENGTH:** Solo está presente si el campo IL está activo.
- **PAYLOAD\_LENGTH:** Este campo especifica la longitud del campo payload.
- **TYPE:** Es un identificador que especifica el tipo de payload de la información transmitida.
- **ID:** Es un identificador en forma de referencia URI. Dicha referencia puede ser relativa o absoluta.
- **PAYLOAD:** Este campo es el que lleva la información útil para las aplicaciones de usuario NDEF.

Un mensaje NDEF puede contener varios registros, esto se puede ver en la siguiente imagen:

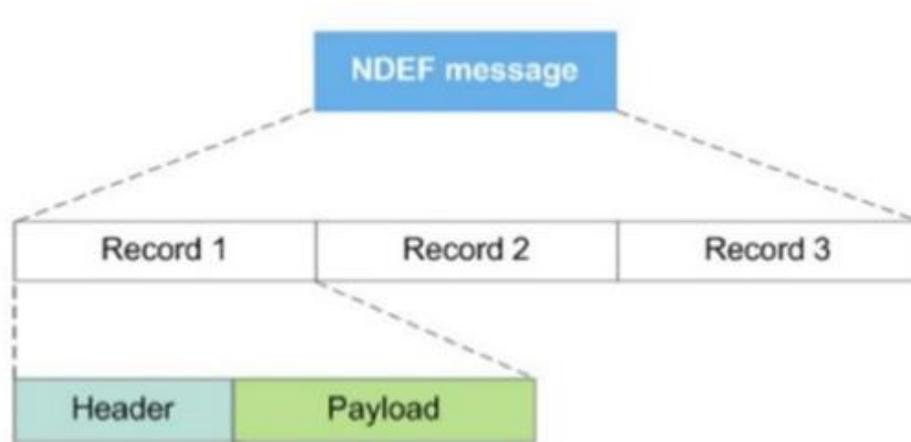


Figura 2.13.- Ejemplo de mensaje NDEF

Para finalizar, cabe destacar que este formato es soportado por la totalidad de dispositivos NFC y constituye un estándar para el intercambio y almacenamiento de información.

### *2.1.6. Ventajas en el uso de tecnología NFC*

A continuación, vamos a exponer las ventajas que conlleva el uso de la tecnología NFC. La primera ventaja que hace que esta tecnología sea de gran interés es el tiempo de conexión necesario para que dos dispositivos que quieran comunicarse, ya que es extremadamente rápido, solo se necesita 0.1 segundos para establecer la conexión y, además, este proceso se realiza de forma automática.

Otras de las ventajas más importantes que presenta esta tecnología es la seguridad de la que dispone dos dispositivos que realicen la comunicación. Las comunicaciones que hacen uso de la tecnología NFC establecen un canal seguro y utilizan cifrado al enviar la información sensible, como por ejemplo información personal.

A consecuencia de la ventaja anterior, surgen numerosas de ventajas para el usuario. La tecnología NFC permite realizar tareas de identificación o pagos por móvil con solo realizar la acción de pasar nuestro móvil por el detector de NFC. Esta es una de las ventajas más importantes de esta tecnología, ya que

permite que numerosas tareas como pueden ser la identificación de personas o la realización de pagos se lleven a cabo de una forma rápida y sencilla. Además, el pago por NFC ha permitido que los usuarios obtengan grandes ventajas y seguridad a la hora de realizar sus compras o pagos, ya que los usuarios pueden disponer de un monedero en sus dispositivos móviles y la pérdida de un móvil se suele detectar antes que la pérdida de una tarjeta de crédito.

Por último, la tecnología NFC es un sistema bastante robusto, que cuenta con estándares internacionales y, además, un lector NFC puede realizar la lectura de varias etiquetas NFC de forma simultánea.

### *2.1.7. Desventajas en el uso de tecnología NFC*

Entre las posibles desventajas de la tecnología NFC podemos encontrar que su velocidad es más lenta que la velocidad que soporta la tecnología Bluetooth. Además, esta tecnología tiene una capacidad de almacenamiento limitada (64 – 924 bytes). Esta desventaja no supondría un problema para nuestra propuesta, ya que, solo se necesitaría almacenar el identificador del vídeo tutorial, por lo que no se requiere de mucha capacidad de almacenamiento.

Por otro lado, esta tecnología tiene un alcance de conexión menor que otras tecnologías como puede ser la tecnología Bluetooth. Además, solo permite la conexión entre dos dispositivos, no existe la posibilidad de crear una red inalámbrica entre varios dispositivos. Esta desventaja puede suponer una gran ventaja para nuestro proyecto, ya que tarjetas NFC se encontrarán en un rango pequeño y no habrá posibilidad de interferencia.

Por último, en la actualidad aún sigue siendo una tecnología recientemente nueva y el número de dispositivos que soportan esta tecnología es pequeño, aunque hay que destacar que cada vez son más los dispositivos que permite el uso de esta tecnología y gracias al empeño de grandes empresas por utilizar NFC para realizar tareas como son los pagos por móvil, poco a poco el número de personas que conocen la tecnología NFC se va incrementando.

## 2.2. Aplicaciones

### 2.2.1. Ejemplos de sistemas basados en tecnología NFC

Uno de los grandes ejemplos de uso NFC son las aplicaciones de pago móvil que están siendo unos de los grandes avances de nuestra época. Estas aplicaciones suponen una revolución en la forma de pago y nos ofrecen una serie de ventajas con respecto al pago tradicional. El empeño de grandes empresas propone que algún día nosotros utilicemos nuestro teléfono móvil como una billetera. El pago móvil por NFC permite proteger nuestros datos financieros y evitar así los robos de datos y demás delitos que ocurren con las tarjetas de crédito. Una de las mayores preocupaciones sobre el pago móvil es si en verdad se cumple esta seguridad, pero la compleja estructura del sistema y las características de esta tecnología hacen que sea un método muy seguro.

En la actualidad existen multitud de aplicaciones que nos permiten realizar esta forma de pago y poco a poco se va convirtiendo en una forma habitual de realizar nuestros pagos y a la vez, gracias a este esfuerzo de las grandes empresas por usar esta tecnología para esta tarea tan habitual en nuestras vidas, la tecnología NFC se va convirtiendo poco a poco en una tecnología menos desconocida.

Otro ejemplo de uso de esta tecnología es la identificación de personas en eventos, este tipo de uso aún sigue siendo menos utilizado que el anterior, pero se espera que poco a poco se vaya implantando. Esto nos permitirá mejorar los procesos de registro y control de acceso a todo tipo de eventos, ya que una persona no tendrá que ir mostrando su acreditación todo el rato. Además, este tipo de sistemas esperan poder implantarse para poder controlar el acceso a hospitales, eventos deportivos y, por supuesto, el acceso a oficinas de trabajo.

Para finalizar, otro ejemplo de uso de esta tecnología que podemos encontrar es la automatización de perfiles. Este uso está muy extendido y poder crear una tarea en Trigger especificando que el activador será una tarjeta NFC. Al escanear esta etiqueta nos permitirá realizar tareas como, por ejemplo:

- **Al llegar a casa:** Desactivar el GPS y activamos el Wifi. Además, podemos configurar una etiqueta para que contenga la contraseña de nuestra red Wifi por lo que podría ser de gran utilidad para las visitas.
- **Para dormir:** Activar el sonido de la alarma y silenciar las notificaciones. También, podremos configurarlo para que nos salga una notificación si no está cargando, para que no se nos olvide cargarlo por la noche o para que baje el brillo de la pantalla y no nos deslumbre si en mitad de la noche necesitamos mirar algo rápido.
- **Al entrar al coche:** Activar el GPS y abrir nuestro navegador favorito. Además, podríamos configurar el Bluetooth para que se active al detectar la etiqueta y así tener el manos libres sincronizado de forma rápida.

Como conclusión, podemos afirmar que son muchos y cada vez más los ejemplos de uso de esta tecnología. Además, se prevé que el uso la tecnología NFC nos aportará grandes beneficios y que se podrá aplicar a una gran variedad de tareas.

### *2.2.2. Ejemplos orientados a manuales y guías*

Existen multitud de ejemplos o de tutoriales de cómo desarrollar una aplicación que utilice NFC. A la hora de desarrollar una aplicación que utilice este tipo de tecnología solo se necesita comprender como realizar el proceso de lectura y el proceso de escritura. Además, debemos aprender a como permitir que nuestra aplicación obtenga los privilegios necesarios para controlar el adaptador NFC de nuestro dispositivo móvil o tablet. Por lo generar implementar una aplicación que haga uso de esta tecnología es relativamente sencillo.

Entre todos los ejemplos disponibles, podemos encontrar la guía nos proporciona Google Android Developers. Esta guía nos explica de un modo detallado como usar la tecnología NFC, explicándonos cada uno de los métodos que debemos utilizar para realizar las tareas necesarias para comunicarnos. Además, nos proporciona algunos ejemplos sencillos de aplicaciones que utilizan es tipo de tecnología.

## Capítulo 3

---

### 3. Android y servicios web

#### 3.1. Introducción

En este capítulo se estudiará el sistema operativo Android, incluyendo una descripción de las características más importantes que hemos de conocer. Además, haremos una introducción a los servicios web, donde indicaremos sus principales ventajas y desventajas de utilizar estos servicios y las distintas tecnologías que utilizan, para centrarnos finalmente en los servicios REST ya que son la base de nuestro sistema.

Como ya hemos mencionado anteriormente, NFC es una tecnología recientemente nueva y que no disfruta de una gran implantación social. Por lo que es muy importante que los sistemas operativos móviles nos proporcionen el soporte necesario para explotar esta tecnología y permitir que los desarrolladores hagan uso de ella.

En el caso del sistema operativo Android, desde su versión 2.3 (Gingerbread API 9) en diciembre de 2010, proporciona funcionalidades NFC. En esta Api solo se permitía la funcionalidad de lectura y no fue hasta la versión 2.3.3 de febrero de 2011 (API 10) donde se introdujo soporte completo para NFC. La API 10 no disfruto de un gran recorrido ya que Google anunció en ese mismo año el lanzamiento de la nueva versión de este sistema operativo (Android 4, Ice Cream Sandwich). Esto supuso que los fabricantes de dispositivos móviles no se centraran en las nuevas funcionalidades que suponía la versión 2.3.3 y que apostaran por el enfoque que Google le dio a Android al fusionar el sistema operativo de Smartphone y tablet en la versión 4. En esta versión, ya no solo se podría usar NFC, sino que existía una API dedicada a la transferencia de información entre terminales vía NFC. Además, este sistema funcionaba en modo Push, lo que el receptor del mensaje no tendría que inicializar la aplicación para recibir el mensaje. Se produjo así el uso de Intent, algo que facilita el uso de esta tecnología.

## 3.2. Android

### 3.2.1. Evolución de Android

Mucho ha cambiado este sistema operativo desde que en 2007 se anunciara el lanzamiento de la versión beta del que sería el sistema operativo más extendido del mundo, con más de un 80% de cuota del mercado móvil [8]. Veamos con un poco más de detalle sus versiones más famosas:

- **Android 1.0 Apple Pie:** Fue la primera versión de este sistema operativo, se presentó como un sistema móvil totalmente gratuito y abierto a diferencia del sistema iOS. A pesar de que esta primera versión no preocupó demasiado a la competencia, introdujo algunos de los conceptos que, a día de hoy, son un estándar de los sistemas operativos móviles como el Android Market o Gmail. El primer dispositivo móvil que llevo este sistema fue el HTC Dream-Google Phone.
- **Android 1.5 Cupcake:** Esta versión llegó un año más tarde que la anterior, supuso una serie de mejoras que para los usuarios eran casi impredecibles. Algunas de las novedades más interesantes llegaron para mejorar la usabilidad de sus características, como son el teclado táctil QWERTY con predicción de texto y el Bluetooth.
- **Android 2:** La segunda generación de Android presentó mejoras importantes en cuenta a la velocidad, diseño y calidad de imagen, con el zoom digital o el flash de cámara. Además, se desarrolló la popular Auto-rotación de la pantalla, pero sin duda la gran aportación fue la tecnología NFC, que permitió realizar pagos por teléfono.
- **Android 4.0 Ice Cream Sandwich:** En esta versión Android quiso montar un sistema funcional para Smartphones y tablets. Esta nueva generación incluía entre otras muchas cosas una barra de notificaciones mejorada y el navegador Google Chrome con capacidad de navegar con 15 pestañas simultáneamente. Otro de los grandes logros de esta versión fue la grabación de vídeo en full HD.
- **Android 4.4 KitKat:** Esta versión permitió mejorar el rendimiento de aplicaciones como Whatsapp y potenciar la funcionalidad multitarea.

Además, corrigió gran parte de los fallos de seguridad de versiones anteriores.

- **Android 5.0 Lollipop:** Esta versión de octubre de 2014, supuso la llegada de quinta generación de este sistema móvil. Con la versión 5 llegaba Material Design; un nuevo lenguaje de diseño que permitirá unificar la experiencia de uso en cualquier dispositivo.
- **Android 6.0 Marshmallow:** La principal novedad de la sexta generación de este sistema móvil es Android Pay, la plataforma de pago de Google. Además, implanta soporte nativo para lector de huellas dactilares y desarrolla la tecnología para USB Type-C. Su sistema “Doze” minimiza el uso de la batería.



Figura 3.1.- Evolución de Android

En la actualidad, el sistema operativo Android domina por completo el mercado de los Smartphones siendo el sistema operativo que está más presente en los teléfonos inteligentes.

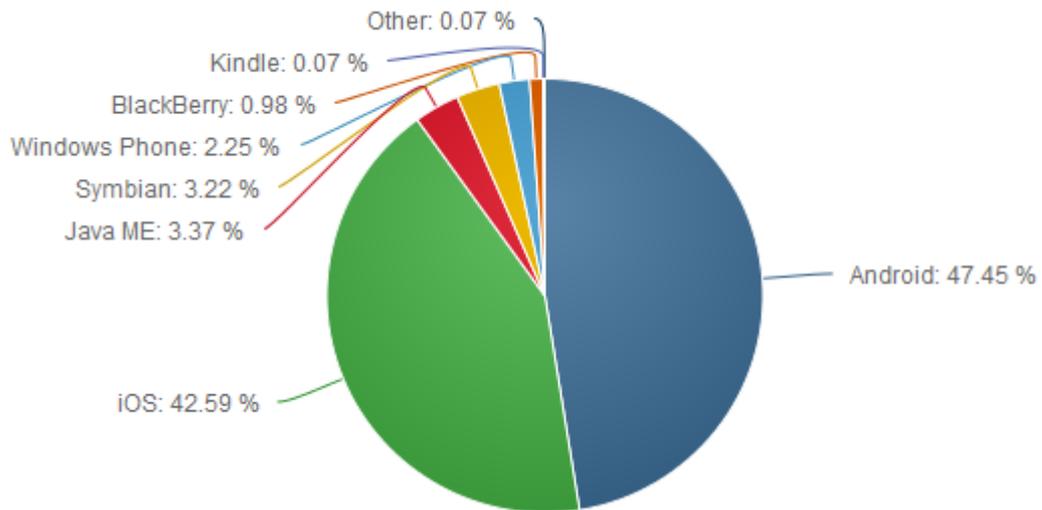


Figura 3.2.- Comparativa de uso de sistemas operativos en Smartphone

### 3.2.2. Arquitectura

Para poder desarrollar aplicaciones en Android es importante conocer cómo está estructurado este sistema operativo [9]. A esto le llamamos arquitectura y en el caso de Android está formada por cinco capas que facilitan al desarrollador la creación de aplicaciones. Además, esta distribución permite acceder a las capas más bajas mediante el uso de librerías para que así el desarrollador no tenga que programar a bajo nivel las funcionalidades necesarias para que una aplicación haga uso de los componentes de hardware de los teléfonos.



Figura 3.3.- Esquema de capas del sistema Android

Cada una de las capas utiliza para realizar sus funciones los elementos de la capa inferior, es por ello que a este tipo de arquitectura se le conoce como pila. Para entender mejor esta arquitectura, a continuación, en la figura 3.4 se muestra el diagrama de la arquitectura Android.

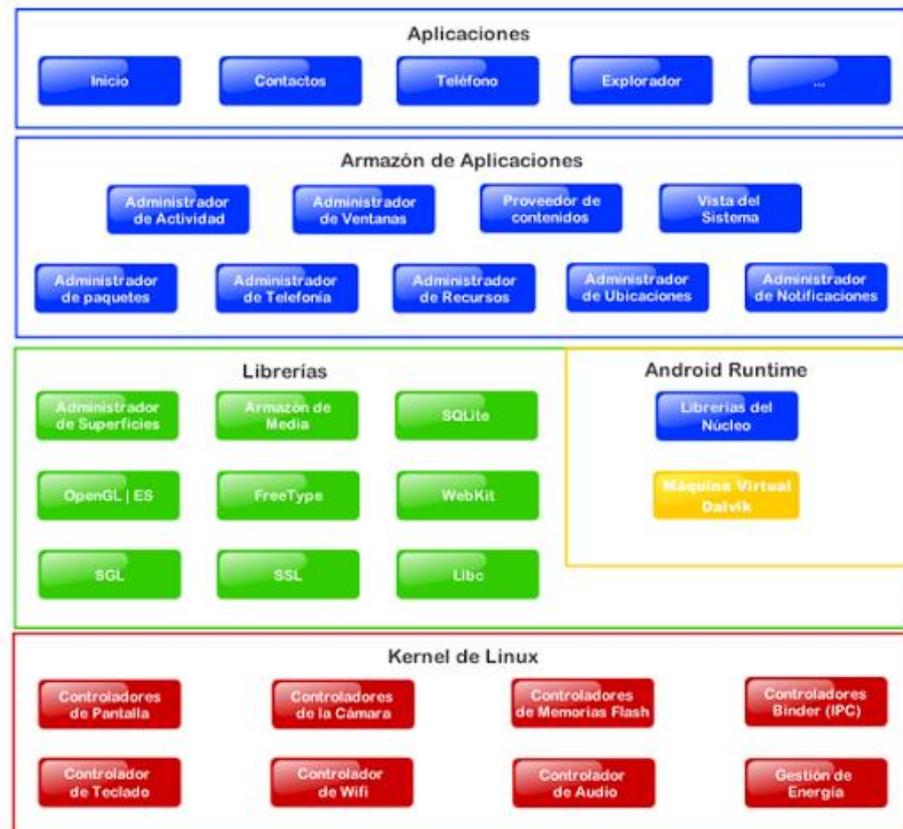


Figura 3.4.- Esquema de la arquitectura Android

Explicamos cada una de las capas, comenzado de abajo hacia arriba:

- **Kernel de Linux:** El núcleo del sistema operativo Android está basado en el kernel de Linux versión 2.6, similar al que se puede incluir en cualquier distribución Linux, solo que adaptado a las características del hardware en el que se ejecutara Android, es decir, para dispositivos móviles.

El núcleo actúa como una capa de abstracción del hardware y el resto de capas de la arquitectura. El desarrollador no accede directamente a esta capa, sino que debe utilizar las librerías disponibles en capas superiores. Con este hecho el desarrollador se quitará el quebradero de

cabeza que puede suponer conocer las características precisas de cada dispositivo móvil. El Kernel también se encarga de gestionar los diferentes recursos del teléfono (energía, memoria...etc.) y el sistema operativo en sí: proceso, elementos de comunicación...etc.

- **Librerías:** Es la siguiente capa que se sitúa sobre el Kernel. Están escritas en C o C++ y compiladas para la escritura hardware específica del teléfono. Estas normalmente están hechas por el fabricante, quien también se encarga de instalarlas en el dispositivo antes de ponerlo a la venta. El objetivo de estas librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando así tener que codificarlas cada vez que se ejecuten y garantizando que se llevan a cabo de un modo más eficiente. Entre las librerías incluidas podemos encontrar: OpenGL (motor gráfico), Bibliotecas multimedia, Webkit (navegador), SSL, FreeType, SQLite...etc.
- **Entornos de ejecución:** No se considera capa dado que también está formado por librerías. Aquí se encuentran las librerías con las funcionalidades habituales en Java, así como otras específicas de Android.

El componente principal del entorno de ejecución Android es la máquina virtual Dalvik. Las aplicaciones Android se codifican en Java y son compiladas en un formato específico para que esta máquina virtual las ejecute. La ventaja de esto es que las aplicaciones se compilan una sola vez y de esta forma se garantiza que estarán listas para distribuirse a cualquier dispositivo Android.

Cabe destacar que Dalvik es una variación de la máquina virtual de Java, por lo que no es compatible con el bytecode Java. Los ejecutables tienen extensión “.dex” que es específico para Dalvik, por lo que no podemos ejecutar aplicaciones Java en Android ni viceversa.

- **Armazón de aplicaciones:** La siguiente capa está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías

Java que acceden a los recursos de las capas anteriores a través de la máquina virtual Dalvik. Siguiendo el diagrama encontramos:

- Administrador de actividad: Se encarga de administrar la pila de actividades de nuestra aplicación, así como su ciclo de vida.
- Administrador de ventanas: Se encarga de administrar la pila de actividades de nuestra aplicación, así como su ciclo de vida.
- Proveedor de contenidos: Esta librería es muy interesante porque crea una capa que encapsula los datos que se compartirán entre las aplicaciones para poder tener así el control de cómo se accede a la información.
- Vista del sistema: Está formada por los elementos que nos ayudaran a construir las interfaces de usuario: botones, cuadro de texto...etc.
- Administrador de notificaciones: Engloba los servicios para notificar al usuario cuando algo requiera su atención.
- Administrador de ubicaciones: Permite manipular los elementos hardware como el acelerómetro, giroscopio...etc.
- Administrador de recursos: Esta librería permite gestionar todos los elementos que forman parte de la aplicación y que están fuera del código.
- Administrador de telefonía: Nos permite realizar llamadas o recibir SMS/MMS.
- Administrador de paquetes: Esta biblioteca permite obtener información sobre paquetes instalados en el dispositivo Android. Además, permite gestionar la instalación de nuevos paquetes.
- **Aplicaciones:** Es la última capa que incluye todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario como las que no, las nativas y las administradas, las que vienen preinstaladas en el dispositivo móvil y aquellas que el usuario ha instalado.

Como podemos observar, Android nos proporciona un entorno sumamente poderoso para que podamos programar aplicaciones que hagan cualquier cosa. El potencial de este sistema se sitúa en el control total por parte del usuario para que haga de su teléfono móvil un dispositivo a su medida.

### 3.2.3. Componentes

Los componentes son elementos clave para el desarrollo de aplicaciones Android ya que representan un punto a través del cual el sistema puede acceder a la aplicación [10]. En este apartado vamos a realizar una descripción de los más importantes:

- **Vista (view):** Son los elementos que componen la interfaz de usuario de una aplicación. Todas las vistas van a ser objetos dependientes de la clase view y serán definidas en un fichero XML.
- **Servicios:** Son componentes sin interfaz gráfica que se ejecutan en segundo plano. Son similares a los servicios presentes en otros sistemas operativos. Estos servicios pueden realizar cualquier tipo de acción, como por ejemplo actualizar datos, lanzar notificaciones...etc.
- **Proveedor de contenidos:** Es el mecanismo que se ha definido en Android para compartir datos entre aplicaciones.
- **Broadcast Receiver:** Es un componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema (“Batería baja”, “SMS recibido” ...etc.) o por otras aplicaciones broadcast. Es decir, no dirigidos a una aplicación concreta sino a cualquiera que quiera escucharlo.
- **Widget:** Son elementos visuales, normalmente interactivos, que pueden mostrarse en la pantalla principal del dispositivo Android y recibir actualizaciones periódicas. Permiten mostrar información de la aplicación al usuario directamente sobre la pantalla principal.
- **Intent:** Es un elemento básico de comunicación entre los dispositivos Android. Son los mensajes o peticiones enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Mediante un Intent se puede iniciar una actividad o servicio, enviar un mensaje broadcast...etc.

Todos los componentes utilizados en el desarrollo de una aplicación Android deben ser declarados en el archivo AndroidManifest.xml. Este archivo

permite al sistema operativo conocer toda la información esencial acerca de la aplicación antes de ejecutarla.

### *3.2.4. Ciclo de vida de una actividad*

Una aplicación Android está compuesta por un conjunto de elementos básicos de interacción con los usuarios, conocidos como actividades [11]. La aplicación corre dentro de un proceso de Linux. Este proceso se inicia con la aplicación y continuará vivo hasta que no sea requerido y el sistema reclame su memoria para asignársela a otra aplicación. Esto implica que en Android el ciclo de vida es controlado principalmente por el sistema, en lugar de ser controlado directamente por el usuario.

El sistema mantiene una pila de actividades previamente visualizadas, esto permite que el usuario pueda retroceder a la actividad anterior pulsando la tecla “retorno”.

Para entender el ciclo de vida de una actividad, que representa cada una de las pantallas de nuestra aplicación. A continuación, se muestra un diagrama en la figura 3.5.

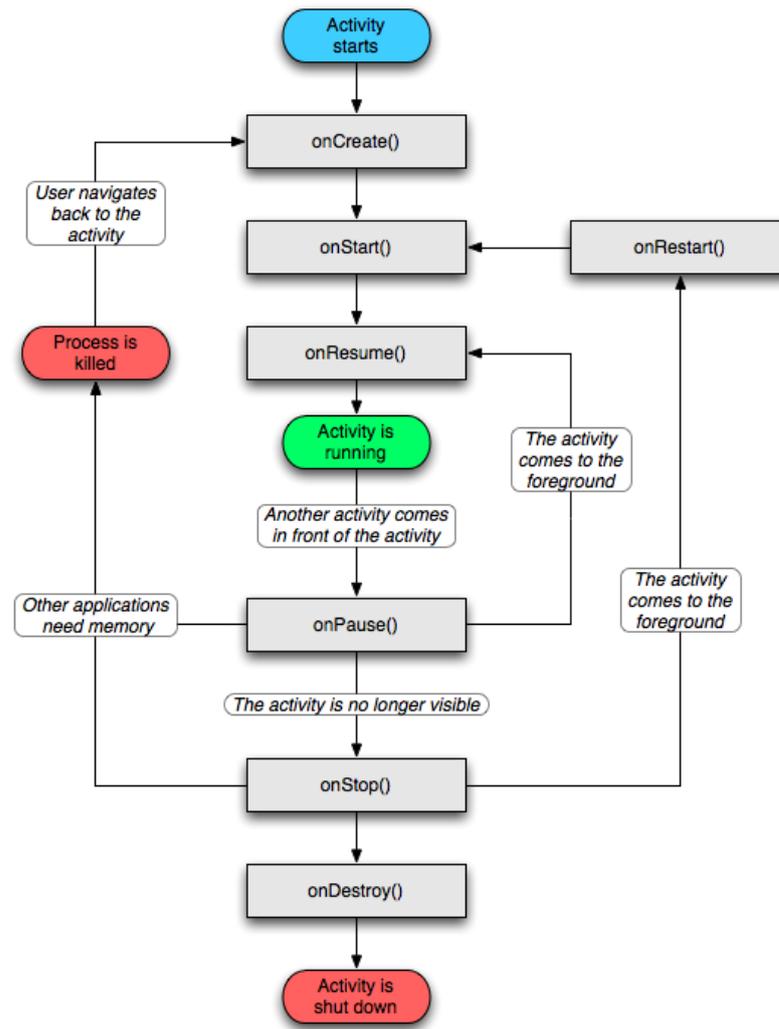


Figura 3.5.- Ciclo de vida de una actividad de Android

Pasemos a explicar brevemente cada uno de los métodos:

- **onCreate():** Se dispara cuando la actividad es llamada por primera vez. En este método es donde se realiza la inicialización de la aplicación. Después de ejecutarse se llama al método onStart().
- **onRestart():** Se ejecuta cuando la actividad ha sido parada, y quieres volver a utilizarla. Como se muestra en el diagrama después de un onStop() se ejecuta este método e inmediatamente se llama al método onStart().
- **onStart():** Se ejecuta cuando la actividad se está mostrando en la pantalla del usuario.

- **onResume():** Se inicia una vez que la actividad ha terminado de cargarse en el dispositivo el usuario empieza a interactuar con la aplicación. Cuando el usuario termina de utilizar la actividad es cuando se llama al método onPause().
- **onPause():** Se ejecuta cuando el sistema inicia una nueva actividad que necesitara los recursos del sistema centrados en ella. Después de esta llamada puede venir un onResume() si la actividad que haya ejecutado el onPause() vuelve a aparecer en primer plano o un onStop() si se hace invisible para el usuario.
- **onStop():** Se ejecuta cuando la actividad ya no es visible para el usuario porque otra actividad ha pasado a primer plano. Después de que se haya ejecutado este método nos quedarían tres opciones: ejecutar el onStart() para que la actividad vuelva a aparecer en primer plano, que el sistema elimine este proceso porque otros procesos requieran memoria o ejecutar el onDestroy() para apagar la aplicación.
- **onDestroy():** Esta es la llamada final de la actividad, después de esta llamada, la actividad es destruida. Si quisiéramos volver a ejecutar una nueva actividad se arrancarían un nuevo ciclo de vida.

### *3.2.5. Estructura de un proyecto Android*

Al crear un nuevo proyecto de Android con nuestra herramienta de desarrollo Android Studio nos aparecerán una serie de carpetas y archivos organizados en una jerarquía bien definida. Es muy importante entender las carpetas fundamentales de esta jerarquía de ficheros para poder así desarrollar un proyecto en Android [12].

En la figura 3.6 se muestra un ejemplo de la jerarquía de ficheros que nuestra herramienta de desarrollo nos crea.

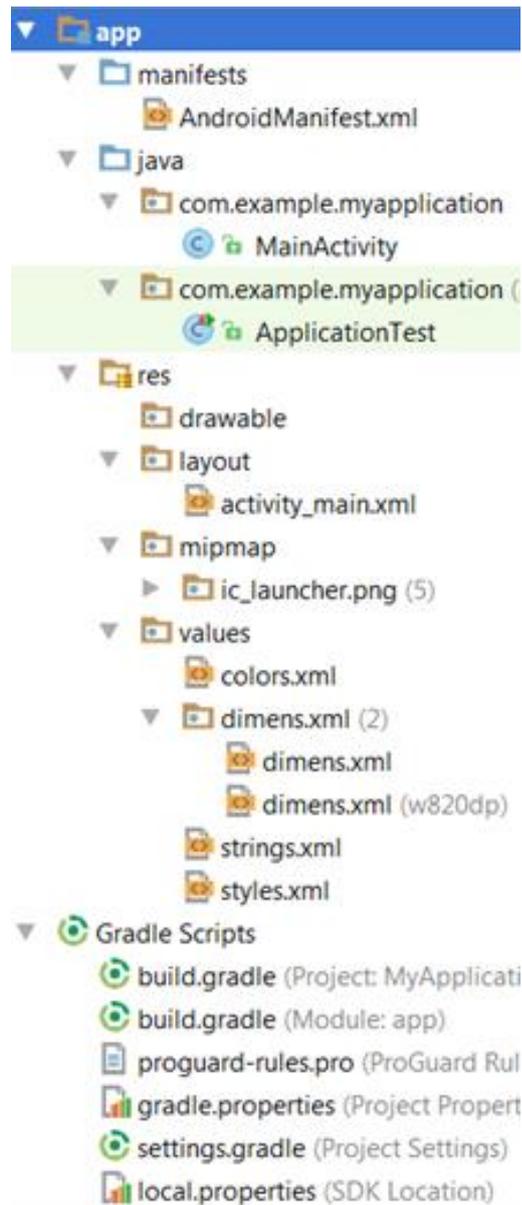


Figura 3.6.- Esquema de la estructura de un proyecto en Android Studio

A continuación, vamos a explicar la utilidad y el contenido de cada una de las carpetas y archivos:

- **Manifests:** En esta carpeta podemos encontrar el archivo `AndroidManifest.xml`. En este fichero se describe la aplicación Android. También, se declaran los permisos que necesitara nuestra aplicación. Se indica la versión mínima de Android para poder ejecutarla, el paquete Java, la versión de la aplicación, se indican las actividades, las intenciones, los servicios...etc.

- **Java:** Esta carpeta contiene el código fuente de la aplicación.
  - **MainActivity:** Clase en Java con el código de la actividad inicial.
  - **AplicacionTest:** Clase en Java pensada para insertar código de testeo de la aplicación utilizando la API JUnit.
- **Res:** Carpeta que contiene los recursos utilizados por la aplicación.
  - **Drawable:** En esta carpeta se almacenan los ficheros de imágenes (JPEG o PNG) y descriptores de imágenes.
  - **Mipmap:** Es una carpeta con la misma finalidad que la drawable. La única diferencia es que, si ponemos los gráficos en esta carpeta, estos no son rescaldados para adaptarlos a la densidad grafica del dispositivo donde se ejecutará la aplicación. Es recomendable utilizar esta carpeta para almacenar los iconos de nuestra aplicación.
  - **Layout:** Contiene ficheros XML con las vistas de la aplicación.
  - **Menu:** Está compuesta por ficheros XML con cada uno de los menús de cada actividad.
  - **Values:** También utilizaremos ficheros XML para indicar valores usados en la aplicación, de esta manera podremos cambiarlos desde estos ficheros sin necesidad de ir al código fuente.
- **Gradle Scripts:** En esta carpeta se almacenan una serie de ficheros Gradle que permiten compilar y construir la aplicación. El fichero más importante es build.gradle (Module:app) que es donde se configuran las opciones de compilación del módulo.

### 3.3. Servicios web

#### 3.3.1. Introducción

Los servicios web son una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Esta tecnología permite que las aplicaciones se comuniquen que no dependan de la plataforma ni del lenguaje de programación.

Existen múltiples definiciones sobre lo que son los servicios web, lo que muestra su complejidad a la hora de dar una adecuada definición que englobe todo lo que son e implican [13].

La World Wide Web Consortium [W3C] define un servicio web como:” Un Servicio Web (Web Service [WS]) es una aplicación software identificada por un URI (Uniform Resource Identifier), cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Los Servicios Web hacen posible la interacción entre” agentes” software (aplicaciones) utilizando mensajes XML intercambiados mediante protocolos de Internet.”

Los servicios web suelen ser considerados como APIs Web que pueden ser accedidos dentro de una red (principalmente internet) y ejecutados en el sistema donde se encuentre alojados.

Entre las características de los servicios web podemos destacar:

- Están basados en el protocolo HTTP.
- Son flexibles.
- Son interoperables.
- Son accesibles desde cualquier lugar.
- Debe contener una descripción del mismo para que cualquier aplicación pueda saber cuál es la funcionalidad del mismo.
- Debe ser localizado para que una aplicación localice al servicio que necesite de forma automática.

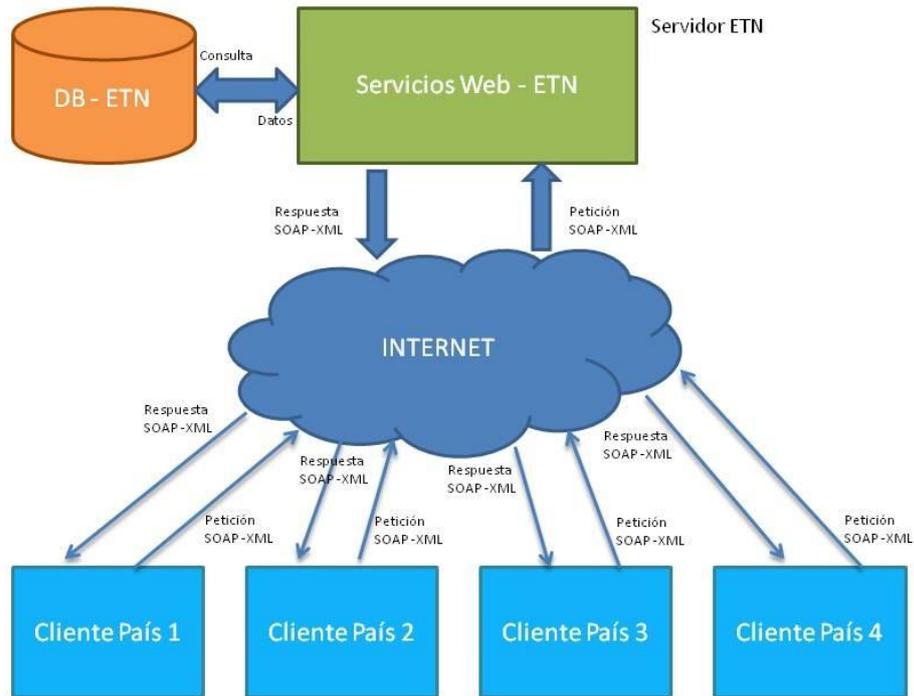


Figura 3.7.- Ejemplo de funcionamiento de los servicios web

### 3.3.2. Arquitectura de los servicios web

Los servicios Web presentan una arquitectura orientada a servicios SOA que permite crear una definición abstracta de un servicio, proporcionar una implementación concreta de dicho servicio, publicar y localizar un servicio, seleccionar una instancia de un servicio, y utilizar dicho servicio con una elevada interoperabilidad [14]. Es posible desacoplar la implementación del servicio Web y su uso por parte de un cliente. También es posible desacoplar la implementación del servicio y de cliente. Las implementaciones concretas del servicio pueden desacoplarse a nivel de lógica y transporte. En la figura 3.8 se muestra el diagrama de una arquitectura orientada a servicios.

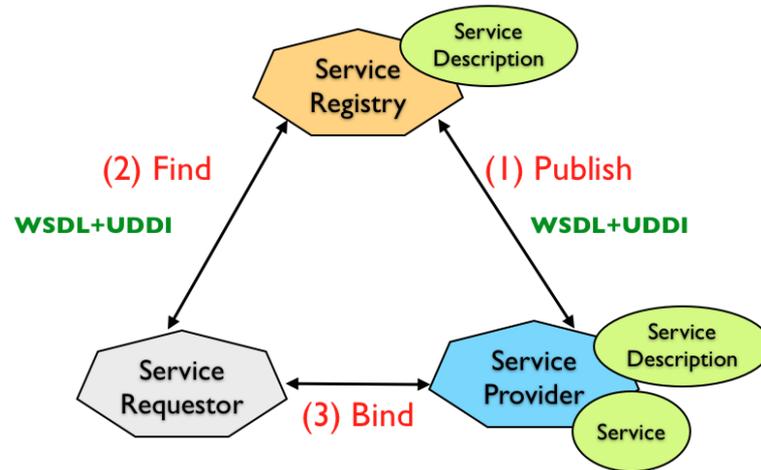


Figura 3.8.- Esquema de la arquitectura de los servicios web

El proveedor del servicio define la descripción abstracta de dicho servicio utilizando un lenguaje de descripción de Servicios Web. Seguidamente, se crea un Servicio concreto a partir de la descripción abstracta del servicio, produciendo así una descripción concreta del servicio en WSDL. Dicha descripción concreta puede entonces publicarse en un servicio de registro como por ejemplo UDDI (Universal Description, Discovery and Integration). Un cliente de un servicio puede utilizar un servicio de registro para localizar una descripción de un servicio, a partir de la cual podrá seleccionar y utilizar una implementación concreta de dicho servicio.

La descripción abstracta se define en un documento WSDL como un PortType. Una instancia concreta de un Servicio se define mediante un elemento port de un WSDL (consistente a su vez en una combinación de un PortType, un binding de codificación y transporte, más una dirección). Un conjunto de ports definen un elemento service de un WSDL.

### 3.3.3. Ventajas de los servicios web

Las ventajas de los servicios web son:

- Proporcionan interoperabilidad entre aplicaciones. Los servicios web se comunican utilizando HTTP y XML, por lo que cualquier dispositivo que soporte estas tecnologías podrá tener y acceder a los servicios web.

- Permiten una implementación sencilla. Los servicios web están basados en conceptos fácilmente entendibles y actualmente existen herramientas que permiten desarrollar y crear un servicio web prácticamente teniendo solo algunas nociones de programación.
- Proporcionan accesibilidad. Los servicios web pueden ser completamente descentralizados, distribuidos sobre internet y accedidos a través de una gran variedad de dispositivos.
- Especificaciones universalmente aceptadas. Los servicios web están basados en especificaciones estándar para el intercambio de datos.
- Integración con sistemas existentes, permiten una mayor agilidad y flexibilidad debido a una mejor integración con los sistemas que ya existen.

#### *3.3.4. Desventajas de los servicios web*

Las desventajas de los servicios web son:

- Para realizar transacciones no pueden compararse con los estándares abiertos de computación distribuida como CORBA (Common Object Request Broker Architecture).
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida, como RMI (Remote Method Invocation), CORBA, o DCOM (Distributed Component Object Model).
- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear la comunicación entre programas.
- Existe poca información de servicios web para algunos lenguajes de programación.

#### *3.3.5. Tipos de servicios web*

Los servicios web pueden implementarse de varias formas. Con lo que podemos distinguir dos tipos de servicios web: los denominados

servicios web “grandes” (“big” Web Services), los llamaremos servicios web SOAP y servicios web REST.

- **Los servicios web SOAP:** Este tipo de servicios utilizan mensajes XML para comunicarse que siguen el estándar SOAP (Simple Object Access Protocol). Dichos sistemas normalmente contienen una descripción legible por la máquina de la descripción de las operaciones ofrecidas por el servicio, escrita en WSDL (Web Services Description Language), que es un lenguaje basado en XML para definir las interfaces sintácticamente.

El formato de mensaje SOAP y el lenguaje de definición de interfaces WSDL se ha extendido bastante, y muchas herramientas de desarrollo, por ejemplo, Netbeans, pueden reducir la complejidad de desarrollar aplicaciones de servicios Web.

El diseño de un servicio basado en SOAP debe establecer un contrato formal para describir la interfaz que ofrece el servicio Web. WSDL puede utilizarse para describir los detalles del contrato, que pueden incluir mensajes, operaciones, bindings, y la localización del servicio Web. También deben tenerse en cuenta los requerimientos no funcionales, como por ejemplo las transacciones, necesidad de mantener el estado (addressing), seguridad y coordinación.

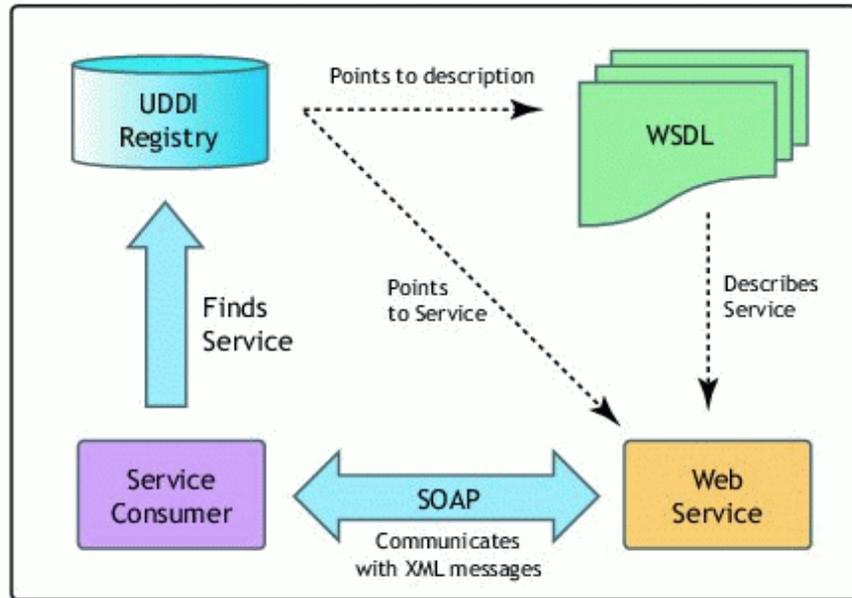


Figura 3.9.- Ejemplo de servicio SOAP

- **Los servicios web REST:** Es tipo de servicio web se apoya totalmente en el estándar HTTP. REST nos permite crear servicios y aplicaciones que puedan ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que son más simples y convencional que otras alternativas como puede ser SOAP.

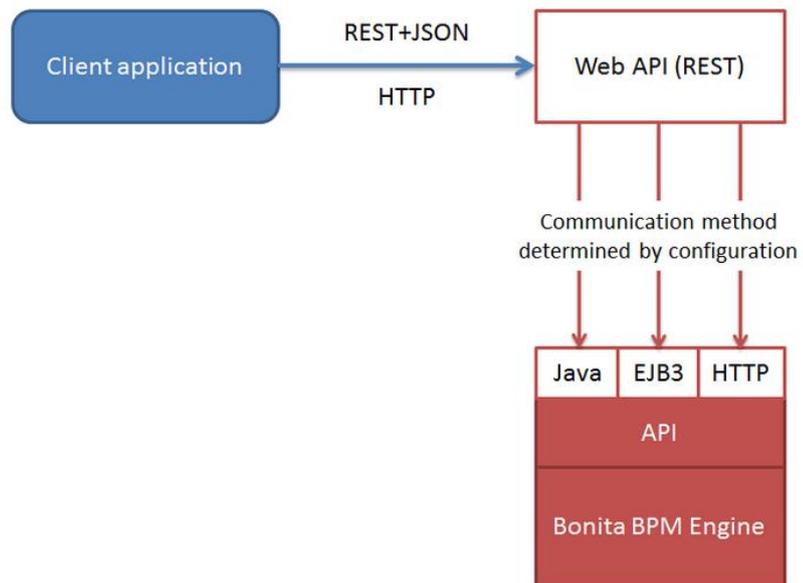


Figura 3.10.- Ejemplo de servicio REST

En el siguiente apartado nos vamos a centrar en los servicios REST ya que son el tipo de servicios utilizado para desarrollar este proyecto.

### 3.3.6. Servicios REST

#### 3.3.6.1. Introducción a los servicios REST

Los servicios REST (Representational State Transfer) es una forma ligera de crear servicios web [15]. Es tipo de servicios fue introducido por primera vez por Roy Fielding (uno de los creadores de HTTP) en la lectura de su tesis para describir un tipo de arquitectura de los servicios en red. La idea es utilizar simplemente HTTP para realizar llamadas entre máquinas, en lugar de otros mecanismos o protocolos más complejos.

Uno de los conceptos más importantes en este tipo de servicios son los recursos (elementos de información de los que se compone la web), que pueden ser accedidos utilizando un identificador URI. Para realizar la manipulación de estos recursos, los clientes y servidores de la red se comunican a través de la interfaz estándar HTTP e intercambian representaciones de estos.

REST ofrece por si solo características de seguridad, encriptación, gestión de sesiones...etc. y por supuesto como en cualquier servicio web, estas funciones se pueden añadir por encima de HTTP: mediante usuarios, contraseñas, SSL, cookies, etc.

Es importante destacar el hecho de que REST no es un estándar, ya que solo es un tipo de arquitectura, pero también está basado en los estándares HTTP, URL, XML, HTML, MIME.

Los servicios REST tienen las siguientes características:

- Las interfaces deben construirse sobre HTTP:
  - Debe soportar los métodos:
    - **HTTP GET**: Es usado para obtener la representación de un recurso.
    - **HTTP DELETE**: Se usa para eliminar la representación del recurso.
    - **HTTP POST**: Usado para actualizar o crear representaciones de un recurso.

- **HTTP PUT:** Es utilizado para crear representaciones de un recurso.
  - Para indicar que acción debe realizar el cliente en relación a la respuesta, debe proporcionar los siguientes códigos de estado HTTP:

<i>Código de estado</i>	<i>Significado</i>
200	Correcto
201	Creado
304	No modificado
400	Petición incorrecta
401	No autorizado
403	Olvidado
404	No encontrado
422	Entidad no procesable
500	Error interno en el servidor

Tabla 3.1.- Códigos de estado HTTP.

- La mayoría de los mensajes se encuentran en XML, pero también pueden venir definidos en el formato JSON.
- Los servicios y los proveedores de servicios deben ser recursos, mientras que los consumidores pueden ser un recurso.
- Se pueden codificar mensajes simples en las URL.

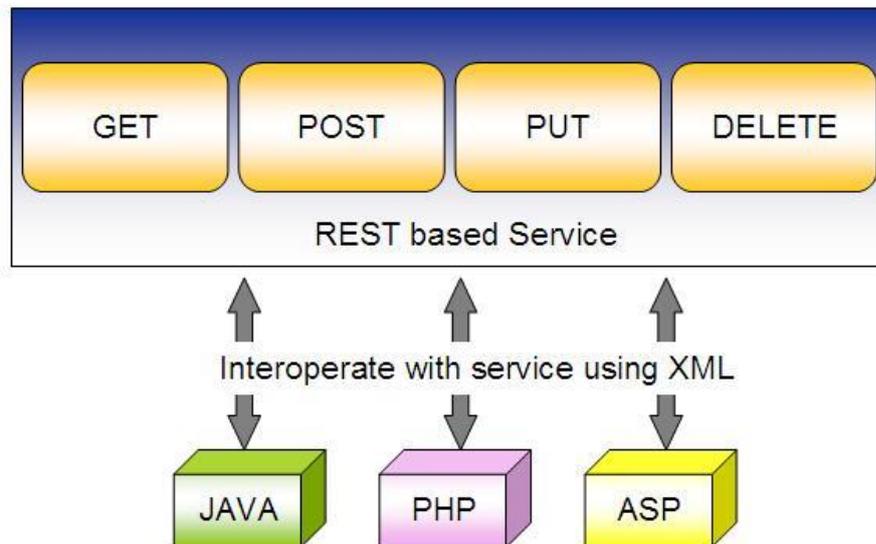


Figura 3.11.- Métodos de los servicios REST

Además, la arquitectura REST tiene que cumplir con 6 principios:

- **No se mantiene el estado:** Cada petición HTTP mantiene toda la información necesaria para ser comprendida. Debido a esto ni el cliente ni el servidor necesitan recordar ningún estado de la comunicación entre mensajes. Esto permite que se mejore el rendimiento de los servicios web y que sea más fácil su implementación y diseño.
- **Cliente-servidor:** El cliente tiene que estar separado del servidor a través de interfaces uniformes, es decir, el cliente no debe saber nada de cómo se encuentra almacenada la información o de cómo se obtiene. Por otro lado, los servidores tampoco deben saber la manera en que la información consultada es presentada.
- **Sistema en capas:** El cliente no debe saber si está conectado directamente a un servidor final o a un intermediario.
- **Un conjunto de operaciones bien definidas:** Estas operaciones se deben aplicar a los recursos de forma inequívoca. En el caso de HTTP son: GET, POST...etc.
- **Una sintaxis universal para identificar los recursos:** En REST cada recurso debe ser accedido únicamente a través de su URI. Estas URIs se definen de forma análoga a una estructura basada en directorios, de forma que la jerarquía es fácilmente comprensible y necesita escasa documentación.

- **Capaz de almacenarse en caché:** Las respuestas deben definirse como almacenables en caché, para evitar que los clientes hagan uso inapropiado de información devuelta por una solicitud.

Las ventajas de los servicios REST son las siguientes:

- Separación de un recurso de su representación.
- Sencillos de diseñar e implementar.
- Bajo consumo de recursos.
- El cliente no necesita información de enrutamiento a partir de la URI inicial.
- Es ligero ya que sus respuestas contienen exactamente la información que se necesita

Las desventajas de los servicios REST son las siguientes:

- No hay estándar en sus respuestas por lo que no se definen tipo de datos.
- Manejar todas las URIs pueden suponer un problema.
- La seguridad puede llegar a ser una tarea difícil de implementar.

### 3.3.6.2. Estructura de las URLs

Desde el punto de vista del cliente que accede a un recurso, la URI determina lo intuitivo que puede llegar a ser el servicio web. Las URI en los servicios REST deben ser intuitivas hasta el punto de que sea fácil adivinarlas.

Una forma de lograr este nivel de usabilidad es definir URIs con una estructura al estilo de directorios. Este tipo de URIs es jerárquica, con una única raíz, y va abriendo ramas a través de las subrutinas para exponer las áreas principales del servicio. Cada URL para cada recurso debería ser identificado de forma única.

En un sistema como en el que se trata en este proyecto, donde se puede necesitar los datos de un vídeo en concreto, un ejemplo de URL sería:

- GET→<http://mipagina.com/video/id>→ Devuelve los datos del vídeo-tutorial con identificador id.

Existen varias reglas a la hora de crear URIs para un servicio web REST:

- Mantener todo en minúsculas.
- Sustituir espacios por guiones o guiones bajos.
- Deben mantener una jerarquía lógica.
- Deben ser únicas, no debemos tener más de una URI para identificar un recurso.
- Deben ser independientes del formato.
- Los nombres de URI no deben implicar una acción, por lo tanto, debe evitarse usar verbos en ellos.
- Deben ser intuitivas.

## Capítulo 4

---

### 4. Proceso de ingeniería del software

#### 4.1. Fases de la ingeniería del software

Las fases del proceso de ingeniería software que se han llevado en nuestra propuesta son las siguientes [16]:

- **Especificación de requerimientos:** Descripción completa del comportamiento del sistema, incluyendo los requisitos y funcionalidades de lo que dispondrá.
- **Análisis del sistema:** Obtendremos un modelo del sistema claro y consistente.
- **Diseño del sistema:** Se determina el funcionamiento del sistema para satisfacer los requerimientos establecidos.
- **Implementación del sistema:** Se realiza el desarrollo del sistema descrito, teniendo en cuenta sus requerimientos y el diseño establecido.
- **Pruebas:** Verificación de que el software desarrollado cumple con los requerimientos establecidos.

#### 4.2. Especificación de requerimientos

Los requerimientos o requisitos de un sistema describen los servicios que debe ofrecer nuestro sistema y las restricciones asociadas a su funcionamiento.

Esta es la primera fase de nuestro proceso de ingeniería, y es de suma importancia que se defina correctamente ya que es la base de nuestro software desarrollado.

Existen dos tipos de requerimientos:

- **Requerimientos funcionales:** Expresan la naturaleza del funcionamiento del sistema.

- **Requerimientos no funcionales:** Definen como debe ser el sistema.

A continuación, se describirán con detalle cada uno de los requerimientos (funcionales y no funcionales) de los que se compone este proyecto.

#### *4.2.1. Requerimientos de la aplicación web*

A continuación, se detallan los requerimientos funcionales y no funcionales necesarios para nuestro sistema web.

##### 4.2.1.1. Funcionales

Los requerimientos funcionales de nuestro sistema web serian:

1. **Login usuario:** El sistema debe proporcionar un formulario para que el usuario pueda acceder a su panel de administración para poder así, realizar las operaciones que necesite sobre sus vídeos. Además, el sistema debe proporcionar la funcionalidad de poder cerrar la sesión.
2. **Registrar usuario:** El sistema debe proporcionar un formulario que permita que un usuario pueda darse de alta en nuestro sistema web.
3. **Gestionar Vídeos:** El sistema debe permitir gestionar los vídeos por parte del administrador de los mismos. El sistema para permitir la gestión de los vídeos-tutoriales ofrecerá las siguientes operaciones:
  - Editar: Permitirá editar la información referente a un vídeo del usuario.
  - Borrar: Permitirá eliminar un vídeo-tutorial del usuario.
  - Añadir: El sistema por medio de un formulario, permitirá que el usuario pueda subir un vídeo al sistema.

Además, el sistema agrupará los vídeos en dos grupos:

- Privados: Este tipo de vídeos solo podrán ser utilizados y visualizados por su propietario.

- Públicos: Este tipo de vídeos podrán ser usados y utilizados por cualquier usuario de nuestro sistema web.
4. **Generar clave privada:** El sistema debe generar automáticamente una clave privada y única para cada uno de los vídeos que el usuario quiera registrar de manera privada.
  5. **Buscar Vídeos:** El sistema debe permitir que el usuario pueda realizar búsqueda de los vídeos disponibles en el sistema web.
  6. **Seguridad:** El sistema ofrecerá al usuario la posibilidad de poder modificar sus credenciales de acceso a nuestro sistema, así como la posibilidad de recordar la contraseña en caso de pérdida u olvido de la misma.
  7. **Contacto:** El sistema permitirá que el usuario a través de un formulario pueda ponerse en contacto con el equipo de desarrollo de este proyecto.

#### 4.2.1.2. No funcionales

Como hemos indicado anteriormente, los requerimientos no funcionales nos describirán las restricciones o limitaciones que han de tenerse en cuenta en el desarrollo de nuestro sistema.

A continuación, se detallan los requerimientos no funcionales necesarios para nuestro sistema web:

1. **Requisitos físicos:** Para entender de una forma correcta los requisitos necesarios, vamos a dividirlos entre los necesarios para el usuario de nuestro sistema web y el servidor donde se aloja:
  - Usuario: El usuario solo necesitara disponer de un dispositivo con conexión a Internet.
  - Servidor: Se necesitará un servidor en donde se pueda alojar nuestro sistema web. Se recomienda que el servidor cumpla con los siguientes requisitos mínimos:
    - Almacenamiento de 10GB
    - RAM 1GB
    - Salida a internet

2. **Requisitos software:** Al igual que en los requerimientos físicos, vamos a dividir en los necesarios para el usuario y el servidor:
  - Usuario:
    - Navegador web
    - Sistema operativo Windows, Linux o Mac OS.
    - Soporte para JavaScript
  - Servidor:
    - Distribución de Linux o Windows
    - MySQL
    - Apache2
    - PHP
3. **Requisitos de la interfaz:** La interfaz de nuestro sistema web debe poseer los siguientes requisitos:
  - Robustez: El sistema debe soportar los posibles fallos producidos por los usuarios.
  - Usabilidad: La interfaz de nuestro sistema web debe ser intuitivo y fácil para el usuario.
  - Flexibilidad: Debe poder adaptarse a cada uno de los dispositivos en los que el usuario pueda visualizar este sistema web.

#### *4.2.2. Requerimientos de la aplicación móvil*

En este apartado, vamos a detallar los requerimientos funcionales y no funcionales necesarios para nuestra aplicación móvil.

##### **4.2.2.1. Funcionales**

Los requerimientos funcionales de nuestra aplicación móvil serán:

1. **Leer etiqueta NFC:** La aplicación móvil debe ser capaz de poder leer el contenido almacenado en una etiqueta NFC y poder visualizar automáticamente el vídeo asociado a esa tarjeta sin la necesidad de manipular el dispositivo por parte del usuario.

2. **Borra etiqueta NFC:** La aplicación móvil debe permitir la funcionalidad de poder borrar el contenido de una etiqueta NFC.
3. **Buscar Vídeo:** Se debe permitir que el usuario busque entre los vídeos disponibles en el sistema web y poder así seleccionar el que más se ajuste a la tarea que quiera realizar. La aplicación móvil permitirá dos tipos de búsqueda:
  - Búsqueda de vídeos: Se realizará una búsqueda entre todos los vídeos públicos del sistema, teniendo en cuenta el tipo de tarea que el usuario desea realizar y la marca o modelo del aparato electrónico con el que se va a realizar la tarea.
  - Vídeo privado: La aplicación buscará el vídeo que contenga la clave privada generada por el usuario.
4. **Visualizar vídeo seleccionado:** La aplicación permitirá que el usuario visualice el vídeo que ha seleccionado.
5. **Escribir tarjeta NFC:** Una vez que el usuario haya seleccionado el vídeo que más se adapte a su tarea, la aplicación debe permitir enlazar una etiqueta NFC con el vídeo seleccionado.
6. **Lanzamiento automático de la aplicación:** Para facilitar la tarea al usuario, nuestra aplicación podrá ejecutarse automáticamente al poner en contacto la parte trasera de nuestro dispositivo móvil o tablet con una etiqueta NFC. De este modo el vídeo asociado a la etiqueta NFC se visualizará en nuestra pantalla de forma automática.

#### 4.2.2.2. No funcionales

Entre los requerimientos no funcionales de nuestra aplicación móvil podemos encontrar:

1. **Requisitos físicos:** El usuario necesitara un dispositivo móvil o tablet con las siguientes características:
  - Android 2.3.3 o superior
  - Conexión a internet
  - Tecnología NFC

- Resolución mínima 800x400 píxeles
2. **Requisitos de la interfaz:** La interfaz de nuestra aplicación móvil debe poseer las siguientes características:
- **Usabilidad:** Debe presentar una interfaz intuitiva y fácil de manejar por el usuario.
  - **Robustez:** La interfaz debe poder soportar los posibles fallos que puedan ocurrir durante el proceso de iteración con el usuario.

### 4.3. Análisis del sistema

A continuación, se va a especificar los casos de uso y escenarios de nuestro sistema web y aplicación móvil.

#### 4.3.1. Casos de uso.

Los diagramas de casos de uso nos describen los pasos o actividades que el usuario deberá realizar para poder llevar a cabo alguna tarea.

Un diagrama de caso de usos estará compuesto por los siguientes elementos:

- **Nombre del caso de uso:** Es un identificador.
- **Actor primario:** Es el usuario que interactúa con el sistema.
- **Condiciones previas:** Requisitos previos que se necesitan cumplir para realizar una determinada tarea.
- **Flujo de eventos:** Los eventos que ocurrirán al realizar una acción.

Como podremos observar, en los diagramas de caso de uso los casos de uso se representan en forma de elipses, los actores en forma de personajes y las relaciones en forma de líneas o flechas. Además, en este tipo de diagramas podemos encontrar cuatro tipos de relaciones:

- **Comunicación:** Es la relación de asociación entre un actor y caso de uso. El estereotipo de este tipo de relación es: <<communicate>>.
- **Inclusión:** Un caso de uso incorpora explícitamente el comportamiento de otro en algún lugar de su secuencia.

- **Extensión:** Un caso de uso base incorpora implícitamente el comportamiento de otro caso de uso en el lugar especificado indirectamente por este otro caso de uso.
- **Generalización:** Un caso de uso hereda el comportamiento y significado de otro, es decir, las relaciones de comunicación, inclusión y extensión del súper-caso de uso.

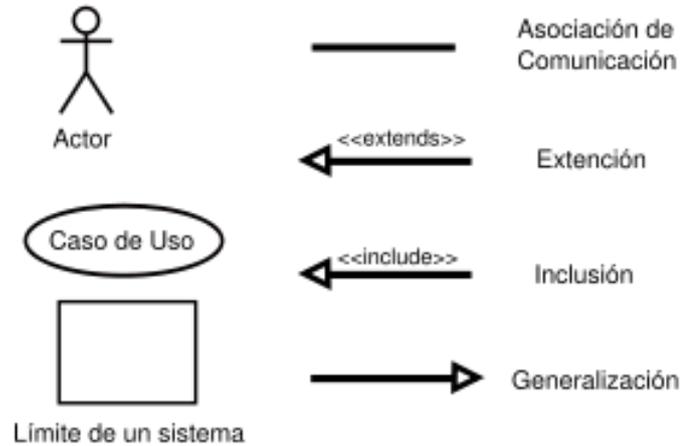


Figura 4.1.- Ejemplo de representación de los elementos de un caso de uso

#### 4.3.1.1. Casos de uso del sistema web

En primer lugar, vamos a mostrar el diagrama frontera de nuestro sistema web. Este diagrama nos permite describir de una forma completa el funcionamiento de nuestro sistema.

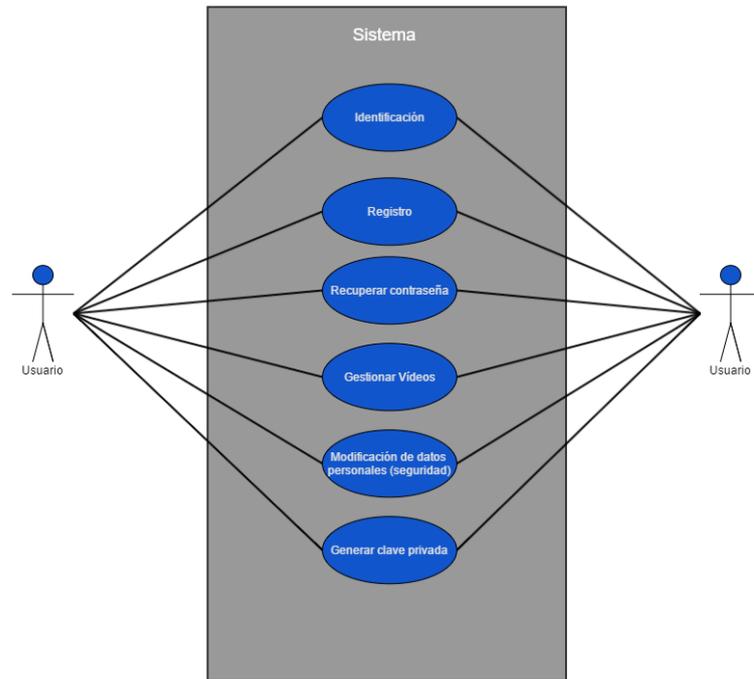


Figura 4.2.- Diagrama frontera del sistema web

En segundo lugar, mostramos los casos con detalle cada uno de los casos de uso mostrados en el diagrama anterior.

#### 1. Caso de uso: Identificación del usuario.

- **Actuadores:** Usuario.
- **Condiciones de entrada:** El usuario no está validado en el sistema y deben existir usuarios en el sistema.
- **Eventos:**
  1. El usuario inicializa el sistema web.
  2. El sistema le muestra al usuario un formulario donde puede introducir sus datos de acceso.
  3. El usuario introduce sus datos.
  4. El sistema comprueba que el usuario existe en el sistema y que sus datos son correctos.
  5. El sistema permite que el usuario pueda acceder a su panel de administración.
- **Excepciones:** El usuario introduce sus datos de manera incorrecta. En este caso el sistema le pedirá al usuario que vuelva a introducir sus datos correctamente.

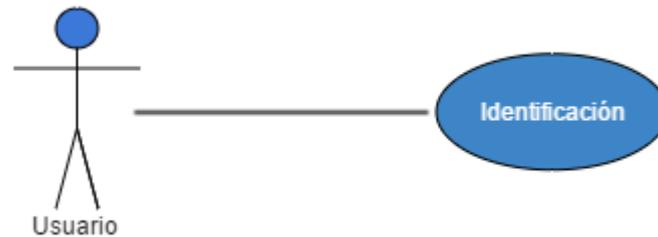


Figura 4.3.-Caso de uso: Identificación del usuario

## 2. Caso de uso: Registro de un nuevo usuario.

- **Actuadores:** Usuario.
- **Condiciones de entrada:** El usuario no debe estar registrado en el sistema.
- **Eventos:**
  6. El usuario inicializa el sistema web.
  7. El sistema le muestra al usuario un formulario de registro donde puede introducir sus datos.
  8. El usuario introduce sus datos.
  9. El sistema comprueba que el usuario no está registrado en el sistema y que sus datos son correctos.
  10. El sistema permite que el usuario pueda acceder a su panel de administración.
- **Excepciones:** El usuario introduce sus datos de manera incorrecta. En este caso el sistema le pedirá al usuario que vuelva a introducir sus datos correctamente.

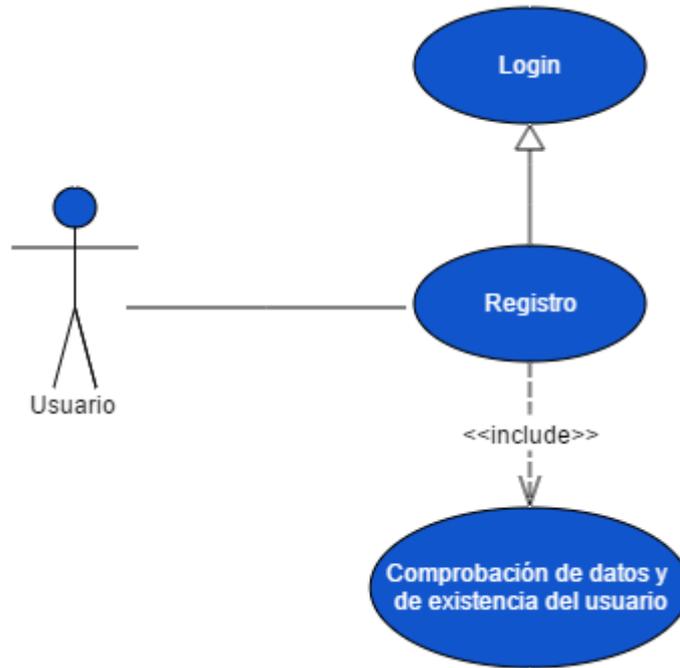


Figura 4.4.- Caso de uso: Registro del usuario.

### 3. Caso de uso: Recuperar contraseña.

- **Actuadores:** Usuario.
- **Condiciones de entrada:** El usuario no debe estar registrado en el sistema.
- **Eventos:**
  1. El usuario inicializa el sistema web.
  2. El sistema muestra al usuario un formulario donde el usuario deberá introducir su cuenta de usuario.
  3. El sistema comprueba que el usuario existe.
  4. El sistema genera una nueva contraseña.
  5. El sistema envía al correo del usuario la nueva contraseña.
- **Excepciones:** El usuario introduce sus datos de manera incorrecta o el usuario no está registrado en el sistema. En este caso el sistema le pedirá al usuario que vuelva a introducir sus datos correctamente o que se registre en el sistema.

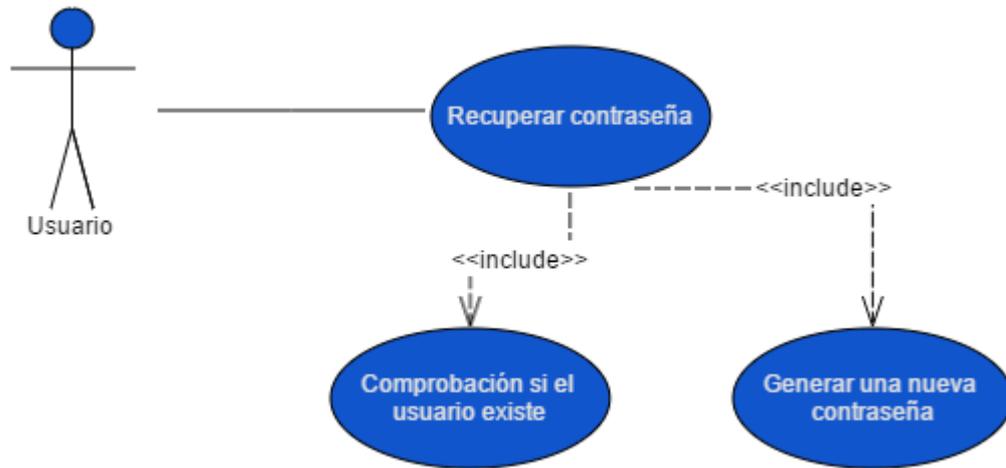


Figura 4.5.-Caso de uso: Recuperar contraseña.

#### 4. Caso de uso: Gestionar vídeos-tutoriales.

- **Actuadores:** Usuario.
- **Condiciones de entrada:** El usuario debe estar identificado en el sistema.
- **Eventos:**
  1. El sistema muestra al usuario todos los vídeos disponibles.
  2. El usuario podrá editar los vídeos que son de su propiedad.
    - a. El sistema muestra al usuario un formulario con los datos del vídeo.
    - b. El usuario realizara los cambios que desee.
    - c. El sistema comprueba que los datos introducidos son correctos.
    - d. El sistema actualizara los datos del vídeo editado.
  3. El usuario podrá eliminar un vídeo de su propiedad.
    - a. El usuario realiza la acción de borrar un vídeo.
    - b. El sistema realizara el borrado del vídeo
    - c. El sistema actualizara la lista de vídeos que pertenecen al usuario del panel de administración.
  4. El usuario podrá subir un vídeo al sistema.

- a. El usuario introduce los datos del vídeo que desea subir en el formulario que el sistema le proporciona.
  - b. El sistema comprueba si los datos introducidos son correctos
  - c. El sistema actualiza la lista de vídeos pertenecientes al usuario.
- **Excepciones:** Los datos introducidos por el usuario son incorrectos, el sistema mostrara un error y pedirá que se vuelvan a introducir de manera correcta.

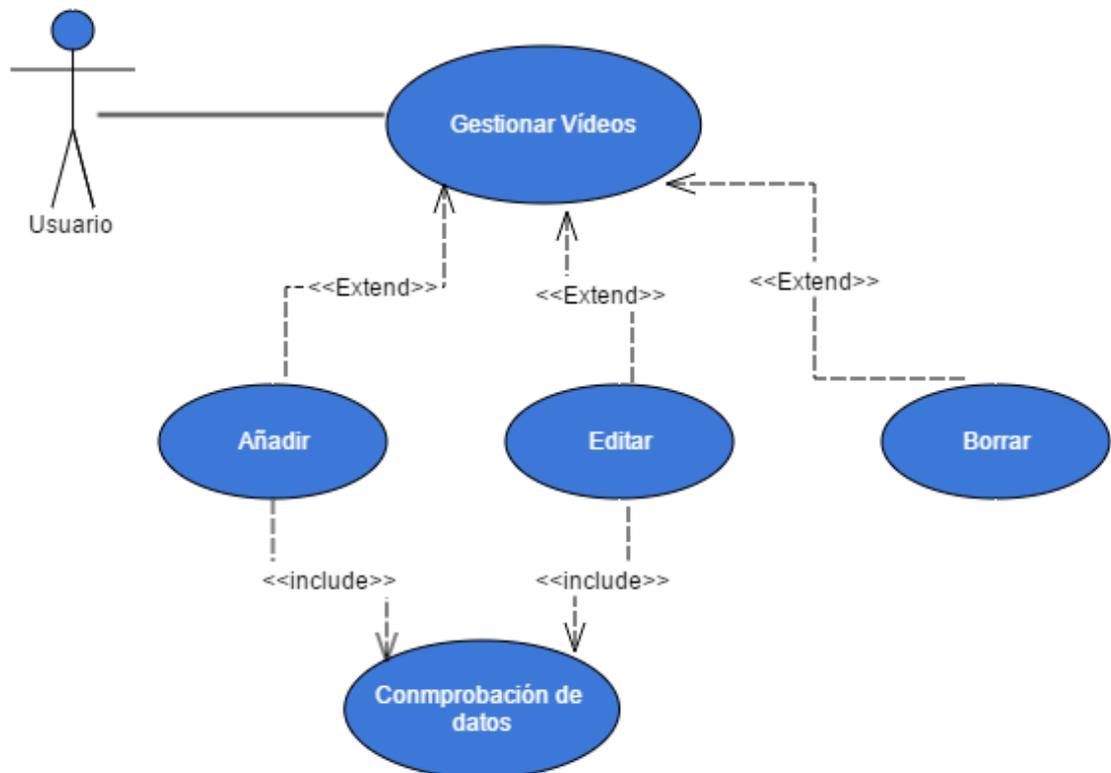


Figura 4.6.- Caso de uso: Gestión de Vídeos.

## 5. Caso de uso: Modificación de datos personales (seguridad).

- **Actuadores:** Usuario.
- **Condiciones de entrada:** El usuario debe estar identificado en el sistema.
- **Eventos:**
  1. El sistema muestra al usuario un formulario con sus datos personales

2. El usuario modifica los datos que considere necesarios, en caso de querer modificar la contraseña el sistema pedirá al usuario que introduzca su contraseña actual y seguidamente la contraseña nueva y una confirmación de esta.
  3. El sistema comprueba si los datos introducidos son correctos.
  4. El sistema realiza la actualización de los datos personales del usuario.
- **Excepciones:** Los datos introducidos por el usuario son incorrectos, el sistema mostrara un error y pedirá de nuevo que los datos se introduzcan de manera correcta.

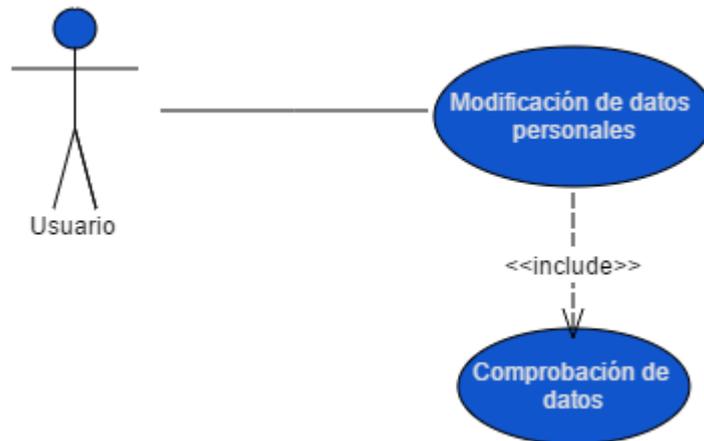


Figura 4.7.- Caso de uso: Modificación de datos personales (seguridad).

## 6. Caso de uso: Generar clave privada para un vídeo-tutorial.

- **Actuadores:** Usuario.
- **Condiciones de entrada:** El usuario debe estar identificado en el sistema.
- **Eventos:**
  1. El usuario pulsa en la opción de vídeo privado en el formulario de subir un nuevo vídeo o de editar un vídeo-tutorial del panel de administración.

2. El sistema automáticamente genera una clave única y privada para identificar el vídeo.
  3. El sistema actualiza los datos del vídeo-tutorial en el panel de administración del usuario.
- **Excepciones:** Los datos introducidos por el usuario son incorrectos, el sistema mostrara un error y pedirá de nuevo que los datos se introduzcan de manera correcta.

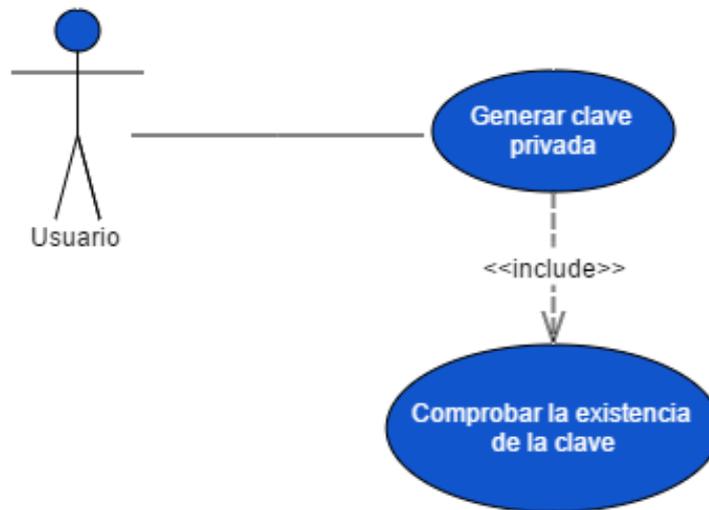


Figura 4.8.- Caso de uso: Generar clave privada

#### 4.3.1.2. Casos de uso de la aplicación móvil

De la misma forma que hemos hecho en el caso del sistema web. En primer lugar, vamos a mostrar el diagrama frontera de nuestra aplicación móvil.

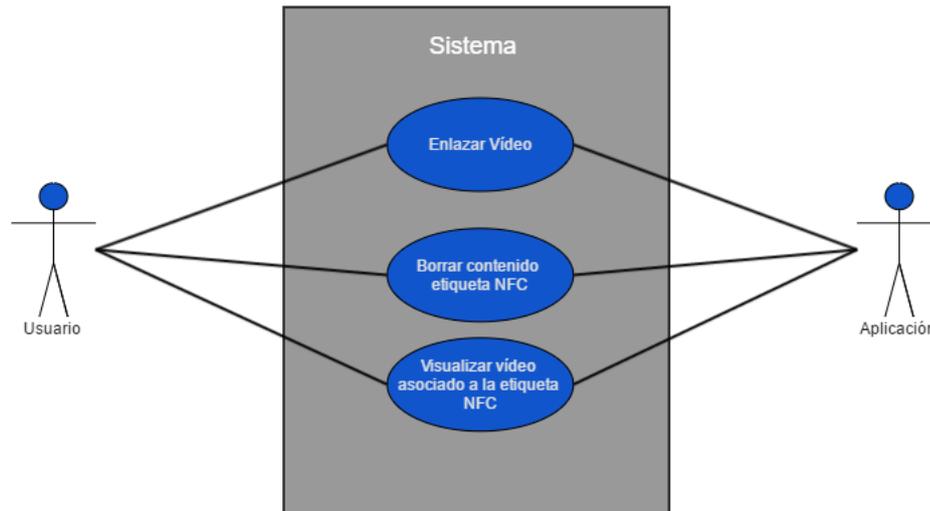


Figura 4.9.- Diagrama frontera de la aplicación móvil

Seguidamente, explicamos con mayor detalle cada uno de los casos de uso de los que se compone nuestra aplicación.

#### 1. Caso de uso: Enlazar vídeo a tarjeta NFC.

- **Actuadores:** Usuario.
- **Condiciones de entrada:** El dispositivo móvil o tablet debe tener conexión a internet y soportar la tecnología NFC.
- **Eventos:**
  1. El usuario realiza la búsqueda del vídeo que más se adapte a su tarea.
  2. La aplicación móvil le mostrara los datos del vídeo seleccionado.
  3. El usuario iniciara la acción de escribir en la tarjeta NFC.
  4. La aplicación realiza la escritura en la etiqueta NFC.
- **Excepciones:** Durante el proceso de escritura puede ocurrir un error. La aplicación mostrara al usuario una alerta indicando que la escritura no se ha realizado correctamente.

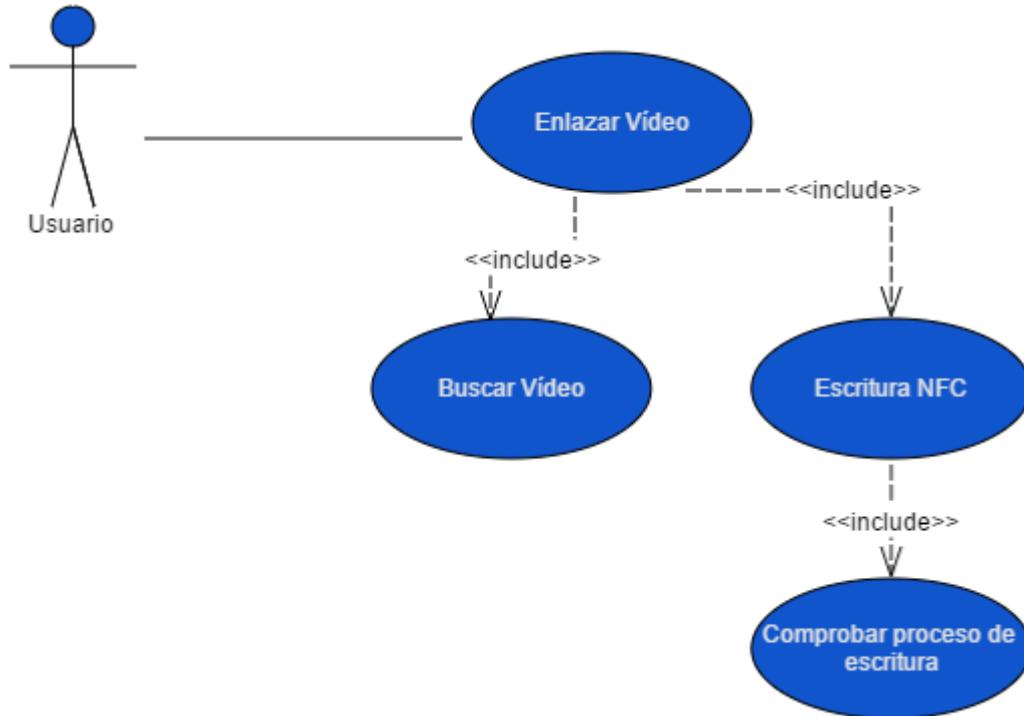


Figura 4.10.- Caso de uso: Enlazar vídeo a tarjeta NFC.

## 2. Caso de uso: Borrar contenido tarjeta NFC.

- **Actuadores:** Usuario.
- **Condiciones de entrada:** El dispositivo móvil o tablet debe soportar la tecnología NFC.
- **Eventos:**
  1. El usuario inicia la acción de eliminar el contenido de la etiqueta NFC.
  2. La aplicación realiza el borrado en la etiqueta NFC.
- **Excepciones:** Durante el proceso de borrado puede ocurrir un error. La aplicación mostrara al usuario una alerta indicando que el borrado no se ha realizado correctamente.

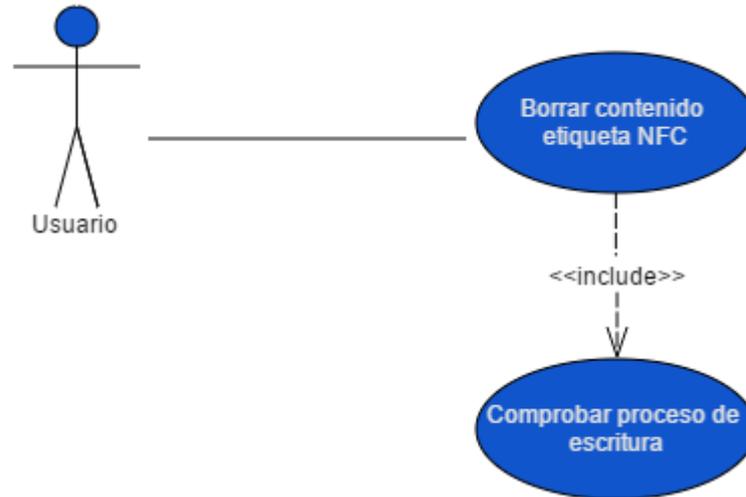


Figura 4.11.- Caso de uso: Borrar contenido etiqueta NFC

### 3. Caso de uso: Reproducir Vídeo asociado a una etiqueta NFC.

- **Actuadores:** Usuario.
- **Condiciones de entrada:** El dispositivo móvil o tablet debe tener conexión a internet y soportar la tecnología NFC.
- **Eventos:**
  1. El usuario coloca la parte trasera de su dispositivo móvil o tablet sobre la etiqueta NFC.
  2. La aplicación móvil leerá el contenido de la tarjeta NFC.
  3. La aplicación móvil reproducirá automáticamente el vídeo-tutorial asociado a la etiqueta NFC.
- **Excepciones:** Durante el proceso de lectura puede ocurrir un error. La aplicación no reproducirá el vídeo-tutorial asociado a esa tarjeta NFC, pero si el usuario vuelve a intentar leer el contenido de dicha tarjeta el proceso de lectura se iniciará automáticamente, con lo que el vídeo se podrá reproducir de nuevo.

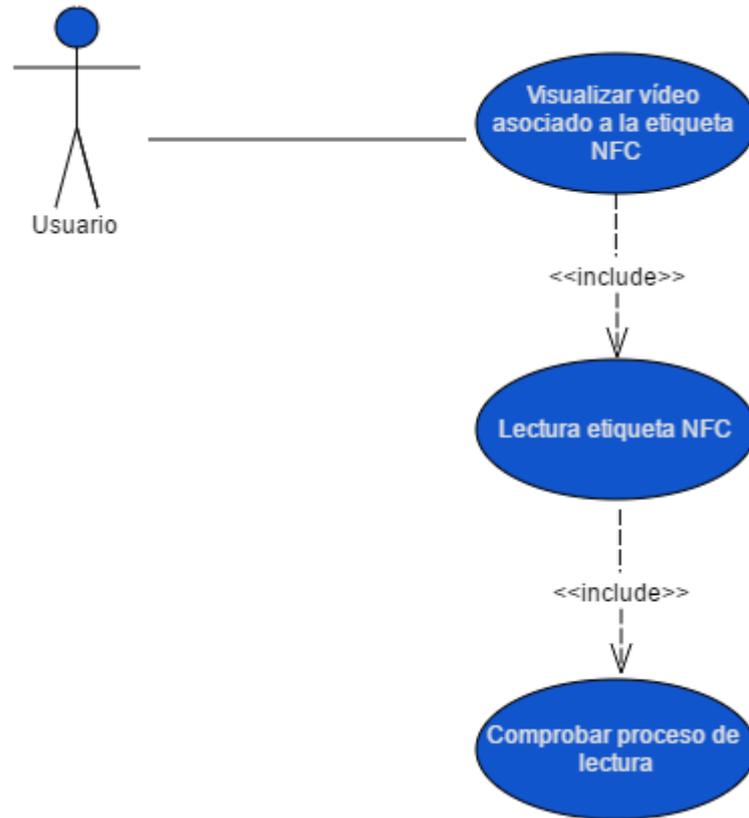


Figura 4.12.- Caso de uso: Reproducir vídeo asociado a una etiqueta NFC.

### 4.3.2. Escenarios

Un escenario es una descripción parcial del comportamiento del sistema en un momento específico. La utilización de escenario implica identificar distintas situaciones y describir la acción a llevar a cabo. La utilización de escenarios es de gran ayuda para especificar requerimientos.

Un escenario está formado por los siguientes elementos:

- Un nombre único
- Una descripción
- Los actores
- El flujo de eventos

A continuación, vamos a describir los escenarios de los que compone nuestro proyecto, para entenderlos de una manera más sencilla, vamos a diferenciar entre el sistema web y la aplicación móvil.

#### 4.3.2.1. Escenarios del sistema web

Al igual que en el apartado anterior, los escenarios más importantes del sistema web son:

##### 1. Escenario: Identificación del usuario.

- Actores: Antonio, Sistema web y Servidor.
- Descripción: Antonio quiere identificarse en el sistema web para poder gestionar sus vídeos.
- Flujo de eventos:
  1. Antonio inicializa el sistema web de gestión de vídeos.
  2. Antonio introduce sus credenciales de acceso (correo y contraseña) en el formulario de inicio de sesión.
  3. El servidor comprueba en la base de datos si las credenciales introducidas por Antonio son correctas. Si son correctas Antonio podrá acceder a su panel de administración.

##### 2. Escenario: Registro de un nuevo usuario.

- Actores: Antonio, Sistema web y Servidor.
- Descripción: Antonio quiere registrarse en el sistema web para poder utilizar sus propios vídeos.
- Flujo de eventos:
  1. Antonio inicializa el sistema web de gestión de vídeos.
  2. Antonio se dirige al formulario de registro que el sistema web le proporciona.
  3. Antonio introduce sus datos.
  4. El sistema web comprueba que los datos introducidos son correctos.

5. El servidor añade a la base de datos un nuevo usuario y el sistema permite que Antonio pueda acceder a su panel de administración.

### 3. Escenario: Recuperar contraseña perdida u olvidada.

- Actores: Antonio, Sistema web y Servidor.
- Descripción: Antonio no recuerda la contraseña de acceso al sistema web.
- Flujo de eventos:
  1. Antonio se dirige al formulario de recuperar contraseña que el sistema web le proporciona.
  2. Antonio introduce su cuenta de correo.
  3. El servidor comprueba que existe un usuario con la cuenta de correo que Antonio ha facilitado.
  4. El sistema web genera una nueva contraseña y se la envía a Antonio por medio de un correo electrónico.
  5. Antonio recibe el correo con su nueva contraseña y puede acceder a su cuenta en nuestro sistema web.

### 4. Escenario: Editar un vídeo.

- Actores: Antonio, Sistema web y Servidor.
- Descripción: Antonio quiere corregir un error en un vídeo que subió al sistema web con anterioridad.
- Flujo de eventos:
  1. Antonio localiza el vídeo que quiere editar en su panel de administración y pulsa sobre el botón editar asociado a ese vídeo.
  2. El sistema web despliega un formulario con los datos asociados al vídeo seleccionado.
  3. Antonio modifica los datos que considere necesarios.
  4. El sistema web comprueba que los datos introducidos son correctos
  5. El servidor actualiza la base de datos para que el vídeo modificado contenga la información

actualizada. Además, el sistema web actualiza la lista de vídeos en el panel de administración para que Antonio pueda visualizar su vídeo modificado.

**5. Escenario:** Eliminar un vídeo.

- Actores: Antonio, Sistema web y Servidor.
- Descripción: Antonio quiere eliminar un vídeo ya que considera que no explica de manera correcta la utilización de tarea que realiza.
- Flujo de eventos:
  1. Antonio localiza el vídeo que desea eliminar en su panel de administración y pulsa sobre el botón borrar.
  2. El servidor realiza la acción de eliminar el registro en la base de datos asociado al vídeo.
  3. El sistema web actualiza la lista de vídeos del panel de administración de Antonio.

**6. Escenario:** Añadir un vídeo.

- Actores: Antonio, Sistema web y Servidor.
- Descripción: Antonio quiere subir un vídeo al sistema web ya que no encuentra un vídeo que se adapte a la tarea que desea realizar.
- Flujo de eventos:
  1. Antonio se dirige al formulario de subida de vídeos que nuestro sistema web le proporciona y rellena todos los campos.
  2. El sistema comprueba que los datos introducidos por Antonio son correctos.
  3. El servidor añade un nuevo registro en la base de datos con la información relacionada con el vídeo que Antonio ha subido.
  4. El sistema actualiza la lista de vídeos pertenecientes a Antonio.

**7. Escenario:** Modificación de datos personales.

- Actores: Antonio, Sistema web y Servidor.

- Descripción: Antonio se ha dado cuenta de que hay errores en sus datos personales. Además, Antonio quiere modificar su contraseña actual por una más intuitiva.
- Flujo de eventos:
  1. Antonio se dirige al formulario de modificación de datos personales.
  2. El sistema web despliega un formulario con los datos del usuario.
  3. Antonio modifica los datos que considere necesarios. A la hora de modificar su contraseña Antonio deberá introducir su contraseña actual, su nueva contraseña y realizar una confirmación de la nueva contraseña.
  4. El sistema web comprueba que los datos introducidos son correctos
  5. El servidor realiza la actualización de los datos en la base de datos.

#### **8. Escenario: Generar clave privada**

- Actores: Antonio, Sistema web y Servidor.
- Descripción: Antonio decide cambiar la privacidad de un vídeo, ya que considera que solo el debería poder usarlo
- Flujo de eventos:
  1. Antonio se dirige a su panel de administración y localiza el vídeo que quiere modificar.
  2. El sistema web despliega un formulario con los datos del vídeo-tutorial
  3. Antonio selecciona la opción de vídeo privado y acepta los cambios.
  4. El sistema web comprueba que los datos introducidos son correctos y genera una clave privada para identificar al vídeo de forma única.
  5. El servidor realiza la actualización de los datos en la base de datos.

6. El sistema actualiza los datos del vídeo en el panel de administración de Antonio.

#### 4.3.2.2. Escenarios de la aplicación móvil

Los escenarios de nuestra aplicación móvil son:

##### 1. **Escenario:** Enlazar un vídeo a una tarjeta NFC.

- Actores: Antonio, Aplicación Móvil y Servidor.
- Descripción: Antonio quiere enlazar un vídeo a una tarjeta NFC.
- Flujo de eventos:
  1. Antonio a través de la aplicación móvil realiza la búsqueda del vídeo que quiere asociar a su tarjeta NFC.
  2. El servidor le envía a la aplicación móvil los datos de los vídeos que mejor se adapten a los criterios de búsqueda de Antonio.
  3. Antonio selecciona el vídeo que mejor se adapte a la tarea que él desea realizar.
  4. La aplicación móvil permite a Antonio poder visualizar el vídeo-tutorial que él ha seleccionado.
  5. Antonio decide asociar el vídeo a su tarjeta NFC y coloca su dispositivo móvil o tablet en contacto con la etiqueta NFC.
  6. La aplicación móvil realiza la escritura en la tarjeta NFC.

##### 2. **Escenario:** Borrar contenido tarjeta NFC.

- Actores: Antonio y Aplicación Móvil.
- Descripción: Antonio quiere borrar el contenido de su tarjeta NFC.
- Flujo de eventos:

1. Antonio coloca su dispositivo móvil o tablet en contacto con la etiqueta NFC.
  2. La aplicación móvil inicia el proceso de borrado.
- 3. Escenario:** Visualizar un vídeo asociado a una tarjeta NFC.
- Actores: Antonio y Aplicación Móvil
  - Descripción: Antonio quiere visualizar el vídeo asociado a su tarjeta NFC.
  - Flujo de eventos:
    1. Antonio pone en contacto la parte trasera de su dispositivo móvil o tablet con su tarjeta NFC.
    2. La aplicación móvil realiza la lectura del contenido de la tarjeta NFC e inicializa la reproducción del vídeo que contiene dicha tarjeta.

## 4.4. Diseño del sistema

Una vez que se han establecido los requisitos del software, el diseño nos permite identificar los objetivos de nuestro sistema.

A continuación, para entender de un modo correcto nuestro diseño, se va a dividir en tres partes, diseño de clases, diseño de datos y finalmente, diseño de la interfaz.

### 4.4.1. Diseño de clases

En este apartado se va a especificar la estructura del sistema. Para ello se va a detallar el diagrama de clases de cada una de las aplicaciones desarrolladas en nuestro proyecto.

Un diagrama de clases es una representación gráfica que permite representar la estructura de un sistema que será implementado utilizando un lenguaje orientados a objetos. Un diagrama de clases está compuesto por los siguientes elementos:

- Clase: Contiene los atributos, métodos y visibilidad. Una clase es la unidad básica que encapsula toda la información de un objeto y

en un diagrama UML es representada por un rectángulo con tres divisiones.

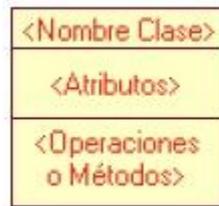


Figura 4.13.- Ejemplo de clase en un diagrama UML.

- Relaciones: Herencia, composición, agregación y uso.

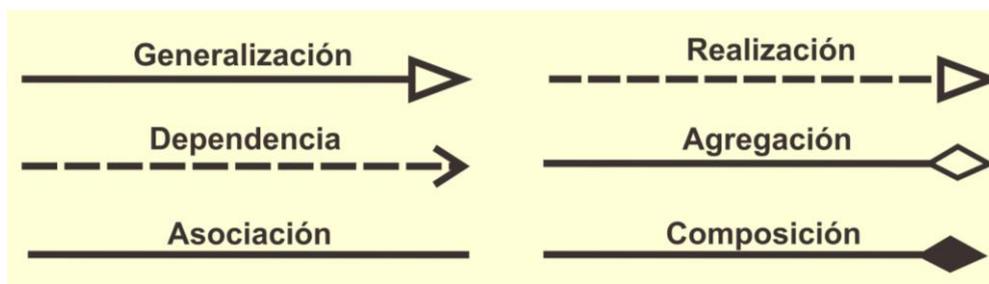


Figura 4.14.- Representación de los tipos de relaciones en un diagrama UML.

Antes de realizar el diseño de clases de nuestro sistema, hay que tener en cuenta que nuestro sistema web utiliza el diseño MVC (modelo-vista-controlador). El objetivo de este esquema es conseguir el mayor grado de independencia entre el modelo de la vista y el controlador.

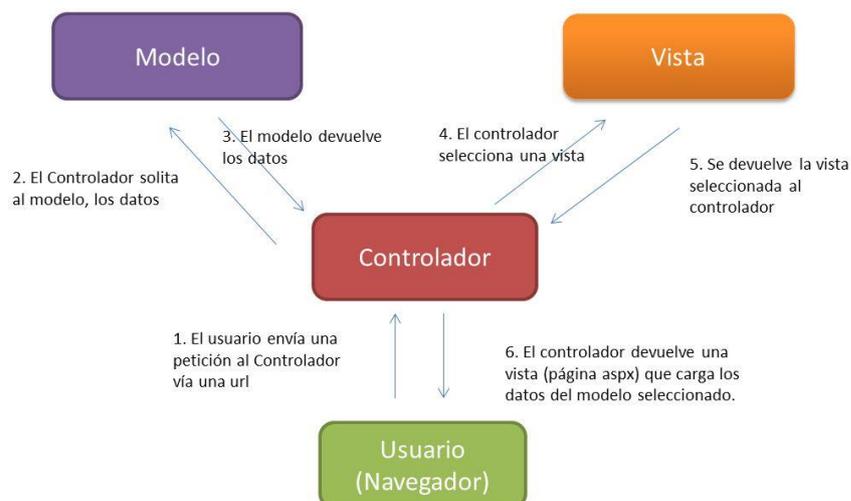


Figura 4.15.- Esquema del diseño modelo-vista-controlador.

En este esquema el modelo contendrá todos los datos contenido en el servidor, las vistas serán las encargadas de mostrarle los datos al cliente y el controlador será el encargado de realizar la comunicación entre la vista y el modelo. El controlador es el elemento fundamental de este esquema, ya que es el encargado de realizar el control del sistema.

A continuación, vamos a detallar el diseño de clases de nuestro sistema web. Debido a que contamos con un gran número de clases, vamos a diferenciar cada diagrama según el controlador implicado:

- **Controlador Login:**

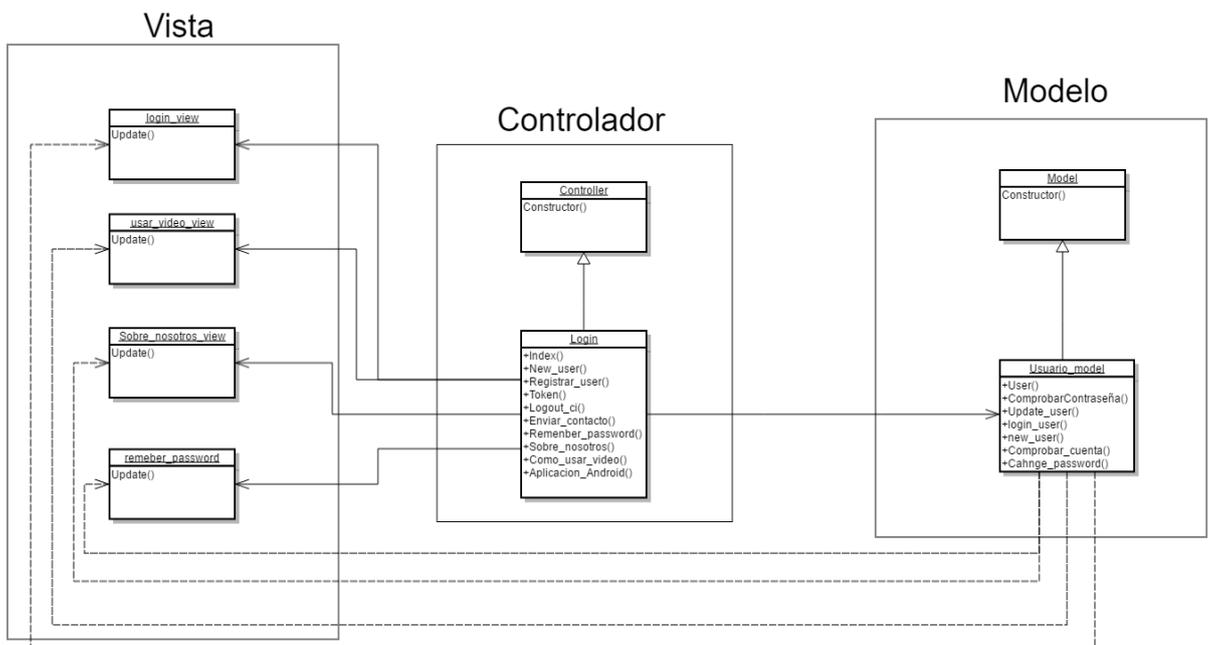


Figura 4.16.- Diagrama UML controlador: Login

- **Controlador Admin:**

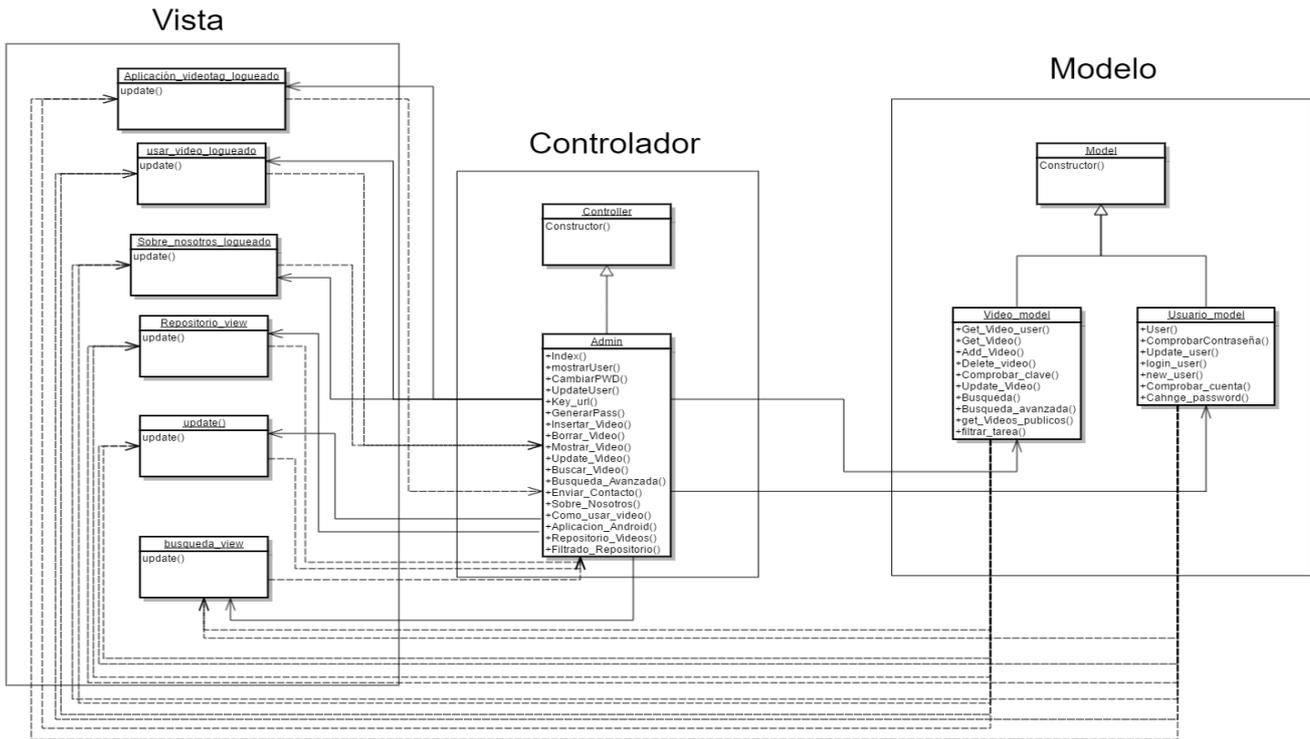


Figura 4.17.- Diagrama UML controlador: Admin

- **Controlador Api:**

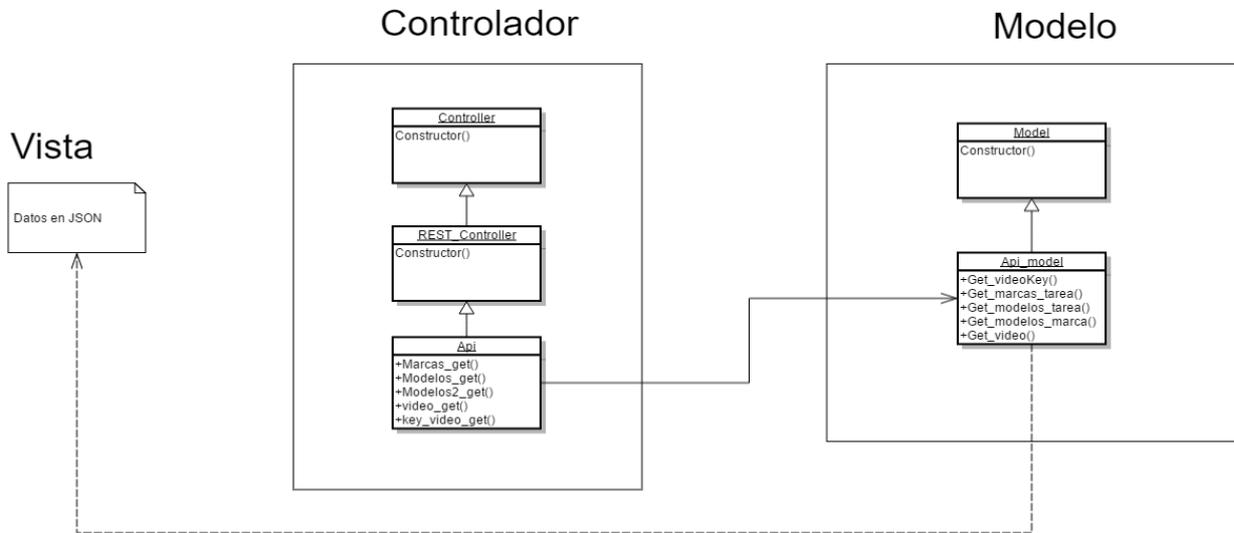


Figura 4.18.- Diagrama UML controlador: API

Para finalizar, se muestra el diseño de clases de nuestra aplicación móvil:

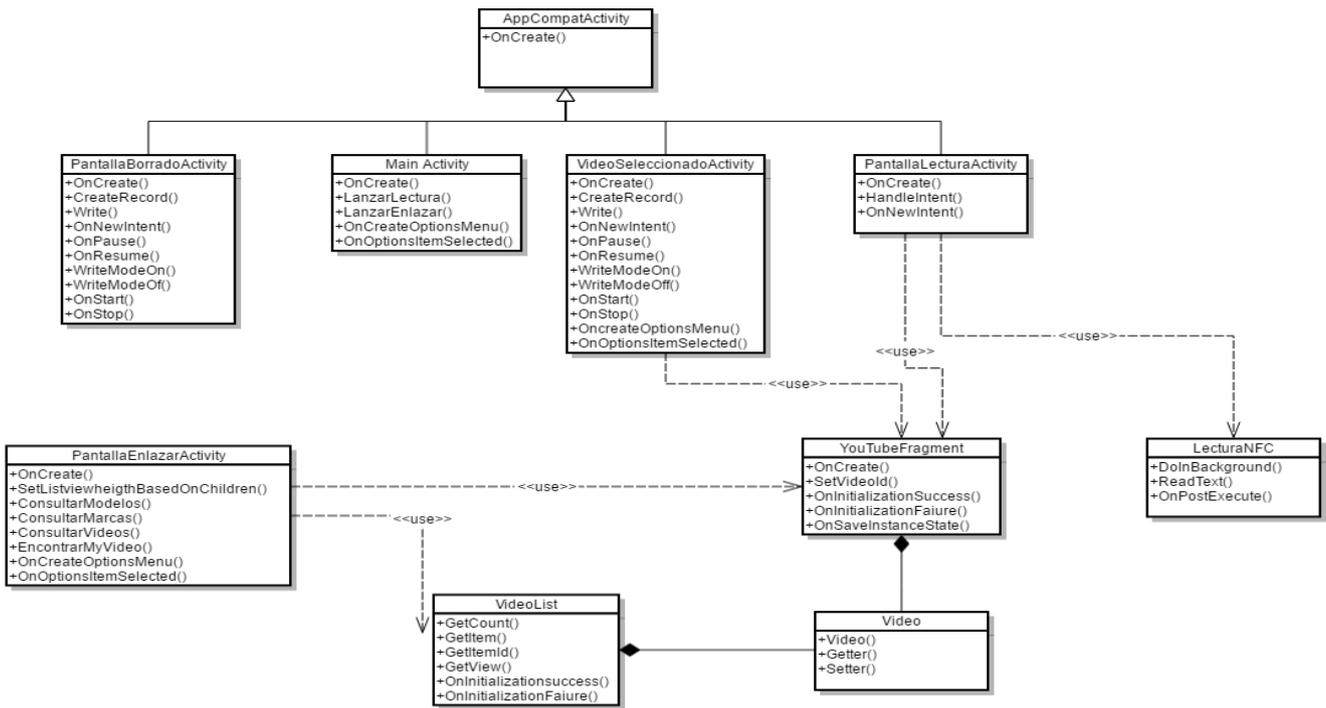


Figura 4.19.- Diagrama UML de la aplicación móvil

#### 4.4.2. Diseño de datos

En esta fase del diseño software se va a detallar como se encuentra estructurada la información existente en nuestro sistema. Nuestro sistema está formado por los siguientes elementos:

- **Videos:** De cada vídeo conoceremos el tipo de tarea que realizan, la marca y modelo del aparato electrónico con el que se realiza la tarea, la URL del vídeo en YouTube, el identificador del usuario al que pertenece el vídeo y un identificador para identificar forma única cada vídeo de nuestro sistema.
- **Usuarios:** De cada usuario conoceremos su nombre, apellidos, correo y contraseña (se almacenará cifrada).

Una vez hemos especificado los elementos principales por los que está formado nuestro sistema, vamos a representar en forma de tablas la base de

datos asociada a este conjunto de información. Para ello, se debe realizar en primer lugar el diseño conceptual de la base de datos, para después obtener las tablas requeridas. Para realizar el diseño conceptual de la base de datos vamos a utilizar el modelo Entidad-Relación (ER).

Un diagrama Entidad-Relación es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades. Este tipo de diagramas está formado por un conjunto de elementos:

- **Entidad:** Las entidades son el fundamento del modelo entidad relación.
- **Atributo:** Se representan mediante un círculo o elipse etiquetado mediante un nombre en su interior.
- **Relaciones:** Se representan mediante un rombo etiquetado en su interior con un verbo.



Figura 4.20.- Elementos de un diagrama entidad-relación

Una vez introducido el diagrama entidad relación, procedemos a realizar el esquema conceptual de nuestra base de datos:

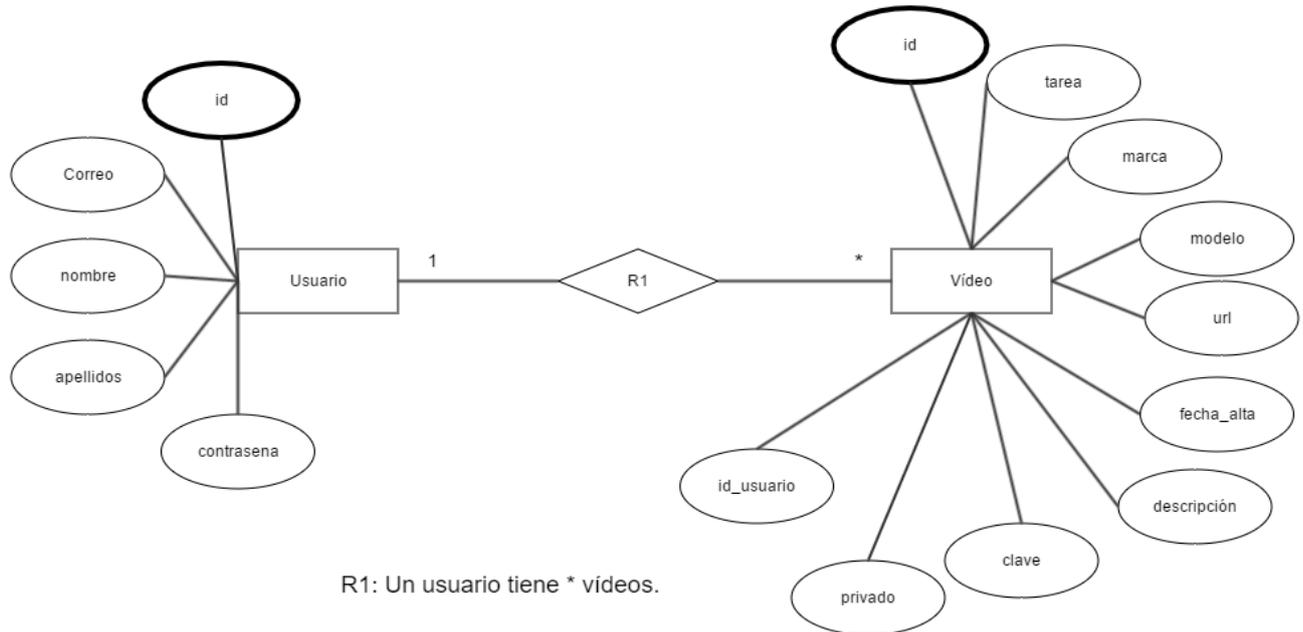


Figura 4.21.- Esquema conceptual de la base de datos.

El siguiente paso sería obtener a partir del esquema conceptual el esquema conceptual modificado. Para ello se deben realizar los siguientes cambios sobre el esquema conceptual:

- Eliminar las posibles entidades débiles.
- Eliminar las posibles relaciones de muchos a muchos.
- Eliminar todas las relaciones con atributos existentes en el esquema conceptual.

Como podemos observar, en nuestro caso no sería necesario realizar ningunas de las correcciones que se han detallado anteriormente, por lo que podemos decir que nuestro esquema conceptual coincide con el esquema conceptual modificado.

Basándonos en el esquema de nuestra base de datos, podemos definir dos tablas con una serie de atributos. A continuación, se detallan las tablas que componen nuestro sistema:

- **Usuario:** Esta tabla está formada por los siguientes atributos:

- **Id:** Clave primaria. Es un identificador único que permite identificar cada usuario de nuestro sistema. Es un entero de 11 caracteres y es de tipo auto incremental.
- **Correo:** Contienen el correo del usuario. Es una cadena de 25 caracteres y tiene una restricción de no nulo.
- **Nombre:** Contiene el nombre del usuario. Es una cadena de 25 caracteres.
- **Apellidos:** Contiene los apellidos del usuario. Es una cadena de 25 caracteres.
- **Contraseña:** Contiene la contraseña del usuario. Este campo por seguridad contendrá los datos cifrados y es una cadena de 100 caracteres.
- **Vídeo:** Esta tabla está formada por los siguientes atributos:
  - **Id:** Clave primaria. Es un identificador único que permite identificar cada vídeo de nuestro sistema. Es un entero de 11 caracteres y es de tipo auto incremental.
  - **Tarea:** Contiene el tipo de tarea que realizar el vídeo. Es una cadena de 50 caracteres y tiene una restricción de no nulo.
  - **Marca:** Contiene la marca del aparato electrónico con el que se ha realizado la tarea que resuelve el vídeo. Es una cadena de 25 caracteres y tiene restricción de no nulo.
  - **Modelo:** Contiene el modelo del aparato electrónico con el que se ha realizado la tarea que resuelve el vídeo. Es una cadena de 25 caracteres y tiene restricción de no nulo.
  - **Fecha\_alta:** Contiene la fecha de registro del vídeo.
  - **URL:** Contiene la URL del vídeo en YouTube. Es una cadena de 1024 caracteres y tiene una restricción de no nulo.
  - **Descripción:** Contiene una descripción del contenido del vídeo. Es una cadena de 1000 caracteres.
  - **Privado:** Es un entero de 1 caracteres y permite determinar si el vídeo es privado o público (1 privado, 0 público). Este campo tiene una restricción de no nulo.

- **Clave:** Contiene una clave privada. Este campo solo estará relleno en los casos en que el vídeo sea privado. Es una cadena de 25 caracteres.
- **Id\_usuario:** Es una clave foránea. Contiene el identificador del usuario que es propietario del vídeo. Es una cadena de 25 caracteres y tiene una restricción de no nulo.

### *4.4.3. Diseño de la interfaz*

El diseño de la interfaz de usuario es una tarea que tiene una gran importancia en el desarrollo de un sistema. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso.

Para realizar el diseño de la interfaz, debemos tener presente cuales son los puntos de nuestro sistema en los que el usuario interactúa: metáforas, estilo y mensajes.

#### **4.4.3.1. Metáforas**

Se trata de la aplicación de un concepto o de una expresión sobre una idea o un objeto al cual no describe de manera directa, con la intención de sugerir una comparación con otro elemento y facilitar su comprensión.

La mayoría de las metáforas que se han utilizado en el sistema web y aplicación móvil son representadas en forma de iconos que permiten hacer la interfaz más sencilla e intuitiva.

Entre las metáforas más importantes podemos destacar:

- **Sistema web:**
  - **Logo del sistema:**



Figura 4.22.- Metáfora icono de logo del sistema web.

Esta metáfora permite que el usuario pueda reconocer el sistema web y que pueda dirigirse a la pantalla principal de este sistema.

- **Mi cuenta:**



Figura 4.23.- Metáfora icono de mi cuenta

Esta metáfora permite que el usuario pueda acceder a sus datos personales y poder así modificarlos.

- **Aplicación Móvil:**

- **Ayuda:**



Figura 4.24.- Metáfora de icono de ayuda.

Esta metáfora aparece en el navbar de nuestra aplicación móvil y permite que el usuario pueda obtener información de ayuda.

- **Información:**



Figura 4.25.- Metáfora de icono de información.

Esta metáfora permite que el usuario pueda conocer información acerca de nuestra aplicación.

○ **Estado de procesos de escritura y borrado:**



Figura 4.26.- Metáforas de iconos para estado del proceso de escritura en NFC.

Este conjunto de metáforas permite que el usuario pueda entender en qué estado se encuentra el proceso de escritura en una tarjeta NFC.

#### 4.4.3.2. Estilo

En este apartado se va a especificar una guía de estilo para que todos los elementos de nuestra interfaz tengan coherencia y permita disponer de una armonía entre las distintas aplicaciones desarrolladas en este proyecto.

Hay que tener en cuenta que una guía de estilo recoge normativas y patrones básicos relacionados con el aspecto de una interfaz para su aplicación en el desarrollo de nuevas pantallas dentro de un entorno concreto.

A continuación, se detallan las reglas establecidas por nuestra guía de estilo.

- **Tipo de letra:**
  - **Sistema web:**
    - **Tamaño de letra:** 14px.
    - **Tipo de fuente:** Helvetica Neue.
  - **Aplicación móvil:**
    - **Tamaño de letra:** 20dp.
    - **Tipo de fuente:** Sans.
  
- **Colores:** Para el desarrollo de nuestro sistema se ha empleado una paleta de colores. De este modo nos aseguramos de que la interfaz de nuestra aplicación móvil y sistema web tienen armonía.

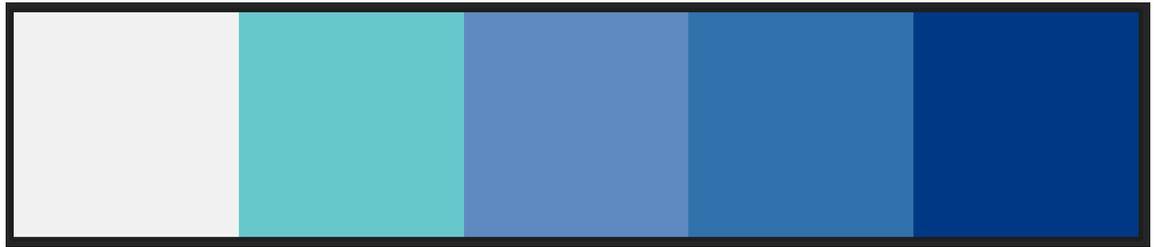


Figura 4.27.- Paleta de colores.

- **Iconos:** Los iconos utilizados en el desarrollo de nuestro sistema web son los proporcionados por el framework de diseño Bootstrap [17].

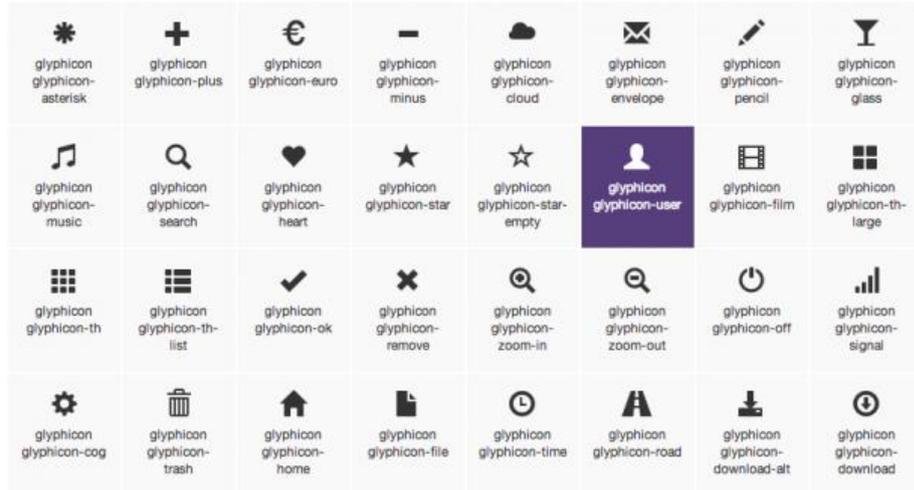


Figura 4.28.- Icono de Bootstrap.

Por otro lado, los iconos utilizados para el desarrollo de nuestra aplicación móvil provienen de Google Desing, en concreto del paquete de iconos Icon Font [18].



Figura 4.29.- Iconos de Google Desing.

- **Botones:** Para nuestro sistema web los botones utilizados provienen del framework de desarrollo Bootstrap [19].

## Botones en Bootstrap



Figura 4.30.- Botones de Bootstrap.

Sin embargo, para la aplicación móvil los botones utilizados son los proporcionados por la herramienta de desarrollo Android Studio.

#### 4.4.3.3. Mensajes

El uso de mensajes es importante para poder comunicarle al usuario los errores que puedan ocurrir durante la utilización de nuestro sistema web. El diseño de los mensajes de error es una tarea fundamental ya que un sistema con mensajes de error mal diseñados puede suponer el rechazo de nuestro sistema por parte del usuario.

Los mensajes deben de ser educados, concisos, consistentes y constructivos. En nuestro sistema web hemos usado el estilo de mensajes que nos proporciona el framework de diseño Bootstrap [19].



Figura 4.31.- Ejemplo de mensajes en Bootstrap.

En concreto hemos usado el estilo “alert-success” (primer mensaje de la figura 4.21) para los mensajes de confirmación de que el proceso realizado por el usuario se ha complementado de forma correcta. Por otro lado, para los mensajes de error, se ha usado el estilo de mensajes “alert-danger” (último mensaje de la figura 4.21).

#### 4.4.3.4. Storyboard de la aplicación web

Un Storyboard nos permite indicar los enlaces a diferentes páginas a partir de los resultados de las iteraciones del usuario. Además, nos permite ilustrar cómo será el comportamiento de nuestro sistema web.

Estas figuras representan una secuencia similar a como se comportaría el sistema web con la iteración del usuario. Hay que tener en cuenta que cada figura contendrá un círculo con un número para indicar donde continuará la iteración del usuario.

En primer lugar, vamos a mostrar la secuencia relacionada con las opciones que el usuario puede realizar en la pantalla principal de nuestra aplicación web.

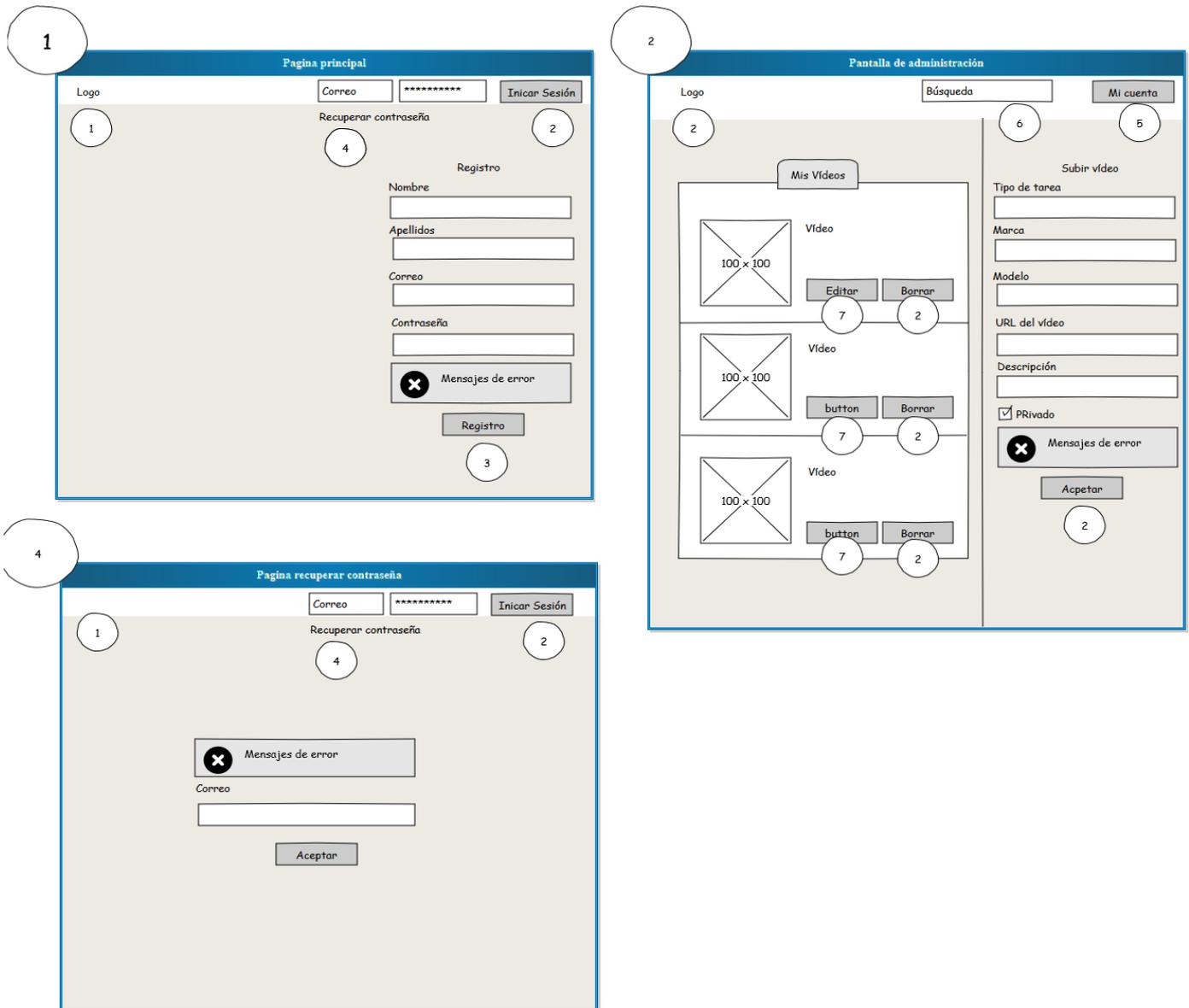


Figura 4.32.- Storyboard pantalla principal del sistema web.

En segundo lugar, una vez que el usuario se encuentra en su panel de administración, vamos a ver cuáles serían las secuencias más importantes que tendría la administración del sistema.

- **Gestión de vídeos:**

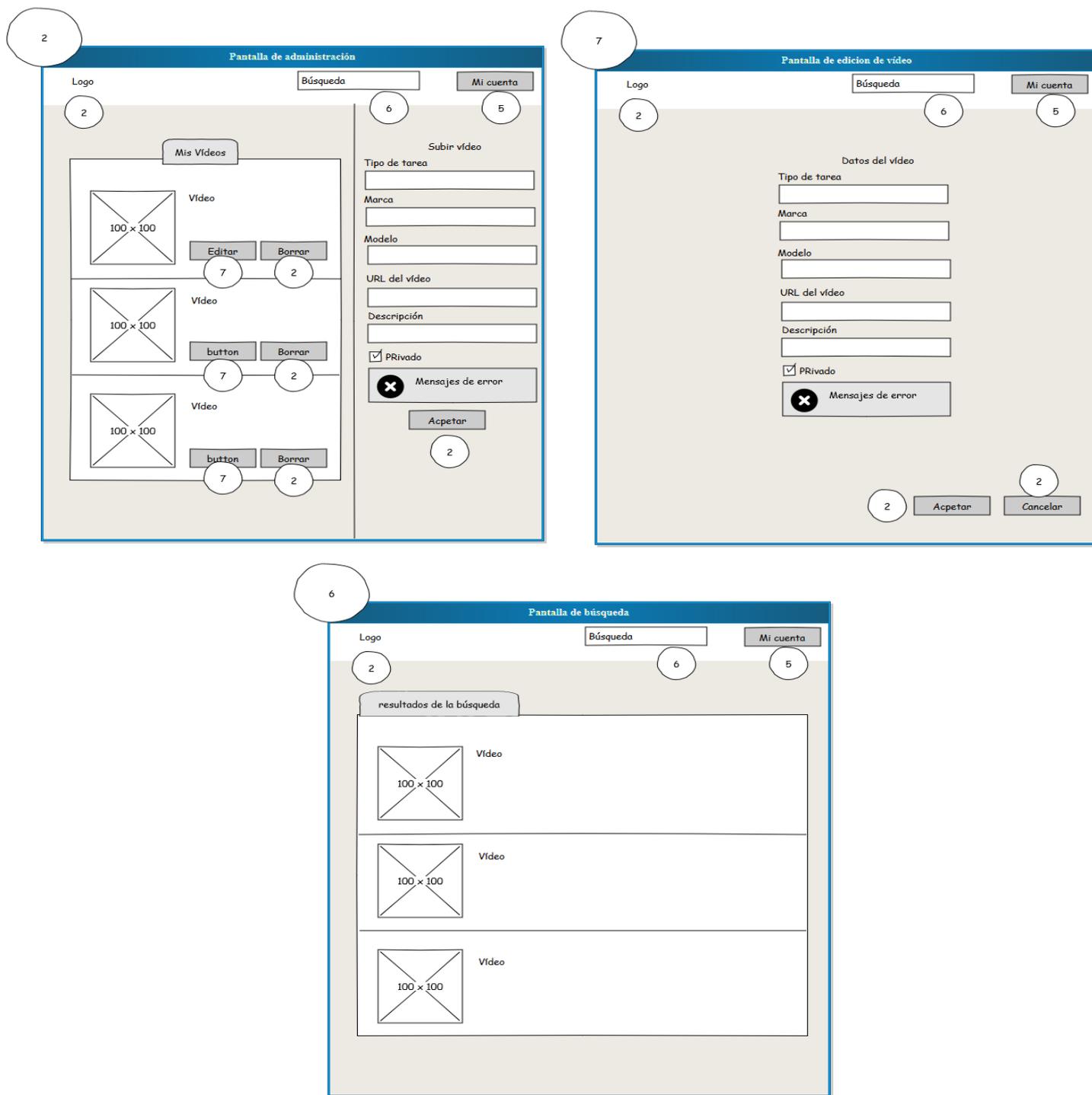


Figura 4.33.- Storyboard Gestión de Vídeos

- **Modificación de datos personales:**

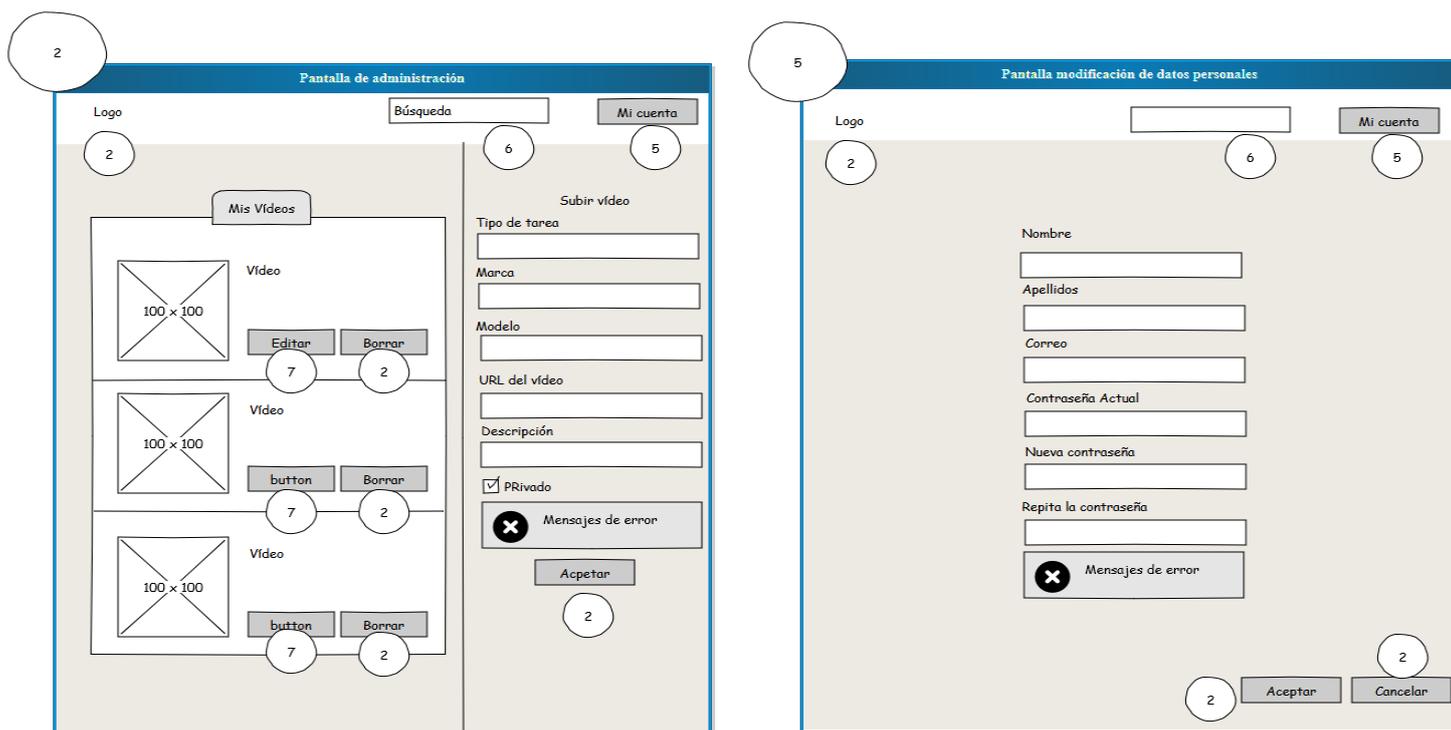


Figura 4.34.- Storyboard Modificación de datos personales

#### 4.4.3.5. Storyboard de la aplicación móvil

En las siguientes figuras se muestra las secuencias más importantes que el usuario podrá realizar con la aplicación móvil.

- **Enlazar vídeo:**



Figura 4.35.- Storyboard enlazar vídeo con tarjeta NFC.

- **Visualizar vídeo asociado a tarjeta NFC:**

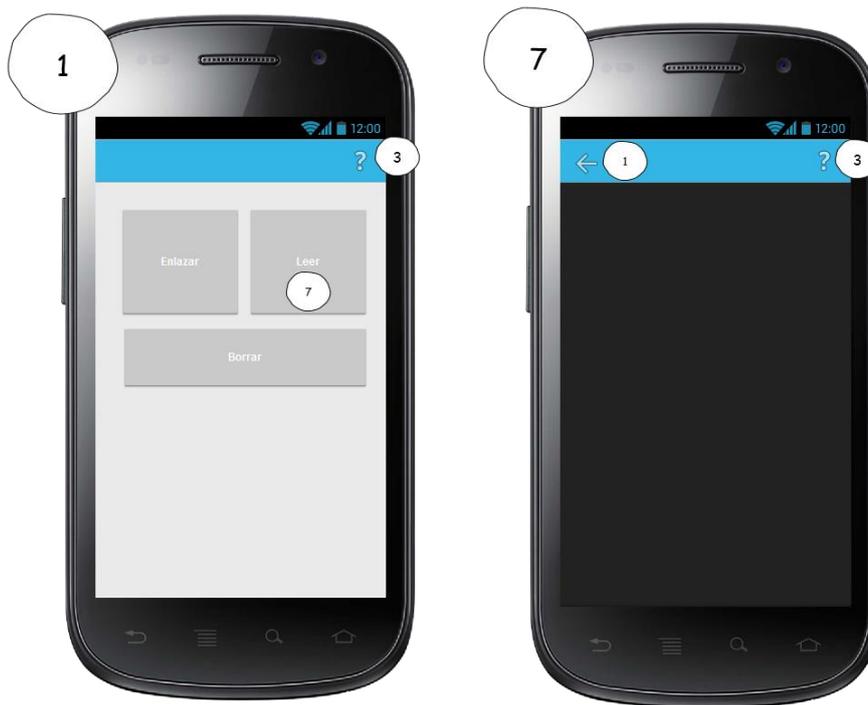


Figura 4.36.- Storyboard Visualizar vídeo.

- **Borrar contenido de tarjeta NFC:**

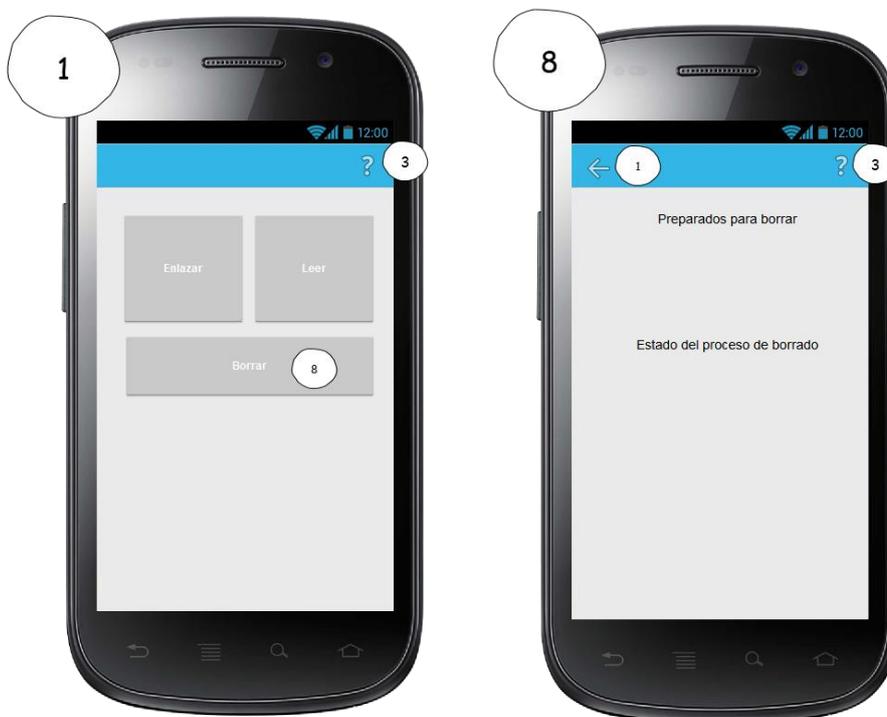


Figura 4.37.- Storyboard borrar contenido tarjeta NFC.

## 4.5. Implementación

La implementación es la etapa final de nuestro proceso de ingeniería Software. En esta etapa se traduce todos los modelos que hemos diseñado anteriormente a código fuente. La implementación es una etapa que debe ser desarrollada de manera correcta, ya que un mal desarrollo puede suponer aumentar los costes y tiempos de nuestro proyecto.

### 4.5.1. Arquitectura del sistema

En nuestro proyecto hemos desarrollado un sistema web de gestión de vídeos tutoriales, tal y como hemos mencionado anteriormente, este sistema está basado en el patrón de arquitectura Modelo-Vista-Controlador. Este patrón nos permite separar los datos y la lógica de nuestro sistema de la interfaz de usuario.

El funcionamiento de este tipo de arquitectura es sencillo ya que nuestro sistema se encuentra almacenado en un servidor, los usuarios acceden a nuestro sistema a través de un software cliente (navegador y aplicación móvil). Además, el sistema ofrecerá un servicio REST para que los usuarios a través de la aplicación móvil puedan realizar peticiones al sistema.

La arquitectura del sistema se puede resumir en la siguiente figura:

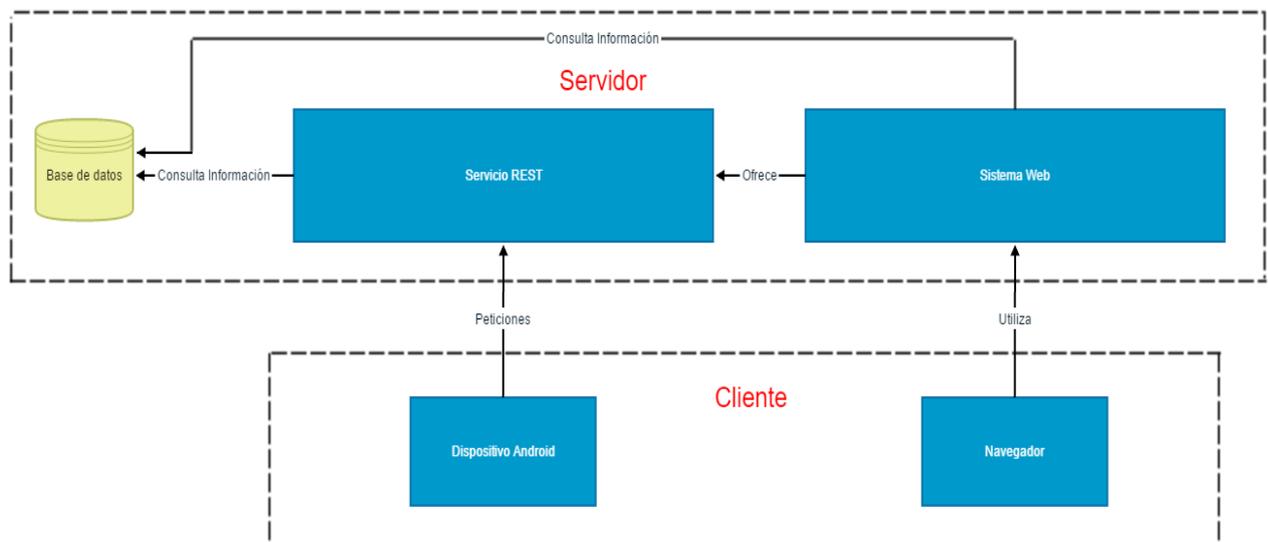


Figura 4.38.- Arquitectura del sistema

### 4.5.2. Lenguajes de programación utilizados

Los lenguajes de programación que hemos utilizado para el desarrollo de nuestro proyecto son:

- **Sistema web:**

- **PHP:** Es lenguaje de código abierto muy popular, adecuado para el desarrollo web y que puede ser incrustado en HTML [20]. La popularidad de este lenguaje se debe a que es utilizado en un gran número de páginas y portales web. Se trata de un lenguaje interpretado que se ejecuta en el servidor y genera contenido dinámico a petición del cliente.

Hemos decidido optar por este lenguaje debido a las grandes ventajas que posee frente a otros lenguajes que realizan funciones parecidas, algunas de las ventajas son:

1. Es un lenguaje multiplataforma.
2. Permite la conexión con la mayoría de manejadores de bases de datos, en especial con MySQL (utilizada para el desarrollo de este proyecto).
3. Posee de una amplia documentación, ejemplos, manuales...etc.
4. Permite un estilo de programación Orientado a objetos.

- **HTML:** Es un lenguaje que pertenece a la familia de lenguajes de marcado y es muy utilizado para la elaboración de páginas web [21]. El estándar HTML lo define la W3C (World Wide Web Consortium) y actualmente se encuentra en su versión HTML5.

El lenguaje HTML es usado en conjunto con otros lenguajes de programación para poder crear páginas web dinámicas.

- **CSS:** Es un lenguaje de estilo que define la presentación de los documentos HTML [22]. Este lenguaje permite que podamos incluir márgenes, tipo de letra, fondos...etc.

- **Javascript:** Es un lenguaje de programación que a diferencia del lenguaje PHP se utiliza en el lado del cliente, es decir, se ejecuta en el ordenador del usuario no en el servidor, permitiendo crear efectos atractivos y dinámicos en las páginas web [23].
- **Aplicación Android:**
  - **Java:** Es un lenguaje orientado a objetos cuyo potencial reside en la compilación de código intermedio o bytecode de las aplicaciones desarrolladas haciendo uso de este lenguaje [24]. Esta característica permite que cualquier aplicación desarrollada en este lenguaje de programación pueda ejecutarse en múltiples plataformas a través de una máquina virtual.  
  
Java es el lenguaje utilizado por nuestra herramienta de desarrollo Android Studio para la creación de nuestra aplicación móvil.
  - **XML:** Es un lenguaje de etiquetas, es decir, cada paquete de información se encuentra delimitado por etiquetas como se hace también en el lenguaje HTML, pero XML separa el contenido de la presentación. Android Studio utiliza este lenguaje para el desarrollo de sus interfaces, aunque también se permite el desarrollo de sus interfaces de una forma gráfica.

### *4.5.3. Herramientas de desarrollo*

Una vez que hemos detallado en el apartado anterior los lenguajes de programación empleados para el desarrollo de nuestro sistema, pasamos a especificar las herramientas empleadas para el desarrollo de cada una de las aplicaciones desarrolladas en este proyecto.

Para el desarrollo del sistema web hemos empleado las siguientes herramientas:

- **CodeIgniter:** Es un framework que permite el desarrollo de aplicaciones en PHP y que utiliza el esquema MVC (modelo-vista-controlador) [25]. Este framework permite a los desarrolladores web mejorar su forma de trabajar y hacerlo a una mayor velocidad.

CodeIgniter es un framework de código libre que contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación.

Algunos de los puntos más interesantes de este framework son:

- **Versatilidad:** Es capaz de trabajar con la mayoría de entornos o servidores, incluso en sistemas de alojamiento compartido.
- **Compatibilidad:** Este framework es compatible con la última versión de PHP disponible, lo que permite que se pueda ejecutar en cualquier servidor, incluso en los antiguos.
- **Facilidad de instalación:** No se necesita na más que una cuenta FTP para subir CodeIgniter al servidor y su configuración se realiza con apenas la edición de un archivo.
- **Documentación:** Posee de una gran documentación que es sencilla de seguir y de asimilar.



Figura 4.39.- CodeIgniter

- **NetBeans:** Es un entorno de desarrollo integrado libre, está diseñado principalmente para el lenguaje de programación Java, pero permite el desarrollo en una gran variedad de lenguajes de

programación como puede ser PHP. Además, esta herramienta permite la integración de módulos, lo que le permite que se pueda extender [26].

Esta herramienta ha sido utilizada conjuntamente con el framework CodeIgniter para el desarrollo de nuestro sistema web. La herramienta NetBeans nos ha permitido poder desarrollar nuestro código en PHP de una forma estructurada y sencilla.

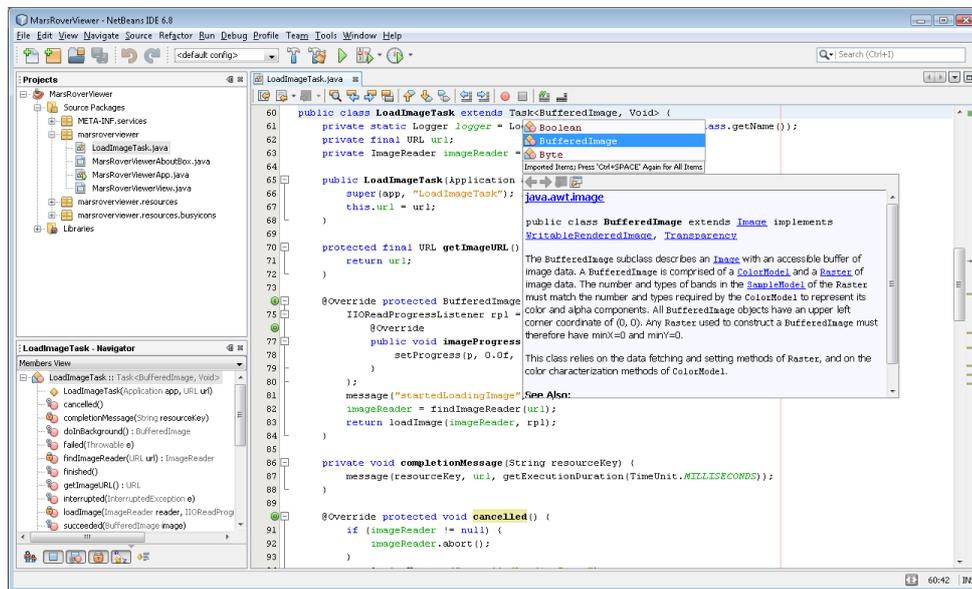


Figura 4.40.- Herramienta NetBeans

- **phpMyAdmin:** Es una herramienta escrita en PHP que permite manejar la administración de MySQL a través de las páginas web. Esta herramienta tiene un uso muy extendido, está disponible en 50 idiomas y es distribuida bajo la licencia GPL [27].

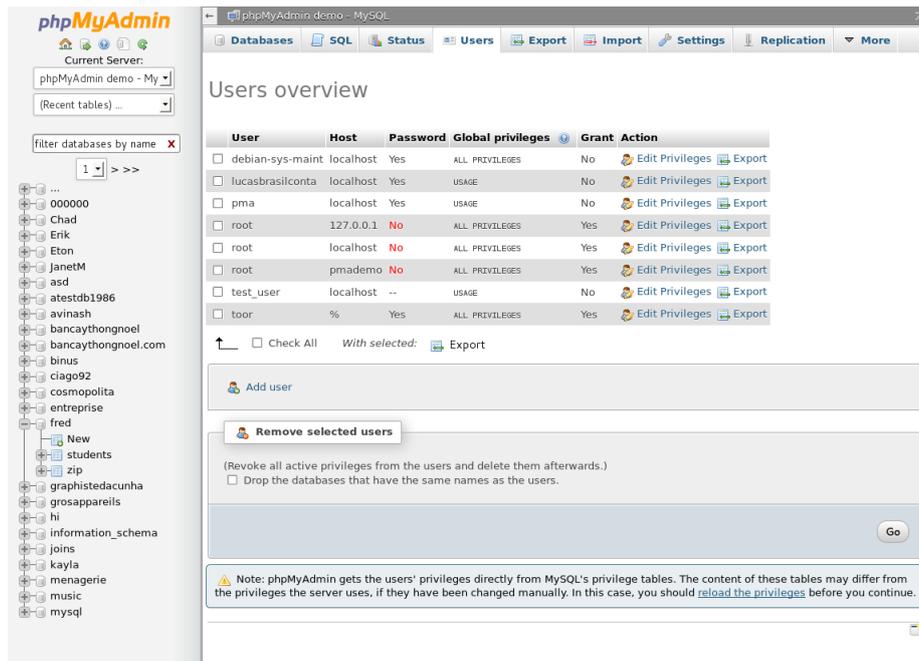


Figura 4.41.- Herramienta phpMyAdmin

Por otro lado, para el desarrollo de nuestra aplicación Android, hemos usado la herramienta de desarrollo Android Studio [28]. Esta herramienta constituye un entorno de desarrollo integrado para la plataforma Android y está basado en IntelliJ IDEA, un IDE para Java de JetBrains. Android Studio permite que el desarrollador pueda ver en directo los cambios de diseño de las aplicaciones en las diferentes resoluciones que soporta Android. Además, nos muestra información útil en el margen de la pantalla de código, como iconos o colores según los mencionemos en el código. Es por todo esto, lo que convierte a esta herramienta en un entorno muy adecuado para el desarrollo de aplicaciones Android.

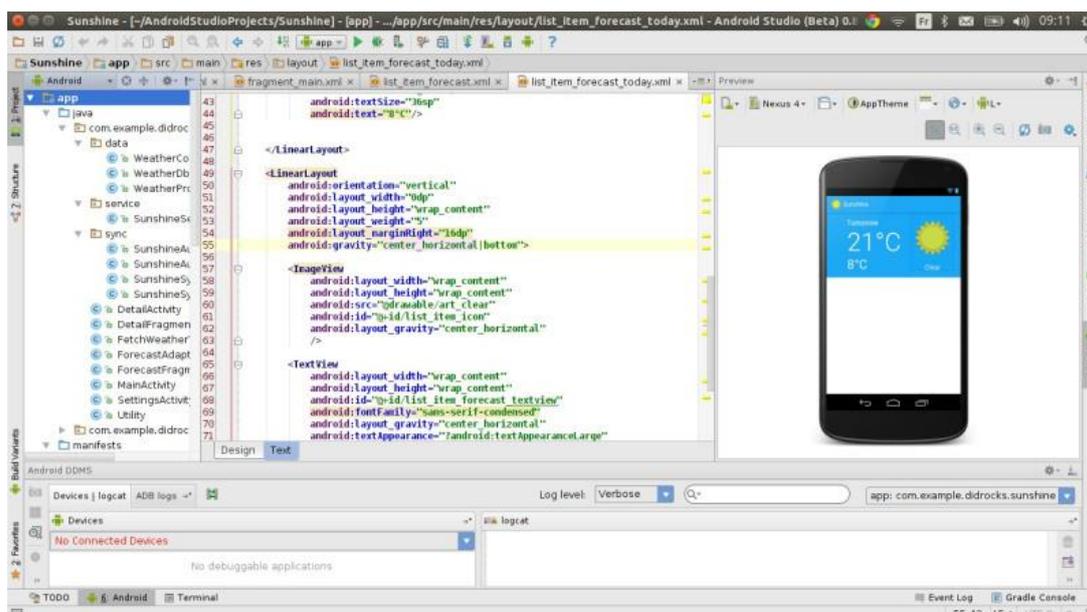


Figura 4.42.- Herramienta Android Studio

Para finalizar, cabe destacar que, para el desarrollo de las aplicaciones de nuestro sistema, hemos usado la herramienta de control de versiones SmartGit. El uso de una herramienta de este tipo nos aporta una serie de ventajas a la hora de realizar el proceso de desarrollo:

- Registro histórico de las acciones realizadas con cada uno de los elementos que forma nuestro proyecto.
- Posibilidad de restaurar nuestro proyecto a un estado anterior.
- Permite la resolución de conflictos.

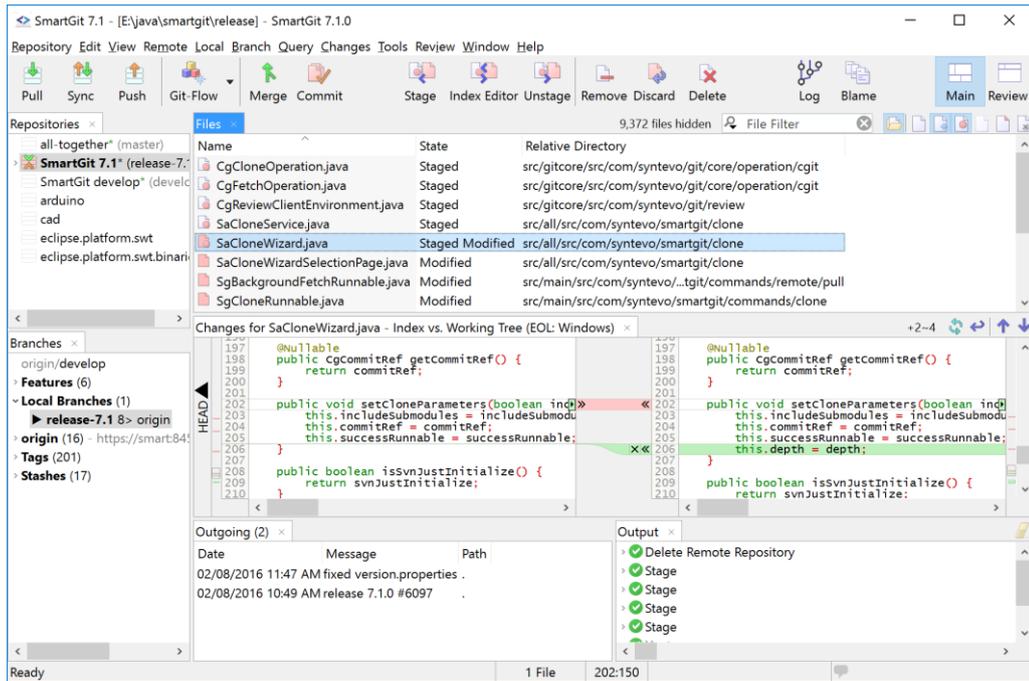


Figura 4.43.- Herramienta SmartGit

#### 4.5.4. Servicios REST implementados

Como se ha introducido anteriormente, nuestro sistema web ofrece una serie de servicios REST de los que hace uso nuestra aplicación móvil. En la tabla 4.1 se muestra una descripción de los servicios implementados:

URL	Método	Parámetros	Descripción
/marcas	GET	tarea	Obtiene todas las marcas de aparatos electrónicos disponibles para una tarea específica
/modelos	GET	tarea	Obtiene todos los modelos de aparatos electrónicos disponibles para una tarea específica
/modelos2	GET	tarea, marca	Obtiene los modelos de

			aparatos electrónicos disponibles para una tarea y marca específica.
/video	GET	tarea, marca, modelo	Obtiene los datos de los vídeos que cumplan con los criterios de búsqueda del usuario (tarea, marca o modelo)
/Key_video	GET	key	Obtiene los datos de un vídeo privado por medio de su clave privada

Tabla 4.1.- Servicios REST Implementados

La implementación de estos servicios se ha realizado haciendo uso de la librería REST\_Controller. Esta librería es de código abierto y nos ofrece una gran variedad de opciones y posibilidades como son:

- Implementación de una tabla de registro de las consultas realizadas por los clientes.
- Gestión de la seguridad, incorporando la posibilidad de restringir el uso de un servicio web a un cliente si el sistema detecta que en poco tiempo está haciendo uso repetido del mismo. Además, permite el uso de Key para usar los servicios web.
- Permite mostrar los datos devueltos por nuestro servicio web en una gran variedad de formatos (XML, HTML, CSV, JSON...etc.)

## 4.6. Pruebas

Con esta etapa cerramos el proceso de ingería del software desarrollado en este proyecto. Las pruebas permiten comprobar que el software desarrollado responda y realice correctamente las tareas indicadas en la especificación.

Esta etapa es de un gran interés ya que nos permite evaluar la calidad de nuestro software y si es necesario poder mejorarlo. Las pruebas realizadas en este apartado van a estar estructuradas en dos grupos. El primer grupo detallara

las pruebas realizadas a nuestro sistema web basándonos en los requisitos funcionales descritos en el apartado 4.2.1.1 y el segundo grupo nos mostrara las pruebas realizadas a nuestra aplicación Android basándonos también en los requisitos funcionales descritos en el apartado 4.2.2.1.

- **Casos probados en el sistema web:**

- Caso:1.- Login del usuario correcto.

<i>Condiciones</i>	<i>El usuario debe estar registrado en el sistema</i>
<i>Acción</i>	<i>El usuario introduce sus credenciales de acceso</i>
<i>Control</i>	<i>El sistema valida los datos y redirige al usuario a su panel de administración</i>

Tabla 4.2.- Pruebas. Caso: Login del usuario (correcto)

- Caso:2.- Login del usuario incorrecto.

<i>Condiciones</i>	<i>El usuario debe estar registrado en el sistema</i>
<i>Acción</i>	<i>El usuario introduce sus credenciales de acceso</i>
<i>Control</i>	<i>El sistema valida los datos y muestra un mensaje de error informando al usuario de que sus datos de acceso no son correctos</i>

Tabla 4.3.-Pruebas. Caso: Login del usuario (incorrecto)

- Caso:3.- Registro del usuario correcto

<i>Condiciones</i>	<i>El usuario no debe estar registrado en el sistema</i>
<i>Acción</i>	<i>El usuario introduce sus datos en el formulario de registro</i>
<i>Control</i>	<i>El sistema valida los datos y redirige al usuario a su panel de administración</i>

Tabla 4.4.-Pruebas. Caso: Registro del usuario (correcto)

- Caso:4.- Registro del usuario incorrecto

<i>Condiciones</i>	<i>El usuario no debe estar registrado en el sistema</i>
<i>Acción</i>	<i>El usuario introduce sus datos en el formulario de registro</i>
<i>Control</i>	<i>El sistema valida los datos y muestra un mensaje de error informando al usuario de que sus datos de registro no son correctos</i>

Tabla 4.5.-Pruebas. Caso: Registro del usuario (incorrecto)

- Caso:5.- Gestión de Vídeos: Editar (correcto)

<i>Condiciones</i>	<i>El usuario debe estar identificado en el sistema</i>
<i>Acción</i>	<i>El usuario edita los campos que considere necesarios de un vídeo de su panel de administración</i>
<i>Control</i>	<i>El sistema valida los datos y muestra en el panel de administración del usuario el vídeo con los datos modificados.</i>

Tabla 4.6.- Pruebas. Caso: Gestión de vídeos: Editar (correcto)

- Caso:6.- Gestión de Vídeos: Editar (incorrecto)

<i>Condiciones</i>	<i>El usuario debe estar identificado en el sistema</i>
<i>Acción</i>	El usuario edita los campos que considere necesarios de un vídeo de su panel de administración
<i>Control</i>	El sistema valida los datos y muestra un error indicando al usuario que los datos modificados no son correctos.

Tabla 4.7.- Pruebas. Caso: Gestión de vídeos: Editar (incorrecto)

- Caso: 7.- Gestión de Vídeos: Borrar(correcto)

<i>Condiciones</i>	<i>El usuario debe estar identificado en el sistema</i>
<i>Acción</i>	El usuario elimina un vídeo de su panel de administración
<i>Control</i>	El sistema realiza la acción y actualiza el panel de administración del usuario

Tabla 4.8.- Pruebas. Caso: Gestión de vídeos: Borrar (correcto)

- Caso: 8.- Gestión de Vídeos: Borrar(incorrecto)

<i>Condiciones</i>	<i>El usuario debe estar identificado en el sistema</i>
<i>Acción</i>	El usuario elimina un vídeo de su panel de administración
<i>Control</i>	El sistema muestra un mensaje de error al usuario, indicando así que la acción no se ha podido realizar de forma correcta

Tabla 4.9.- Pruebas. Caso: Gestión de vídeos: Borrar (incorrecto)

- Caso: 9.- Gestión de Vídeos: Añadir(correcto)

<i>Condiciones</i>	<i>El usuario debe estar identificado en el sistema</i>
<i>Acción</i>	El usuario rellena los campos del formulario de añadir un nuevo vídeo al sistema.
<i>Control</i>	El sistema valida los datos y actualiza la lista de vídeos del panel de administración del usuario

Tabla 4.10.- Pruebas. Caso: Gestión de vídeos: Añadir (correcto)

- Caso: 10.- Gestión de Vídeos: Añadir(incorrecto)

<i>Condiciones</i>	<i>El usuario debe estar identificado en el sistema</i>
<i>Acción</i>	El usuario rellena los campos del formulario de añadir un nuevo vídeo al sistema.
<i>Control</i>	El sistema valida los datos y muestra un mensaje de error indicando al usuario que los datos introducidos no son correctos

Tabla 4.11.- Pruebas. Caso: Gestión de vídeos: Añadir (incorrecto)

- Caso: 11.- Gestión de Vídeos: Añadir sin URL del vídeo (incorrecto)

<i>Condiciones</i>	<i>El usuario debe estar identificado en el sistema</i>
<i>Acción</i>	El usuario rellena los campos del formulario de añadir un nuevo vídeo al sistema.

<i>Control</i>	El sistema valida los datos y muestra un mensaje alertando al usuario de que el campo URL es necesario
----------------	--

Tabla 4.12.- Pruebas. Gestión de vídeos: Añadir sin URL del vídeo (incorrecto)

- Caso: 12.- Modificación de datos personales (correcto)

<i>Condiciones</i>	<i>El usuario debe estar identificado en el sistema</i>
<i>Acción</i>	El usuario modifica sus datos personales
<i>Control</i>	El sistema valida los datos y actualiza los datos personales del usuario

Tabla 4.13.- Pruebas. Caso: Modificación de datos personales (correcto)

- Caso: 13.- Modificación de datos personales (incorrecto)

<i>Condiciones</i>	<i>El usuario debe estar identificado en el sistema</i>
<i>Acción</i>	El usuario modifica sus datos personales
<i>Control</i>	El sistema valida los datos y muestra un mensaje de error indicando al usuario que los datos introducidos no son correctos

Tabla 4.14.- Pruebas. Caso: Modificación de datos personales (incorrecto)

- Caso: 14.- El usuario se pone en contacto con el equipo de desarrollo (correcto)

<i>Acción</i>	<i>El usuario rellena el formulario de contacto</i>
<i>Control</i>	El sistema valida los datos y muestra al usuario un mensaje de confirmación.

Tabla 4.15.- Pruebas. Caso: Contacto (correcto)

- Caso: 15.- El usuario se pone en contacto con el equipo de desarrollo (incorrecto)

<i>Acción</i>	<i>El usuario rellena el formulario de contacto</i>
<i>Control</i>	El sistema valida los datos y muestra al usuario un mensaje de error, informándole de que los datos introducidos no son correctos

Tabla 4.16.- Pruebas. Caso: Contacto (incorrecto)

- Caso: 16.- Cambiar la privacidad de un vídeo-tutorial (correcto)

<i>Condiciones</i>	<i>El usuario debe estar identificado en el sistema</i>
<i>Acción</i>	El usuario cambia la privacidad de un vídeo-tutorial
<i>Control</i>	En el caso de que cambiemos a una publicación privada, el sistema validará los datos y generará una clave privada. En el caso de que sea pública el sistema eliminará la clave privada de la que dispone.

Tabla 4.17.-Pruebas. Caso: Cambiar privacidad (correcto)

- Caso: 17.- Cambiar la privacidad de un vídeo-tutorial (incorrecto)

<i>Condiciones</i>	<i>El usuario debe estar identificado en el sistema</i>
<i>Acción</i>	<i>El usuario cambia la privacidad de un vídeo-tutorial</i>
<i>Control</i>	<i>El sistema valida los cambios y muestra un mensaje de error indicando al usuario que el cambio de tipo de privacidad no se ha podido realizar de manera correcta</i>

Tabla 4.18.- Pruebas. Caso: Cambiar privacidad (incorrecto)

- **Casos probados en la Aplicación Android:**

- Caso:1.- Visualizar vídeo asociado a etiqueta NFC (correcto)

<i>Condiciones</i>	<i>El usuario debe disponer de un dispositivo móvil o tablet que soporte tecnología NFC</i>
<i>Acción</i>	<i>El usuario pone la parte trasera de su dispositivo en contacto con la etiqueta NFC</i>
<i>Control</i>	<i>La aplicación lee el contenido de la etiqueta NFC e inicializa la reproducción del vídeo asociado a la tarjeta</i>

Tabla 4.19.- Pruebas. Caso: Visualizar vídeo de etiqueta NFC (correcto)

- Caso:2.- Visualizar vídeo asociado a etiqueta NFC (incorrecto)

<i>Condiciones</i>	<i>El usuario debe disponer de un dispositivo móvil o tablet que soporte tecnología NFC</i>
<i>Acción</i>	<i>El usuario pone la parte trasera de su dispositivo en contacto con la etiqueta NFC</i>
<i>Control</i>	<i>La aplicación lee el contenido de la etiqueta NFC e y muestra al usuario una alerta indicando el problema ocurrido.</i>

Tabla 4.20.- Pruebas. Caso: Visualizar vídeo de etiqueta NFC (incorrecto)

- Caso:3.- Borrar contenido etiqueta NFC (correcto)

<i>Condiciones</i>	<i>El usuario debe disponer de un dispositivo móvil o tablet que soporte tecnología NFC</i>
<i>Acción</i>	<i>El usuario pone la parte trasera de su dispositivo en contacto con la etiqueta NFC</i>
<i>Control</i>	<i>La aplicación realiza el borrado de la etiqueta NFC y muestra al usuario un mensaje confirmando que el proceso de borrado se ha completado correctamente</i>

Tabla 4.21.- Pruebas. Caso: Borrar contenido etiqueta NFC (correcto)

- Caso:4.- Borrar contenido etiqueta NFC (incorrecto)

<i>Condiciones</i>	<i>El usuario debe disponer de un dispositivo móvil o tablet que soporte tecnología NFC</i>
<i>Acción</i>	<i>El usuario pone la parte trasera de su dispositivo en contacto con la etiqueta NFC</i>
<i>Control</i>	<i>La aplicación realiza el borrado de la etiqueta NFC y muestra al usuario un mensaje de error para informar de que el contenido de la etiqueta NFC no se ha borrado.</i>

Tabla 4.22.- Pruebas. Caso: Borrar contenido etiqueta NFC (incorrecto)

- Caso:5.- Enlazar vídeo con etiqueta NFC (correcto)

<i>Condiciones</i>	<i>El usuario debe disponer de un dispositivo móvil o tablet que soporte tecnología NFC</i>
<i>Acción</i>	<i>El usuario pone la parte trasera de su dispositivo en contacto con la etiqueta NFC</i>
<i>Control</i>	<i>La aplicación escribe el vídeo seleccionado por el usuario en su tarjeta NFC y muestra al usuario un mensaje de confirmación</i>

Tabla 4.23.- Pruebas. Caso: Enlazar vídeo con etiqueta NFC (correcto)

- Caso:6.- Enlazar vídeo con etiqueta NFC (incorrecto)

<i>Condiciones</i>	<i>El usuario debe disponer de un dispositivo móvil o tablet que soporte tecnología NFC</i>
<i>Acción</i>	<i>El usuario pone la parte trasera de su dispositivo en contacto con la etiqueta NFC</i>
<i>Control</i>	<i>La aplicación muestra al usuario un mensaje de error, indicándole que el proceso de escritura no se ha completa de manera correcta.</i>

Tabla 4.24.- Pruebas. Caso: Enlazar vídeo con etiqueta NFC (incorrecto)

- Caso:7.- Búsqueda de vídeo usando clave privada (correcto)

<i>Condiciones</i>	<i>El usuario debe tener registrado en el sistema web un vídeo de forma privada</i>
<i>Acción</i>	<i>El usuario introduce la clave privada de su vídeo en el campo de búsqueda</i>
<i>Control</i>	<i>La aplicación muestra al usuario el vídeo que busca</i>

Tabla 4.25.- Pruebas. Caso: vídeo usando clave privada (correcto)

- Caso:8.- Búsqueda de vídeo usando clave privada (incorrecto)

<i>Condiciones</i>	<i>El usuario debe tener registrado en el sistema web un vídeo de forma privada</i>
<i>Acción</i>	<i>El usuario introduce la clave privada de su vídeo en el campo de búsqueda</i>
<i>Control</i>	<i>La aplicación muestra al usuario un mensaje de error para informarle que la clave suministrada no es correcta</i>

Tabla 4.26.- Pruebas. Caso: vídeo usando clave privada (incorrecto)

### 4.6.1. Resultados

A continuación, se muestran los resultados obtenidos tras realizar cada una de las pruebas detalladas en el apartado anterior.

- **Resultados obtenidos en las pruebas realizadas al sistema web:**

<i>Casos</i>	<i>Problemas detectados</i>	<i>Acciones</i>	<i>Resultado</i>
<i>Caso1</i>			Superado
<i>Caso2</i>			Superado
<i>Caso3</i>			Superado
<i>Caso4</i>	No comprueba si los campos de contraseña (contraseña y confirmación), son iguales	Solucionar problema encontrado	Corregido
<i>Caso5</i>			Superado
<i>Caso6</i>			Superado
<i>Caso7</i>			Superado
<i>Caso8</i>			Superado
<i>Caso9</i>			Superado
<i>Caso10</i>			Superado
<i>Caso11</i>			Superado
<i>Caso12</i>			Superado
<i>Caso13</i>			Superado
<i>Caso14</i>	No limpia el formulario de contacto una vez que se ha realizado la acción	Solucionar problema encontrado	Corregido
<i>Caso15</i>			Superado
<i>Caso16</i>			Superado
<i>Caso17</i>			Superado

Tabla 4.27.- Resultados obtenidos en el sistema web

- **Resultados obtenidos en las pruebas realizadas a la aplicación Android:**

<i>Casos</i>	<i>Problemas detectados</i>	<i>Acciones</i>	<i>Resultado</i>
<i>Caso1</i>	No muestra mensaje de error en caso de que el contenido de la etiqueta NFC sea vacío		Corregido
<i>Caso2</i>			Superado
<i>Caso3</i>			Superado
<i>Caso4</i>			Superado
<i>Caso5</i>			Superado
<i>Caso6</i>			Superado
<i>Caso7</i>			Superado
<i>Caso8</i>	No muestra mensaje de error de clave no valida	Solucionar problema encontrado	Corregido

Tabla 4.28.- Resultados obtenidos en la aplicación Android

## Capítulo 5

---

### 5. Conclusiones y líneas de trabajo futuras

En este trabajo fin de grado se ha desarrollado un sistema web que permite la gestión de vídeos tutoriales y una aplicación Android que permite asociar etiquetas NFC con vídeo-tutoriales.

La propuesta de este trabajo supone que personas discapacitadas que son dependientes de otras para realizar determinadas tareas cotidianas del hogar puedan realizarlas de una manera más independiente. Esto puede suponer una gran apuesta si tenemos en cuenta que en la actualidad el número de personas dependientes en España está en torno al millón y que en otros países la cifra puede aumentar. Además, cabe destacar que la mayor parte de las personas dependientes deben esta dependencia a enfermedades como la demencia.

La tecnología NFC es muy sencilla de utilizar y su implementación no es muy costosa. Además, esta tecnología no requiere que el usuario que disfrute de ella necesite tener grandes conocimientos sobre dispositivos móviles o tablet, ya que como es en el caso de nuestra aplicación, el usuario con solo tocar con su dispositivo Android una etiqueta NFC la aplicación le mostrará la información que necesita.

Por todo esto, mi opinión personal de la tecnología NFC es que brinda una amplia variedad de aplicaciones en este campo y que puede aportar una gran cantidad de beneficios en personas que sufren esta dependencia.

Como hemos mencionado anteriormente, nuestro sistema está enfocado a ayudar a este tipo de personas permitiendo que sean capaces de realizar tareas cotidianas donde están implicados el uso de aparatos eléctricos y electrónicos que hasta ahora solo podían realizar con el apoyo de una persona física que les indicara las instrucciones. Nuestro sistema funciona de manera correcta y permite conseguir el objetivo para él que ha sido diseñado. Dicho sistema podría ser extendido para aumentar su funcionalidad: realizar una serie de mejoras que en mi opinión aumentarían la calidad del sistema desarrollado.

En primer lugar, sería muy interesante que nuestro sistema web funcionase como si fuera una red social, es decir, los usuarios pudiesen subir vídeos tutoriales a nuestro sistema web de la misma forma que se realiza actualmente y además realizar valoraciones sobre los vídeos-tutoriales de otros usuarios. Esto permitiría que la persona encargada de asociar los vídeos con las tarjetas NFC pudiera escoger entre los vídeos con mejor valoración. Además, se podría permitir que los usuarios pudieran comunicarse entre ellos, esto permitiría que los vídeos privados no solo puedan ser usados por el usuario que los creó, sino que este usuario podría establecer una relación con otro usuario y poder cederle la visualización de este vídeo-tutorial.

En segundo lugar, se podría aumentar la funcionalidad de nuestra aplicación Android y permitir que las tarjetas que sean escritas con nuestra aplicación estén protegidas contra el borrado o la sobrescritura limitando su escritura solamente al usuario que las enlazó. Esta ampliación evitaría que las tarjetas puedan ser reescritas por otros dispositivos NFC con otro fin.

Otro aspecto a ampliar es la interfaz de nuestra ampliación Android ya que su visualización en dispositivos con tamaño de pantalla superior a 10 pulgadas podría mejorarse.

Para finalizar, como conclusión personal, me gustaría poder expresar mi gran satisfacción de haber podido desarrollar este trabajo fin de grado, ya que me ha supuesto un reto y una posibilidad de aplicar todo lo que he aprendido en estos años en la universidad. La realización de este trabajo me ha permitido darme cuenta que gracias a los conocimientos aprendidos durante estos años soy capaz de enfrentarme a un problema y de aprender nuevas tecnologías de una manera rápida. Además, es una satisfacción saber que el trabajo que uno ha desempeñado en este trabajo fin de grado puede ser utilizado para mejorar la vida de muchas personas que por desgracia no pueden valerse por sí mismas.

## REFERENCIAS

---

### Para la redacción de esta memoria:

- [1] “NFC Forum”. <http://nfc-forum.org/>
- [2] “Mundo NFC”. <https://mundonfc.wordpress.com/2012/02/09/como-funciona-el-nfc/>
- [3] “NFC Connection”. [http://www.nfconnection.cl/que\\_es\\_nfc.php](http://www.nfconnection.cl/que_es_nfc.php)
- [4] “Universidad Nacional Abierta y a Distancia”.  
[http://datateca.unad.edu.co/contenidos/256594/256594\\_MOD/11cdigo\\_de\\_barras.html](http://datateca.unad.edu.co/contenidos/256594/256594_MOD/11cdigo_de_barras.html)
- [5] “Escuela de organización industrial (eoi)”.  
<http://www.eoi.es/blogs/scm/2013/03/08/codigos-qr-quick-response-barcode/>
- [6] “Sistemas de identificación por radiofrecuencias”.  
<http://www.it.uc3m.es/jmb/RFID/rfid.pdf>
- [7] “NFC Ampliando horizontes, S21sec”.
- [8] “Tuanity”. <https://tuanity.com/es/blog/49/android--la-r-evolucion-de-un-sistema>
- [9] “Androideity”. <http://androideity.com/2011/07/04/arquitectura-de-android/>
- [10] “Diploma de especialización en el desarrollo de aplicaciones Android. Universidad politécnica de valencia”.  
<http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/149-componentes-de-una-aplicacion>
- [11] “Diploma de especialización en el desarrollo de aplicaciones Android. Universidad politécnica de valencia”.  
<http://www.androidcurso.com/index.php/tutoriales-android/37-unidad-6-multimedia-y-ciclo-de-vida/158-ciclo-de-vida-de-una-actividad>

- [12] “Android Studio Faqs”. <http://androidstudiofaqs.com/conceptos/cual-es-la-estructura-de-un-proyecto-en-android-studio>
- [13] “SCS.-Sistemas Cliente/Servidor”.  
<http://ccia.ei.uvigo.es/docencia/SCS/0910/transparencias/Tema4.pdf>
- [14] “Titulo de Experto Universitario en el desarrollo de aplicaciones y servicios web”. <http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion01-apuntes.html>
- [15] “Dos Ideas. Personas y software”.  
<http://www.dosideas.com/noticias/java/314-introduccion-a-los-servicios-web-restful.html>
- [16] “Ingeniería del Software. Séptima edición. Ian Sommerville, 2005”.
- [17] “Bootstrap”. <http://getbootstrap.com/css/>
- [18] “Google Desing”. <https://design.google.com/>
- [19] “w3schools”. <http://www.w3schools.com/bootstrap/>
- [20] “php”. <http://php.net/manual/es/intro-what-is.php>
- [21] “Acerca de html”. <http://www.acercadehtml.com/manual-html/que-es-html.html>
- [22] “w3c”. <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
- [23] “Librosweb”. [http://librosweb.es/libro/javascript/capitulo\\_1.html](http://librosweb.es/libro/javascript/capitulo_1.html)
- [24] Jon Byous, Java technology: “The early years. Sun Developer Network”, sin fecha [ca. 1998]. Recuperado 21 de abril de 2005.
- [25] “Asociación de desarrolladores web de España”.  
<http://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>
- [26] M. Domínguez-Dorado, Todo Programación. Nº 13. Págs. 32-34. Editorial Iberprensa (Madrid). DL M-13679-2004. Noviembre, 2005. NetBeans IDE 4.1. “La alternativa a Eclipse”.
- [27] “phpmyadmin”. <https://www.phpmyadmin.net/>

[28] “Academia Android”. <http://academiaandroid.com/ide-android-intellij-android-studio-aide/>

**Para el desarrollo de nuestras aplicaciones:**

[1] “Android Developers”.

<https://developer.android.com/guide/topics/connectivity/nfc/index.html?hl=es>

[2] Vedat Coskun, Kerem Ok, Busra Ozdenizci, “Professional NFC Application Development for Android”, abril 2013.

[3] “Bootstrap”. <http://getbootstrap.com/css/>

[4] “Google Desing”. <https://design.google.com/>

[5] “w3schools”. <http://www.w3schools.com/bootstrap/>

## Anexos

---

### A. Contenido del CD-ROM

En este apartado se detalla el contenido del disco adjunto en la documentación de esta memoria de trabajo fin de grado.

- **Aplicaciones finales:**
  - **Aplicación web:** Carpeta con el contenido de nuestro sistema web.
  - **Aplicación móvil:** Fichero con extensión. apk para permitir la instalación en un sistema Android.
- **Base de datos:** En esta carpeta se almacena un archivo con extensión .sql, este archivo contiene una copia actual de toda la base de datos del sistema.
- **Código Fuente:** En esta carpeta podemos encontrar el código perteneciente a cada una de las aplicaciones desarrolladas en este proyecto. El código de nuestro sistema web se puede abrir con el IDE de programación NetBeans. Para abrir el código perteneciente a nuestra aplicación móvil, deberemos usar la IDE Android Studio.
- **Memoria:** Fichero pdf que contiene la documentación desarrollada en este trabajo fin de grado.
- **Demostración de funcionamiento:** Contiene un vídeo demostrativo del funcionamiento de nuestro sistema.

### B. Manual de instalación

En el siguiente manual de instalación, se describe los pasos a seguir para realizar el despliegue de las aplicaciones que constituyen el sistema desarrollado en este proyecto.

Para entender de una forma correcta este manual, vamos a diferenciar entre los pasos necesarios para realizar la instalación de nuestro sistema web y los pasos necesarios para llevar a cabo la instalación de nuestra aplicación móvil.

## B.1. Sistema web

En primer lugar, se debe tener en cuenta que nuestro sistema web requiere un servidor con soporte para PHP, MySQL y preferiblemente un a distribución Linux.

El primer paso que debemos realizar es la creación de la base de datos que utiliza nuestro sistema web para su funcionamiento. Para la creación de esta base de datos podemos usar phpMyAdmin que nos permite realizar el proceso de una forma sencilla. Para ello, inicializamos nuestro gestor de base de datos y pulsamos en el botón “Nueva” de la barra lateral de nuestro panel de administración (véase en la figura Manual 1).

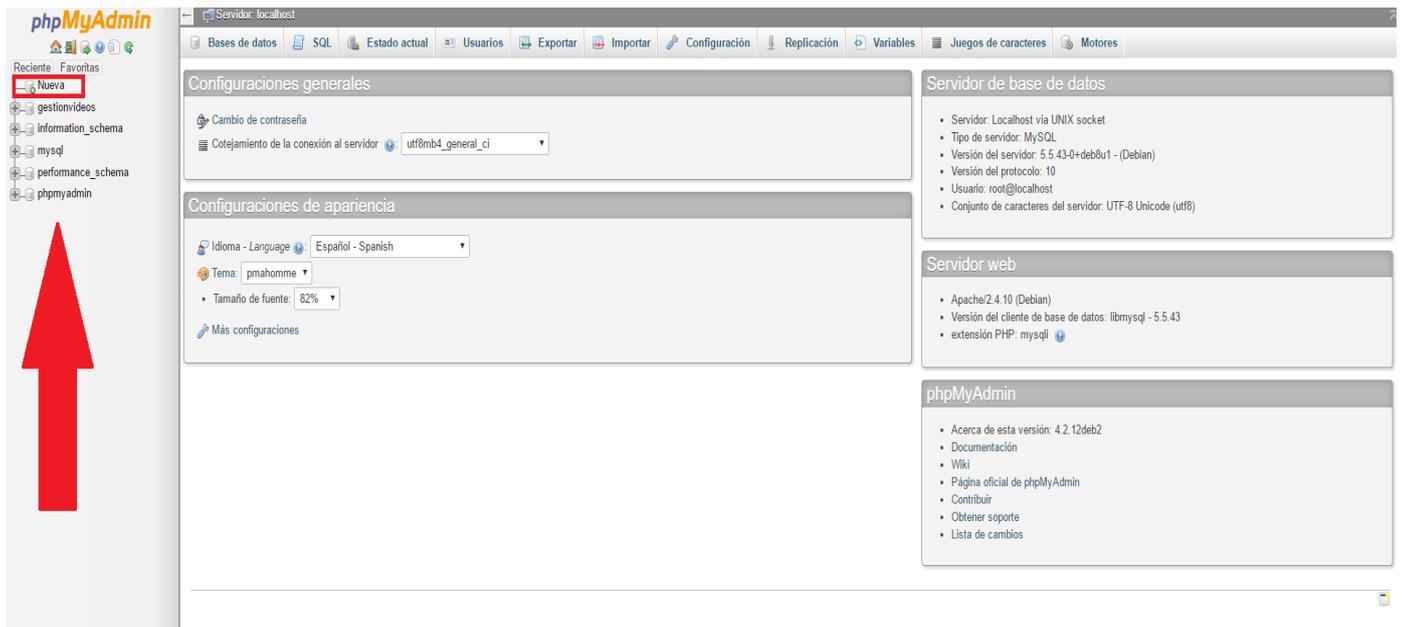


Figura Manual 1.- Panel de administración de phpMyAdmin

Seguidamente, creamos una base de datos con el nombre “gestionvideos” y un usuario para acceder a nuestra base de datos.

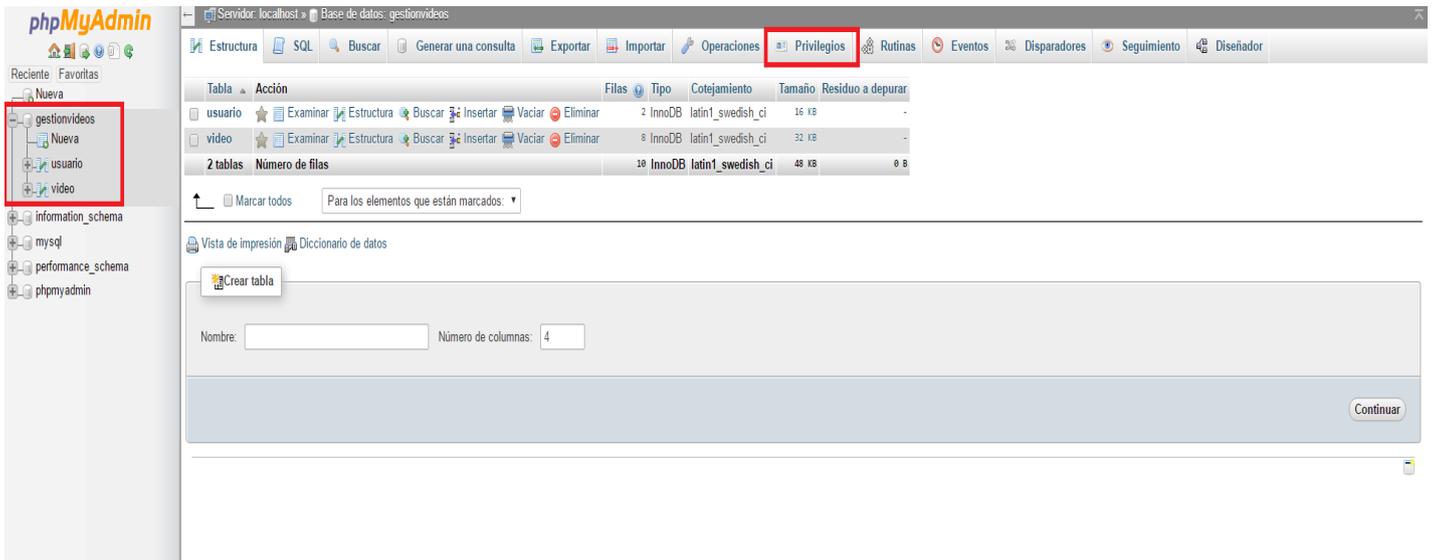


Figura Manual 2.- Panel de administración de nuestra base de datos.

Para crear un usuario, debemos pulsar en nuestra base de datos y después pulsar en el botón “Privilegios” de la pantalla de administración de nuestra base de datos (véase en la figura Manual 2) y seguidamente en la opción “Agregar usuario”. El usuario creado deberá tener todos los tipos de privilegios.

Una vez hayamos creado la base de datos, debemos importar el archivo con extensión .sql que se aporta en contenido del CD-ROM. Este archivo contiene una copia del contenido actual de la base de datos y al importarlo nos creara las tablas necesarias para la utilización de nuestro sistema web.

Para importar archivo, debemos pulsar en el botón “importar” del panel de administración de nuestra base de datos (véase en la figura Manual 3) y después, seleccionar el archivo .sql y seguir los pasos que nuestro gestor nos indica.

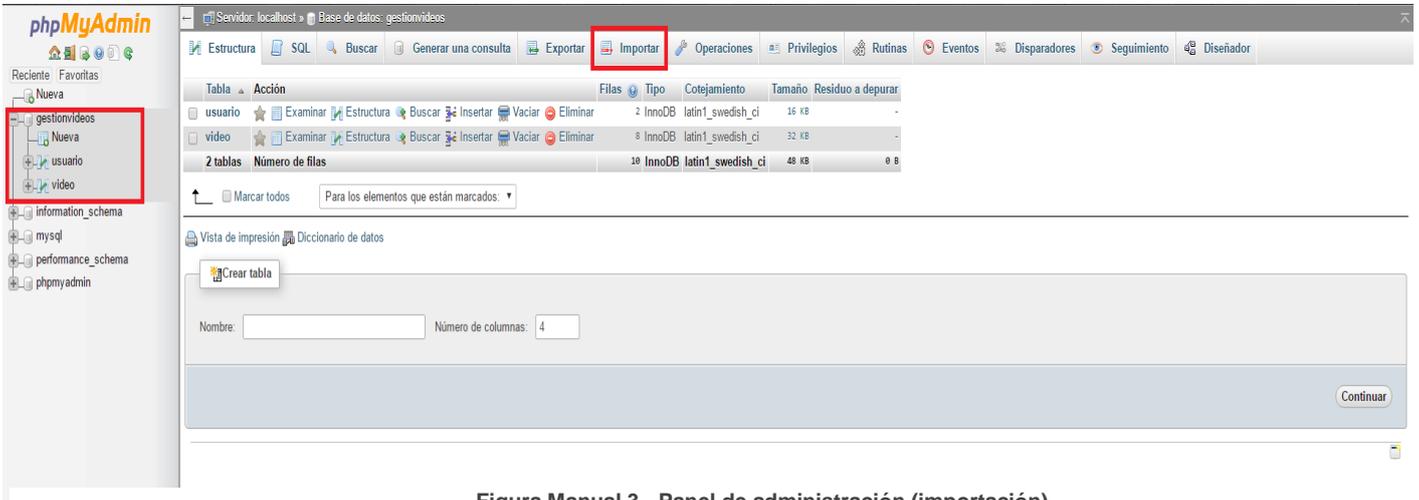


Figura Manual 3.- Panel de administración (importación)

El segundo paso, será conectarnos a nuestro servidor vía FTP y copiar la carpeta de nuestro sistema web en la carpeta www de nuestro servidor Apache. Para ello podemos usar algún cliente FTP como puede ser FileZilla o WinSCP.

Una vez que la carpeta se haya subido de forma correcta a nuestro servidor, debemos realizar los siguientes cambios:

1. Cambiar el contenido del archivo .htacces, localizado en la dirección “GestionVideos/htacces”, por lo siguiente:

```
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule .* index.php/$0 [PT,L]
```

2. Debemos cambiar la configuración de nuestra aplicación, para ello nos dirigiremos al archivo config.php localizado en la siguiente dirección “GestionVideos/application/config/config.php” y realizaremos las siguientes modificaciones:

- a. `$config['base_url'] = 'http://ceatic.ujaen.es:8027';`
- b. `$config['index_page'] = '';`

- c. `$config['uri_protocol']='AUTO';`
3. Debemos configurar la conexión a nuestra base de datos, para ello nos dirigiremos al archivo `database.php` localizado en la dirección “`GestionVideos/application/config/database.php`” y realizar los siguientes cambios:
  - a. `'username'=>'nombre del usuario que hayamos creado'`
  - b. `'password'=>'la contraseña del usuario'`
4. Para que no tengamos problemas en la visualización de algunas imágenes que forman nuestro sistema, debemos realizar las siguientes modificaciones:
  - a. Dirigirnos al archivo `gv.css` localizado en la dirección “`GestionVideos/css/gv.css`” y cambiar la localización de las imágenes de fondos por los siguiente:  
`/imagen/nombre_de_la_imagen`
  - b. Dirigirnos al archivo `admin.js` localizado en la dirección “`GestionVideos/js/admin.js`” y cambiar la localización de las imágenes por la siguiente:  
`/imagen/nombre_de_la_imagen`

Una vez realizados todos estos cambios, ya podremos acceder de forma correcta a nuestro sistema web.

## B.2. Aplicación móvil

Para realizar la instalación de nuestra aplicación móvil, deberemos ejecutar el archivo `TutorialTag.apk` en nuestro dispositivo móvil o tablet y automáticamente se iniciará la instalación de nuestra aplicación.

Si por lo contrario, preferimos instalar nuestra aplicación Android directamente con nuestro IDE de desarrollo Android Studio, debemos conectar nuestro dispositivo móvil o tablet con un cable USB a nuestro ordenador y activar el modo depuración de nuestro dispositivo. Seguidamente, nos dirigiremos a carpeta de código fuente de nuestro CD y abriremos con Android Studio la carpeta de nuestra aplicación móvil. Para terminar, solo tendremos que pulsar

en el botón de ejecutar nuestro proyecto en Android Studio y la aplicación se instalara e iniciara en nuestro dispositivo móvil o tablet.

## C. Manual de usuario del sistema web

### C.1. Acceso

Para entrar al panel de administración de nuestro sistema web, el usuario solo deberá rellenar el formulario de acceso (ver figura Manual 4) que se muestra en la página principal de nuestra aplicación web. En este formulario el usuario deberá introducir sus credenciales de acceso.



El formulario de acceso muestra dos campos de entrada: 'Correo electrónico' con un ícono de persona y 'Contraseña' con un ícono de candado. A la derecha hay un botón azul con el texto 'Iniciar sesión'. Debajo de los campos hay un enlace azul que dice '¿Olvidates la contraseña?'.

Figura Manual 4.- Formulario de acceso

Si los datos son correctos, automáticamente el sistema nos redirigirá a nuestro panel de administración (figura Manual 8).

### C.2. Registro del usuario

Para poder registrarse en nuestro sistema web, el usuario deberá rellenar correctamente el formulario de registro (véase en la figura Manual 5) que nuestro sistema muestra en la página principal. En este formulario el usuario deberá introducir sus datos personales que pasaran a formar parte de su cuenta de usuario.



Regístrate

Nombre

Apellidos

Correo electrónico

Contraseña

Confirmar contraseña

Registrarme

Figura Manual 5.- Formulario de registro

Si los datos introducidos por el usuario son correctos, el sistema permitirá al usuario acceder a su panel de administración (figura).

### C.3. Recuperación de clave de acceso

En caso de que el usuario hay olvidado su contraseña de acceso a nuestro sistema web. El sistema le proporciona al usuario la posibilidad de recuperar su contraseña, para ello deberá pulsar en el botón “¿Olvidaste tu contraseña? Del formulario de acceso y el sistema mostrará un formulario (figura Manual 6) donde el usuario deberá introducir su cuenta de correo con la que está registrado en nuestro sistema.



¿Quieres recordar su contraseña?

Introduzca la cuenta de correo que usted utiliza para acceder a su cuenta. Te enviaremos un correo con instrucciones para modificar tu contraseña.

@ Dirección de correo

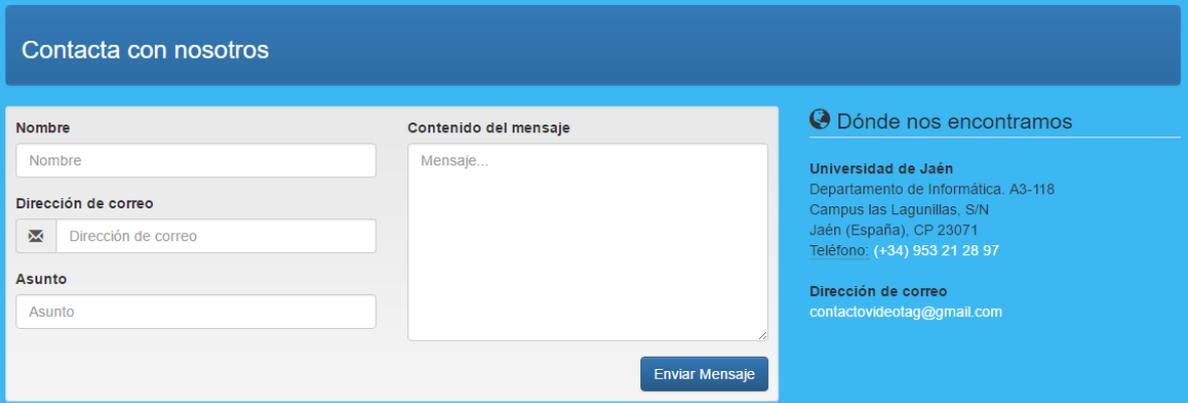
Continuar

Figura Manual 6.- Formulario de recuperación de contraseña

Si la cuenta de usuario proporcionada por el usuario es correcta, el sistema enviara automáticamente a la cuenta de correo del usuario un mensaje con su nueva contraseña de acceso.

## C.4. Contacto

El usuario o cualquier visitante de nuestro sistema web, puede ponerse en contacto con el equipo de desarrollo de nuestro sistema. Para ello, el usuario deberá rellenar el formulario de contacto (véase en la figura Manual 7) que nuestro sistema le proporciona en el apartado “Sobre nosotros” de nuestro sistema web.



Formulario de contacto con los siguientes campos:

- Nombre:** Campo de texto con el valor "Nombre".
- Dirección de correo:** Campo de texto con un ícono de correo y el valor "Dirección de correo".
- Asunto:** Campo de texto con el valor "Asunto".
- Contenido del mensaje:** Área de texto grande con el valor "Mensaje...".
- Enviar Mensaje:** Botón de envío.

Información de contacto adicional:

- Dónde nos encontramos:** Universidad de Jaén, Departamento de Informática, A3-118, Campus las Lagunillas, S/N, Jaén (España), CP 23071, Teléfono: (+34) 953 21 28 97.
- Dirección de correo:** [contactovideotag@gmail.com](mailto:contactovideotag@gmail.com)

Figura Manual 7.- Formulario de contacto

## C.5. Gestionar vídeos-tutoriales

El usuario al acceder a su panel de administración, podrá visualizar todos sus vídeos y realizar las operaciones que considere necesarias.

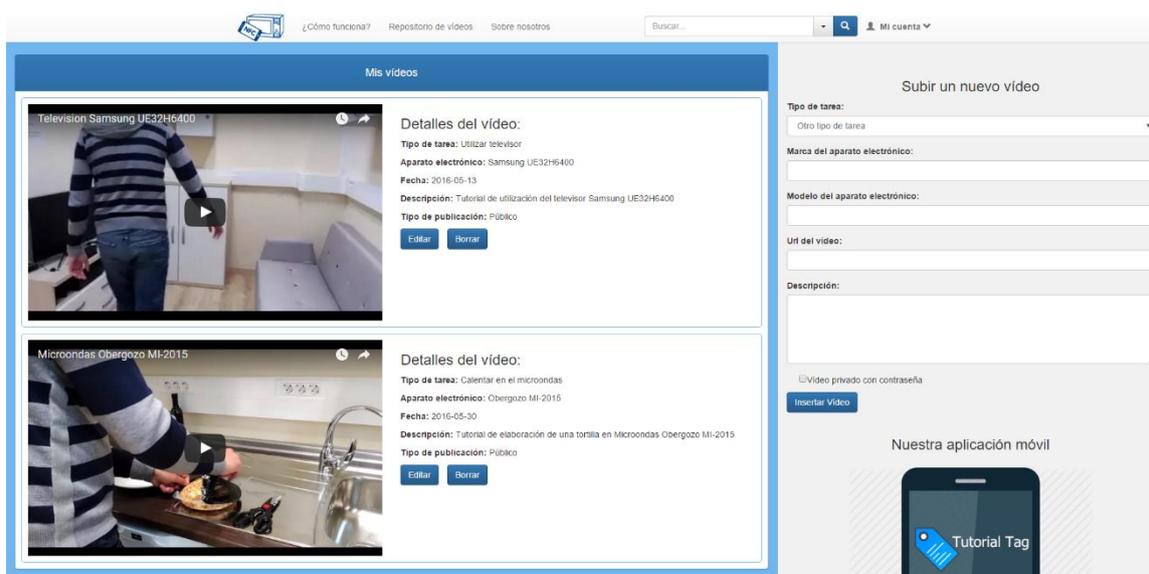


Figura Manual 8.- Panel de administración de nuestro sistema web

### *C.5.1. Añadir un nuevo vídeo-tutorial*

En primer lugar, el usuario deberá crear su vídeo y subirlo de forma correcta a su cuenta de YouTube. Una vez que el vídeo se encuentre en YouTube, el usuario deberá copiar la URL de su vídeo y rellenar los datos el formulario “subir un nuevo vídeo” (figura Manual 9) que nuestro sistema web le proporciona en su panel de administración. Para finalizar, el usuario deberá pulsar el botón “insertar vídeo” para que se realice el proceso de inserción en nuestro sistema web.

El usuario podrá obtener ayuda de cómo realizar la subida de un nuevo vídeo a nuestro sistema en el apartado “¿Cómo funciona?” de nuestro sistema web.



Subir un nuevo vídeo

Tipo de tarea:  
Otro tipo de tarea

Marca del aparato electrónico:  
[Campo de texto]

Modelo del aparato electrónico:  
[Campo de texto]

Uri del vídeo:  
[Campo de texto]

Descripción:  
[Campo de texto]

Vídeo privado con contraseña

Insertar Vídeo

Figura Manual 9.- Formulario de subir un nuevo vídeo

Si los datos introducidos por el usuario son correctos, el sistema actualizará la lista de vídeos del panel de administración del usuario.

### *C.5.2. Editar un vídeo-tutorial*

Nuestro sistema web permite al usuario poder editar cualquier vídeo que sea de su propiedad. Para ello el usuario deberá localizar el vídeo que desea editar en su panel de administración y seguidamente, deberá pulsar en el botón “Editar” que tiene asociado el vídeo seleccionado. Al pulsar en este botón, el sistema muestra al usuario un formulario (figura Manual 10) con los datos que contiene el vídeo seleccionado en nuestro sistema, permitiendo así la edición de los mismos. Para finalizar, el usuario deberá pulsar el botón “Guardar”, para que de esta forma los cambios realizados queden almacenados en nuestro sistema.

Repositorio de videos Sobre nosotros Buscar...

**Editar vídeo**

**Tipo de tarea:**  
Utilizar televisor

**Marca del aparato electrónico:**  
Samsung

**Modelo del aparato electrónico:**  
UE32H6400

**Url del vídeo:**  
https://www.youtube.com/watch?v=IbLbke09t5E

**Descripción:**  
Tutorial de utilización del televisor Samsung UE32H6400

Vídeo privado con contraseña

Guardar Cerrar

Fecha: 2016-06-30

Figura Manual 10.- Formulario para editar los datos de un vídeo

Si los datos modificados por el usuario son correctos, el sistema actualizara el panel de admiración para que el usuario pueda visualizar que el vídeo modificado contiene los datos actualizados.

### *C.5.3. Eliminar un vídeo-tutorial*

Para realizar la eliminación de un vídeo, el usuario deberá localizar el vídeo que desea eliminar en su panel de administración y pulsar el botón “Borrar” que tiene asociado el vídeo seleccionado. Seguidamente, el usuario deberá aceptar el mensaje de confirmación (figura Manual 11) que el sistema le proporciona para asegurarse de que verdaderamente se quiere eliminar el vídeo seleccionado.

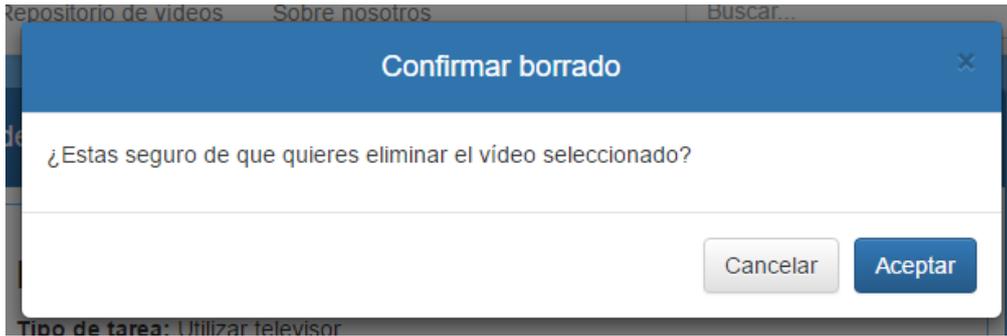
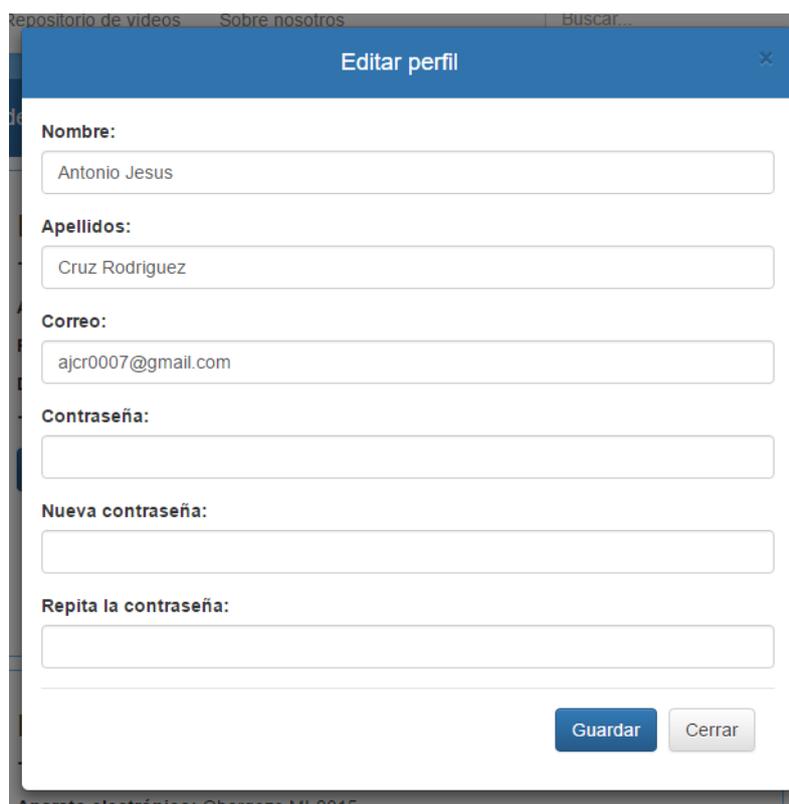


Figura Manual 11.- Mensaje de confirmación de borrado

Al confirmar este mensaje, el sistema realizara la eliminación del vídeo seleccionado y actualizara la lista de vídeos del panel de administración del usuario.

## C.6. Modificación de datos personales

El usuario podrá modificar los datos personales de sus cuentas en nuestro sistema web, para ello deberá pulsar el botón “Mi cuenta” del panel de administración y seguidamente en el botón “Actualizar datos”. El sistema web le mostrara al usuario un formulario (figura Manual 12) con sus datos personales (nombre, apellidos...etc.) y el podrá editar lo datos que considere necesarios. Al finalizar, el usuario deberá pulsar sobre el botón “Guardar” y el sistema actualizara sus datos personales.



repositorio de videos Sobre nosotros Buscar...

**Editar perfil** ×

**Nombre:**

**Apellidos:**

**Correo:**

**Contraseña:**

**Nueva contraseña:**

**Repita la contraseña:**

Aparato electrónico: Chango ML-2015

Figura Manual 12.- Formulario para editar los datos personales del usuario

## D. Manual de usuario de la aplicación móvil

Nuestra ampliación móvil, nos proporciona una pantalla principal (véase en la figura Manual 13) intuitiva para el usuario, que nos permite acceder a cada una de las opciones de nuestra aplicación de una forma sencilla.

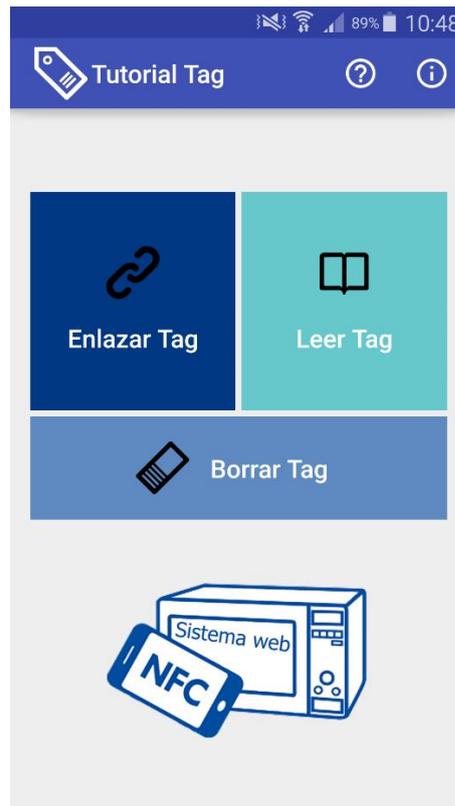


Figura Manual 13.- Pantalla principal de nuestra aplicación Android

## D.1. Enlazar vídeo-tutorial con etiqueta NFC

Para realizar el proceso de enlazar un vídeo tutorial con una etiqueta NFC, el usuario dispondrá de dos opciones dependiendo del tipo de vídeo que quiere enlazar.

Estas dos opciones se explican a continuación:

- El usuario quiere enlazar un vídeo que el registro en nuestro sistema web de forma pública o simplemente el usuario no se encuentra registrado en nuestro sistema web y por lo tanto busca un vídeo entre los disponibles en nuestro sistema.

En este caso, el usuario deberá pulsar en el botón “Enlazar Tag” de la pantalla principal de nuestra aplicación Android (véase en la figura Manual 13). Seguidamente, deberá pulsar en la pestaña “Vídeos públicos” y después, usará los criterios de búsqueda

(tarea, marca y modelo) que nuestra aplicación le proporciona (figura Manual 14).

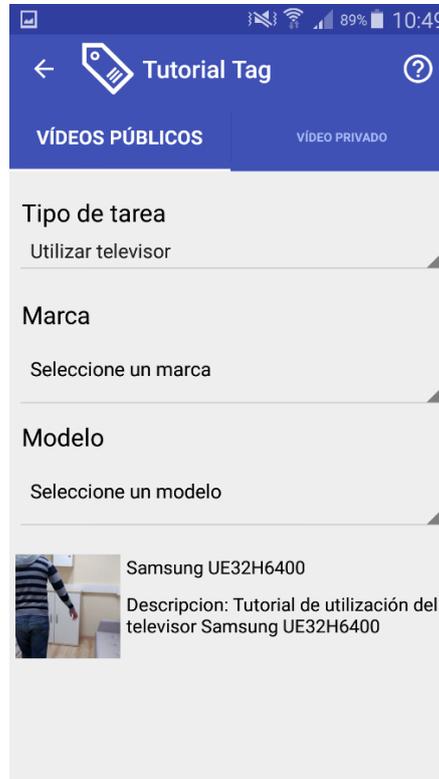


Figura Manual 14.- Buscar vídeos públicos

Una vez que el usuario hay realizado la búsqueda y haya encontrado un vídeo que se adapte a la tarea que quiere resolver, el usuario pulsará en la miniatura del vídeo y automáticamente nuestra aplicación lo redirigirá a la pantalla de enlazado (figura Manual 15).



Figura Manual 15.- Pantalla de enlace con tarjeta NFC

Cabe destacar que si el usuario no encuentra un vídeo que se adapte a su tarea, deberá registrarse en nuestro sistema web y de esta forma podrá crear sus vídeos y usarlos en sus etiquetas NFC.

- El usuario quiere usar un vídeo personal que el registro en nuestro sistema de forma privada.

En este caso, el usuario también deberá pulsar en el botón “Enlazar Tag” de la pantalla principal de nuestra aplicación, pero a diferencia del caso anterior, deberá pulsar en la pestaña “Vídeo privado” y después, usará la clave privada que el sistema web le proporcione a la hora de registrar su vídeo. Si por algún motivo el usuario no recuerda esta clave, deberá dirigirse a su panel de administración en nuestro sistema web, donde podrá visualizar dicha clave en los detalles del vídeo que quiere enlazar (véase en la figura Manual 16).

Figura Manual 16.- Panel de administración de nuestro sistema web

Una vez que el usuario haya introducido la clave asociada a su vídeo, pulsara en el botón “Aceptar” y de esta forma nuestra aplicación realizara la búsqueda del vídeo seleccionado y permitirá al igual que en el caso anterior, que el usuario pulse en la miniatura del vídeo y automáticamente se le redirigirá a la pantalla de enlazado.

Finalmente, el usuario deberá poner en contacto la parte trasera de su dispositivo móvil o tablet con su etiqueta NFC y pulsar en el botón “Escribir Tag” de la pantalla de enlazado y automáticamente, la aplicación le mostrará una ventana con el estado del proceso de escritura en la etiqueta NFC (véase en la figura Manual 17 y figura Manual 18).

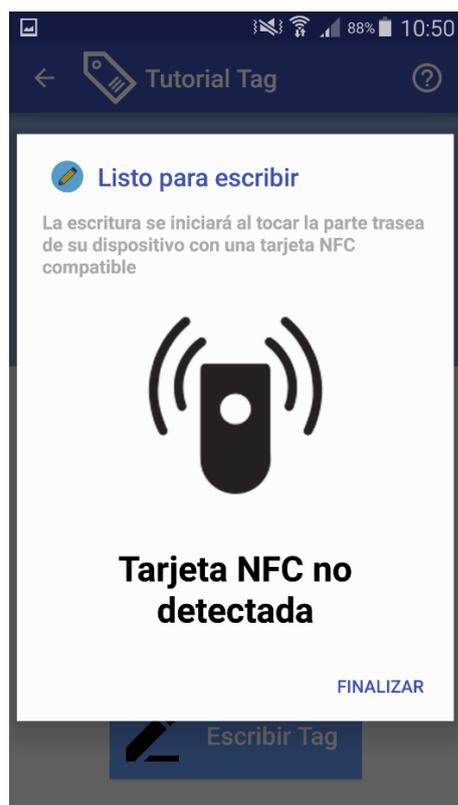


Figura Manual 17.- Pantalla de estado de escritura en etiqueta NFC

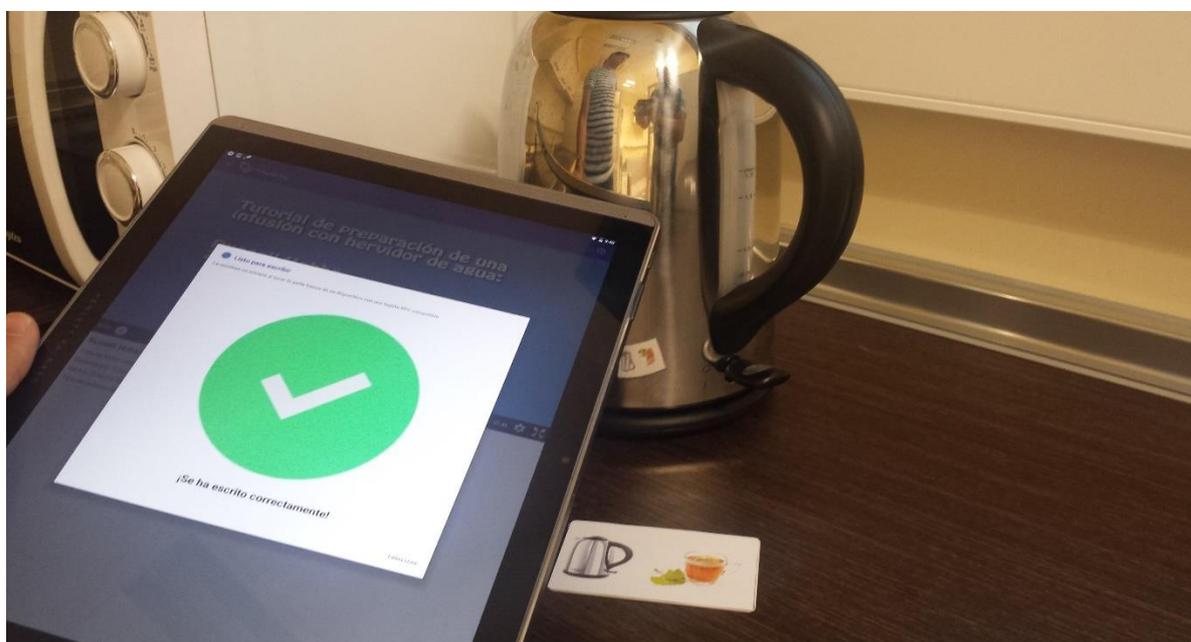


Figura Manual 18.- Demostración de escritura en NFC

En el proceso de escritura en la etiqueta NFC, existen 4 posibles estados (véase en la figura Manual 19).



Figura Manual 19.- Estados en el proceso de escritura

Cada uno de estos estados implica una acción por parte del usuario:

- **Preparado:** La aplicación se encuentra preparada para realizar la escritura en la etiqueta NFC. El usuario deberá poner en contacto la parte trasera de su dispositivo móvil o tablet con su etiqueta NFC.
- **Detectado:** La aplicación avisa al usuario de que ha detectado correctamente su etiqueta NFC. El usuario deberá esperar unos segundos hasta que el proceso de escritura se realice de forma correcta.
- **Error en la escritura:** Este estado indica al usuario que ha ocurrido un error en el proceso de escritura. El usuario deberá poner de nuevo en contacto la parte trasera de su dispositivo móvil o tablet y el proceso de escritura se reiniciará de forma automática.
- **Escritura correcta:** Este estado indica que el proceso de escritura se ha completado de forma correcta y que el usuario ya puede usar su tarjeta NFC para visualizar su vídeo.

## D.2. Visualizar vídeo-tutorial asociado a etiqueta NFC

Para visualizar el vídeo-tutorial asociado a una etiqueta NFC, el usuario dispone de dos opciones:

- Solo deberá poner en contacto la parte trasera de su dispositivo móvil o tablet con su etiqueta NFC y la visualización del vídeo se iniciará de forma automática (véase en la figura Manual 20 y 21).

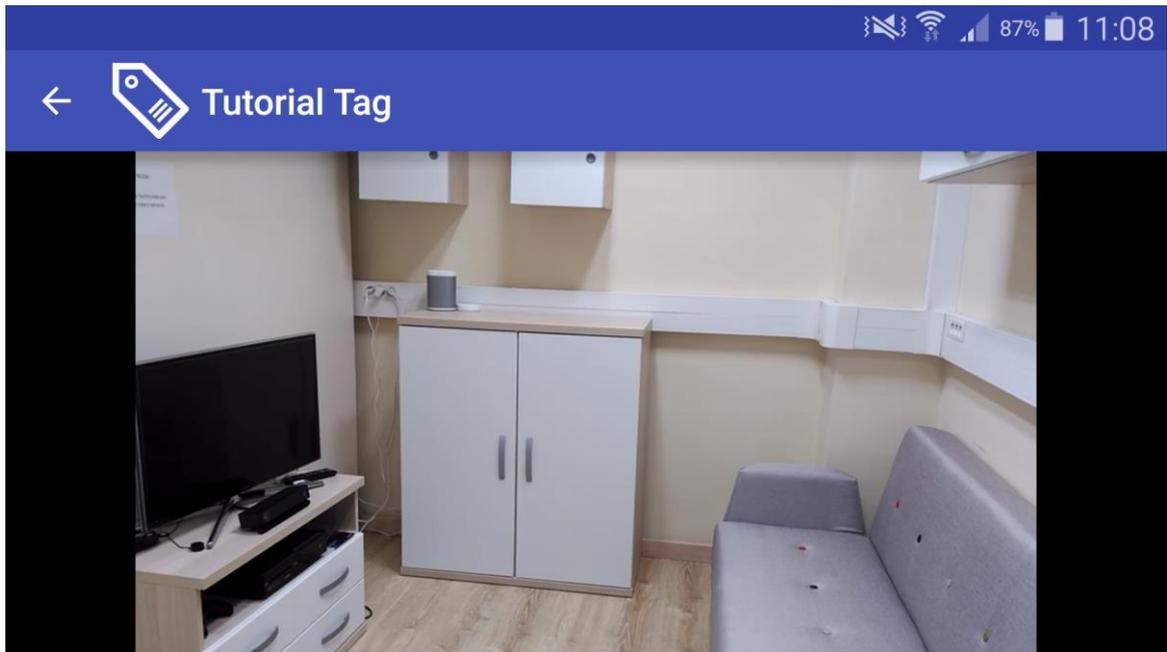


Figura Manual 20.- Pantalla de visualización de vídeo



Figura Manual 21. Demostración de visualización de un vídeo-tutorial

- El usuario pulsará en el botón “Leer Tag” de la pantalla principal de nuestra aplicación (figura Manual 13), seguidamente, el usuario pondrá la parte trasera de su dispositivo móvil o tablet con la tarjeta NFC y el vídeo se reproducirá de forma automática.

### D.3. Borrar contenido de tarjeta NFC

Para borrar el contenido de la tarjeta NFC, el usuario deberá pulsar en el botón “Borrar Tag” de la pantalla principal de nuestra aplicación (figura Manual 13) y la aplicación lanzará la pantalla de borrado (véase en la figura Manual 19).

Después, el usuario deberá colocar la parte trasera de su dispositivo móvil o tablet en contacto con su tarjeta NFC.



Figura Manual 22.- Pantalla de estado de borrado

Durante el proceso de borrado, la aplicación podrá mostrar al usuario los cuatro estados detallados en el proceso de escritura, pero aplicados al proceso de borrado.